



KEMENTERIAN
PENDIDIKAN
MALAYSIA

ASAS SAINS KOMPUTER

TINGKATAN

2





RUKUN NEGARA

Bahawasanya Negara Kita Malaysia
mendukung cita-cita hendak;

Mencapai perpaduan yang lebih erat dalam kalangan
seluruh masyarakatnya;

Memelihara satu cara hidup demokrasi;

Mencipta satu masyarakat yang adil di mana kemakmuran negara
akan dapat dinikmati bersama secara adil dan saksama;

Menjamin satu cara yang liberal terhadap
tradisi-tradisi kebudayaannya yang kaya dan pelbagai corak;

Membina satu masyarakat progresif yang akan menggunakan
sains dan teknologi moden;

MAKA KAMI, rakyat Malaysia,
berikrar akan menumpukan
seluruh tenaga dan usaha kami untuk mencapai cita-cita tersebut
berdasarkan prinsip-prinsip yang berikut:

**KEPERCAYAAN KEPADA TUHAN
KESETIAAN KEPADA RAJA DAN NEGARA
KELUHURAN PERLEMBAGAAN
KEDAULATAN UNDANG-UNDANG
KESOPANAN DAN KESUSILAAN**

KURIKULUM STANDARD SEKOLAH MENENGAH (KSSM)

ASAS SAINS KOMPUTER

TINGKATAN

2

PENULIS

Phang Sau Fong
Looi Kam Fatt
Rohaizan Bin Abd Jalil

PEREKA BENTUK

Ng Peck Foong

EDITOR

Yeoh Joue Huey

ILUSTRATOR

Ng Ying Tong



PERCETAKAN RINA SDN. BHD.
2017



KEMENTERIAN PENDIDIKAN MALAYSIA

NO. SIRI BUKU: 0063

KPM2017 ISBN 978-967-14509-3-2

Cetakan Pertama 2017

© Kementerian Pendidikan Malaysia

Hak Cipta Terpelihara. Mana-mana bahan dalam buku ini tidak dibenarkan diterbitkan semula, disimpan dalam cara yang boleh dipergunakan lagi, ataupun dipindahkan dalam sebarang bentuk atau cara, baik dengan cara bahan elektronik, mekanik, penggambaran semula mahupun dengan cara perakaman tanpa kebenaran terlebih dahulu daripada Ketua Pengarah Pelajaran Malaysia, Kementerian Pendidikan Malaysia. Perundingan tertakluk kepada perkiraan royalti atau honorarium.

Diterbitkan untuk Kementerian Pendidikan Malaysia oleh:

Percetakan Rina Sdn. Bhd. (31964-X)

Lot 45, Persiaran Mewah,

Bandar Tun Razak,

56000 Cheras, Kuala Lumpur.

Tel: 03-91721444

Faks: 03-91721421

Laman Web: www.percetakanrina.com

Reka Letak dan Atur Huruf:

Percetakan Rina Sdn. Bhd.

Muka Taip Teks: Optima

Saiz Muka Taip Teks: 11 poin

Dicetak oleh:

Percetakan Rina Sdn. Bhd.

Lot 45, Persiaran Mewah,

Bandar Tun Razak,

56000 Cheras, Kuala Lumpur.

Penghargaan

Penerbitan buku teks ini melibatkan kerjasama banyak pihak. Sekalung penghargaan dan terima kasih ditujukan kepada semua pihak yang terlibat:

- Jawatankuasa Penambahbaikan Prof Muka Surat, Bahagian Buku Teks, Kementerian Pendidikan Malaysia.
- Jawatankuasa Penyemakan Pembedaan Prof Muka Surat, Bahagian Buku Teks, Kementerian Pendidikan Malaysia.
- Jawatankuasa Penyemakan Naskhah Sedia Kamera, Bahagian Buku Teks, Kementerian Pendidikan Malaysia.
- Pegawai-pegawai Bahagian Buku Teks dan Bahagian Pembangunan Kurikulum, Kementerian Pendidikan Malaysia.
- Murphy Ng: www.mysabah.com
- Moley Robotics
- User: Mario: Wikimedia Commons

KANDUNGAN

PENDAHULUAN

Ciri-ciri Istimewa dalam Buku Ini dan Fungsinya

Panduan Penggunaan Peta Pemikiran i-THINK Semasa Pembelajaran

v

vi

viii

BAB 1

PERWAKILAN DATA

1

1.1 Sistem Nombor Perlapanan

2

1.1.1 Perbezaan antara nombor perlapanan dan nombor perpuluhan

2

1.1.2 (i) Penukaran nombor perpuluhan kepada nombor perlapanan

5

1.1.2 (ii) Penukaran nombor perlapanan kepada nombor perpuluhan

6

1.1.2 (iii) Penukaran nombor perlapanan kepada nombor perduaan

8

1.1.2 (iv) Penukaran nombor perduaan kepada nombor perlapanan

9

1.1.3 Penukaran pengekodan ASCII kepada nombor perlapanan berdasarkan aksara yang diberi

12

1.1.4 Hubung kait antara nombor perlapanan dengan kod ASCII dalam bahasa mesin

13

Praktis 1.1

16

1.2 Sistem Nombor Perenambelasan

18

1.2.1 Perbezaan antara nombor perenambelasan dan nombor perpuluhan

19

1.2.2 (i) Penukaran nombor perpuluhan kepada nombor perenambelasan

20

1.2.2 (ii) Penukaran nombor perenambelasan kepada nombor perpuluhan

22

1.2.2 (iii) Penukaran nombor perenambelasan kepada nombor perduaan

23

1.2.2 (iv) Penukaran nombor perduaan kepada nombor perenambelasan

25

1.2.3 Penukaran pengekodan ASCII kepada nombor perenambelasan berdasarkan aksara yang diberi

28

1.2.4 Hubung kait nombor perenambelasan dengan kod ASCII dalam bahasa mesin

30

Praktis 1.2

33

Peta Konsep Bab 1

34

Senarai Semak Bab 1

35

Pentaksiran Bab 1

36

**BAB
2****ALGORITMA**

38

2.1 Pembangunan Algoritma

40

2.1.1 (i) Struktur kawalan pilihan bersarang

41

2.1.1 (ii) Struktur kawalan ulangan

47

2.1.2 Mengesan dan membaiki ralat yang terdapat dalam pseudokod dan carta alir dalam penyelesaian masalah

59

2.1.3 Menggunakan pseudokod dan carta alir dengan menggabungkan pelbagai struktur kawalan untuk menyelesaikan masalah

65

Praktis 2.1

69

Peta Konsep Bab 2

70

Senarai Semak Bab 2

70

Pentaksiran Bab 2

71

**BAB
3****KOD ARAHAN**

74

3.1 Persekitaran Kod Arahan

76

3.1.1 Jenis data dalam segmen kod

77

3.1.2 Perbezaan antara pemboleh ubah dan pemalar dalam segmen kod

90

3.1.3 (i) Menghasilkan segmen kod menggunakan fungsi input dan fungsi output

95

3.1.3 (ii) Operator perbandingan

98

3.1.3 (iii) Operator logik

99

3.1.4 Mengesan dan membaiki ralat pada segmen kod dalam penyelesaian masalah

102

3.1.5 Menggunakan gabungan jenis data, pemboleh ubah, pemalar, dan operator untuk menyelesaikan masalah

115

Praktis 3.1

119

3.2 Struktur Kod Arahan

121

3.2.1 (i) Struktur kawalan jujukan dalam penyelesaian masalah

122

3.2.1 (ii) Struktur kawalan pilihan dalam penyelesaian masalah

127

3.2.1 (iii) Struktur kawalan pilihan bersarang dalam penyelesaian masalah

136

3.2.1 (iv) Struktur kawalan ulangan dalam penyelesaian masalah

143

3.2.2 Menguji atur cara dan membaiki ralat pada kod arahan yang dihasilkan

149

3.2.3 Atur cara yang melibatkan gabungan pelbagai struktur kawalan

159

Praktis 3.2

166

Peta Konsep Bab 3

168

Senarai Semak Bab 3

170

Pentaksiran Bab 3

172

LAMPIRAN

176

JAWAPAN

182

GLOSARI

189

INDEKS

191

SENARAI RUJUKAN

192

PENDAHULUAN

Buku Teks Asas Sains Komputer Tingkatan 2 ini ditulis berdasarkan Dokumen Standard Kurikulum dan Pentaksiran (DSKP) untuk Kurikulum Standard Sekolah Menengah (KSSM) yang diterbitkan oleh Kementerian Pendidikan Malaysia. Buku teks ini ditulis mengikut fokus kurikulum Asas Sains Komputer bagi menyediakan murid dengan pengetahuan, kemahiran dan nilai dalam empat bidang pembelajaran, iaitu Konsep Asas Pemikiran Komputasional, Perwakilan Data, Algoritma dan Kod Arahan. Murid-murid perlu dipupuk dengan fikiran komputasional supaya mereka dapat menyelesaikan masalah, mereka bentuk dan mencipta aplikasi baharu.

Buku ini berorientasikan strategi pembelajaran dalam Asas Sains Komputer. Pelbagai aktiviti dan tugas dirancang berbandaskan pembelajaran berasaskan inkuiri yang dapat mencetuskan pemikiran kritis dan kreatif murid. Murid didedahkan dengan pelbagai bentuk soalan dalam latihan formatif dan sumatif supaya mereka dapat mentaksir pembelajaran sendiri. Strategi ini membolehkan murid terlibat secara mendalam dan lebih bertanggungjawab ke atas pembelajaran mereka. Pembelajaran berasaskan penyelesaian masalah, pembelajaran koperatif dan kolaboratif, pembelajaran masteri, pembelajaran konstruktivisme dan pembelajaran berasaskan projek diberi penekanan dalam buku ini.

Amalan STEM (*Science, Technology, Engineering and Mathematics*) diperkenalkan dalam buku ini melalui tugas dan projek yang melibatkan penyelesaian masalah dalam konteks kehidupan harian. Buku ini diterapkan dengan Elemen Merentas Kurikulum (EMK) untuk mengukuhkan kemahiran dan keterampilan modal insan supaya murid-murid mampu menangani cabaran semasa dan masa hadapan.

Penekanan pada Kemahiran Abad Ke-21 secara tersirat dan tersurat dalam buku ini bermatlamat untuk melahirkan murid yang mampu bersaing di peringkat global. Elemen Kemahiran Berfikir Aras Tinggi (KBAT) ditunjukkan dengan eksplisit bagi merangsang pemikiran berstruktur dalam kalangan murid. KBAT diaplikasikan dalam proses pembelajaran melalui aktiviti atau projek. Penggunaan peta pemikiran i-THINK dan bentuk soalan aras tinggi dalam buku ini bertujuan untuk menggalakkan murid berfikir secara kritis, kreatif dan inovatif.

Ciri-ciri Istimewa dalam Buku Ini dan Fungsinya



Membekalkan pelbagai aktiviti atau tugas yang membantu murid menguasai kemahiran yang dipelajari.



Membekalkan kerja amali supaya murid dapat mengaplikasikan kemahiran dan teori yang dipelajari.



Mendedahkan kata-kata khas yang akan dipelajari oleh murid dalam bab yang berkenaan.



Membekalkan maklumat lanjut atau video tutorial untuk membantu murid memahami dan menguasai kemahiran yang dipelajari. Pautan laman sesawang dan QR code dibekalkan untuk memudahkan akses.



Memberi penerangan bahagian teks yang pernah dipelajari oleh murid.



Membekalkan maklumat tambahan untuk meluaskan pengetahuan murid semasa pembelajaran.



Memberi penerangan tambahan yang melibatkan penyelesaian masalah atau mengaplikasikan teori yang dipelajari melalui contoh aktiviti harian.



Memberi huraian tambahan kepada murid semasa pembelajaran atau semasa melakukan tugas.



Mengingatkan murid bahawa teknik-teknik pemikiran komputasional yang dipelajari sentiasa diaplikasikan semasa pembelajaran dan penyelesaian masalah.

Kemahiran Abad Ke-21

Mendedahkan ciri-ciri pembelajaran abad ke-21 kepada murid.



Penerapan elemen tambahan semasa pembelajaran secara tersurat atau tersirat.

Cabar Diri Anda

Menyediakan soalan untuk menguji tahap kefahaman murid tentang pembelajaran.



Menyediakan latihan formatif pada akhir setiap subtopik.



Memaklumkan bahawa soalan berjenis Kemahiran Berfikir Aras Tinggi (KBAT).



PETA KONSEP

Memberikan rumusan bab dalam bentuk lembaran grafik.



SENARAI SEMAK

Membolehkan murid melakukan tafsiran sendiri melalui senarai semak pembelajaran.



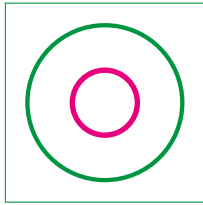
PENTAKSIRAN

Menyediakan latihan sumatif dalam bentuk objektif dan soalan berstruktur bagi menguji tahap pencapaian murid.

Panduan Penggunaan Peta Pemikiran i-THINK Semasa Pembelajaran

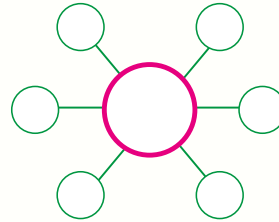
Peta Bulatan

Sesuai digunakan semasa aktiviti sumbang saran untuk menyenaraikan idea



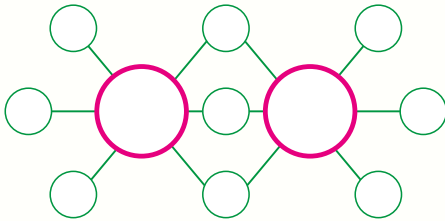
Peta Buih

Sesuai digunakan untuk menunjukkan ciri-ciri sesuatu item



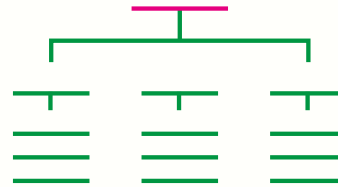
Peta Buih Berganda

Sesuai digunakan untuk membuat perbandingan dan perbezaan



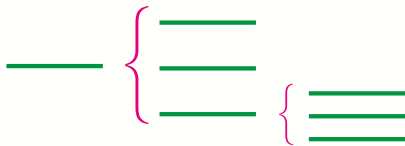
Peta Pokok

Sesuai digunakan untuk mengelaskan item



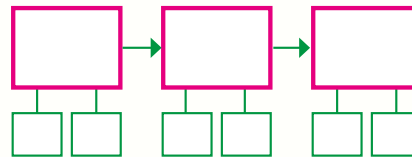
Peta Dakap

Sesuai digunakan untuk menganalisis bahagian sesuatu item



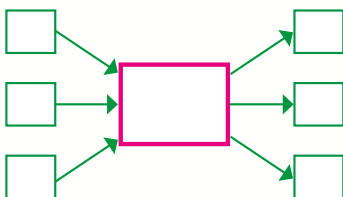
Peta Alir

Sesuai digunakan untuk membuat urutan sesuatu proses



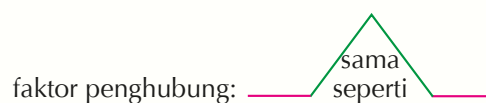
Peta Pelbagai Alir

Sesuai digunakan untuk menunjukkan punca dan kesan sesuatu proses atau situasi



Peta Titi

Sesuai digunakan untuk membuat analogi dengan faktor penghubung



BAB 1

PERWAKILAN DATA

Kebanyakan sistem komputer moden beroperasi dengan sistem pengendalian 32 bit atau 64 bit. Sistem pengendalian 32 bit bermaksud sistem komputer berupaya mengendalikan 32 bit data pada masa yang sama. Bayangkan perwakilan data 32 bit dengan menggunakan nombor perduaan, perwakilan data ini adalah panjang dan susah dibaca. Sistem nombor perlapanan dan sistem nombor perenambelasan dicipta bagi mengatasi masalah ini.

Standard Kandungan

- 1.1 Sistem Nombor Perlapanan
- 1.2 Sistem Nombor Perenambelasan

Cikgu, sistem nombor perduaan.

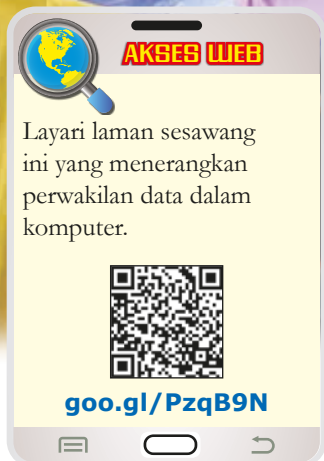
Cikgu, sistem nombor perpuluhan.

Apakah sistem nombor yang telah anda pelajari?

Terdapat sistem nombor lain yang terlibat dalam perwakilan data. Apakah sistem nombor itu?

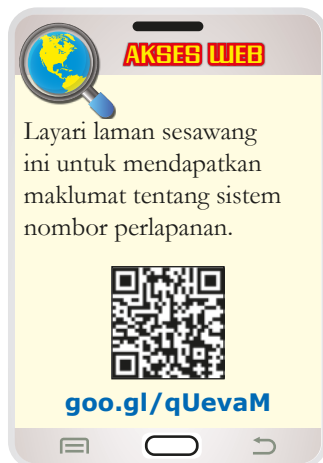
Kata Kunci

- Aksara
- Bit
- Bait
- Kod ASCII
- Sistem nombor perlapanan
- Sistem nombor perenambelasan





1.1 Sistem Nombor Perlapanan



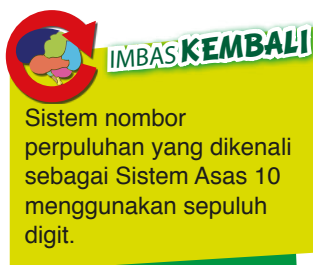
Sistem nombor perlapanan juga disebut **Sistem Asas 8** atau **sistem nombor oktal**. Ini disebabkan sistem ini hanya menggunakan lapan pilihan digit, iaitu 0, 1, 2, 3, 4, 5, 6 dan 7.

Dalam pengkomputeran, sistem nombor perlapanan adalah penting kerana merupakan salah satu sistem nombor yang digunakan untuk mewakili nombor perduaan yang bertindak sebagai bahasa mesin. Sistem nombor perlapanan digunakan kerana dapat saling bertukar kepada sistem nombor perduaan dan sebaliknya dengan mudah. Dengan ini, nombor perduaan yang panjang dapat diwakili oleh nombor perlapanan yang lebih pendek dan kemas.

1.1.1 Perbezaan antara nombor perlapanan dan nombor perpuluhan

Standard Pembelajaran

Murid boleh:
1.1.1 Membezakan nombor perlapanan dan nombor perpuluhan.



Sistem nombor perpuluhan

Sistem nombor perpuluhan atau disebut **sistem nombor desimal** menggunakan digit-digit 0, 1, 2, 3, 4, 5, 6, 7, 8 dan 9 untuk mewakili sebarang nombor. Sistem nombor perpuluhan digunakan secara meluas dalam kehidupan seharian kita. Contohnya dalam bidang perbankan, kita mesti mengira wang dengan menggunakan sistem nombor perpuluhan. Oleh yang demikian, murid diperkenalkan sistem nombor semasa di sekolah lagi.

Sistem nombor ini juga dikenali sebagai **Sistem Asas 10** kerana terdapat sepuluh pilihan digit, iaitu bermula daripada 0 hingga 9. Sistem nombor ini ditanda dengan subskrip 10 di hujung nombor tersebut. Contohnya, nombor perpuluhan 1025 ditulis sebagai 1025_{10} .

Setiap digit dalam sistem nombor perpuluhan mempunyai nilai tempatnya yang tertentu. Nilai tempat sistem nombor perpuluhan ialah 1, 10, 100, 1 000, 10 000 dan seterusnya. Nilai sesuatu digit dikira dengan mendarabkannya dengan nilai tempatnya. Jadual 1.1 menunjukkan cara memperoleh nilai digit-digit bagi nombor perpuluhan 1025_{10} .

Jadual 1.1 Cara memperoleh nilai digit-digit bagi nombor perpuluhan 1025_{10}

10^3	10^2	10^1	10^0
1000	100	10	1
1×1000	0×100	2×10	5×1
Nilai digit 1 = 1000	Nilai digit 0 = 0	Nilai digit 2 = 20	Nilai digit 5 = 5

Penambahan untuk nilai-nilai setiap digit ($1000 + 0 + 20 + 5$) akan memperoleh nombor perpuluhan 1025_{10} .

Sistem nombor perlapanan

Sistem nombor perlapanan atau disebut **sistem nombor oktal** menggunakan digit-digit 0, 1, 2, 3, 4, 5, 6 dan 7 untuk mewakili sebarang nombor. Sistem nombor perlapanan juga dikenali sebagai **Sistem Asas 8** kerana terdapat lapan pilihan digit bermula daripada 0 hingga 7. Apabila kita membilang secara perlapanan, selepas 7 adalah 10 kerana sistem perlapanan tidak mempunyai digit 8.

Nombor dalam sistem nombor perlapanan ditanda dengan subskrip 8 pada hujung nombor tersebut. Contohnya, nombor perlapanan 2001 akan ditulis sebagai 2001_8 .

Setiap digit dalam sistem nombor perlapanan mempunyai nilai tempatnya yang tertentu. Nilai tempat sistem nombor perlapanan ialah 1, 8, 64, 512 dan seterusnya. Nilai sesuatu digit dikira dengan mendarabkannya dengan nilai tempatnya. Jadual 1.2 menunjukkan cara memperoleh nilai digit-digit bagi nombor perlapanan 2001_8 .

Jadual 1.2 Cara memperoleh nilai digit-digit bagi nombor perlapanan 2001_8

8^3	8^2	8^1	8^0
512	64	8	1
2×512	0×64	0×8	1×1
Nilai digit 2 = 1024	Nilai digit 0 = 0	Nilai digit 0 = 0	Nilai digit 1 = 1

Penambahan untuk nilai-nilai setiap digit ($1024 + 0 + 0 + 1$) akan memperoleh nombor perpuluhan 1025_{10} . Maka, nombor perlapanan 2001_8 bersamaan dengan nombor perpuluhan 1025_{10} .



Fakta Segera

Pengiraan dalam sistem perlapanan wujud dalam kalangan kaum Yuki yang merupakan orang asli Amerika. Kaum Yuki mengira dengan menggunakan lapan ruang di antara jari bagi kedua-dua belah tangan.

(Dipetik dan disesuaikan dari: goo.gl/efgxvc)



Jadual 1.3 Perbezaan nombor perpuluhan dan nombor perlapanan

Sistem nombor perpuluhan	Sistem nombor perlapanan
Dikenali sebagai Sistem Asas 10 .	Dikenali sebagai Sistem Asas 8 .
Juga dikenal sebagai sistem nombor desimal .	Juga dikenal sebagai sistem nombor oktal .
Nombor ditandakan dengan subskrip 10, seperti 74_{10} .	Nombor ditandakan dengan subskrip 8, seperti 112_8 .
Mempunyai sepuluh pilihan digit bermula daripada 0 hingga 9.	Mempunyai lapan pilihan digit bermula daripada 0 hingga 7.
Nilai tempat: 1, 10, 100, 1 000 dan seterusnya.	Nilai tempat: 1, 8, 64, 512 dan seterusnya.



Aktiviti 1.1

Individu

Membilang secara perlapanan.

Nombor dalam sistem nombor perlapanan dibilang daripada 0 hingga 7. Berdasarkan corak pembilangan nombor dalam sistem nombor perpuluhan, senaraikan 20 nombor perlapanan yang seterusnya.



Teknik
Pengecaman
Corak



Aktiviti 1.2

Kumpulan

Membandingkan antara sistem nombor perpuluhan dan sistem nombor perlapanan.

- Lakukan sumbang saran bersama-sama ahli kumpulan anda tentang perkara-perkara berikut:
 - Pilihan digit yang digunakan dalam nombor perpuluhan dan nombor perlapanan.
 - Cara mendapatkan nilai nombor perpuluhan dan nombor perlapanan.
 - Kegunaan sistem nombor perpuluhan dan sistem nombor perlapanan.
- Catatkan hasil sumbang saran kumpulan anda dalam jadual perbandingan antara sistem nombor perpuluhan dan sistem nombor perlapanan.

1.1.2 (ii) Penukaran nombor perpuluhan kepada nombor perlapangan

Semua sistem komputer dan peralatan elektronik menggunakan sistem nombor perduaan untuk menyimpan data. Sistem nombor perlapangan boleh digunakan untuk mewakili nombor perduaan yang panjang. Oleh yang demikian, pemahaman tentang kaedah penukaran nombor perpuluhan kepada nombor perlapangan dan sebaliknya adalah penting.

Kaedah yang paling lazim digunakan untuk menukarkan nombor perpuluhan kepada nombor perlapangan ialah **kaedah bahagi dengan 8 dan gunakan bakinya**.

Dalam kaedah ini, nombor perpuluhan dibahagi dengan 8. Catatkan hasil bahagi dan bakinya. Hasil bahagi pertama dibahagi dengan 8 sekali lagi dan hasil bahagi serta bakinya dicatatkan. Proses ini diulang sehingga hasil bahaginya tidak boleh dibahagi lagi dan setiap bakinya dicatatkan. Nilai nombor perlapangan diambil berdasarkan bakinya secara menyongsang.

Standard Pembelajaran

Murid boleh:

- 1.1.2** Menukar nombor:
(i) perpuluhan kepada nombor perlapangan.



Contoh 1.1

Penukaran nombor perpuluhan 2266_{10} kepada nombor perlapangan.

Langkah-langkah bagi menggunakan kaedah bahagi dengan 8 dan gunakan bakinya ditunjukkan di bawah:

Langkah 1: Suatu nombor perpuluhan diberi.

Langkah 2: Jika nombor adalah lebih besar daripada 7, maka bahagikan dengan 8.

Langkah 3: Hasil bahagi pertama dan bakinya dicatatkan.

Langkah 4: Langkah 2 dan 3 diulangi sehingga hasil bahaginya tidak boleh dibahagi lagi.

Langkah 5: Tulis semua hasil baki yang tercatat dari bawah ke atas.

2266	÷	8	=	283	baki	2
283	÷	8	=	35	baki	3
35	÷	8	=	4	baki	3
4	÷	8	=	0	baki	4

Baki dibaca dari bawah ke atas

Baki yang dibaca dari bawah ke atas akan menghasilkan nombor perlapangan 4332_8 .

Semakan:

Baris pertama	8^3	8^2	8^1	8^0
Baris kedua	512	64	8	1
Baris ketiga	4	3	3	2

Jumlahkan hasil darab baris kedua dan baris ketiga dalam jadual di atas.

$$(4 \times 512) + (3 \times 64) + (3 \times 8) + (2 \times 1) = 2048 + 192 + 24 + 2 = \mathbf{2266}$$

Maka, nombor perlapanan bagi nombor perpuluhan 2266_{10} ialah $\mathbf{4332_8}$.



Aktiviti 1.3

Individu

Penukaran nombor perpuluhan kepada nombor perlapanan.

Tukarkan nombor-nombor perpuluhan yang berikut kepada nombor perlapanan dengan menggunakan kaedah bahagi dengan 8 dan gunakan bakinya. Tunjukkan cara kerja yang lengkap.

- (a) 58_{10}
(b) 128_{10}

- (c) 379_{10}
(d) 1008_{10}

1.1.2 (ii) Penukaran nombor perlapanan kepada nombor perpuluhan

Standard Pembelajaran

Murid boleh:

- 1.1.2** Menukar nombor:
(ii) perlapanan kepada nombor perpuluhan.

Penukaran nombor perlapanan kepada nombor perpuluhan dapat dilakukan dengan mendarabkan digit dalam nombor perlapanan dengan nilai tempat nombor perlapanan. Jadual 1.4 menunjukkan nilai tempat dalam nombor perlapanan.

Jadual 1.4 Nilai tempat nombor perlapanan

8^4	8^3	8^2	8^1	8^0
4 096	512	64	8	1



IMBAS KEMBALI

Kaedah yang digunakan untuk penukaran nombor perduaan kepada nombor perpuluhan adalah sama seperti kaedah penukaran nombor perlapanan kepada nombor perpuluhan.



Cabar Diri Anda

Bina satu jadual yang menunjukkan nilai tempat nombor perpuluhan. Kemudian, nyatakan satu ciri kesamaan antara jadual yang dibina dengan Jadual 1.4.



Teknik Pengitlakan

Contoh 1.2

Penukaran nombor perlapanan 326_8 kepada nombor perpuluhan.

Langkah-langkah untuk penukaran nombor perlapanan 326_8 kepada nombor perpuluhan.

Langkah 1:

Bina satu jadual nilai tempat nombor perlapanan. Bermula dari lajur kanan, isikan nombor perlapanan 326_8 dengan bermula daripada digit paling kanan ke dalam jadual yang dibina.

Langkah 2:

Dapatkan hasil darab baris kedua dan baris ketiga, iaitu digit dalam nombor perlapanan dan nilai tempatnya.

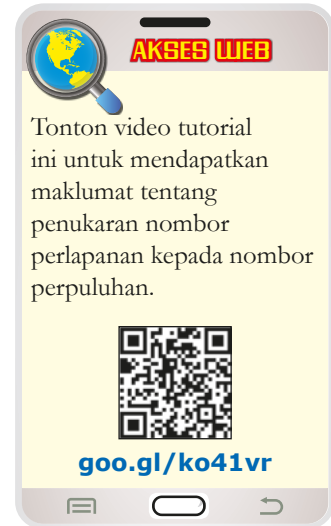
8^2	8^1	8^0
64	8	1
3	2	6
$3 \times 64 = 192$	$2 \times 8 = 16$	$6 \times 1 = 6$

Langkah 3:

Jumlahkan hasil darab yang dilakukan dalam Langkah 2.

$$192 + 16 + 6 = 214$$

Maka, nombor perpuluhan bagi nombor perlapanan 326_8 ialah 214_{10} .

**Aktiviti 1.4**

Individu

Penukaran nombor perlapanan kepada nombor perpuluhan.

Tukarkan nombor-nombor perlapanan yang berikut kepada nombor perpuluhan dengan menggunakan jadual nilai tempat perlapanan. Tunjukkan cara kerja yang lengkap.

- (a) 25_8
- (b) 226_8
- (c) 745_8
- (d) 5630_8

1.1.2 (iii) Penukaran nombor perlapanan kepada nombor perduaan

Standard Pembelajaran

Murid boleh:

- 1.1.2** Menukar nombor:
(iii) perlapanan kepada
nombor perduaan.

Nilai tempat keempat bagi nombor perduaan ialah $2^3 = 8$. Sistem nombor perlapanan diwakili dengan digit-digit 0, 1, 2, 3, 4, 5, 6 dan 7. Oleh yang demikian, tiga digit nombor perduaan bersamaan dengan satu digit nombor perlapanan. Jadual 1.5 menunjukkan digit-digit dalam nombor perlapanan dan perwakilannya dalam nombor perduaan.

Jadual 1.5 Digit-digit dalam nombor perlapanan dan perwakilannya dalam nombor perduaan

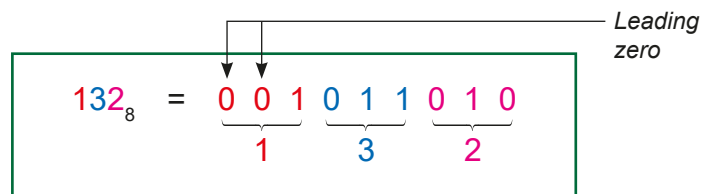
Digit-digit dalam nombor perlapanan	Perwakilan digit-digit ini dalam nombor perduaan
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111



Fakta Segera

Leading zero ialah nombor sifar yang ditambahkan pada hadapan nombor perduaan bagi tujuan pengiraan atau penukaran nombor. Semasa penulisan suatu nombor perduaan, leading zero tidak ditulis kerana ia adalah tidak penting semasa pembacaan. Contohnya, 001 akan dibaca sebagai 1.

Rajah 1.1 mengilustrasikan cara nombor perlapanan dapat diwakili oleh nombor perduaan. Penukaran dilakukan dengan memecahkan digit-digit dalam nombor perlapanan kemudian menukarkan tiap-tiap digit kepada nombor perduaan yang diwakilinya. Akhir sekali, kesemua nombor perduaan yang diperoleh dicantumkan.



Setiap tiga digit nombor perduaan (dari kanan) mewakili satu digit nombor perlapanan

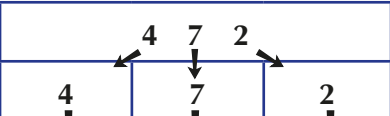
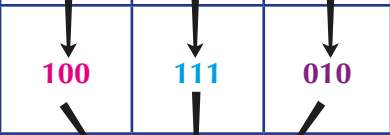
Rajah 1.1 Cara penukaran nombor perlapanan 132_8 kepada nombor perduaan

Maka, nombor perduaan bagi nombor perlapanan 132_8 ialah 1011010_2 .

Contoh 1.3

Penukaran nombor perlapanan 472_8 kepada nombor perduaan.

Langkah-langkah untuk penukaran nombor perlapanan 472_8 kepada nombor perduaan.

Langkah 1	Kenal pasti dan pisahkan digit-digit dalam nombor perlapanan	
Langkah 2	Tukarkan digit-digit dalam nombor perlapanan kepada nombor perduaan (rujuk Jadual 1.5)	
Langkah 3	Cantumkan nombor-nombor perduaan yang diperoleh	

Maka, nombor perduaan bagi nombor perlapanan 472_8 ialah 100111010_2 .

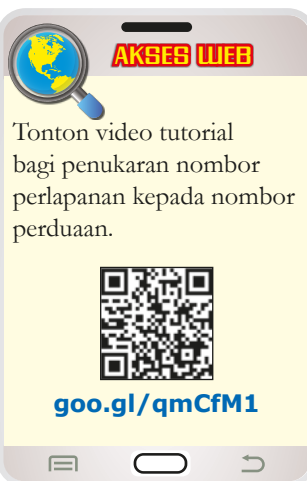
**Aktiviti 1.5**

Individu

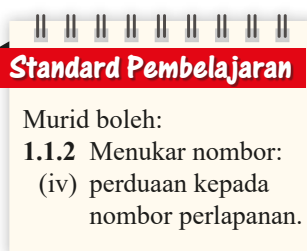
Penukaran nombor perlapanan kepada nombor perduaan.

Tukarkan nombor-nombor perlapanan yang berikut kepada nombor perduaan dengan mengikuti langkah-langkah dalam Contoh 1.3.

- (a) 537_8
- (b) 614_8
- (c) 3021_8
- (d) 4325_8

**1.1.2 (iv) Penukaran nombor perduaan kepada nombor perlapanan**

Pembelajaran tentang sistem nombor perlapanan adalah amat penting kerana sistem nombor perlapanan digunakan untuk meningkatkan nombor perduaan yang panjang. Penukaran suatu nombor perduaan kepada nombor perlapanan dapat dilakukan dengan dua kaedah seperti yang ditunjukkan berikut.





IMBAS KEMBALI

Nombor perduaan terdiri daripada dua digit sahaja, iaitu 0 dan 1. Dalam perwakilan data dalam komputer, satu digit nombor perduaan akan menggunakan 1 bit dalam ingatan unit pemprosesan pusat komputer.

(a) Kaedah pengumpulan tiga digit nombor perduaan dan tukarkan kepada nilai perlapanaan

Bahagikan digit-digit dalam nombor perduaan kepada kumpulan tiga digit bermula dari kanan. Jika perlu, satu atau dua sifar boleh ditambahkan di bahagian paling kiri untuk membentuk kumpulan tiga digit. Kemudian, setiap kumpulan tiga digit ditukarkan kepada nombor perlapanaan yang setara dengannya (sila rujuk Jadual 1.5).

Contoh 1.4

Penukaran nombor perduaan 1111001011_2 kepada nombor perlapanaan dengan menggunakan kaedah pengumpulan tiga digit nombor perduaan dan tukarkan kepada nilai perlapanaan.

Langkah-langkah untuk penukaran nombor perduaan 1111001011_2 kepada nombor perlapanaan dengan kaedah pengumpulan tiga digit nombor perduaan dan tukarkan kepada nilai perlapanaan (rujuk Jadual 1.5).

Langkah 1:

Nombor perduaan 1111001011_2 dibahagikan kepada kumpulan tiga digit.

Langkah 2:

Masukkan bagi kumpulan tiga digit perduaan dengan nombor perlapanaan yang setara dengannya.

Nombor perduaan			1	1	1	1	0	0	1	0	1	1
Kumpulan tiga digit	0	0	1	1	1	1	0	0	1	0	1	1
Nombor perlapanaan	1			7			1			3		

Maka, nombor perlapanaan bagi nombor perduaan 1111001011_2 ialah 1713_8 .

(b) Kaedah pengiraan

Bahagikan digit-digit dalam nombor perduaan kepada kumpulan tiga digit. Secara berkumpulan, setiap digit didarabkan dengan nilai tempat bagi nombor perduaan, iaitu mengikut urutan $[2^2, 2^1, 2^0]$. Dalam kumpulan masing-masing, hasil darab bagi setiap digit dan nilai tempat dijumlahkan. Hasil akhir dicantumkan menjadi nombor perlapanaan bagi nombor perduaan yang diberi.

Contoh 1.5

Penukaran nombor perduaan 1111001011_2 kepada nombor perlapanan dengan menggunakan kaedah pengiraan.

Langkah-langkah untuk penukaran nombor perduaan 1111001011_2 kepada nombor perlapanan dengan kaedah pengiraan (rujuk Jadual 1.5).

Langkah 1:

Nombor perduaan 1111001011_2 dibahagikan kepada kumpulan tiga digit.

Langkah 2:

Masukkan bagi kumpulan tiga digit perduaan dengan nombor perlapanan yang setara dengannya.

Nombor perduaan			1	1	1	1	0	0	1	0	1	1
Kumpulan tiga digit	0	0	1	1	1	1	0	0	1	0	1	1
Nilai tempat	2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0
Hasil darab	0×4	0×2	1×1	1×4	1×2	1×1	0×4	0×2	1×1	0×4	1×2	1×1
Hasil tambah	$0 + 0 + 1 = 1$			$4 + 2 + 1 = 7$			$0 + 0 + 1 = 1$			$0 + 2 + 1 = 3$		
Nombor perlapanan	1			7			1			3		

Maka, nombor perlapanan bagi nombor perduaan 1111001011_2 ialah **1713₈**.

**Cabar Diri Anda**

Nyatakan satu kekuatan dan satu kelemahan kaedah pengumpulan tiga digit nombor perduaan dan tukarkan kepada nilai perlapanan berbanding dengan kaedah pengiraan.

**Aktiviti 1.6**

Individu

Penukaran nombor perduaan kepada nombor perlapanan.

Tukarkan nombor-nombor perduaan di bawah kepada nombor perlapanan dengan menggunakan kaedah pengumpulan tiga digit nombor perduaan dan tukarkan kepada nilai perlapanan, dan menggunakan kaedah pengiraan.

- (a) 01000011_2 (c) 1011101010_2
 (b) 110111000_2 (d) 111000001100_2



1.1.3 Penukaran pengekodan ASCII kepada nombor perlapangan berdasarkan aksara yang diberi

Standard Pembelajaran

Murid boleh:

1.1.3 Menukar pengekodan ASCII kepada nombor perlapangan berdasarkan aksara yang diberi.



Fakta Segera

1 bait = 8 bit
1 bit = 1 atau 0
Bilangan kombinasi untuk 1 bait = $2^8 = 256$



IMBAS KEMBALI

Kod ASCII ialah satu set kod piawai yang paling lazim digunakan sebagai perwakilan data untuk papan kekunci komputer.

Komputer hanya boleh memproses nombor perduaan yang mempunyai dua digit, iaitu 1 dan 0 sahaja. Setiap digit 1 dan 0 dikenali sebagai bit dan merupakan unit terkecil untuk data yang diwakilkan oleh komputer.

Satu bit sebenarnya tidak mewakili apa-apa kecuali “on” dan “off”. Namun, satu bait, iaitu kombinasi 8 bit boleh digunakan untuk mewakili banyak data atau maklumat. Ini disebabkan gabungan 0 dan 1 yang berbeza dalam 1 bait boleh mewakili sehingga 256 aksara yang berbeza. Set gabungan ini dikenali sebagai **Skema Pengekodan**.

Pengekodan **ASCII** merupakan salah satu sistem pengekodan yang paling luas digunakan dalam dunia pengkomputeran. ASCII ialah singkatan untuk *American Standard Code for Information Interchange*. Kod ASCII digunakan untuk mewakili teks dalam komputer dan alatan komunikasi yang lain.

Kod utama ASCII adalah dalam bentuk nombor perpuluhan. Maka, penukaran kod ASCII nombor perpuluhan kepada nombor perlapangan adalah penting. Terdapat dua kaedah penukaran, pertama ialah menggunakan kaedah bahagi dengan 8 dan gunakan bakinya. Kaedah penukaran ini sama seperti yang anda telah pelajari dalam Subtopik 1.1.2(i).

Contoh 1.6

Penukaran aksara “B” daripada kod ASCII kepada nombor perlapangan. Kod ASCII “B” ialah 66_{10} .

Gunakan kaedah bahagi dengan 8 dan gunakan bakinya untuk penukaran.

66	÷	8	=	8	baki	2
8	÷	8	=	1	baki	0
1	÷	8	=	0	baki	1

Baki dibaca dari bawah ke atas

Baki yang dibaca dari bawah ke atas akan menghasilkan jawapan 102_8 . Maka, aksara “B” diwakili dengan 102_8 .

Kaedah kedua untuk menukarkan kod ASCII nombor perpuluhan kepada nombor perlapanan dapat dilakukan dengan merujuk kepada Lampiran A.

Contoh 1.7

Penukaran aksara “G” daripada kod ASCII nombor perpuluhan kepada nombor perlapanan. Kod ASCII “G” ialah 71_{10} .

Nombor perpuluhan	Nombor perduaan	Nombor perlapanan	Aksara
69	01000101	105	E
70	01000110	106	F
71	01000111	107	G
72	01001000	110	H
73	01001001	111	I

Rujuk kepada Lampiran A, aksara “G” dalam kod ASCII 71_{10} bersamaan 107_8 .



Aktiviti 1.7

Individu

Menukarkan pengkodan ASCII kepada nombor perlapanan dengan menggunakan kaedah pengiraan.

- Huraikan kaedah pengiraan yang digunakan untuk menukar pengkodan ASCII kepada nombor perlapanan.
- Gunakan kaedah pengiraan, kemudian tukarkan kod ASCII yang berikut kepada nombor perlapanan.

84	79	76	79	78	71
----	----	----	----	----	----

- Gunakan Lampiran A dan terjemahkan hasil nombor perlapanan yang diperoleh daripada Soalan 2 kepada aksara. Apakah perkataan yang diperoleh?

1.1.4 Hubung kait antara nombor perlapanan dengan kod ASCII dalam bahasa mesin

Bahasa mesin juga disebut sebagai kod mesin. Kod ini ialah bahasa asas yang difahami oleh komputer. Kod ini terdiri daripada urutan digit 0 dan 1 yang disebut nombor perduaan. Suatu atur cara komputer boleh ditulis dalam pelbagai bahasa pengaturcaraan seperti Scratch, C++, Java, Visual Basic dan Python. Komputer tidak boleh memahami kod-kod bahasa pengaturcaraan ini secara terus. Oleh yang demikian, kod-kod atur cara perlu diterjemah terlebih dahulu kepada bahasa mesin.

Standard Pembelajaran

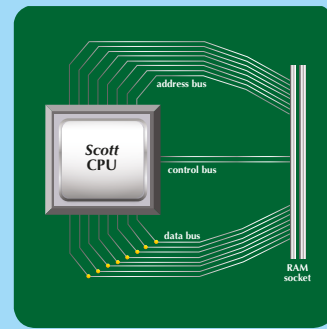
Murid boleh:
1.1.4 Menghubungkan nombor perlapanan dengan pengkodan ASCII yang digunakan dalam bahasa mesin.



Kefahaman tentang data yang dimasukkan (input) dan yang dipaparkan (output) daripada sebarang komputer adalah penting supaya manusia dapat berkomunikasi dengan komputer tanpa sebarang masalah. Pengekoden ASCII yang menggunakan nombor perduaan digunakan untuk memberitahu alatan digital hari ini tentang data yang dimasukkan (input). Kebanyakan papan kekunci komputer di Malaysia telah diipiawaikan dengan kod ASCII. Rajah 1.2 menunjukkan pemindahan satu aksara dari papan kekunci ke unit pemprosesan pusat dan dipaparkan dalam dokumen perisian pemprosesan perkataan pada monitor.



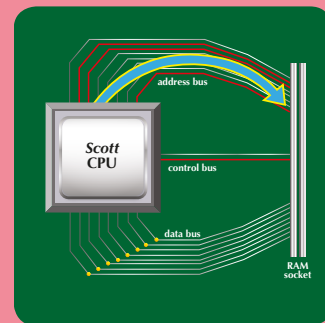
1 Apabila pengguna menekan papan kekunci (Shift + D) untuk menghasilkan aksara besar "D" (input).



2 Pada masa yang sama, isyarat elektronik 01000100 (kod ASCII nombor perduaan) untuk aksara "D" dihantar ke unit pemprosesan pusat melalui sistem bas.



4 Selepas pemprosesan, kod ASCII bagi aksara "D" ditukarkan semula menjadi aksara "D" dan dipaparkan pada skrin monitor (output).



3 Kemudian, isyarat elektronik dalam bentuk kod ASCII nombor perduaan untuk aksara "D" dihantar ke storan ingatan untuk disimpan.

Rajah 1.2 Pemindahan satu aksara dari papan kekunci ke unit pemprosesan pusat dan dipaparkan dalam dokumen perisian pemprosesan perkataan pada skrin monitor.

Rajah 1.2 menunjukkan kepentingan saling penukaran nombor perduaan (kod ASCII) dalam komputer. Walaupun untuk satu aksara, komputer perlu memproses dan menyimpan satu siri urutan nombor perduaan yang amat panjang. Oleh yang demikian, sistem nombor perlapanan digunakan untuk memudahkan proses ini.

Contoh 1.8

Penukaran perkataan TOLONG dari kod ASCII perpuluhan kepada nombor perlapanan.

Berdasarkan Lampiran A, kod ASCII dalam nombor perduaan dan nombor perlapanan bagi perkataan “TOLONG” adalah seperti yang ditunjukkan berikut.

Aksara	Kod ASCII dalam nombor perpuluhan	Kod ASCII dalam nombor perduaan	Kod ASCII dalam nombor perlapanan
T	84	01010100	124
O	79	01001111	117
L	76	01001100	114
O	79	01001111	117
N	78	01001110	116
G	71	01000111	107

Berdasarkan jadual di atas, perkataan “TOLONG” boleh ditulis dalam nombor perlapanan seperti ini: 124_8 117_8 114_8 117_8 116_8 107_8 .

Satu perkataan “TOLONG” mempunyai enam aksara. Enam aksara ini diwakili oleh 48 digit nombor perduaan dalam kod ASCII. Manakala, hanya 18 digit dalam perwakilan nombor perlapanan.



Cabar Diri Anda

Berdasarkan penerangan dalam Contoh 1.8, penyimpanan dan pembacaan yang manakah lebih efisien? Berikan pandangan anda dan terangkan.

Selain kod ASCII, terdapat kod-kod lain yang digunakan untuk mewakili bahasa lain seperti bahasa Arab, Cina, Tamil dan Jepun. Kod yang digunakan untuk bahasa tersebut adalah Unicode. Kod ini membolehkan kita menggunakan bahasa Arab, Cina, Tamil dan Jepun yang merangkumi 65536 aksara di dalam perisian pemprosesan perkataan dan perisian-perisian lain.





Aktiviti 1.8

Kumpulan

Menyediakan folio digital mengenai sistem nombor yang berkaitan perwakilan ASCII, EBCDIC dan Unicode.

- Lakukan sumbang saran dengan rakan sekumpulan anda untuk mengetahui tentang perkara berikut:
 - Pengenalan pengekodan ASCII, EBCDIC dan Unicode.
 - Hubung kait pengekodan-pengekodan ASCII, EBCDIC dan Unicode.
- Kumpulkan sumbang saran kumpulan anda dan sediakan satu folio digital dengan menggunakan perisian persembahan yang sesuai. Anda digalakkan untuk menggunakan peta pemikiran bagi perwakilan data anda.



Praktis 1.1

- Nyatakan sistem nombor yang telah anda pelajari.
- Tuliskan turutan nombor daripada 0 sehingga nombor ke-40 dalam sistem nombor perlapanan.
- Tukarkan nombor perpuluhan berikut kepada nombor perlapanan.

(a) 83_{10}	(c) 250_{10}
(b) 614_{10}	(d) 543_{10}
- Padankan jawapan yang betul.

(a) 314_8	• 646_{10}
(b) 701_8	• 449_{10}
(c) 1206_8	• 1062_{10}
(d) 2046_8	• 204_{10}
- Tandakan (✓) bagi hasil penukaran nombor perlapanan kepada nombor perdua yang betul sahaja.

(a) $356_8 = 11101110_2$	☐
(b) $172_8 = 1101010_2$	☐
(c) $4702_8 = 100111000010_2$	☐
(d) $3610_8 = 1111010000_2$	☐

6. Tukarkan nombor-nombor perduaan yang berikut kepada nombor perlapanan dengan menggunakan kaedah pengumpulan tiga digit nombor perduaan dan tukarkan kepada nombor perlapanan, dan menggunakan kaedah pengiraan.

(a) 110101001_2

(c) 11010101100_2

(b) 100000110_2

(d) 110011001000_2

7. Tukarkan kod ASCII nombor perpuluhan yang berikut kepada nombor perlapanan dengan menggunakan kaedah pengiraan.

(a) 50

(c) 99

(b) 89

(d) 119

8. Lengkapkan peta titi di bawah.

Nombor perlapanan	101_8	sama	103_8	sama	105_8
Aksara yang diwakili	"A"	seperti	"C"	seperti	

9. (a) Lengkapkan jadual berikut dengan bantuan daripada Lampiran A.

Aksara	Kod ASCII dalam nombor perduaan	Kod ASCII dalam nombor perlapanan
B		
E		
S		
T		
A		
R		
I		

- (b) Tukarkan perkataan yang berikut kepada nombor perlapanan dengan menggunakan kaedah yang ditunjukkan dalam jadual di atas.

(i) p@55word

(ii) komputer

(iii) Malaysia

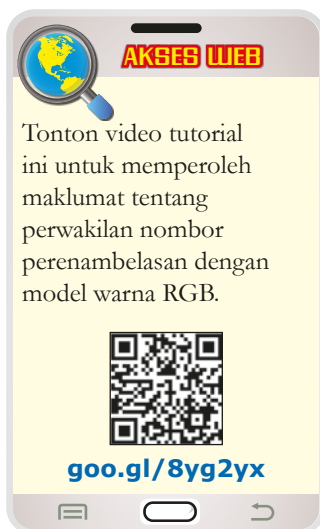


10. Terangkan kelebihan dan kekurangan sistem nombor perlapanan jika digunakan dalam perwakilan data.



1.2

Sistem Nombor Perenambelasan

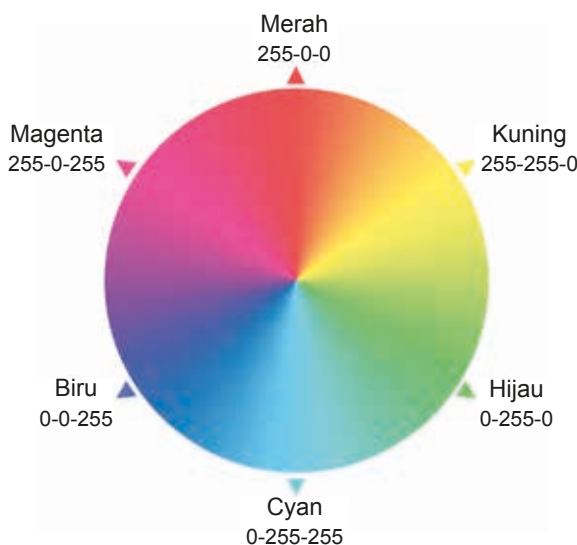


Fakta Segera

Hexadecimal numbering system setara dengan sistem nombor perenambelasan dalam bahasa Melayu.

Sistem nombor perenambelasan ialah salah satu sistem nombor yang digunakan untuk mewakili nombor perduaan yang digunakan sebagai bahasa mesin. Sistem nombor ini digunakan kerana dapat saling bertukar kepada sistem nombor perduaan dan sebaliknya dengan mudah.

Penggunaan sistem nombor perenambelasan membolehkan komputer menyimpan data dengan efisien. Salah satu kepentingan nombor perenambelasan adalah untuk mewakili warna pada alatan digital dalam model warna RGB, iaitu **Red**, **Green**, **Blue**, di mana setiap warna diwakili oleh dua nombor perenambelasan. Ini bermaksud, setiap warna mempunyai sebanyak $16^2 = 256$ varian warna. Secara keseluruhan, jumlah spektrum warna yang dapat diwakili ialah 256 merah $\times$ 256 hijau $\times$ 256 biru, iaitu melebihi 16 juta warna.



Rajah 1.3 Roda spektrum warna R-G-B

Selain itu, sistem nombor perenambelasan juga digunakan untuk pembangunan asas sistem mikropemproses. Segala mesej ralat tentang lokasi ingatan komputer ditunjukkan dengan nombor perenambelasan. Ini memudahkan pembangun sistem mengenal pasti dan membetulkan ralat tersebut.

Sistem nombor perenambelasan menggunakan sepuluh digit dan enam abjad. Oleh itu, sistem nombor perenambelasan dikenali sebagai **Sistem Asas 16**. Sepuluh digit dan enam abjad yang digunakan dalam nombor perenambelasan ialah 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

1.2.1 Perbezaan antara nombor perenambelasan dan nombor perpuluhan

Sistem nombor perenambelasan

Sistem nombor perenambelasan atau disebut **sistem nombor heksadesimal** menggunakan digit-digit 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E dan F untuk mewakili sebarang nombor. Sistem nombor perenambelasan juga dikenali sebagai **Sistem Asas 16** kerana terdapat 16 pilihan digit bermula daripada 0 hingga 9 dan A hingga F. Jadual 1.6 menunjukkan perwakilan digit A, B, C, D, E dan F semasa melakukan pengiraan.

Jadual 1.6 Perwakilan digit A, B, C, D, E dan F dalam pengiraan

Digit	Perwakilan digit semasa melakukan pengiraan
A	10
B	11
C	12
D	13
E	14
F	15

Nombor dalam sistem nombor perenambelasan ditanda dengan subskrip 16 pada hujung nombor tersebut. Contohnya nombor perenambelasan 401 akan ditulis sebagai 401_{16} .

Tiap-tiap digit dalam sistem nombor perenambelasan mempunyai nilai tempatnya tertentu. Nilai tempat sistem nombor perenambelasan ialah sa, enam belas, dua ratus lima puluh enam, dan seterusnya. Nilai sesuatu digit dikira dengan mendarabkannya dengan nilai tempatnya. Jadual 1.7 menunjukkan cara memperoleh nilai digit-digit bagi nombor perenambelasan 401_{16} .

Jadual 1.7 Cara memperoleh nilai digit-digit bagi nombor perenambelasan 401_{16}

16^2	16^1	16^0
256	16	1
4×256	0×16	1×1

Nilai digit **4**
= 1024

Nilai digit **0**
= 0

Nilai digit **1**
= 1

Penambahan untuk nilai-nilai setiap digit ($1024 + 0 + 1$) akan memperoleh nombor perpuluhan 1025_{10} . Maka, nombor perenambelasan 401_{16} bersamaan dengan nombor perpuluhan 1025_{10} .

Standard Pembelajaran

Murid boleh:
1.2.1 Membezakan nombor perenambelasan dan nombor perpuluhan.



IMBAS KEMBALI

Anda telah pelajari tentang sistem nombor perpuluhan dalam subtopik 1.1.1.





Aktiviti 1.9

Individu

Membilang secara perenambelasan.

Nombor dalam sistem nombor perenambelasan dibilang daripada 0 hingga 9 diikuti A hingga F. Berdasarkan corak pembilangan nombor dalam sistem nombor perpuluhan, senaraikan 20 nombor perenambelasan yang seterusnya.



Teknik
Pengecaman
Corak

Jadual 1.8 Perbezaan nombor perpuluhan dan nombor perenambelasan

Sistem nombor perpuluhan	Sistem nombor perenambelasan
Dikenali sebagai Sistem Asas 10 .	Dikenali sebagai Sistem Asas 16 .
Juga dikenali sebagai sistem nombor desimal .	Juga dikenali sebagai sistem nombor heksadesimal .
Nombor ditandakan dengan subskrip 10, seperti 74_{10} .	Nombor ditandakan dengan subskrip 16, seperti $4A_{16}$.
Mempunyai sepuluh pilihan digit bermula daripada 0 hingga 9.	Mempunyai enam belas pilihan digit bermula daripada 0 hingga 9 dan A hingga F.
Nilai tempat: 1, 10, 100, 1000 dan seterusnya.	Nilai tempat: 1, 16, 256, 4096 dan seterusnya.

1.2.2 (ii) Penukaran nombor perpuluhan kepada nombor perenambelasan

|| || || || || || || || || ||

Standard Pembelajaran

Murid boleh:

1.2.2 Menukar nombor

- (i) perpuluhan kepada nombor perenambelasan.

Kaedah penukaran nombor perpuluhan kepada nombor perenambelasan adalah serupa dengan kaedah yang digunakan semasa penukaran nombor perpuluhan kepada nombor perlapanan. Kaedah ini disebut **kaedah bahagi dengan 16 dan gunakan bakinya**.

Dalam kaedah ini, nombor perpuluhan dibahagi dengan 16. Kemudian hasil bahagi dan bakinya dicatatkan. Hasil bahagi pertama dibahagi dengan 16 sekali lagi dan hasil bahagi serta baki dicatatkan. Proses ini diulang sehingga hasil bahaginya tidak boleh dibahagi lagi. Nilai nombor perenambelasan diambil berdasarkan baki yang tercatat secara menyongsang.

Contoh 1.9

Penukaran nombor perpuluhan 423_{10} kepada nombor perenambelasan.

423	÷	16	=	26	baki	7
26	÷	16	=	1	baki	10
1	÷	16	=	0	baki	1

Arah membaca

Dalam nombor perenambelasan, nombor 10 tidak wujud. Nombor 10 adalah diwakili oleh digit A.

Baki yang dibaca dari bawah ke atas akan menghasilkan nombor perenambelasan $1A7_{16}$.

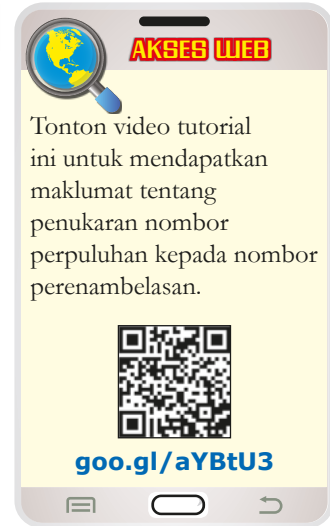
Semakan:

Baris pertama	16^2	16^1	16^0
Baris kedua	256	16	1
Baris ketiga	1	10	7

Jumlahkan hasil darab baris kedua dan baris ketiga dalam jadual di atas.

$$(1 \times 256) + (10 \times 16) + (7 \times 1) = 256 + 160 + 7 = 423$$

Maka, nombor perenambelasan bagi nombor perpuluhan 423_{10} ialah $1A7_{16}$.

**Aktiviti 1.10**

Individu

Tukarkan nombor-nombor perpuluhan yang berikut kepada nombor perenambelasan dengan menggunakan kaedah bahagi dengan 16 dan gunakan bakinya.

1. 486

486	÷	16	=		baki	
	÷	16	=		baki	
	÷	16	=		baki	

Nombor perenambelasan =

2. 678

678	÷	16	=		baki	
	÷	16	=		baki	
	÷	16	=		baki	

Nombor perenambelasan =



Aktiviti 1.11

Kumpulan

Membandingkan kaedah penukaran nombor perpuluhan kepada nombor perlapanan dan nombor perpuluhan kepada nombor perenambelasan.

1. Rujuk Subtopik 1.1.2 (i) tentang penukaran nombor perpuluhan kepada nombor perlapanan dan perhatikan kaedah yang digunakan.
2. Bandingkan kaedah yang telah anda perhatikan dalam langkah 1 dengan kaedah penukaran nombor perpuluhan kepada nombor perenambelasan.
3. Sediakan satu lembaran grafik yang menunjukkan persamaan dan perbezaan dengan menggunakan satu contoh nombor perpuluhan yang sesuai.
4. Apakah kesimpulan yang dapat anda buat tentang perbandingan ini?
5. Adakah kaedah yang sama dapat digunakan untuk penukaran nombor perpuluhan kepada sistem nombor yang lain seperti nombor pertigaan dan nombor perempatan?
6. Berikan justifikasi kepada jawapan anda.

1.2.2 (ii) Penukaran nombor perenambelasan kepada nombor perpuluhan

|| || || || || || || || ||

Standard Pembelajaran

Murid boleh:

- 1.2.2** Menukar nombor
(ii) perenambelasan
kepada nombor
perpuluhan.

Nombor perenambelasan boleh ditukar kepada nombor perpuluhan dengan mendarabkan digit dan abjad dalam nombor perenambelasan dengan nilai tempat nombor perenambelasan. Nilai tempat nombor perenambelasan ditunjukkan dalam Jadual 1.9.

Jadual 1.9 Nilai tempat nombor perenambelasan

16^3	16^2	16^1	16^0
4 096	256	16	1

Contoh 1.10

Penukaran nombor perenambelasan $4F3_{16}$ kepada nombor perpuluhan.

Abjad F dalam nombor perenambelasan mewakili nombor 15 dalam nombor perpuluhan. Oleh itu, nilai perwakilan abjad digunakan semasa pengiraan.

16^2	16^1	16^0
256	16	1
4	15	3
$4 \times 256 = 1024$	$15 \times 16 = 240$	$3 \times 1 = 3$

Nombor perpuluhan = $1024 + 240 + 3 = 1267$

Semakan:

1267	÷	16	=	79	baki	3
79	÷	16	=	4	baki	15
4	÷	16	=	0	baki	4

Arah membaca

Nilai 15 diwakili dengan abjad F dalam nombor perenambelasan.

Baki yang dibaca dari bawah ke atas akan menghasilkan nombor perenambelasan $4F3_{16}$. Maka, nombor perpuluhan bagi nombor perenambelasan $4F3_{16}$ ialah 1267_{10} .



Aktiviti 1.12

Individu

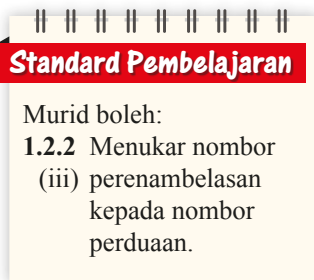
Tukarkan nombor-nombor perenambelasan yang berikut kepada nombor perpuluhan.

- (a) 432_{16}
- (b) $89E_{16}$
- (c) $A546_{16}$
- (d) $8B40A_{16}$
- (e) $CEF123_{16}$



1.2.2 (iii) Penukaran nombor perenambelasan kepada nombor perduaan

Nilai tempat kelima bagi nombor perduaan ialah $2^4 = 16$. Maka empat digit nombor perduaan bersamaan dengan satu digit nombor perenambelasan. Jadual 1.10 menunjukkan digit-digit dalam nombor perenambelasan dan perwakilannya dalam nombor perduaan (rujuk Lampiran A).



Fakta Segera

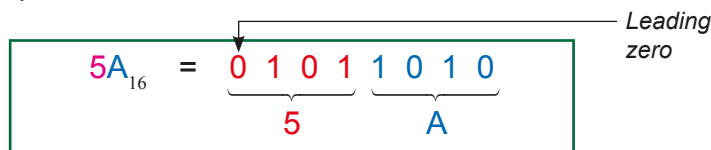
Penukaran nombor perenambelasan kepada nombor perduaan boleh diselesaikan dengan membuat penukaran nombor perenambelasan kepada nombor perpuluhan dahulu, kemudian tukar nombor perpuluhan itu kepada nombor perduaan.

Jadual 1.10 Digit-digit dalam nombor perenambelasan dan perwakilannya dalam nombor perduaan



Digit-digit dalam nombor perenambelasan	Perwakilan digit-digit ini dalam nombor perduaan
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A (mewakili 10)	1010
B (mewakili 11)	1011
C (mewakili 12)	1100
D (mewakili 13)	1101
E (mewakili 14)	1110
F (mewakili 15)	1111

Rajah 1.4 mengilustrasikan cara nombor perenambelasan dapat diwakili oleh nombor perduaan. Penukaran dilakukan dengan memecahkan digit-digit dalam nombor perenambelasan dan menukarkan setiap digit kepada nombor perduaan. Akhir sekali, kesemua nombor perduaan dicantumkan.



Setiap empat digit nombor perduaan (dari kanan) mewakili satu digit nombor perenambelasan

Rajah 1.4 Cara penukaran nombor perenambelasan $5A_{16}$ kepada nombor perduaan

Maka, nombor perduaan bagi nombor perenambelasan $5A_{16}$ ialah 1011010_2 .



Aktiviti 1.13

Pasangan

Pertandingan permainan interaktif nombor perenambelasan.

Laman sesawang ini menunjukkan satu permainan interaktif untuk menukarkan nombor perenambelasan kepada nombor perduaan. Cuba bertanding dengan rakan anda dan tentukan siapakah yang mendapat markah paling tinggi.



goo.gl/QZPWlh

Contoh 1.11

Penukaran nombor perenambelasan $9F2_{16}$ kepada nombor perduaan.

Langkah-langkah untuk penukaran nombor perenambelasan $9F2_{16}$ kepada nombor perduaan.

Langkah 1	Kenal pasti dan pisahkan digit-digit dalam nombor perenambelasan.	
Langkah 2	Tukarkan digit-digit dalam nombor perenambelasan kepada nombor perduaan (rujuk Jadual 1.10).	
Langkah 3	Cantumkan nombor-nombor perduaan yang diperoleh.	

Maka, nombor perduaan bagi nombor perenambelasan $9F2_{16}$ ialah **100111110010₂**.

**Aktiviti 1.14**

Individu

Penukaran nombor perenambelasan kepada nombor perduaan.

Tukarkan nombor-nombor perenambelasan yang berikut kepada nombor perduaan dengan mengikuti langkah-langkah dalam Contoh 1.11.

- (a) $A35_{16}$
- (b) $6D4_{16}$
- (c) $3B7F_{16}$
- (d) $8E21_{16}$

1.2.2 (iv) Penukaran nombor perduaan kepada nombor perenambelasan

Anda telah mempelajari nombor perduaan yang digunakan untuk mewakili data dalam komputer. Maka, pembelajaran tentang penukaran nombor perduaan kepada nombor perenambelasan adalah penting supaya anda mengetahui cara nombor perduaan diwakili oleh nombor perenambelasan. Penukaran nombor perduaan kepada nombor perenambelasan boleh dilakukan dengan dua kaedah seperti yang ditunjukkan berikut.

|| || || || || || || || || ||

Standard Pembelajaran

Murid boleh:

- 1.2.2** Menukar nombor
(iv) perduaan
kepada nombor
perenambelasan.



(a) Kaedah pengumpulan empat digit nombor perduaan dan tukarkan kepada nilai perenambelasan

Bahagikan digit-digit yang terdapat dalam nombor perduaan kepada kumpulan empat digit bermula dari kanan. Jika perlu, sifar boleh ditambahkan di bahagian paling kiri untuk membentuk kumpulan empat digit. Kemudian setiap kumpulan empat digit ditukarkan kepada nombor perenambelasan yang setara dengannya (rujuk Jadual 1.10).

Contoh 1/2

Penukaran nombor perduaan 111110011010_2 kepada nombor perenambelasan dengan menggunakan kaedah pengumpulan empat digit nombor perduaan dan tukarkan kepada nilai perenambelasan.

Langkah-langkah untuk penukaran nombor perduaan 111110011010_2 kepada nombor perenambelasan dengan kaedah pengumpulan empat digit nombor perduaan dan tukarkan kepada nilai perenambelasan (rujuk Jadual 1.10).

Langkah 1:

Nombor perduaan 111110011010_2 dibahagikan kepada kumpulan empat digit.

Langkah 2:

Gantikan setiap kumpulan empat digit perduaan kepada nombor perenambelasan yang setara dengannya.

Nombor perduaan	1	1	1	1	1	0	0	1	1	0	1	0
Kumpulan empat digit	1	1	1	1	1	0	0	1	1	0	1	0
Nombor perenambelasan	F				9				A			

Maka, nombor perenambelasan bagi nombor perduaan 111110011010_2 ialah $F9A_{16}$.

(b) Kaedah pengiraan

Bahagikan digit-digit dalam nombor perduaan kepada kumpulan empat digit. Secara berkumpulan, setiap digit didarabkan dengan nilai tempat bagi nombor perduaan, iaitu mengikut urutan $[2^3, 2^2, 2^1, 2^0]$. Dalam kumpulan masing-masing, hasil darab bagi setiap digit dan nilai tempat dijumlahkan. Hasil akhir dicantumkan menjadi nombor perenambelasan bagi nombor perduaan yang diberi.

Contoh 1.13

Penukaran nombor perduaan 111110011010_2 kepada nombor perenambelasan dengan menggunakan kaedah pengiraan.

Langkah-langkah untuk penukaran nombor perduaan 111110011010_2 kepada nombor perenambelasan dengan kaedah pengiraan (rujuk Jadual 1.10).

Langkah 1:

Nombor perduaan 111110011010_2 dibahagikan kepada kumpulan empat digit.

Langkah 2:

Darabkan setiap kumpulan empat digit perduaan dengan nilai tempatnya. Kemudian tambahkan hasil darab setiap kumpulan empat digit perduaan itu.

Nombor perduaan	1	1	1	1	1	0	0	1	1	0	1	0
Kumpulan empat digit	1	1	1	1	1	0	0	1	1	0	1	0
Nilai tempat	2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
Hasil darab	1×8	1×4	1×2	1×1	1×8	0×4	0×2	1×1	1×8	0×4	1×2	0×1
Hasil tambah	$8 + 4 + 2 + 1 = 15$				$8 + 0 + 0 + 1 = 9$				$8 + 0 + 2 + 0 = 10$			
Nombor perenambelasan	F				9				A			

Maka, nombor perenambelasan bagi nombor perduaan 111110011010_2 ialah **F9A**₁₆.



Aktiviti 1.15

Individu

Penukaran nombor perduaan kepada nombor perenambelasan.

Tukarkan nombor-nombor perduaan yang berikut kepada nombor perenambelasan dengan menggunakan dua kaedah, iaitu kaedah pengumpulan empat digit nombor perduaan dan tukarkan kepada nilai perenambelasan, dan menggunakan kaedah pengiraan.

- 101000000111_2
- 110101100001_2
- 1000011101010010_2
- 1110000011001101_2



Aktiviti 1.16

Kumpulan

Membandingkan kaedah penukaran nombor perduaan kepada nombor perlapanan dan nombor perenambelasan.

1. Rujuk Subtopik 1.1.2 (iv) tentang penukaran nombor perduaan kepada nombor perlapanan dan perhatikan kaedah-kaedah yang digunakan.
2. Bandingkan kaedah penukaran nombor perduaan kepada nombor perlapanan dan nombor perenambelasan.
3. Dengan menggunakan satu contoh nombor perduaan yang sesuai, sediakan satu lembaran grafik yang menunjukkan persamaan dan perbezaan.
4. Apakah kesimpulan yang dapat anda buat tentang perbandingan ini?
5. Adakah kaedah yang sama dapat digunakan untuk penukaran nombor perduaan kepada sistem nombor yang lain seperti Sistem Asas 4 dan Sistem Asas 32?
6. Berikan justifikasi bagi jawapan anda.

1.2.3 Penukaran pengekodan ASCII kepada nombor perenambelasan berdasarkan aksara yang diberi

|| || || || || || || || ||

Standard Pembelajaran

Murid boleh:
1.2.3 Menukar pengekodan ASCII kepada nombor perenambelasan berdasarkan aksara yang diberi.

Pengekodan utama ASCII adalah dalam bentuk nombor perpuluhan, maka aksara yang diwakili dengan kod ASCII nombor perpuluhan perlu ditukarkan kepada nombor perenambelasan untuk perwakilan. Kod ASCII nombor perpuluhan dapat ditukarkan kepada nombor perenambelasan dengan mudah berdasarkan dua kaedah.

Kaedah pertama ialah melibatkan pengiraan dengan menggunakan kaedah bahagi dengan 16 dan gunakan bakinya. Kaedah ini sama seperti apa yang anda belajar dalam Subtopik 1.2.2(i).

Contoh 1.14

Penukaran aksara "a" daripada kod ASCII kepada nombor perenambelasan. Kod ASCII "a" ialah 97.

Gunakan kaedah bahagi dengan 16 dan gunakan bakinya untuk penukaran.

97	÷	16	=	6	baki	1
6	÷	16	=	0	baki	6

Baki dibaca dari bawah ke atas

Baki yang dibaca dari bawah ke atas akan menghasilkan jawapan **61₁₆**.

Maka, aksara "a" dari kod ASCII diwakili dengan **61₁₆**.

Kaedah kedua dilakukan dengan merujuk kepada Lampiran A yang menunjukkan perwakilan kod ASCII dalam nombor perpuluhan dan nombor perenambelasan dan mencari nombor perenambelasan yang setara dengannya.

Contoh 1.15

Penukaran aksara “D” daripada kod ASCII kepada nombor perenambelasan.

Nombor perpuluhan	Nombor perduaan	Nombor perenambelasan	Aksara
66	01000010	42	B
67	01000011	43	C
68	01000100	44	D
69	01000101	45	E

Berdasarkan Lampiran A, kod ASCII “D” diwakili dengan 44_{16} .



Aktiviti 1.17

Individu

Menukar pengkodan ASCII kepada nombor perenambelasan dengan menggunakan kaedah pengiraan.

- Huraikan kaedah pengiraan yang digunakan untuk menukar pengkodan ASCII kepada nombor perenambelasan.
- Gunakan kaedah pengiraan, kemudian tukarkan kod ASCII yang berikut kepada nombor perenambelasan.

82	65	74	73	78
----	----	----	----	----

- Gunakan Lampiran A dan terjemahkan hasil nombor perenambelasan yang diperoleh daripada Soalan 2 kepada aksara. Apakah perkataan yang diperoleh?



Aktiviti 1.18

Individu

Penukaran kod ASCII yang mewakili aksara.

- Berdasarkan Lampiran A, tukarkan aksara-aksara yang berikut kepada kod ASCII.
 - P
 - Y
 - @
 - =
- Kemudian tukar setiap kod ASCII kepada nombor perenambelasan dengan menggunakan kaedah pengiraan.

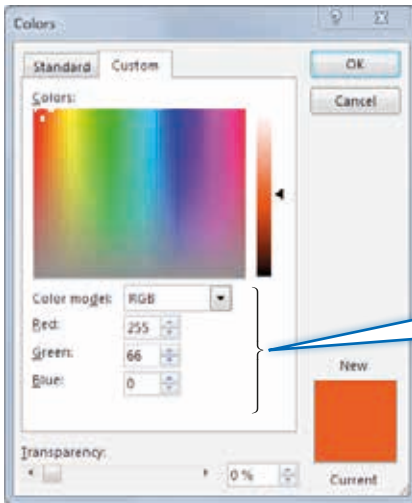


Aktiviti 1.19

Individu

Meneka warna berdasarkan nombor perenambelasan.

Nombor perenambelasan digunakan untuk mewakili warna pada laman sesawang dan perisian penyunting imej dengan menggunakan format #RRGGBB, di mana RR = merah, GG = hijau, BB = biru. Simbol # menunjukkan nombor-nombor ini telah ditulis dalam format nombor perenambelasan. Sistem ini menggunakan dua digit nombor perenambelasan untuk setiap warna, contohnya #FF4200. Rajah di bawah menunjukkan paparan yang diperolehi daripada perisian penyunting kata apabila ingin menukar warna teks.



- Perisian ini menggunakan model warna RGB
- Julat nombor ialah 0–255, iaitu setiap warna mempunyai sebanyak 256 varian warna.
- Menggunakan nombor perpuluhan untuk mewakili nombor perenambelasan kerana pengguna, iaitu manusia lebih memahami sistem nombor ini.

Layari laman sesawang <http://yizzle.com/whatthehex/> untuk permainan interaktif meneka warna berdasarkan nombor perenambelasan yang diberi.

1.2.4 Hubung kait nombor perenambelasan dengan kod ASCII dalam bahasa mesin

Standard Pembelajaran

Murid boleh:

1.2.4 Menghubungkan nombor perenambelasan dengan pengkodan ASCII yang digunakan dalam bahasa mesin.

Komputer hanya memahami satu bahasa, iaitu bahasa mesin. Bahasa mesin hanya terdiri daripada nilai 1 dan 0 sahaja. Jika manusia perlu menulis atur cara dalam bahasa mesin untuk difahami komputer, proses ini adalah sangat rumit dan tidak efisien. Maka satu bahasa yang dapat menterjemah bahasa mesin yang boleh difahami oleh manusia diperlukan. Pengkodan ASCII adalah bahasa penterjemahan.

Setiap perkataan dibentuk berdasarkan pelbagai simbol atau aksara. Apabila sesuatu butang ditekan pada papan kekunci, satu rentetan nombor yang mewakili sesuatu simbol atau aksara akan dijanakan. Nombor tersebut dikenali sebagai kod aksara.

Kod ASCII merupakan salah satu sistem pengekodan yang membolehkan manusia berinteraksi dengan sebuah komputer. Sistem piawai ASCII membolehkan komponen-komponen komputer yang dihasilkan oleh pelbagai pengeluar dapat beroperasi dengan betul dalam sesebuah komputer. Sistem ini juga membolehkan jenis-jenis peralatan pemprosesan data yang berlainan berfungsi dengan sempurna. Maka, peralatan-peralatan dapat berkomunikasi di antara satu sama lain dengan sempurna.

Kegunaan utama nombor perenambelasan ialah mewakili nombor perduaan dalam bentuk yang lebih pendek dan lebih mudah dibaca oleh manusia. Nombor perenambelasan dipilih kerana proses penukaran daripada nombor perduaan yang lebih mudah, iaitu nombor-nombor perduaan dapat dipisahkan kepada kumpulan-kumpulan empat digit dan setiap kumpulan boleh diwakilkan dalam nombor perenambelasan. Misalnya, nombor perduaan 1001111000001010_2 amat susah dibaca dan diingati. Maka, dengan menukarkannya kepada nombor perenambelasan, ia akan jadi $9E0A_{16}$. Ini akan menjadi lebih senang dibaca dan diingati.

Fakta Segera

Mikropemproses ialah pemproses yang terkandung dalam satu atau beberapa cip yang merupakan bahan separa pengalir. Cip ialah komputer terkecil. Mikropemproses yang pertama diperkenalkan oleh Intel pada tahun 1978 dengan kelajuan 8 MHz.

Fakta Segera

BASIC dan Bahasa peringkat tinggi yang lain selalu memaparkan nombor dalam bentuk perpuluhan. Bahasa himpunan pula biasa memaparkan nombor perenambelasan seperti kandungan lokasi ingatan.

Contoh 1.16

Penukaran perkataan BAKTI dari kod ASCII nombor perduaan kepada nombor perenambelasan.

Berdasarkan Lampiran A, kod ASCII dalam nombor perduaan dan nombor perenambelasan bagi perkataan “BAKTI” adalah seperti berikut.

Aksara	Kod ASCII dalam nombor perduaan	Kod ASCII dalam nombor perenambelasan
B	01000010	42
A	01000001	41
K	01001011	4B
T	01010100	54
I	01001001	49

Satu perkataan “BAKTI” mengandungi 40 digit dalam nombor perduaan. Manakala, hanya 10 digit dalam perwakilan nombor perenambelasan. Perwakilan bagi BAKTI diletakkan secara mendatar.

Contohnya, $BAKTI = 42_{16} 41_{16} 4B_{16} 54_{16} 49_{16}$



Cabar Diri Anda

Berdasarkan pemerhatian anda, bandingkan Contoh 1.8 dan Contoh 1.16. Penyimpanan dan pembacaan sistem nombor yang manakah lebih efisien? Berikan pandangan anda.



Aktiviti 1.20

Individu

Hasilkan satu ayat daripada bahasa mesin kepada nombor perenambelasan.

1. Berikut ialah satu ayat yang diberi.

Saya suka belajar ASK.

2. Berdasarkan Lampiran A, kenal pasti nombor perduaan bagi perwakilan aksara-aksara dalam ayat di atas. Kemudian kenal pasti nombor perenambelasan yang setara dengannya.
3. Catatkan semua hasil anda dalam jadual dengan kemas.
4. Apakah pemerhatian kepada hasil dapatan anda?
5. Apakah kesimpulan yang dapat anda buat berdasarkan pemerhatian di atas?

TIP

Jangan lupa bahawa ruang kosong ialah satu aksara dan diwakili oleh satu nombor perduaan.



Aktiviti 1.21

Kumpulan

Sediakan folio digital mengenai sistem nombor perenambelasan dan perwakilannya dalam pengekodan ASCII.

1. Lakukan sumbang saran dengan rakan sekumpulan anda untuk perkara-perkara yang tersenarai berikut:
 - (a) Kepentingan nombor perenambelasan
 - (b) Hubungan kait pelbagai pengekodan ASCII dengan nombor perenambelasan
 - (c) Kepentingan perhubungan pelbagai pengekodan ASCII dan nombor perenambelasan
2. Kemaskan hasil sumbang saran kumpulan anda dan sediakan satu folio digital dengan perisian persembahan yang sesuai. Anda digalakkan untuk menggunakan peta pemikiran untuk perwakilan data anda.



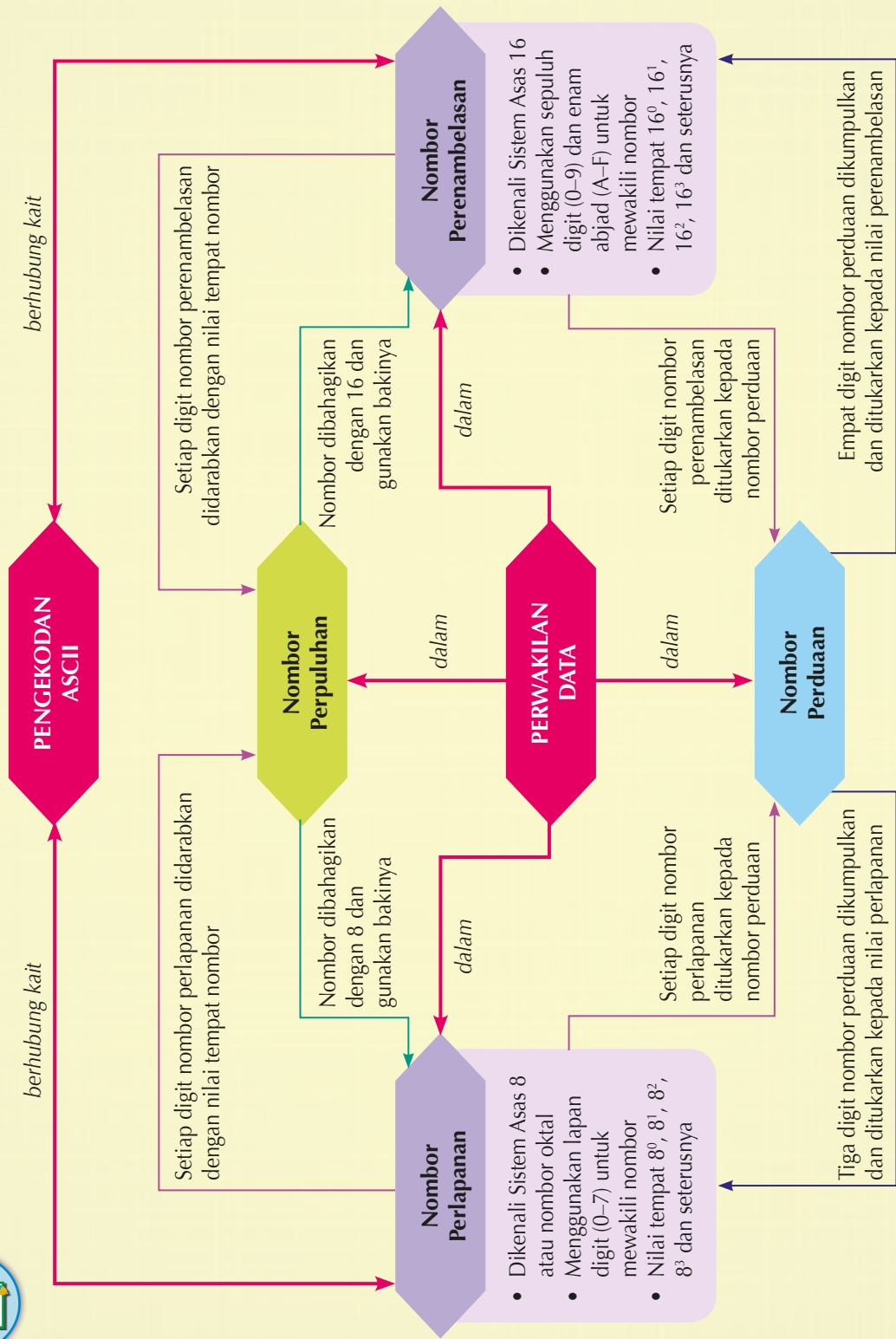
Praktis 1.2

- Tukarkan nombor perpuluhan berikut kepada nombor perenambelasan.
 - 415_{10}
 - 562_{10}
 - 5316_{10}
 - 2714_{10}
- Tukarkan nombor perenambelasan berikut kepada nombor perpuluhan.
 - 156_{16}
 - $4A7_{16}$
 - 1234_{16}
 - $F10F_{16}$
- Tukarkan nombor perenambelasan berikut kepada nombor perduaan.
 - $A72_{16}$
 - $3F8_{16}$
 - $2B15_{16}$
 - $40E6_{16}$
- Tukarkan nombor perduaan berikut kepada nombor perenambelasan dengan menggunakan kaedah pengumpulan empat digit nombor perduaan dan tukarkan kepada nilai perenambelasan, dan menggunakan kaedah pengiraan.
 - 11010100_2
 - 10000011_2
 - 11010101100_2
 - 110000111000_2
- Lengkapkan peta titi di bawah.

Nombor perenambelasan	$4A_{16}$	sama	$4D_{16}$	sama	50_{16}
Aksara yang diwakili	"J"	seperti	"M"	seperti	
- Tunjukkan langkah menukar kod ASCII 01100010 kepada nombor perenambelasan melalui kaedah pengiraan dan tuliskan aksara tersebut.
- Gunakan kod ASCII untuk mewakili aksara-aksara dalam perkataan-perkataan yang berikut. Kemudian tukarkan kepada nombor perenambelasan berdasarkan Lampiran A.
 - Hello!
 - BESTARI
 - Bunga Raya
 - Hari Gawai
- Dengan merujuk Lampiran A, terjemahkan nombor perenambelasan di bawah kepada aksara yang diwakilinya.

54_{16} 61_{16} 68_{16} $6E_{16}$ 69_{16} 61_{16} 68_{16} 21_{16}

PETA KONSEP BAB 1





SENARAI SEMAK BAB 1

Pada akhir bab ini, saya dapat...

menghuraikan nombor perpuluhan.	
menerangkan nilai tempat sistem perpuluhan.	
menghuraikan nombor perlawanan.	
menerangkan nilai tempat sistem perlawanan.	
membezakan nombor perlawanan dan nombor perpuluhan.	
menukarkan nombor perpuluhan kepada nombor perlawanan.	
menukarkan nombor perlawanan kepada nombor perpuluhan.	
menukarkan nombor perlawanan kepada nombor perduaan.	
menukarkan nombor perduaan kepada nombor perlawanan.	
menukarkan pengekodan ASCII kepada nombor perlawanan berdasarkan aksara yang diberi.	
menghubungkan nombor perlawanan dengan pengekodan ASCII yang digunakan dalam bahasa mesin.	
menghuraikan nombor perenambelasan.	
menerangkan nilai tempat sistem perenambelasan.	
membezakan nombor perenambelasan dan nombor perpuluhan.	
menukarkan nombor perpuluhan kepada nombor perenambelasan.	
menukarkan nombor perenambelasan kepada nombor perpuluhan.	
menukarkan nombor perenambelasan kepada nombor perduaan.	
menukarkan nombor perduaan kepada nombor perenambelasan.	
menukarkan pengekodan ASCII kepada nombor perenambelasan berdasarkan aksara yang diberi.	
menghubungkan nombor perenambelasan dengan pengekodan ASCII yang digunakan dalam bahasa mesin.	



PENTAKSIRAN BAB 1

Soalan Objektif

1. Sistem nombor perlapangan juga dikenali sebagai sistem nombor _____.
A perduaan
B perpuluhan
C heksa
D oktal
2. Sistem nombor perenambelasan juga dikenali sebagai sistem _____.
A desimal
B binari
C asas 16
D oktal
3. Apakah nombor perlapangan bagi nombor perpuluhan 179_{10} ?
A 263_8
B 236_8
C 326_8
D 632_8
4. Apakah nombor perenambelasan bagi nombor perduaan 1110110001011_2 ?
A $1C6B_{16}$
B $1C8B_{16}$
C $1D8B_{16}$
D $1D9B_{16}$
5. Apakah nombor perpuluhan bagi nombor perlapangan 1234_8 ?
A 667_{10}
B 668_{10}
C 669_{10}
D 670_{10}
6. Apakah nombor perenambelasan bagi nombor perpuluhan 678_{10} ?
A $2A6_{16}$
B $2B6_{16}$
C $2C6_{16}$
D $2D6_{16}$
7. Berikut ialah nombor perpuluhan yang telah ditukar daripada nombor perenambelasan dengan betul, **kecuali**
A $473_{16}=1139_{10}$
B $356_{16}=854_{10}$
C $514_{16}=1200_{10}$
D $289_{16}=649_{10}$
8. Nombor perenambelasan $6A_{16}$ mewakili aksara "j" dan $6B$ mewakili aksara "k". Apakah nombor perenambelasan bagi aksara "o"?
A $6C_{16}$ C $6E_{16}$
B $6D_{16}$ D $6F_{16}$
9. Nombor perlapangan 60 mewakili nombor "0" dan 65_8 mewakili nombor "5". Apakah nombor yang diwakili oleh 71_8 ?
A 6 C 8
B 7 D 9
10. Apakah perwakilan nombor perlapangan bagi perkataan "Bapa"?
A $102_8 141_8 160_8 141_8$
B $142_8 141_8 160_8 141_8$
C $102_8 100_8 160_8 100_8$
D $142_8 100_8 160_8 100_8$
11. Antara nombor-nombor perenambelasan yang berikut, yang manakah mewakili perkataan "Aminah"?
A $41_{16} 6D_{16} 60_{16} 6E_{16} 60_{16} 68_{16}$
B $41_{16} 6D_{16} 69_{16} 6E_{16} 61_{16} 68_{16}$
C $61_{16} 4D_{16} 49_{16} 4E_{16} 61_{16} 68_{16}$
D $61_{16} 4D_{16} 69_{16} 4E_{16} 60_{16} 68_{16}$
12. Kod ASCII 65 mewakili aksara "A". Apakah aksara yang diwakili oleh kod ASCII 71?
A E C G
B F D H

Soalan Berstruktur

1. Dengan menggunakan contoh yang sesuai, terangkan maksud nombor perlapangan dan nombor perenambelasan.
2. (a) Tukarkan nombor perpuluhan berikut kepada nombor perlapangan dan nombor perenambelasan.
 - (i) 184_{10}
 - (ii) 234_{10}
 - (iii) 4516_{10}
 - (iv) 3827_{10}
 (b) Cadangkan **satu** corak kesamaan semasa penukaran nombor perpuluhan yang dilakukan di (a).
3. (a) Tukarkan nombor perlapangan berikut kepada nombor perduaan.
 - (i) 321_8
 - (ii) 476_8
 - (iii) 1245_8
 - (iv) 5343_8
 (b) Tukarkan nombor perenambelasan berikut kepada nombor perduaan.
 - (i) $A23_{16}$
 - (ii) $DE9_{16}$
 - (iii) $374F_{16}$
 - (iv) $A56B_{16}$
 (c) Bandingkan kaedah penukaran yang digunakan dalam (a) dan (b). Nyatakan satu persamaan dan satu perbezaan bagi kaedah penukaran tersebut.
4. (a) Tukarkan nombor perduaan berikut kepada nombor perlapangan dan nombor perenambelasan dengan kaedah perwakilan digit-digit dalam nombor perduaan.
 - (i) 111101111010_2
 - (ii) 100001011101_2
 - (iii) 1100001100101001_2
 - (iv) 100101100011110_2
 (b) Cadangkan **satu** corak kesamaan semasa penukaran nombor perduaan yang dilakukan di (a).
5. (a) Apakah yang dimaksudkan dengan kod ASCII?
 (b) Nyatakan hubungan kait antara nombor perduaan, kod ASCII dan bahasa mesin.
6. (a) Apakah kepentingan sistem nombor perlapangan dan nombor perenambelasan dalam perwakilan data?
 (b) Bagaimanakah pembelajaran tentang sistem nombor perlapangan dan sistem nombor perenambelasan membantu perwakilan data dalam komputer?
7. Dengan merujuk Lampiran A, terjemahkan nombor-nombor perenambelasan yang berikut kepada perkataan yang diwakilinya.
 - (a) 41_{16} $6E_{16}$ 64_{16} 61_{16} 20_{16} 74_{16} 65_{16} $6C_{16}$ 61_{16} 68_{16} 20_{16} 62_{16} 65_{16} 72_{16} $6A_{16}$ 61_{16} 79_{16} 61_{16} 21_{16}
 - (b) 53_{16} 61_{16} 79_{16} 61_{16} 20_{16} 73_{16} 75_{16} $6B_{16}$ 61_{16} 20_{16} 73_{16} 75_{16} 62_{16} $6A_{16}$ 65_{16} $6B_{16}$ 20_{16} 41_{16} 53_{16} $4B_{16}$ $2E_{16}$

Dalam kehidupan seharian, kita selalu menyaksikan dan mengalami banyak situasi yang melibatkan pilihan dan ulangan. Perhatikan situasi-situasi dalam gambar foto yang ditunjukkan. Setiap satunya dapat dihubungkan dengan algoritma dan struktur kawalan yang akan anda pelajari dalam bab ini. Berdasarkan gambar-gambar foto yang ditunjukkan, terangkan algoritma yang terlibat. Bolehkah anda memberi contoh-contoh lain dalam kehidupan seharian yang melibatkan langkah ulangan dan pilihan?

Standard Kandungan**2.1 Pembangunan Algoritma**

Dalam perlumbaan kereta profesional bertaraf dunia, Formula One (F1), para peserta perlu memandu di litar berdasarkan pusingan yang telah ditetapkan.

Lampu isyarat yang terdapat di jalan-jalan raya akan sentiasa berulang menunjukkan lampu merah, kuning dan hijau untuk mengurus trafik. Perhatikan juga butang melintas jalan yang terdapat pada tiang lampu isyarat. Jika butang itu ditekan, ulangan lampu isyarat akan terganggu bagi membenarkan pejalan kaki melintas jalan.



Kata Kunci

- Algoritma
- Carta alir
- Output
- Pseudokod
- Ralat
- Semakan meja
- Struktur kawalan pilihan bersarang
- Struktur kawalan ulangan
- Ujian syarat



AKSES WEB

Pembangunan algoritma yang kompleks bagi sebuah kereta membolehkannya bergerak dengan sendiri. Tonton video ini untuk menyaksikan pergerakan kereta tersebut.



goo.gl/iPejHW

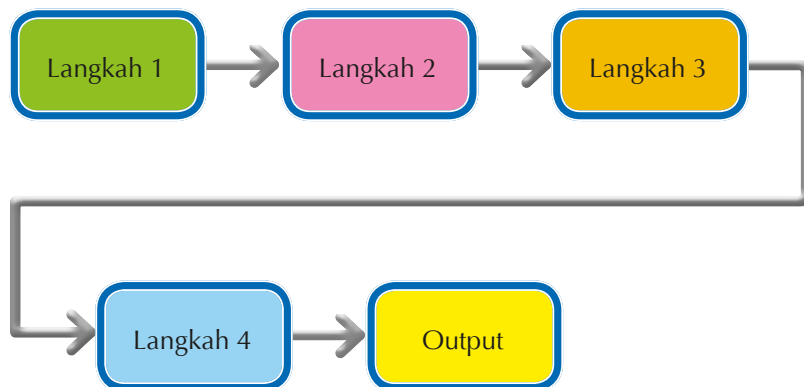


2.1 Pembangunan Algoritma

Algoritma ialah salah satu teknik utama dalam pemikiran komputasional.



Algoritma ialah satu siri langkah untuk menyelesaikan sesuatu masalah atau melengkapkan sesuatu tugas. Oleh itu, algoritma terdiri daripada satu siri prosedur langkah demi langkah yang tersusun untuk menghasilkan output yang diperlukan seperti yang ditunjukkan dalam Rajah 2.1.



Rajah 2.1 Algoritma ialah prosedur langkah demi langkah yang tersusun untuk menghasilkan output

Dalam pengkomputeran, seorang pengatur cara perlu membangunkan algoritma untuk mengarahkan komputer menyelesaikan suatu tugas. Langkah ini dilakukan pada peringkat awal sebelum pembangunan atur cara yang menggunakan bahasa pengaturcaraan. Dalam proses ini, pengatur cara tidak perlu mengetahui tentang cara komputer akan melaksanakan arahan-arahan yang diberi.

Semasa pembangunan algoritma, aspek-aspek penting bagi masalah atau tugas yang perlu diselesaikan harus dikenal pasti dan dipertimbangkan untuk menghasilkan output yang dikehendaki.



Lebih daripada satu teknik pemikiran komputasional digunakan semasa penyelesaian masalah



Fakta Segera

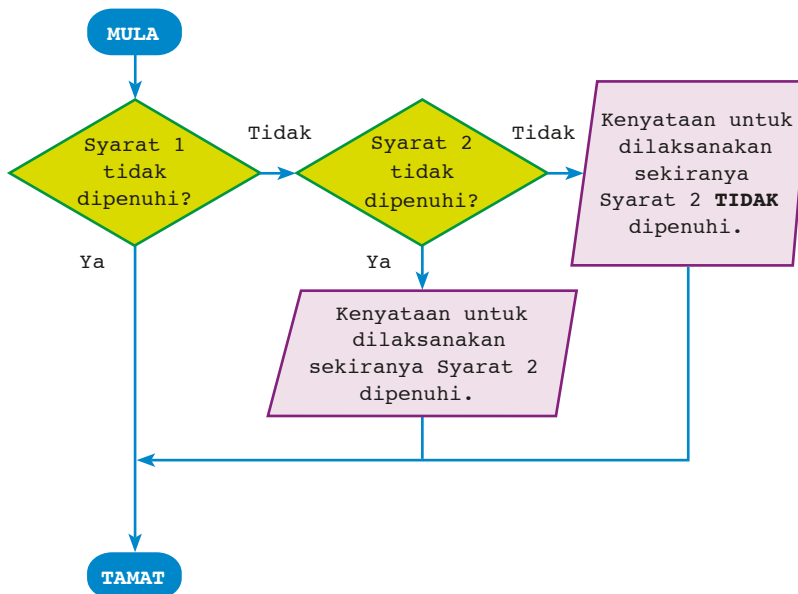
Contoh aktiviti dalam kehidupan seharian kita yang mempunyai algoritma:

- Peraturan yang wajib diikuti semasa menggunakan jalan raya
- Membakar kek
- Menanam pokok

Kesemua ini ialah algoritma atau peraturan yang harus diikuti untuk menyelesaikan sesuatu masalah dan menghasilkan output yang diinginkan.

2.1.1 (i) Struktur kawalan pilihan bersarang

Struktur kawalan pilihan bersarang terdiri daripada satu struktur kawalan yang terbenam dalam satu struktur kawalan yang lain. Satu struktur kawalan pilihan bersarang berlaku apabila kejadian satu struktur kawalan diikuti dengan kejadian satu struktur kawalan yang lain dengan serta-merta seperti dalam Rajah 2.2.



Rajah 2.2 Struktur kawalan pilihan bersarang

Dalam satu struktur kawalan pilihan bersarang, struktur kawalan pertama akan menyebabkan struktur kawalan kedua yang terbenam bermula. Dalam keadaan ini, struktur kawalan kedua akan dilaksanakan sehingga tamat dahulu baharu diikuti dengan pelaksanaan struktur kawalan pertama sehingga tamat. Rajah 2.3 menunjukkan pseudokod dan Rajah 2.4 menunjukkan carta alir bagi algoritma mencari kek lapis Sarawak.

```
Mula
Cari kedai kek
Masuk ke kedai kek
Jika ada kek lapis Sarawak
    Semak corak kek
    Jika menarik
        Beli kek lapis Sarawak
    Jika Tidak
        Keluar dari kedai kek
Jika Tidak
    Keluar dari kedai kek
Tamat
```

Rajah 2.3 Pseudokod bagi algoritma mencari kek lapis Sarawak

Standard Pembelajaran

Murid boleh:

2.1.1 Menulis pseudokod dan melukis carta alir menggunakan

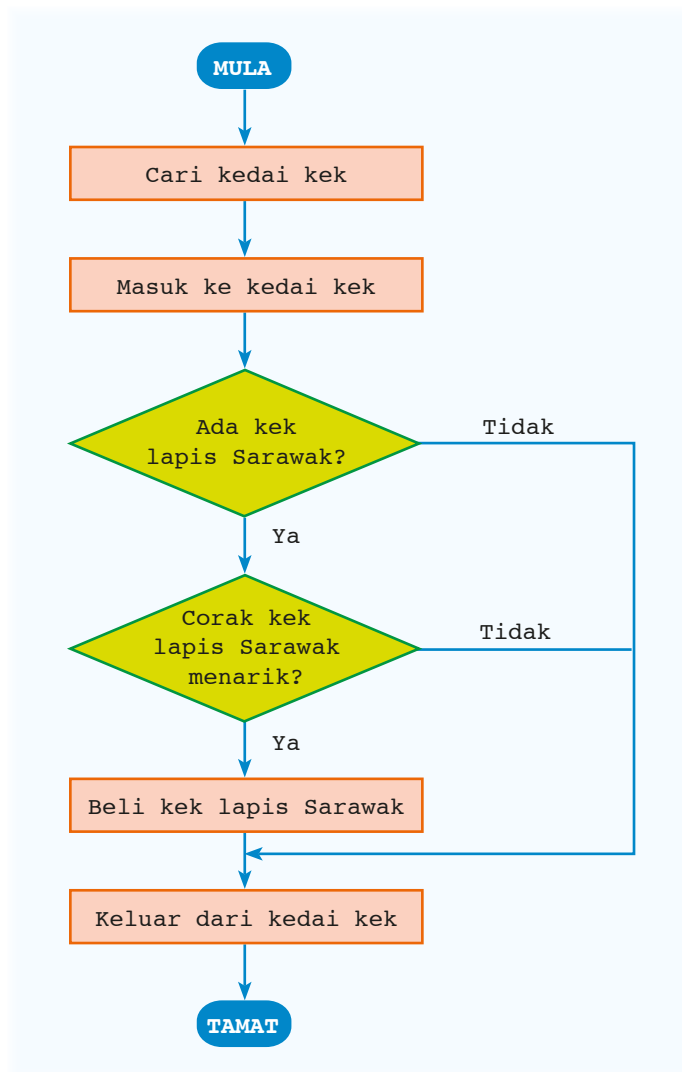
- (i) struktur kawalan pilihan bersarang dalam penyelesaian masalah.



Semasa di Tingkatan 1, jenis-jenis struktur kawalan yang telah anda pelajari ialah:

1. struktur kawalan pilihan tunggal
2. struktur kawalan dwipilihan
3. struktur kawalan pelbagai pilihan
4. struktur kawalan ulangan





Rajah 2.4 Carta alir bagi algoritma mencari kek lapis Sarawak

Struktur kawalan yang pertama mesti dipenuhi supaya struktur kawalan kedua dapat dilaksanakan. Berdasarkan Rajah 2.4 terdapat dua keputusan yang perlu dilakukan, iaitu:

- ① Ada kek lapis Sarawak?
- ② Corak kek lapis Sarawak menarik?

Rajah 2.4 menunjukkan satu struktur kawalan pilihan bersarang yang mempunyai dua struktur kawalan dwipilihan.



Contoh 2.1 menunjukkan contoh yang paling lazim digunakan untuk menerangkan struktur kawalan pilihan bersarang.

Contoh 2.1 Penentuan sama ada integer ialah positif, negatif atau sifar.

Katakan integer x perlu diuji.

Jika $x < 0$, maka integer mempunyai tanda $-$

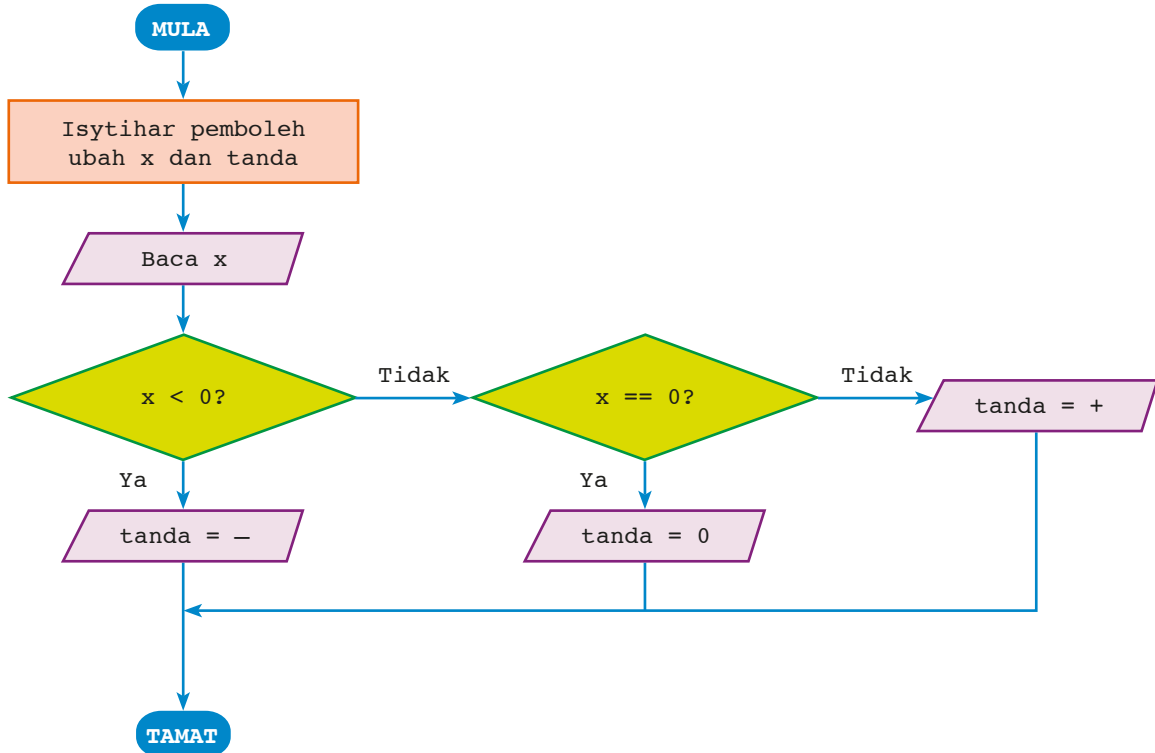
Jika $x > 0$, maka integer mempunyai tanda $+$

Jika $x = 0$, maka integer ialah 0

Menulis pseudokod

1. **Mula**
2. **Isytihar** pemboleh ubah x dan tanda
3. **Baca** x
 Jika $(x < 0)$
 tanda = -
 Jika tidak
 Jika $(x == 0)$
 tanda = 0
 Jika tidak
 tanda = +
4. **Tamat**

Melukis carta alir





Aktiviti 2.1

Pasangan

Menulis pseudokod dan melukis carta alir bagi satu struktur pilihan bersarang.

Anda ialah ketua kepada sekumpulan jurujual sebuah syarikat pemasaran. Pada setiap akhir bulan, anda dikehendaki mengira komisen bagi setiap jurujual berdasarkan jumlah jualan bulanan mereka dengan merujuk jadual di bawah.

Jumlah jualan sebulan (RM)	Komisen (%)
Kurang daripada 400	7.0
400 dan ke atas tetapi kurang daripada 750	10.0
750 dan ke atas tetapi kurang daripada 1000	12.5
1000 dan ke atas	16.0

Algoritma bagi masalah di atas adalah seperti berikut:

1. Jika jualan kurang daripada RM400, beri komisen 7.0%.
2. Jika jualan ialah RM400 atau lebih tetapi kurang daripada RM750, beri komisen 10.0%.
3. Jika jualan ialah RM750 atau lebih tetapi kurang daripada RM1000, beri komisen 12.5%.
4. Jika jualan ialah RM1000 dan ke atas, beri komisen 16.0%

Tulis pseudokod dan lukis carta alir untuk mewakili algoritma di atas.

Menyelesaikan masalah menggunakan struktur kawalan pilihan bersarang

Sekolah Menengah Kebangsaan Bandar Sri Purnama ingin mempraktikkan dasar pembelian tanpa tunai di kantin dan kedai koperasi sekolah. Sistem yang digunakan disebut **Point of Sale (POS)**. Semua transaksi berlaku pada skrin sentuh atau skrin komputer yang disebut sebagai terminal bersama kad bayaran.

Warga SMK Bandar Sri Purnama perlu menggunakan kad bayaran elektronik untuk setiap transaksi. Setiap kali kad bayaran ini dimasukkan ke dalam sistem, pengesahan untuk dua syarat dilakukan. Pertama, tempoh sah laku kad akan disemak. Kedua, nilai minimum kad, iaitu sekurang-kurangnya RM5.00 akan disemak. Apabila kedua-dua syarat itu dipenuhi, pengesahan berjaya dan kad boleh digunakan untuk transaksi. Jika salah satu syarat tidak dipenuhi, pengesahan tidak berjaya. Mesej untuk pengesahan yang tidak berjaya akan dipaparkan dan transaksi diberhentikan.



Langkah 1 Penyiasatan

Proses-proses yang terlibat.

- Mengenal pasti kad masih belum melebihi tempoh sah laku.
- Mengenal pasti kad mempunyai nilai minimum sebanyak RM5.00.
- Melaksanakan transaksi jika kedua-dua syarat pertama dan kedua dipenuhi.
- Membatalkan transaksi jika satu syarat tidak dipenuhi.

Langkah 2 Mendraf algoritma dan melakukan semakan semula

Proses-proses dan keputusan-keputusan dimurnikan untuk menghasilkan draf algoritma.

- 1 Terminal membaca kad.
- 2 Terminal mengesahkan kad masih sah digunakan.
- 3 Jika kad sah digunakan, terminal akan memeriksa nilai minimum.
- 4 Jika kad mempunyai nilai bersamaan atau lebih daripada nilai minimum, terminal akan memapar mesej “Jalankan urusan niaga” pada skrin.
- 5 Jika kad mempunyai nilai bersamaan atau kurang daripada nilai minimum, terminal akan memapar mesej “Tambah nilai” pada skrin.
- 6 Jika tidak terminal akan memaparkan mesej “Kad ditolak” pada skrin.

Langkah 3 Memurnikan langkah-langkah algoritma

1. Baca tempoh sah laku kad.
2. Jika kad melebihi tempoh sah laku, mesej “Kad ditolak” dipaparkan.
3. Jika kad belum melebihi tempoh sah laku, periksa nilai kad.
4. Jika nilai kad melebihi nilai RM5.00, transaksi boleh dimulakan.
5. Jika nilai kad kurang daripada RM5.00, mesej “Tambah nilai” dipaparkan.

Rajah 2.5 Algoritma menguji kad bayaran elektronik



Semakan meja (*desk check*) ialah satu teknik semakan kualiti yang digunakan untuk semak logik dalam sesuatu algoritma. Semakan ini dilakukan secara manual dengan menggunakan pen dan kertas sahaja. Anda telah mempelajari teknik ini semasa Tingkatan 1.



Menulis pseudokod

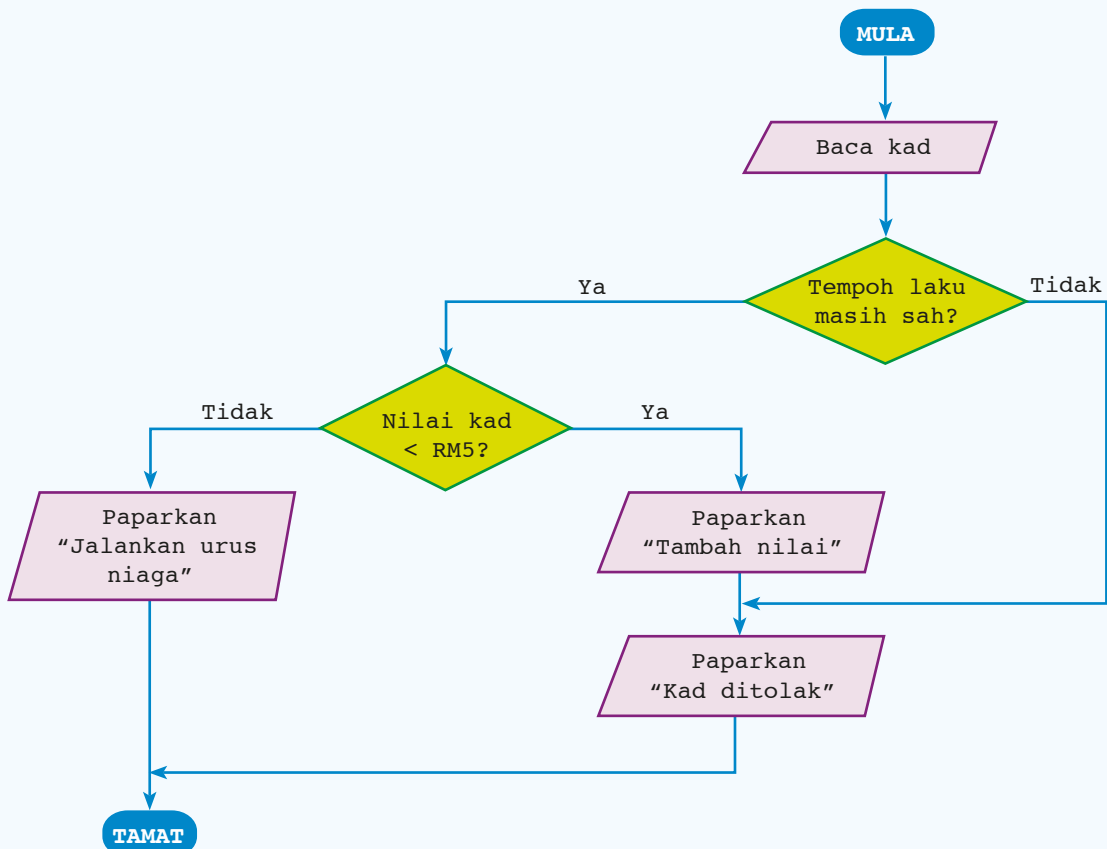
1. **Mula**
2. **Baca kad**
3. **Jika** kad sah digunakan
 Periksa nilai minimum kad
 Jika nilai minimum mencukupi
 Paparkan "Jalankan urusan niaga"
 Jika tidak
 Paparkan "Tambah nilai"
 Tamat jika
 Jika tidak
 Paparkan "Kad ditolak"
4. **Tamat jika**
5. **Tamat**

TIP

Apabila menggunakan struktur kawalan pilihan bersarang, kenyataan **Jika** mesti diakhiri dengan **Tamat jika** di tempat yang betul. Kalau tidak, akan timbul ralat dalam skrip.

Rajah 2.6 Pseudokod bagi menguji kad bayaran elektronik

Melukis carta alir



Rajah 2.7 Carta alir bagi menguji kad bayaran elektronik



Aktiviti 2.2

Kumpulan

Membangunkan algoritma untuk menyelesaikan masalah penggunaan sistem biometrik di sekolah.

Sekolah anda ingin melaksanakan sistem biometrik, iaitu imbasan cap ibu jari untuk merekodkan kedatangan harian murid dan guru ke sekolah. Pihak sekolah telah pun selesai mengumpulkan semua imbasan cap ibu jari kanan guru dan murid serta dimuatkan ke dalam pangkalan data. Terdapat dua keadaan yang perlu dikenal pasti untuk menjayakan sistem biometrik ini, iaitu:

1. Memastikan imbasan cap ibu jari adalah sah dan berada dalam pangkalan data.
2. Memastikan sistem ini menyokong murid OKU yang tiada tangan kanan, jari kanan atau ibu jari kanan.

Bangunkan satu algoritma dalam perwakilan pseudokod dan carta alir untuk menyelesaikan masalah penggunaan sistem biometrik di sekolah.

2.1.1 (ii) Struktur kawalan ulangan

Struktur kawalan ulangan terdiri daripada satu set langkah atau arahan yang sentiasa berulang sehingga suatu syarat dipenuhi. Set langkah yang berulang disebut gelung (*loop*) dan setiap ulangan gelung disebut lelaran (*iteration*).

Banyak aktiviti harian yang melibatkan ulangan contohnya tarian dan lampu isyarat. Gambar foto 2.1 menunjukkan tarian sumazau yang merupakan tarian tradisi suku kaum Kadazan Dusun untuk meraikan Tadau Kaamatan yang bermaksud Pesta Menuai.



Gambar foto 2.1 Tarian sumazau

Standard Pembelajaran

Murid boleh:

- 2.1.1** Menulis pseudokod dan melukis carta alir menggunakan
- (ii) struktur kawalan ulangan (*for*, *while*) dalam penyelesaian masalah.



Rajah 2.8 menunjukkan contoh algoritma bagi tarian sumazau yang menunjukkan langkah-langkah yang berulang.

1. Mulakan muzik.
2. Berdiri dalam satu barisan.
3. Enjut-enjut kaki kanan tanpa henti.
4. Enjut-enjut kaki kiri tanpa henti.
5. Ayun kedua-dua belah tangan ke hadapan dan ke belakang.
6. **Ulang langkah 2 hingga 5 sebanyak lima kali.**
7. Enjut-enjut sambil menari dengan pasangan.
8. Angkat kedua-dua belah tangan ke paras bahu.
9. Ayun kedua-dua pergelangan tangan seperti sayap burung sedang terbang.
10. **Ulang langkah 7 hingga 9 sebanyak 10 kali.**
11. Enjut-enjut ke kedudukan asal.
12. **Ulang langkah 7 hingga 11 sehingga muzik tamat.**

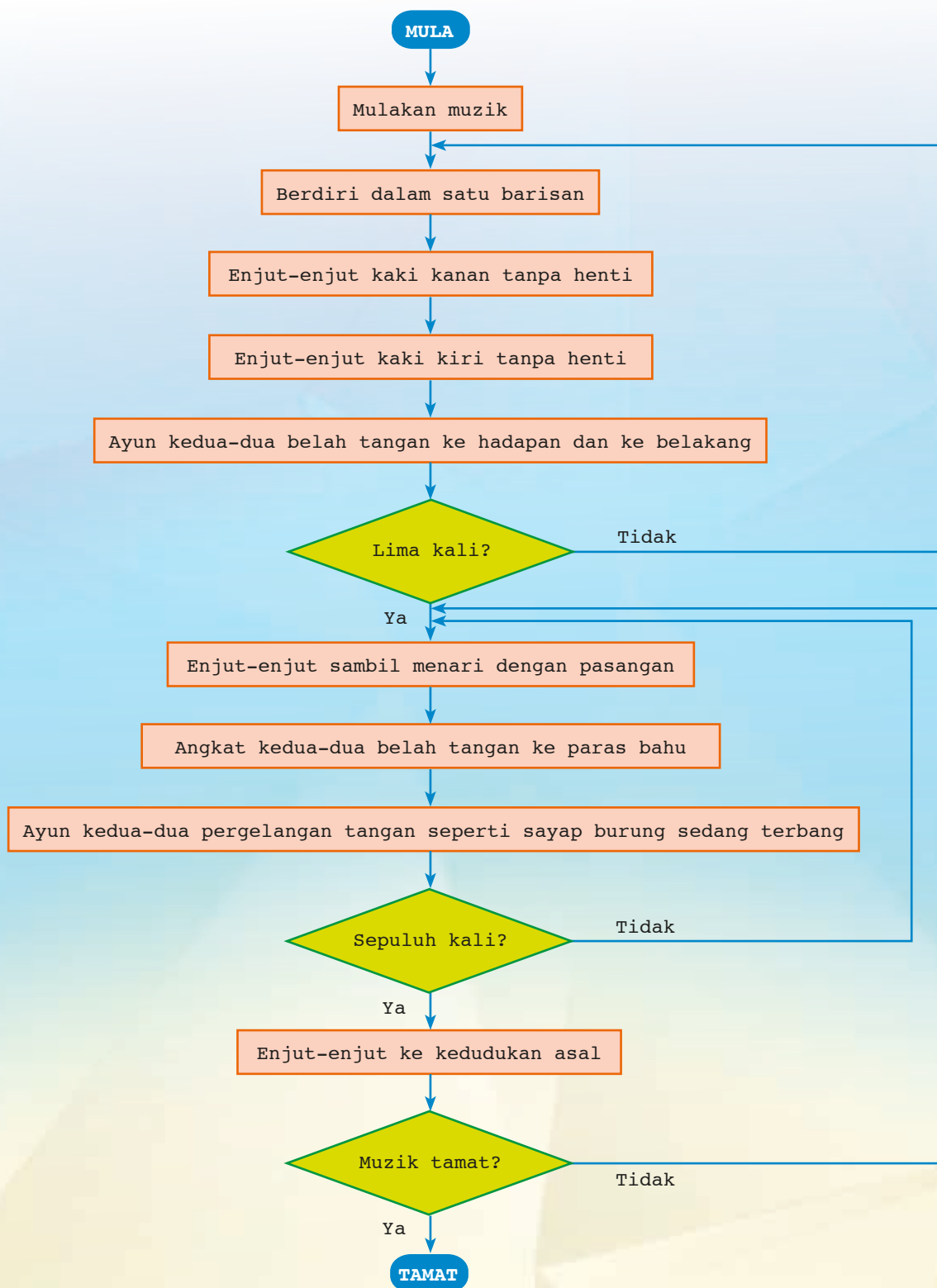
Langkah-langkah nombor 6, 10 dan 12 memberitahu ulangan bagi jujukan langkah yang berlainan

Rajah 2.8 Algoritma bagi tarian sumazau

Rajah 2.9 menunjukkan pseudokod dan Rajah 2.10 menunjukkan carta alir bagi algoritma tarian sumazau.

1. **Mula**
2. **Mulakan muzik**
3. **Berdiri** dalam satu barisan.
4. **Enjut-enjut** kaki kanan tanpa henti.
5. **Enjut-enjut** kaki kiri tanpa henti.
6. **Ayun** kedua-dua belah tangan ke hadapan dan ke belakang.
7. **Ulang** Langkah 3 hingga 6 sebanyak lima kali?
 Jika ya, teruskan Langkah 8 hingga 10.
 Jika tidak, ulang Langkah 3 hingga 6.
8. **Enjut-enjut** sambil menari dengan pasangan.
9. **Angkat** kedua-dua belah tangan ke paras bahu.
10. **Ayun** kedua-dua pergelangan tangan seperti sayap burung sedang terbang.
11. **Ulang** Langkah 8 hingga 10 sebanyak sepuluh kali?
 Jika ya, enjut-enjut ke kedudukan asal.
 Jika tidak, ulang Langkah 8 hingga 10.
12. **Muzik tamat?**
 Jika ya, teruskan Langkah 13.
 Jika tidak, ulang Langkah 8 hingga 12.
13. **Tamat**

Rajah 2.9 Pseudokod bagi tarian sumazau

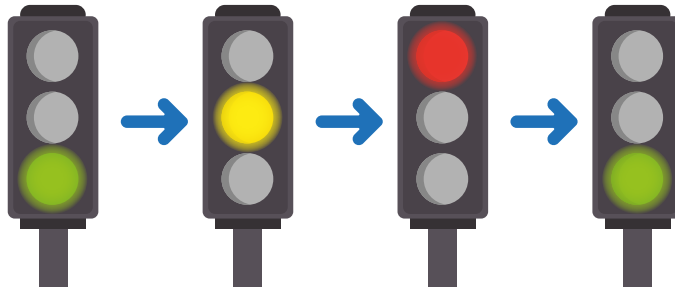


Rajah 2.10 Carta alir bagi tarian sumazau

Selain tarian, sistem lampu isyarat di persimpangan jalan juga mempunyai tindakan yang berulang-ulang.

Contoh 2.2

Tindakan yang berulang-ulang dalam sistem lampu isyarat di persimpangan jalan raya.

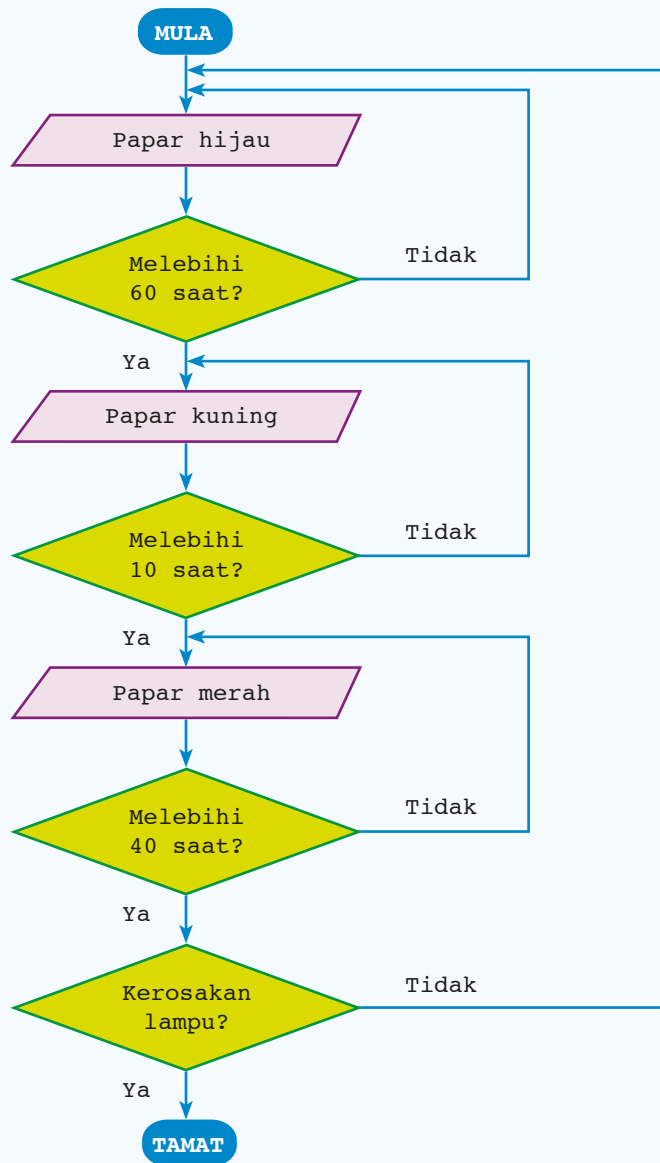


Dalam sistem lampu isyarat, nyalaan lampu hijau memberi isyarat bergerak untuk tempoh masa yang ditetapkan. Kemudian, nyalaan bertukar kepada lampu kuning untuk seketika. Seterusnya, nyalaan bertukar kepada lampu merah untuk tempoh yang ditetapkan. Nyalaan lampu akan berulang semula dengan lampu hijau dan seterusnya untuk sepanjang masa selagi sistem lampu isyarat masih berfungsi.

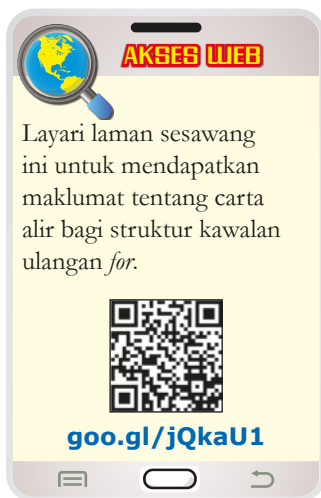
Menulis pseudokod

1. **Mula**
2. **Papar** hijau
3. **Paparan** melebihi 60 saat?
 Jika ya, papar kuning
 Jika tidak, ulang Langkah 2
4. **Papar** kuning
5. **Paparan** melebihi 10 saat?
 Jika ya, papar merah
 Jika tidak, ulang Langkah 4
6. **Papar** merah
7. **Paparan** melebihi 40 saat?
 Jika ya, periksa lampu
 Jika tidak, ulang Langkah 6
8. **Periksa** kerosakan lampu
 Jika ya, tamat
 Jika tidak, ulang Langkah 2
9. **Tamat**

Melukis carta alir



Dalam kedua-dua contoh tarian sumazau dan sistem lampu isyarat, terdapat langkah-langkah yang perlu diulang. Dalam pengaturcaraan, terdapat juga segmen-segmen kod yang berulang. Sintaks kod atur cara yang membolehkan set langkah berulang adalah penting supaya segmen kod yang serupa tidak perlu ditulis berulang kali. Ini memberi faedah kepada pengatur cara kerana jika terdapat penambahbaikan pada kod yang diulang, mereka hanya perlu lakukan pada satu segmen kod sahaja. Dua struktur kawalan ulangan yang sering digunakan ialah **ulangan for** dan **ulangan while**.



Struktur kawalan ulangan *for*

Struktur kawalan ulangan *for* melakukan ulangan bagi sesuatu pernyataan untuk bilangan kali yang tertentu. Bilangan ulangan dikawal oleh suatu nombor yang disebut pembilang yang akan berubah pada setiap ulangan. Setiap struktur kawalan ulangan *for* mempunyai rangka umum seperti yang ditunjukkan dalam Rajah 2.11.

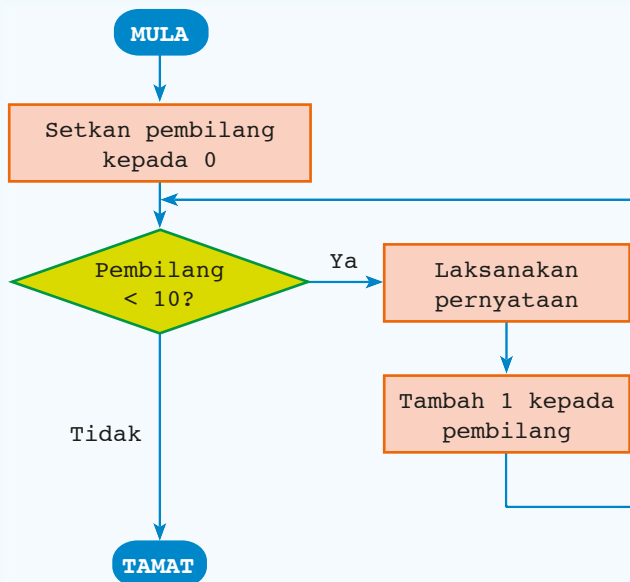
1. Isytihar pembilang.
2. Syarat diuji.
 Jika benar
 Pernyataan yang berulang dilaksanakan.
 Kemas kini pembilang *for*.
 Jika tidak benar
 Atur cara keluar dari struktur kawalan ulangan *for*.
3. Ulangi langkah 2 sehingga syarat menjadi tidak benar.

Rajah 2.11 Rangka umum struktur kawalan *for*

Rajah 2.12 menunjukkan pseudokod dan Rajah 2.13 menunjukkan carta alir bagi struktur kawalan ulangan *for* yang akan berulang selagi pembilang kurang daripada 10. Ini bermakna pernyataan akan berulang sebanyak 10 kali.

1. Mula
2. Setkan pembilang kepada 0
3. **for** pembilang < 10
 Laksanakan pernyataan
 Tambah 1 kepada pembilang
4. Tamat

Rajah 2.12 Pseudokod bagi struktur kawalan ulangan *for*



Rajah 2.13 Carta alir bagi struktur kawalan ulangan *for*

Contoh 2.3

Pengiraan hasil tambah nombor sehingga nombor n , di mana n ialah nombor bulat yang dimasukkan oleh pengguna.

Langkah 1 Penyiasatan

Proses-proses yang terlibat.

- Pengguna diminta memasukkan satu nombor.
- Sekiranya pengguna memasukkan 10, maka atur cara akan menjumlahkan sepuluh nombor yang pertama, seperti $(1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10)$.

Langkah 2 Mendraf algoritma dan melakukan semakan semula

Proses dan keputusan yang berikut dimurnikan untuk menghasilkan draf algoritma.

- 1 Isytihar pemboleh ubah.
- 2 Isytihar pernyataan pengawalan *for*.
- 3 Input/output.
- 4 Uji syarat.

Langkah 3 Memurnikan langkah-langkah algoritma

1. Isytihar pemboleh ubah **nom**, **kira**, **jum**.
2. Setkan nilai **kira** kepada 1.
3. Setkan nilai **jum** kepada 0.
4. Pengguna diminta memasukkan satu nombor.
5. Setkan nilai **nom** kepada nombor yang dimasukkan.
6. **for** **kira <= nom**
 - Kirakan $\text{jum} = \text{jum} + \text{kira}$
 - Kirakan $\text{kira} = \text{kira} + 1$
7. Ulang langkah 6.
8. Paparkan nilai **jum**.

Tiga pemboleh ubah

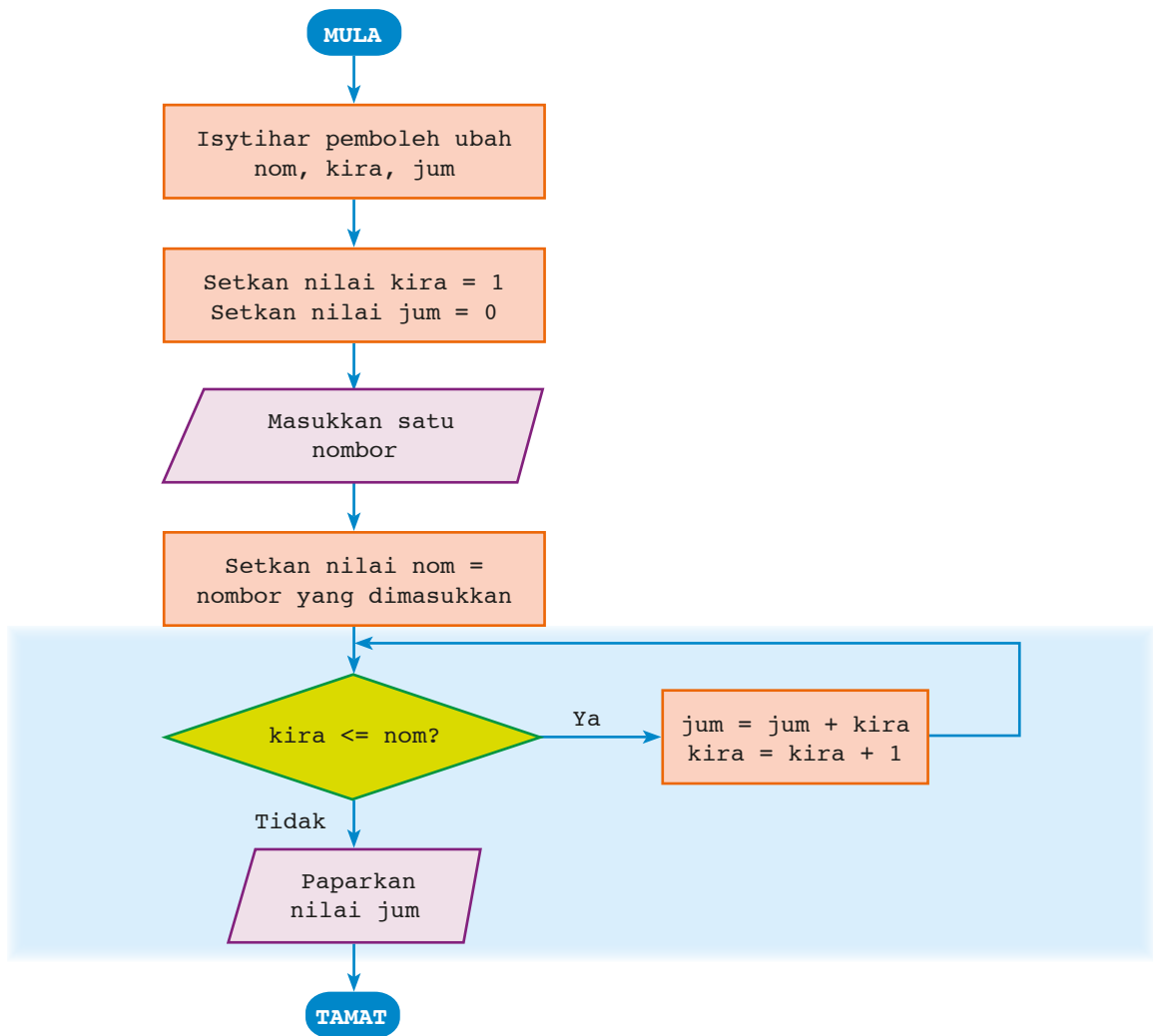
Syarat untuk gelung

Pernyataan yang berulang

Menulis pseudokod

1. **Mula**
2. **Isytihar** pemboleh ubah nom, kira, jum
3. **Setkan** nilai kira kepada 1
4. **Setkan** nilai jum kepada 0
5. **Masukkan** satu nombor
6. **Setkan** nilai nom kepada nombor yang dimasukkan
7. **for** kira $\leq$ nom
 Kirakan jum = jum + kira
 Kirakan kira = kira + 1
8. **Paparkan** nilai jum
9. **Tamat**

Melukis carta alir



Penerangan:

- Pemboleh ubah kira = 1 dan jum = 0.
- Pengguna diminta memasukkan satu nombor.
- Jika nilai yang dimasukkan oleh pengguna ialah 10, maka pemboleh ubah nom = 10.
- Syarat ulangan ialah kira $\leq$ nom.
- Dalam ujian syarat kali pertama, kira = 1 adalah kurang daripada nom = 10. Syarat dipenuhi, struktur kawalan ulangan dilaksanakan dan nilai-nilai jum dan kira dikemas kini menjadi jum = 1 dan kira = 2. Syarat diuji sekali lagi.
- Dalam ujian syarat kali kedua, kira = 2 adalah kurang daripada nom = 10. Syarat dipenuhi, struktur kawalan ulangan dilaksanakan dan nilai-nilai jum dan kira dikemas kini menjadi jum = 3 dan kira = 3. Syarat diuji sehingga syarat tidak dipenuhi.
- Proses ulangan ini akan berterusan sehingga kira = 11, di mana syarat bagi ulangan kira $\leq$ nom menjadi tidak benar. Maka ulangan dihentikan dan jum dipaparkan.



Cabar Diri Anda

Lakukan semakan langkah demi langkah untuk pseudokod dalam Contoh 2.3 sehingga mendapat output jum yang dipaparkan.



Aktiviti 2.3

Kumpulan

Menghasilkan pseudokod dan carta alir untuk masalah yang diberi.

1. Teliti masalah yang berikut.

Seorang murid diminta melakukan ujian matematik secara dalam talian. Jika markah ujian kurang daripada 85, murid akan diminta mengulangi ujian tersebut. Bilangan maksimum ulangan yang dapat dilakukan ialah tiga kali. Selepas melakukan ujian, markah akan dipaparkan. Jika murid mengambil lebih daripada satu ujian, markah yang dipaparkan ialah purata markah.

2. Tulis pseudokod dan lukis carta alir untuk masalah di atas.
3. Berdasarkan Contoh 2.3, bincangkan langkah-langkah penyelesaian masalah bersama-sama rakan sekumpulan anda.
4. Sediakan satu folio digital untuk langkah-langkah penyelesaian masalah bagi kumpulan anda dengan menggunakan perisian persembahan.



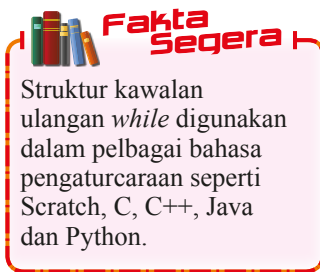
Struktur kawalan ulangan *while*

Dalam struktur kawalan ulangan *while*, ulangan dilaksanakan selagi syarat yang diuji adalah benar. Ini serupa dengan struktur kawalan *for*, yang mana pernyataan syarat diuji sebelum pernyataan yang berulang dilaksanakan. Perbezaannya, bagi struktur kawalan *for*, bilangan ulangan adalah ditetapkan manakala bagi struktur kawalan *while*, bilangan ulangan tidak diketahui dan ditentukan sama ada syarat dipenuhi ataupun tidak. Rajah 2.14 menunjukkan struktur umum bagi suatu struktur kawalan ulangan *while*.

1. Pengguna memasukkan input
2. Input ini diuji berdasarkan syarat yang ditetapkan
3. Selagi syarat adalah benar, pernyataan yang berulang dilaksanakan
4. Ulang langkah 1

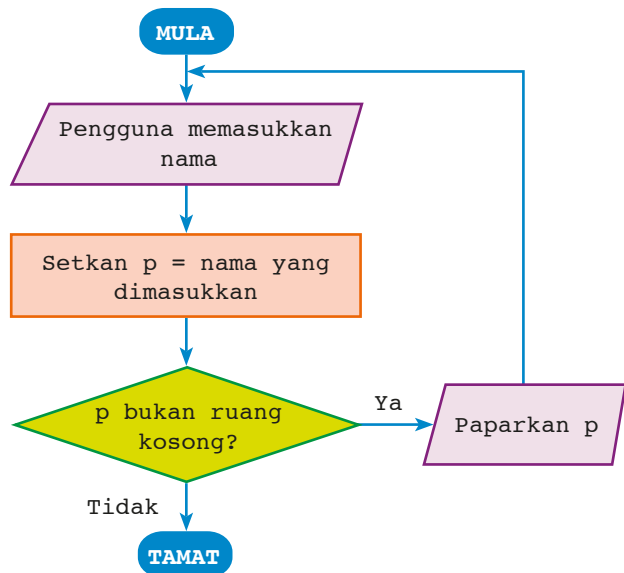
Rajah 2.14 Algoritma bagi suatu struktur kawalan ulangan *while*

Rajah 2.15 menunjukkan pseudokod dan Rajah 2.16 menunjukkan carta alir struktur kawalan *while* bagi satu algoritma yang menerima dan mencetak nama yang ditaipkan oleh pengguna selagi nama yang ditaip bukan ruang kosong.



1. **Mula**
2. **Minta** pengguna memasukkan nama
3. **Setkan** p = nama yang dimasukkan
4. **while** p bukan ruang kosong
 Paparkan p
5. **Tamat**

Rajah 2.15 Pseudokod yang menerima dan mencetak nama yang ditaipkan oleh pengguna selagi nama yang ditaip bukan ruang kosong



Rajah 2.16 Carta alir yang menerima dan mencetak nama yang ditaipkan oleh pengguna selagi nama yang ditaip bukan ruang kosong

Semasa pelaksanaan, pengguna diminta memasukkan satu nombor. Kemudian syarat diuji, jika syarat tidak dipenuhi, pembilang akan ditambah satu. Selagi syarat tidak dipenuhi, pernyataan akan berulang. Apabila syarat telah dipenuhi, gelung akan berhenti dan pembilang yang mengira bilangan ulangan akan dipaparkan.

Contoh 2.4

Pengiraan hasil tambah nombor yang dimasukkan sehingga pengguna menaip 0.

Langkah 1 Penyiasatan

Proses-proses yang terlibat.

- Pengguna diminta memasukkan nombor-nombor yang hendak dijumlahkan.
- Apabila pengguna menaip 0, ulangan akan berhenti dan jumlah bagi semua nombor yang telah dimasukkan akan dipaparkan.

Langkah 2 Mendraf algoritma dan melakukan semakan semula

Proses dan keputusan yang berikut dimurnikan untuk menghasilkan draf algoritma.

- 1 Isytihar pemboleh ubah.
- 2 Isytihar pernyataan pengawalan *while*.
- 3 Input/output.
- 4 Uji syarat.

Langkah 3 Memurnikan langkah-langkah algoritma

1. Isytihar pemboleh ubah **jumlah** dan **nombor**
2. Setkan nilai **jumlah** kepada 0
3. Setkan nilai **nombor** kepada 0
4. Paparkan mesej "Masukkan nombor:"
5. Setkan nilai nombor kepada nombor yang dimasukkan
6. **while** **nombor bukan sifar**
 - Kirakan $\text{jumlah} = \text{jumlah} + \text{nombor}$
 - Ulang langkah 5
7. Paparkan jumlah

Dua pemboleh ubah

Syarat untuk gelung berulang

Pernyataan yang berulang selagi syarat yang diuji menjadi benar

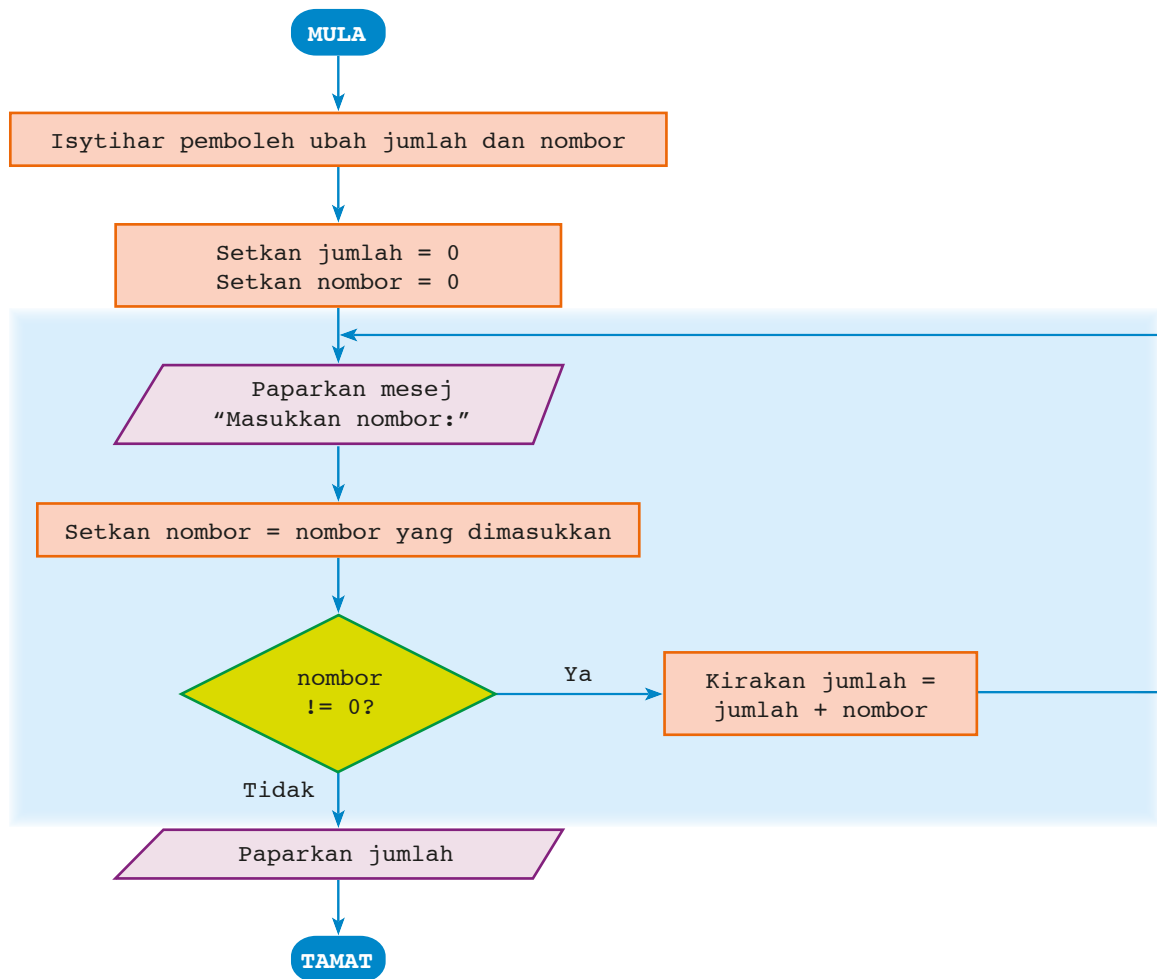
TIP

Simbol $\neq$ bermaksud tidak sama dengan. Oleh itu, syarat nombor $\neq$ sifar bermaksud selagi nombor tidak bersamaan dengan sifar, pernyataan dalam gelung *while* akan dilaksanakan.

Menulis pseudokod

1. **Mula**
2. **Isytihar** pemboleh ubah jumlah dan nombor
3. **Setkan** jumlah = 0
4. **Setkan** nombor = 0
5. **Paparkan** mesej "Masukkan nombor:"
6. **Setkan** nombor = nombor yang dimasukkan
7. **while** nombor $\neq$ sifar
 Kirakan jumlah = jumlah + nombor
 Ulang langkah 5
8. **Paparkan** jumlah
9. **Tamat**

Melukis carta alir



Penerangan:

- Pemboleh ubah jumlah = 0 dan nombor = 0.
- Pengguna diminta memasukkan satu nombor dan nilai nombor tersebut diumpukkan kepada pemboleh ubah nombor.
- Ujian syarat untuk nombor bukan 0 dilakukan. Jika ya, proses penambahan jumlah = jumlah + nombor dilaksanakan. Kemudian, mesej yang meminta pengguna memasukkan nombor berulang.
- Ulangan berhenti apabila ujian syarat nombor bukan 0 menjadi tidak benar.
- Nilai jumlah dipaparkan.

**Aktiviti 2.4****Kumpulan**

Menghasilkan pseudokod dan carta alir untuk permainan matematik.

1. Cuba permainan matematik dalam talian daripada goo.gl/o0vr3n.

**Penerangan ringkas**

Dalam masa 60 saat, setiap jawapan yang dijawab dengan betul diberikan satu markah.

2. Lakukan sumbang saran bersama-sama rakan sekumpulan anda untuk langkah-langkah menulis pseudokod dan melukis carta alir untuk permainan tersebut.
3. Sediakan satu folio digital untuk langkah-langkah penyelesaian masalah bagi kumpulan anda dengan menggunakan perisian persembahan. Folio digital perlu menunjukkan pseudokod dan carta alir sebagai hasil penyelesaian masalah ini.

2.1.2 Mengesan dan membaiki ralat yang terdapat dalam pseudokod dan carta alir dalam penyelesaian masalah**Standard Pembelajaran**

Murid boleh:

2.1.2 Mengesan dan membaiki ralat daripada pseudokod dan carta alir dalam penyelesaian masalah.

Sesuatu atur cara ditulis bagi menyelesaikan suatu masalah dengan jangkaan output yang tertentu. Kadang kala atur cara akan memberikan output yang tidak dijangka, berhenti secara tiba-tiba ataupun struktur kawalan ulangan berulang tanpa berhenti. Apabila perkara-perkara sebegini berlaku, maka atur cara dikatakan mempunyai ralat.

Ralat ditakrifkan sebagai kesilapan atau kesalahan dalam proses pembangunan atur cara segmen kod. Ralat selalu dikenal pasti dalam fasa pengujian. Terdapat tiga jenis ralat, iaitu ralat sintaks, ralat masa larian dan ralat logik.



Anda boleh merujuk semula Bab 3.1.3 dalam buku teks Asas Sains Komputer Tingkatan 1 untuk mengetahui lebih lanjut tentang teknik mengesan ralat dalam pseudokod dan carta alir.



Ralat sintaks merujuk kepada kesilapan dalam penggunaan bahasa pengaturcaraan. Semasa pelaksanaan atur cara, kesilapan yang menyebabkan ralat sintaks akan dipaparkan dalam bentuk mesej ralat. Oleh itu, ralat ini seakan-akan kesilapan ejaan atau tatabahasa dalam bahasa manusia.

Ralat masa larian merujuk kepada kesilapan yang akan menyebabkan atur cara terhenti secara tiba-tiba tanpa diduga semasa jalanan ujian (*test run*). Ralat masa larian boleh juga disebabkan oleh perkakasan komputer seperti memori yang tidak mencukupi. Selain itu, virus komputer dan perisian hasad juga boleh menyebabkan ralat masa larian. Virus dan perisian hasad akan mengubah *settings* pada komputer dan ini akan menyebabkan ralat masa larian.

Ralat logik merujuk kepada kesilapan yang menyebabkan atur cara memberi output yang tidak dijangka dan ini sering kali berlaku semasa pembangunan algoritma. Ini termasuk kesalahan urutan sesuatu tindakan atau syarat ditulis salah. Kecuaian ialah faktor utama kepada ralat logik. Lazimnya, ralat logik yang wujud dalam atur cara sukar untuk dikesan kerana tiada paparan mesej ralat.

Pengesanan ralat logik dapat dilakukan dengan menggunakan dua teknik, iaitu **teknik semakan meja** (*desk check*) dan **teknik langkah demi langkah** (*step through*). Kedua-dua teknik ini ialah teknik manual yang digunakan oleh pengatur cara untuk menyemak logik setiap langkah dalam pseudokod atau carta alir. Satu set sampel data input akan digunakan dan output yang dijana akan disemak. Teknik manual ini dijalankan dengan menggunakan beberapa helai kertas dan pensel sahaja.



Rajah 2.17 Tiga jenis ralat

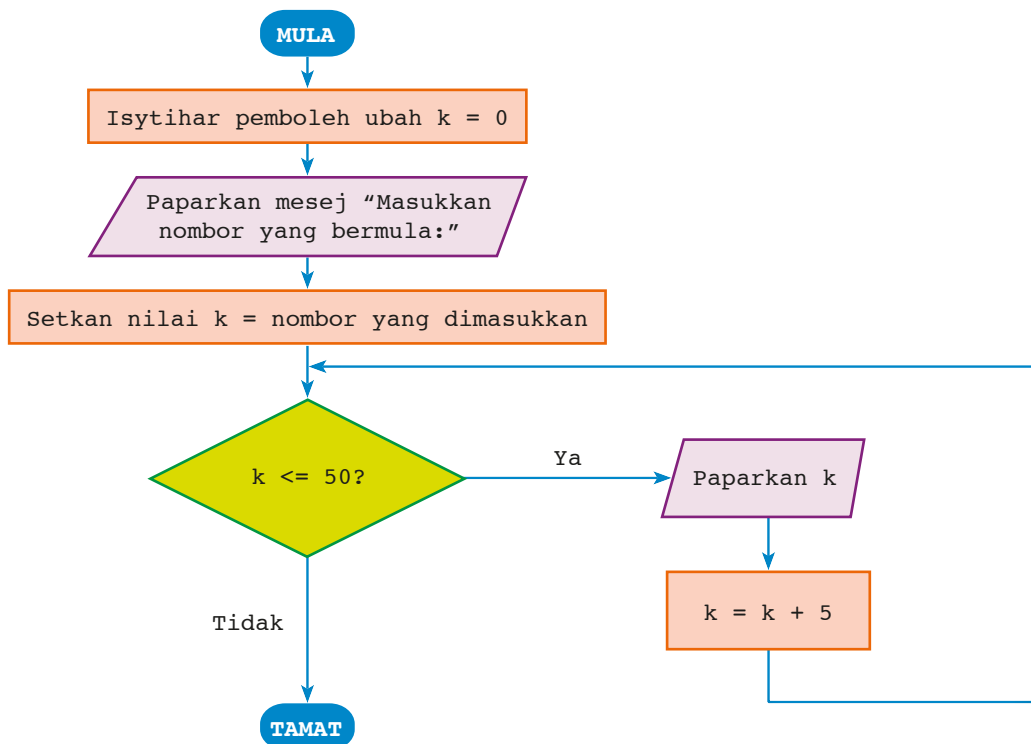
Contoh 2.5

Pengesanan ralat dalam pseudokod dan carta alir dalam penyelesaian masalah.

Murni dikehendaki menghasilkan satu atur cara yang menyenaraikan nombor secara menaik bermula daripada nombor yang dimasukkan oleh pengguna. Urutan nombor menaik yang dihasilkan adalah langkah 5 dan tidak melebihi 50. Berikut menunjukkan pseudokod dan carta alir yang dihasilkan oleh Murni.

Menulis pseudokod

1. **Mula**
2. **Isytihar** pemboleh ubah $k = 0$
3. **Paparkan** mesej "Masukkan nombor yang bermula:"
4. **Setkan** nilai $k =$ nombor yang dimasukkan
5. **while** $k \leq 50$
 Paparkan k
 $k = k + 5$
6. **Tamat**

Melukis carta alir

Apabila Murni membuat semakan ke atas pseudokod dan carta alir yang dihasilkan dengan memasukkan nombor 20 sebagai nombor bermula, dia mendapati hasil output tidak seperti yang diingini.

Contoh output

Output yang diinginkan

20
25
30
35
40
45

Output yang didapati

20
25
30
35
40
45
50

Murni meneliti semula pseudokod dan carta alir yang dilukisnya untuk mengesan dan membaiki ralat yang ada dengan teknik semakan meja.

Contoh 2.6

Penambahbaikan pseudokod dan carta alir dalam Contoh 2.5 menggunakan teknik semakan meja.

Dengan menggunakan kertas dan pensel, dan katakan nombor yang dimasukkan oleh pengguna ialah 20, teknik semakan meja dilakukan. Berikut menunjukkan nilai-nilai k apabila melalui pseudokod dalam Contoh 2.5.

1. Apabila nombor yang dimasukkan ialah 20, $k = 20$
2. Paparkan k , seterusnya $k = 20 + 5$
3. Paparkan k , seterusnya $k = 25 + 5$
4. Paparkan k , seterusnya $k = 30 + 5$
5. Paparkan k , seterusnya $k = 35 + 5$
6. Paparkan k , seterusnya $k = 40 + 5$
7. Paparkan k , seterusnya $k = 45 + 5$
8. Paparkan k , seterusnya $k = 50 + 5$

Berikut ialah output yang diperoleh daripada semakan pseudokod dalam Contoh 2.5.

20
25
30
35
40
45
50

Output ini berlainan dengan hasil yang dikehendaki, iaitu senarai nombor berhenti pada nombor 45.

Ralat logik ini disebabkan syarat bagi ulangan ($k \leq 50$) adalah tidak tepat. Syarat ulangan harus ditulis ($k \leq 45$). Pseudokod yang dibaiki adalah seperti di bawah.

1. **Mula**
2. **Isytihar** pemboleh ubah $k = 0$
3. **Paparkan** mesej "Masukkan nombor yang bermula:"
4. **Setkan** nilai $k =$ nombor yang dimasukkan
5. **while** $k \leq 45$ Syarat ulangan dibaiki
 Paparkan k
 $k = k + 5$
6. **Tamat**

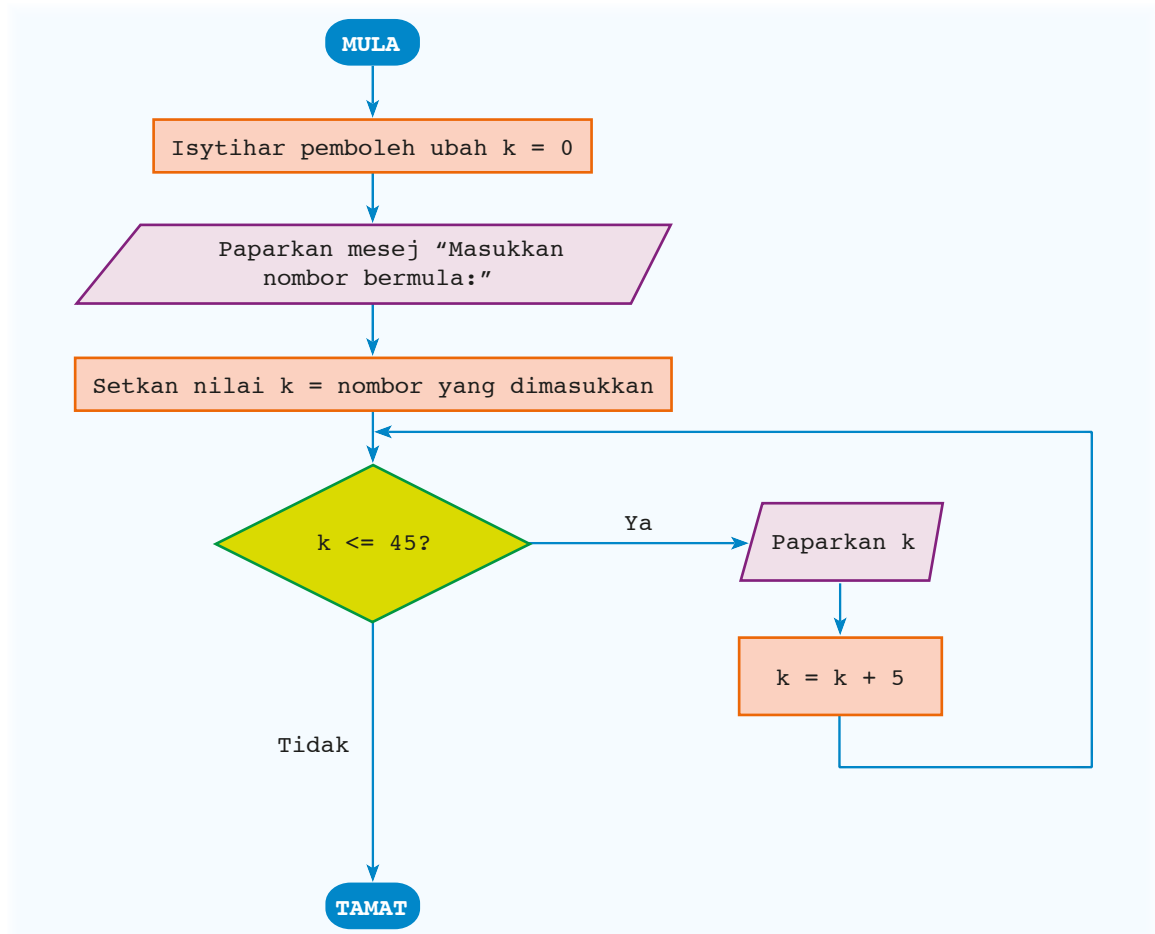
Semakan meja dilakukan sekali lagi pada pseudokod yang telah dibaiki. Berikut menunjukkan hasil semakan meja yang dilakukan.

1. Apabila nombor yang dimasukkan ialah 20, $k = 20$
2. Paparkan k , seterusnya $k = 20 + 5$
3. Paparkan k , seterusnya $k = 25 + 5$
4. Paparkan k , seterusnya $k = 30 + 5$
5. Paparkan k , seterusnya $k = 35 + 5$
6. Paparkan k , seterusnya $k = 40 + 5$
7. Paparkan k , seterusnya $k = 45 + 5$

Output:

20
25
30
35
40
45

Setelah selesai memperbaiki ralat yang terdapat dalam pseudokod, Murni membandingkannya dengan carta alir untuk memperbaiki ralat yang terdapat pada carta alir. Rajah 2.18 menunjukkan carta alir yang telah dibaiki.



Rajah 2.18 Carta alir bagi menyenaraikan nombor secara menaik sehingga 45 dengan perbezaan antara dua nombor ialah 5

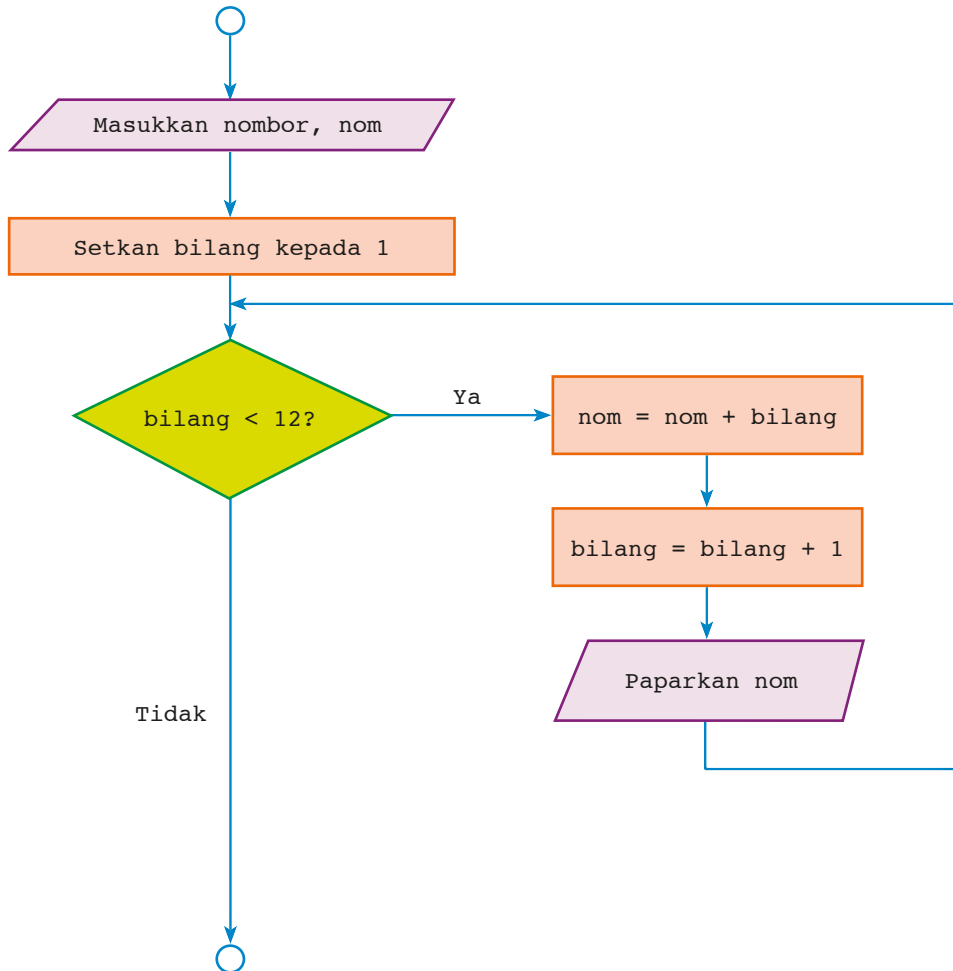


Aktiviti 2.5

Individu

Mengesan dan memperbaiki ralat dengan teknik semakan meja.

Carta alir di bawah mewakili algoritma yang menjana sifir. Sebagai contoh, jika nombor 5 dimasukkan, maka satu urutan 12 nombor bermula daripada 5 akan dipaparkan.



1. Jalankan semakan meja ke atas carta alir di atas dengan menggunakan sampel data. Adakah output yang diperoleh adalah seperti yang dikehendaki?
2. Jika tidak, kenal pasti ralat yang wujud dalam carta alir dan baikinya.
3. Lukis semula carta alir yang telah anda baiki.

2.1.3 Menggunakan pseudokod dan carta alir dengan menggabungkan pelbagai struktur kawalan untuk menyelesaikan masalah

Dalam penyelesaian sesuatu masalah yang kompleks, masalah itu harus dilarikan kepada masalah-masalah yang lebih kecil dan mudah melalui pembentukan algoritma. Terdapat banyak strategi dan teknik dalam mencari penyelesaian kepada sesuatu masalah. Setiap penyelesaian kepada sesuatu masalah bermula dengan sesuatu rancangan. Rancangan tersebut ialah algoritma yang digunakan untuk menyelesaikan masalah.

Langkah-langkah untuk menghasilkan pseudokod dan carta alir bagi menggabungkan struktur kawalan pilihan bersarang dan struktur kawalan ulangan.

Mengenal pasti masalah

Nadira ialah seorang murid Asas Sains Komputer Tingkatan Dua. Makanan kegemarannya ialah capati. Dia mengetahui terdapat mesin yang boleh membuat capati. Nadira berminat untuk mengetahui tentang algoritma membuat capati yang dapat mengarahkan mesin untuk menghasilkan capati. Oleh itu, dengan menggunakan langkah-langkah menghasilkan algoritma yang dipelajari, Nadira akan menulis sebuah algoritma membuat capati untuk mesin tersebut.

Dalam penggunaan mesin yang menghasilkan capati ini, pengguna hanya perlu memasukkan semua bahan yang diperlukan dan menekan butang 'Mula'. Mesin tersebut akan mengadun bahan-bahan yang dimasukkan dan membuat capati secara automatik selagi adunan masih ada.

Langkah 1 Penyiasatan

Dalam langkah penyiasatan, Nadira perlu menentukan input yang diperlukan, proses-proses yang terlibat dan output yang dihasilkan.

Input yang diperlukan

Input merujuk kepada bahan-bahan yang diperlukan untuk membuat adunan capati. Nadira perlu menyediakan bahan-bahan yang diperlukan terlebih dahulu. Sebelum dapat membuat capati, mesin hendaklah menentukan bahawa terdapat bahan yang telah dimasukkan. Jika ada, mesin akan memulakan proses mengadun bahan-bahan. Jika tidak, mesin akan memaparkan mesej memberitahu pengguna untuk memasukkan bahan-bahan yang diperlukan.

Standard Pembelajaran

Murid boleh:

2.1.3 Menyelesaikan masalah menggunakan pseudokod dan carta alir dengan menggabungkan pelbagai struktur kawalan.



Fakta Segera

Berdasarkan kertas kerja Jorge Vasconcelos yang bertajuk *Basic Strategy for Algorithmic Problem Solving*, lima langkah penting dalam strategi penyelesaian masalah adalah seperti berikut:

1. Baca dan fahami masalah.
2. Pilih konsep-konsep teori yang mungkin boleh diguna pakai untuk menyelesaikan masalah.
3. Hurai secara kualitatif tentang masalah.
4. Membina strategi bagi penyelesaian masalah.
5. Uji dan hurai jawapan kepada masalah.

(Dipetik dan disesuaikan daripada: <http://www.cs.jhu.edu/~jorgev/cs106/ProblemSolving.html>)

Langkah 2 Mendraf algoritma dan melakukan semakan semula

Proses-proses yang terlibat

Proses-proses yang terlibat ialah mengadun bahan-bahan dan membuat capati daripada adunan.

1. Ketika proses mengadun bahan-bahan, mesin akan memeriksa sama ada adunan sudah sehati ataupun tidak. Jika sudah, mesin akan mula membuat capati. Jika tidak, proses ini akan berulang sehingga adunan telah sehati.
2. Ketika proses membuat capati, mesin akan memeriksa sama ada adunan masih ada atau tidak. Jika adunan masih ada, mesin akan terus membuat capati. Jika adunan sudah habis, mesin akan paparkan jumlah capati yang dihasilkan.
3. Pengguna akan semak jumlah capati yang dihasilkan. Jika jumlah capati mencukupi, mesin akan dimatikan. Jika tidak mencukupi, pengguna harus menyediakan dan memasukkan bahan baharu dan seluruh proses diulang semula.

Output yang dihasilkan

Output yang dikehendaki ialah capati.

Langkah 3 Memurnikan langkah-langkah algoritma

1. Sediakan bahan.
2. Masukkan bahan.
3. Hidupkan mesin.
4. Proses menyemak bahan.
Jika ada bahan, mula mengadun bahan.
Jika tidak, paparkan mesej.
5. Proses mengadun bahan sehingga sehati.
Jika telah sehati, mula buat capati.
Jika tidak, terus adun bahan.
6. Proses menyemak adunan.
Jika masih ada, terus membuat capati.
Jika tiada, paparkan jumlah capati yang dihasilkan.
7. Semak jumlah capati.
Jika cukup, matikan mesin.
Jika tidak cukup, ulang Langkah 1.

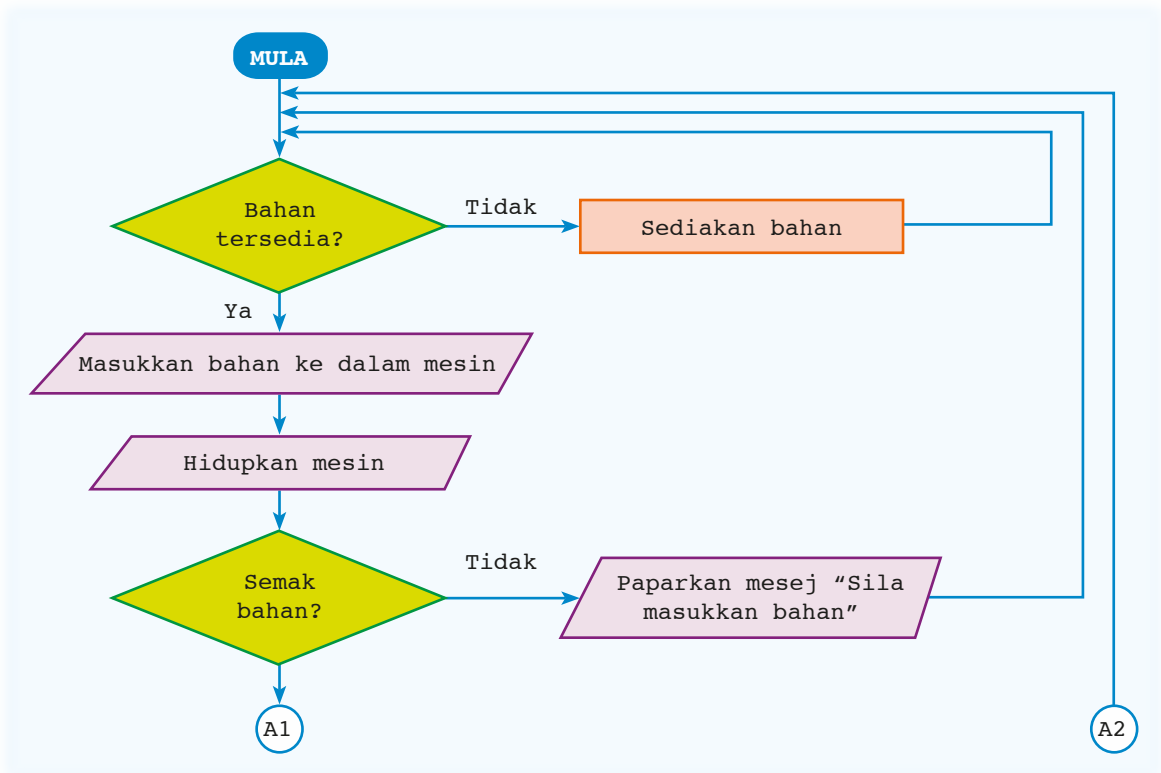
Rajah 2.19 Algoritma membuat capati

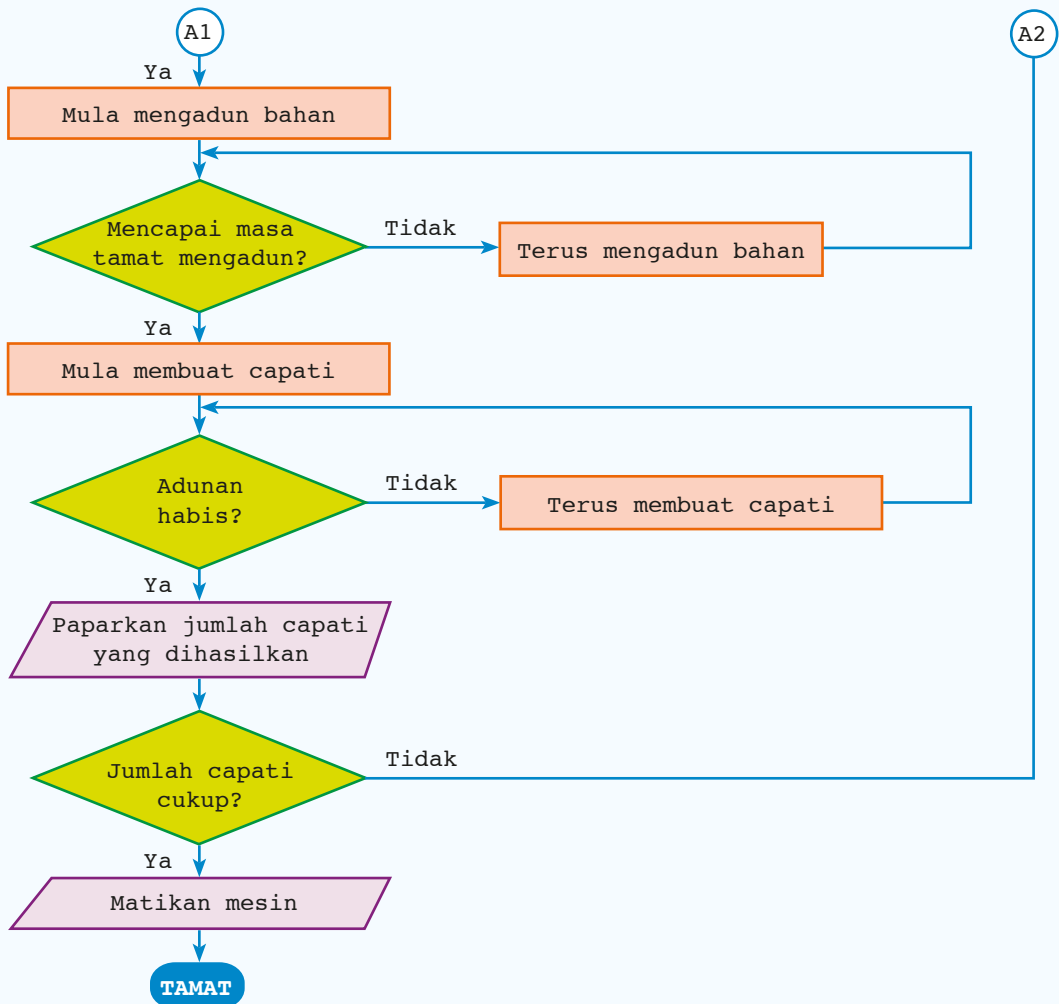
Menulis pseudokod

1. **Mula**
2. **Sediakan** bahan.
 Jika ada, masukkan bahan ke dalam mesin. Hidupkan mesin.
 Jika tidak, sediakan bahan dahulu. Ulang Langkah 2.
3. **Semak** bahan.
 Jika ada, mula mengadun bahan.
 Jika tidak, paparkan mesej "Sila masukkan bahan". Ulang Langkah 2.
4. **Semak** mencapai masa tamat mengadun.
 Jika ya, mula membuat capati.
 Jika tidak, terus mengadun bahan. Ulang Langkah 4.
5. **Semak** adunan habis.
 Jika ya, paparkan jumlah capati yang dihasilkan.
 Jika tidak, terus membuat capati. Ulang Langkah 5.
6. **Semak** jumlah capati.
 Jika mencukupi, matikan mesin.
 Jika tidak mencukupi, ulang Langkah 2.
7. **Tamat**

Rajah 2.20 Pseudokod membuat capati

Melukis carta alir





Rajah 2.21 Carta alir membuat capati



Aktiviti 2.6

Kumpulan

Syarat-syarat untuk mendapat lesen memandu (Kelas D).

Algoritma berikut menerangkan langkah-langkah dan syarat-syarat untuk memperoleh lesen memandu (Kelas D).

1. Minimum umur untuk mendaftar ialah 17 tahun.
2. Perlu menghadiri kursus Kurikulum Pendidikan Pemandu selama 6 jam.
3. Lakukan Ujian Buta Warna.
Jika lulus, boleh daftar untuk Ujian Teori Undang-undang Berkomputer.
Jika tidak lulus, anda tidak dibenarkan memperoleh lesen memandu.

4. Lakukan Ujian Teori Undang-undang Berkomputer.
Jika lulus, boleh mendapat lesen belajar memandu (Lesen L).
Jika tidak lulus, ulangi ujian.
5. Gunakan lesen L untuk belajar memandu sekurang-kurangnya 10 jam.
Jika sampai 10 jam, lakukan Ujian Penilaian Pra-Ujian.
Jika belum sampai 10 jam, teruskan belajar memandu.
6. Lakukan Ujian Penilaian Pra-Ujian.
Jika lulus, boleh daftar untuk Ujian Praktikal JPJ.
Jika tidak lulus, ulangi Ujian Penilaian Pra-Ujian.
7. Lakukan Ujian Praktikal JPJ.
Jika lulus, boleh mendapatkan Lesen Memandu Percubaan (PDL).
Jika tidak lulus, ulangi Ujian Praktikal JPJ.
8. Memandu dengan PDL selama 2 tahun.
Jika sudah 2 tahun memegang PDL, layak dinaik taraf kepada Lesen Memandu Kompeten (CDL).
Jika belum 2 tahun, memandu dengan PDL.
9. Pemandu PDL tertakluk kepada sistem mata demerit 10 mata KEJARA.
Jika mencapai 10 mata, lesen dibatalkan dan pemandu perlu mengulangi kesemua prosedur pembelajaran memandu.
Jika tidak mencapai 10 mata, lesen PDL dinaik taraf kepada CDL.

Tulis pseudokod dan lukis carta alir untuk mewakili algoritma ini.

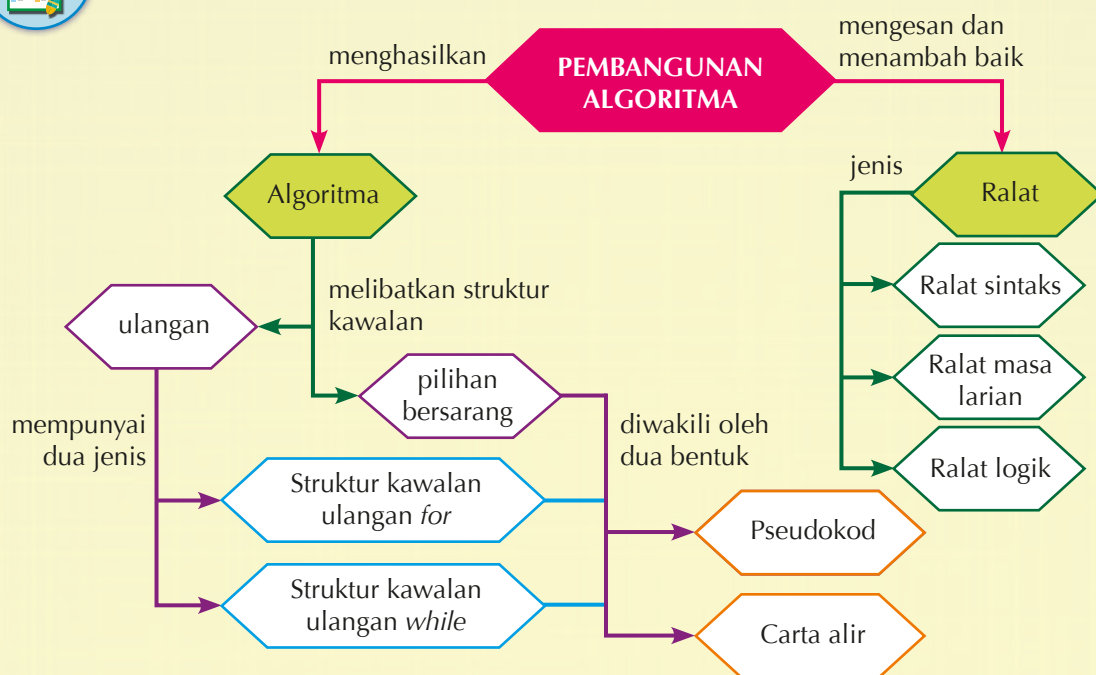


Praktis 2.1

1. Terangkan maksud algoritma.
2. Terangkan bagaimana algoritma boleh membantu menyelesaikan masalah?
3. Tulis **satu** pseudokod dan lukis **satu** carta alir yang menunjukkan
 - (a) struktur kawalan pilihan bersarang.
 - (b) struktur kawalan ulangan *for*.
 - (c) struktur kawalan ulangan *while*.
 - (d) gabungan ketiga-tiga jenis struktur kawalan.
4. Jelaskan **tiga** jenis ralat utama yang boleh didapati semasa pembangunan algoritma.



PETA KONSEP BAB 2



SENARAI SEMAK BAB 2

Pada akhir bab ini, saya dapat...

menjelaskan jenis-jenis struktur kawalan dengan menggunakan contoh yang sesuai.	
menghasilkan pseudokod dan carta alir untuk menyelesaikan masalah yang melibatkan struktur kawalan bersarang.	
menghasilkan pseudokod dan carta alir untuk menyelesaikan masalah yang melibatkan struktur kawalan ulangan <i>for</i> .	
menghasilkan pseudokod dan carta alir untuk menyelesaikan masalah yang melibatkan struktur kawalan <i>while</i> .	
menunjukkan keupayaan mengesan dan membaiki ralat dalam pseudokod dan carta alir yang diberi dalam penyelesaian masalah.	
menggabungkan pelbagai struktur kawalan untuk menyelesaikan masalah dengan menggunakan pseudokod dan carta alir secara kreatif.	



PENTAKSIRAN BAB 2

Soalan Objektif

1. Satu siri langkah yang tersusun untuk menyelesaikan sesuatu masalah dipanggil _____.
A algebra
B algoritma
C pseudokod
D pengaturcaraan
2. Antara berikut, yang manakah perwakilan algoritma?
I Pseudokod
II Peta minda
III Carta alir
IV Segmen kod
A I dan II
B II dan III
C I dan III
D I, II dan IV
3. Antara situasi berikut, yang manakah boleh diwakili dengan algoritma?
I Bersedia untuk tidur.
II Mengikat tali kasut.
III Membakar roti.
IV Bersedia untuk pergi jogging.
A I dan II
B II dan III
C III dan IV
D I, II, III dan IV
4. Antara pilihan berikut, yang manakah bukan suatu struktur kawalan?
A Ulangan *for*
B Pelbagai pilihan
C Ulangan *while*
D Proses
5. Struktur kawalan pilihan yang terbenam dalam struktur kawalan pilihan yang lain disebut _____.
A struktur kawalan pelbagai pilihan
B struktur kawalan ulangan
C struktur kawalan pilihan bersarang
D struktur kawalan dwipilihan
6. Perhatikan peralatan di bawah. Alatan yang manakah melakukan proses yang berulang?
I

II

III

IV

A I dan II
B I, II dan III
C I, II dan IV
D I dan IV

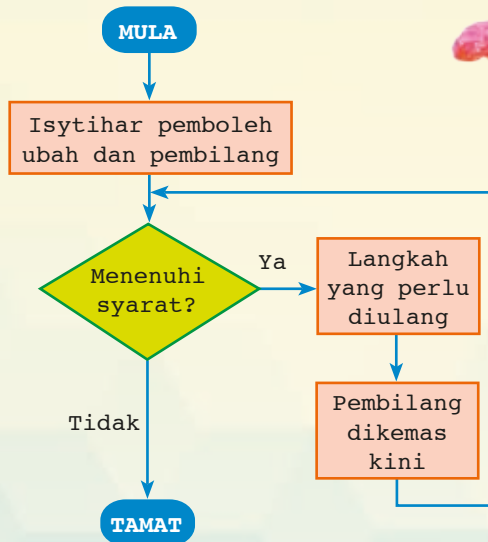
7. Baca dan teliti situasi di bawah ini.

Nadia Natasya berada di taman permainan kanak-kanak. Dia memerhatikan seorang kanak-kanak yang sedang duduk di atas buaian. Buaian tersebut dibuai oleh ibu kanak-kanak tersebut.

Apakah contoh struktur kawalan dalam situasi di atas?

- A Struktur kawalan pilihan tunggal
- B Struktur kawalan dwipilihan
- C Struktur kawalan ulangan
- D Struktur kawalan pilihan bersarang

8. Teliti carta alir di bawah.



Struktur kawalan yang manakah ditunjukkan dalam carta alir ini?

- A Struktur kawalan pilihan bersarang
- B Struktur kawalan ulangan *for*
- C Struktur kawalan ulangan *while*
- D Struktur kawalan pelbagai pilihan

9. Baca dan teliti situasi di bawah.

Warganegara Malaysia yang berumur 21 tahun dan ke atas layak mengundi jika individu ini telah mendaftar sebagai pengundi.

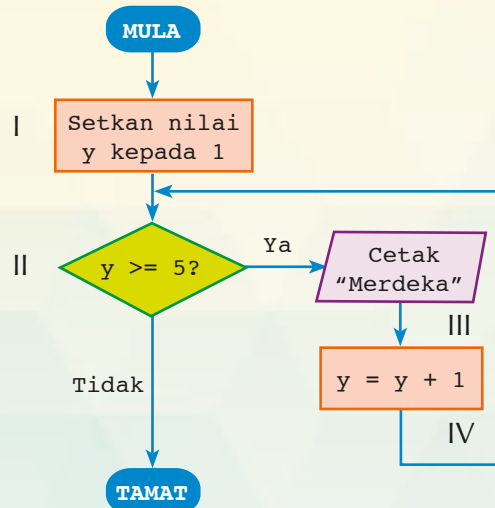
Struktur kawalan yang manakah sesuai digunakan untuk menyelesaikan masalah yang diberi?

- A Struktur kawalan pelbagai pilihan
- B Struktur kawalan pilihan bersarang
- C Struktur kawalan ulangan *for*
- D Struktur kawalan ulangan *while*

10. Teknik yang manakah digunakan untuk mengesan ralat logik dalam algoritma?

- I Teknik semakan meja
- II Teknik semakan manual
- III Teknik langkah demi langkah
- IV Teknik logik
- A I dan II
- B I dan III
- C I, II dan III
- D I, II, III dan IV

11. Carta alir berikut dibina untuk mencetak "Merdeka" sebanyak lima kali. Kewujudan ralat menyebabkan carta alir ini tidak memaparkan output yang sepatutnya.

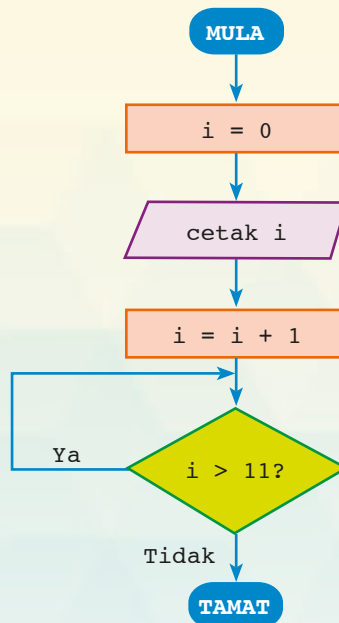


Kesan ralat yang terdapat pada carta alir di atas.

- A I
- B II
- C III
- D IV

Soalan Berstruktur

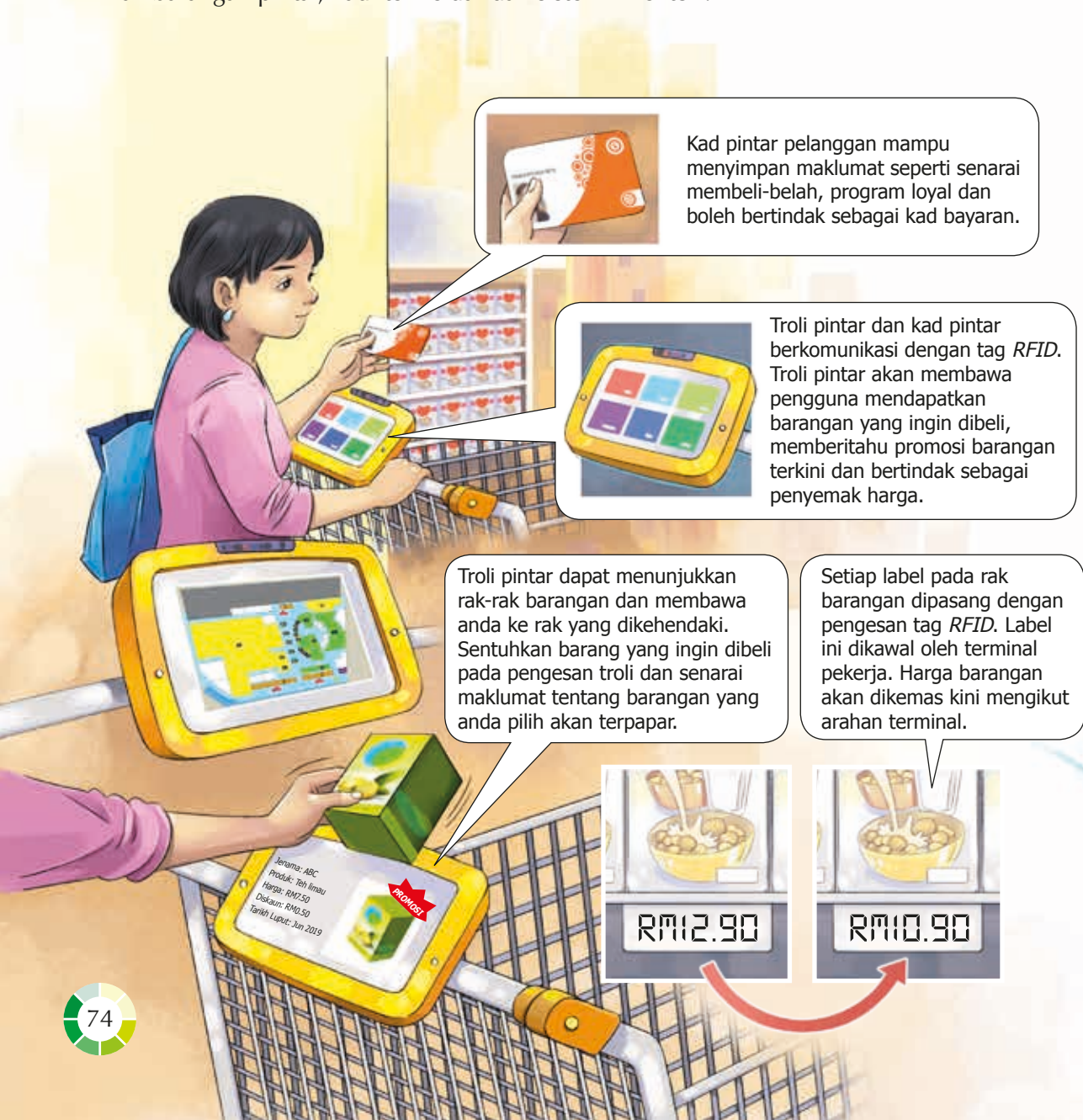
1. Huraikan langkah-langkah yang terlibat untuk membangunkan algoritma bagi menyelesaikan sesuatu masalah.
2. Jelaskan maksud struktur kawalan pilihan bersarang. Berikan satu contoh yang sesuai dalam penjelasan anda.
3. Terdapat aktiviti harian yang melibatkan langkah-langkah yang berulang. Beri **satu** contoh aktiviti harian yang berulang. Jelaskan kepentingan langkah-langkah yang berulang.
4. Apabila anda memasukkan kad bank ke dalam mesin ATM, mesin akan meminta anda memasukkan nombor pin. Jika anda masukkan nombor pin yang salah sebanyak tiga kali, mesin ATM akan menahan kad bank anda. Bina **satu** algoritma dalam bentuk carta alir untuk menunjukkan proses penahanan kad bank pada mesin ATM.
5. Bina **satu** algoritma dalam bentuk pseudokod untuk membolehkan Surendren mencetak ayat "Saya bernama Surendren!" sebanyak tiga kali.
6. Carta alir di bawah melibatkan struktur kawalan ulangan. Berdasarkan struktur kawalan, selepas semakan meja dilakukan, carta alir yang dibina tidak membuat ulangan. Teliti carta alir di bawah dan kesan ralat yang ada. Kemudian lukis semula carta alir yang telah dibuat penambahbaikan.



Teknologi telah mengubah cara kita berbelanja. Pembangunan sistem *Radio-Frequency IDentification (RFID)* menjadikan proses pembelian di pasar raya menjadi lebih mudah, teratur dan tanpa tunai (*cashless*). Teknologi *RFID* yang menggunakan tag pintar mampu menyimpan lebih banyak maklumat tentang barangan dan boleh dikesan melalui frekuensi radio. Barangan yang dilabel dengan tag *RFID* akan dikesan oleh terminal seperti troli pintar, rak barangan pintar, kaunter keluar dan sistem inventori.

Standard Kandungan

- 3.1** Persekitaran Kod Arahan
- 3.2** Struktur Kod Arahan



Kata Kunci

- Boolean
- Char
- Input
- Integer
- Operator logik
- Operator perbandingan
- Output
- Pemalar
- Pemboleh ubah
- Pernyataan bersyarat
- Ralat logik
- Ralat masa larian
- Ralat sintaks
- *String*
- Struktur kawalan jujukan
- Struktur kawalan pilihan
- Struktur kawalan pilihan bersarang
- Struktur kawalan ulangan



Setiap label pada rak barangan dipasang pengesan tag *RFID*. Jika bilangan barangan tinggal dua unit, tag *RFID* akan menghantar mesej ke monitor terminal pekerja melalui frekuensi radio.

Kaunter pembayaran



Senarai barangan pada troli



Semasa pembayaran, senarai barangan pada troli akan dipindahkan ke skrin kaunter pembayaran. Pelanggan boleh memilih mod bayaran yang dikehendaki. Resit akan dicetak di kaunter pembayaran pada akhir transaksi.

Selepas pembayaran, pelanggan akan keluar melalui pintu penyemak. Tag *RFID* barangan di dalam troli akan disemak dengan senarai barangan dalam transaksi. Kemudian semua tag *RFID* barangan akan dikemas kini.



3.1 Persekitaran Kod Arah

Kod arahan ialah satu arahan yang ditulis dalam bahasa komputer untuk mengarahkan komputer melaksanakan sesuatu tugas. Satu set kod arahan yang disebut segmen kod digunakan untuk membina satu atur cara bagi menyelesaikan masalah.

Bahasa komputer terdiri daripada bahasa pengaturcaraan yang digunakan oleh pengatur cara bagi menulis kod arahan. Bahasa pengaturcaraan yang telah dipelajari termasuklah **Scratch** dan **HTML**. Dalam bab ini, anda akan diperkenalkan dengan bahasa pengaturcaraan **Python** yang mempunyai banyak persamaan dengan Scratch. Dalam pengaturcaraan Python, murid perlu menulis kod-kod yang mudah difahami.

Persekitaran kod arahan merujuk kepada persediaan terhadap peralatan dan perisian yang diperlukan untuk menulis kod arahan. Oleh yang demikian, persekitaran kod arahan merujuk kepada perkakasan komputer, sambungan kepada Internet, pelayar web dan perisian aplikasi bahasa pengaturcaraan yang diperlukan untuk pembangunan atur cara, penterjemahan dan pelaksanaan atur cara.

Setiap bahasa pengaturcaraan mempunyai penterjemah tersendiri. Semasa pelaksanaan atur cara, kod arahan akan menjalani pengalihan kepada bahasa mesin yang boleh difahami oleh komputer. Terdapat tiga jenis penterjemah, iaitu **penghimpun** (*assembly*), **pengkompil** (*compiler*) dan **pentaafsir** (*interpreter*). Peta pemikiran pada Rajah 3.1 mengelaskan tiga jenis penterjemah ini.



Rajah 3.1 Tiga jenis penterjemah

Rujuk Lampiran B untuk amali memasang perisian aplikasi Python 3.5.2.



Pembelajaran seterusnya dalam bab ini akan menggunakan bahasa pengaturcaraan Python dalam penulisan kod arahan yang menggunakan sintaks dan beberapa contoh yang terlibat. Oleh yang demikian, perisian aplikasi **Python 3.5.2** diperlukan untuk membaca dan melaksanakan kod arahan Python.

3.1.1 Jenis data dalam segmen kod

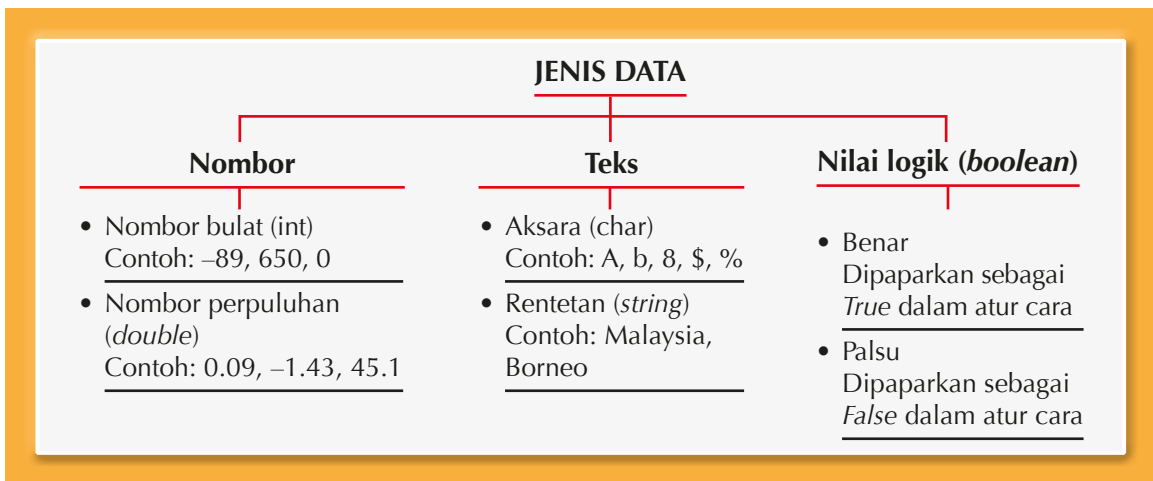
Setiap bahasa pengaturcaraan mempunyai satu senarai jenis data. Jenis data digunakan semasa pengisytiharan suatu pemboleh ubah untuk membenarkan atur cara mengenal pasti kegunaan pemboleh ubah tersebut dan membolehkan pemboleh ubah tersebut disimpan dalam storan ingatan dengan berkesan.

Jenis-jenis data yang akan dipelajari ialah integer, *double*, *boolean*, *char* dan *string*. Jenis-jenis data ini dapat dikelaskan kepada tiga bahagian utama, iaitu nombor, teks dan nilai logik. Nilai **nombor** atau nilai berangka terdiri daripada nombor bulat dan nombor perpuluhan. Nilai **teks** adalah seperti satu aksara dan satu rentetan aksara. Nilai **logik** mempunyai dua keadaan sahaja, iaitu “Benar” dan “Palsu”. Rajah 3.2 menunjukkan peta pemikiran pokok bagi pengelasan jenis data.

Standard Pembelajaran

Murid boleh:

3.1.1 Menggunakan jenis data (*integer*, *boolean*, *double*, *char* dan *string*) dalam segmen kod.



Rajah 3.2 Pengelasan tiga jenis data

Dalam pengaturcaraan, pemboleh ubah ditakrifkan sebagai jenis data pada permulaan atur cara. Jenis-jenis data ini perlu diketahui oleh murid supaya murid dapat membezakan setiap jenis data dan kegunaannya. Murid juga perlu mampu menggunakan pemboleh ubah yang dipilih dengan jenis data yang sesuai bagi atur cara yang bakal dibangunkan.

Integer

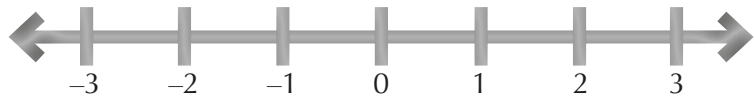
Integer ialah nombor yang tidak mengandungi bahagian pecahan dan tidak mempunyai titik perpuluhan. Integer juga dikenali sebagai nombor bulat. Integer boleh terdiri daripada nombor negatif, sifar dan nombor positif.



Pemboleh ubah penting dalam pengaturcaraan komputer. Pemboleh ubah menyimpan tiga jenis data, iaitu nilai berangka, teks dan nilai logik. Pemboleh ubah perlu diisytiharkan dahulu sebelum digunakan dalam atur cara.



Rajah 3.3 menunjukkan garis nombor bagi nilai integer bermula daripada nilai -3 hingga nilai 3 .



Rajah 3.3 Garis nombor bagi nilai integer -3 hingga 3

Dalam pengaturcaraan, integer selalu digunakan dalam penyelesaian masalah matematik yang melibatkan operasi-operasi matematik yang tertentu. Jadual 3.1 menunjukkan operator matematik, simbol dan contoh.

Jadual 3.1 Operator matematik, simbol dan contoh

Nama operator	Simbol operator matematik	Simbol operator dalam komputer	Contoh pengiraan dalam atur cara
Tambah	+	+	$88 + 22 = 110$
Tolak	-	-	$42 - 25 = 17$
Darab	$\times$	*	$23 * 8 = 184$
Bahagi	$\div$	/	$66 / 3 = 22$



Bahasa pengaturcaraan Python, C dan Java menyokong jenis data integer. Sintaks umum yang digunakan dalam pengisytiharan nombor integer ialah **int**. Fungsi **int** digunakan khusus untuk membuat penetapan sesuatu nilai pemboleh ubah adalah sentiasa integer.

Contoh 3.1

Kod arahan Python yang menggunakan fungsi **int** dengan menggunakan perisian aplikasi Python 3.5.2.

```

Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900
32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> a = int(10)
>>> b = int(29)
>>> c = a + b
>>> print(c)
39
>>>
  
```

Pemboleh ubah a akan sentiasa memegang nilai integer apabila fungsi **int** digunakan.

Ln: 8 Col: 4



Aktiviti 3.1

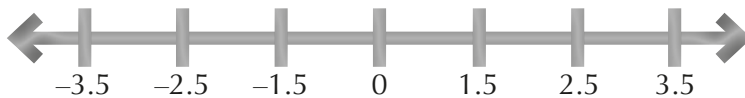
Individu

Tulis segmen kod Python dengan menggunakan operator matematik.

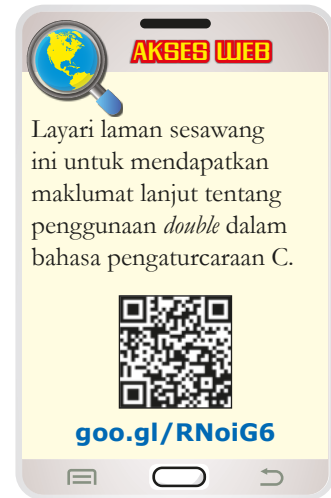
1. Dapatkan hasil tolak bagi $a = 50$ dan $b = 36$.
2. Berapakah hasil darab bagi $x = 49$ dengan $y = 23$?
3. Berapakah hasil bahagi bagi $m = 128$ dengan $n = 12$?

Double

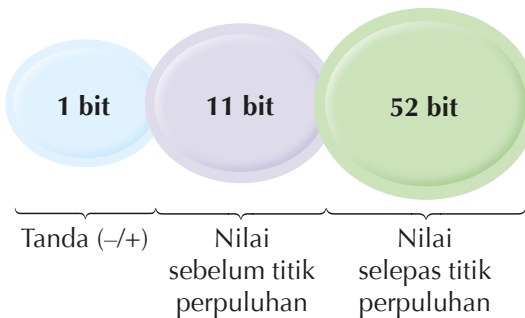
Double juga sejenis data yang mewakili nombor. Perbezaan antara *double* dan integer ialah nombor yang diwakili oleh *double* ialah nombor yang mengandungi bahagian pecahan, iaitu nombor yang mempunyai titik perpuluhan. *Double* boleh terdiri daripada nombor negatif dan nombor positif yang bertitik perpuluhan. Ini membolehkan kod arahan membuat pengiraan yang lebih jitu semasa membuat pengiraan wang ataupun menyelesaikan masalah matematik yang kompleks. Rajah 3.4 menunjukkan garis nombor bagi nilai *double* bermula daripada nilai -3.5 hingga 3.5 .



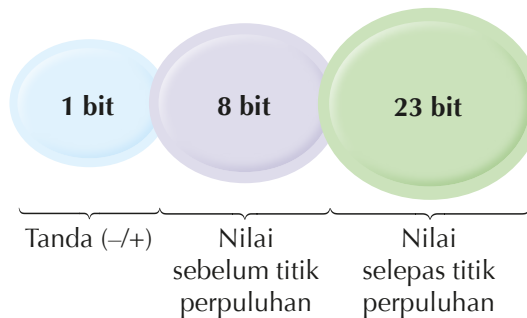
Rajah 3.4 Garis nombor bagi nilai *double* -3.5 hingga 3.5



Bahasa-bahasa pengaturcaraan C, C++ dan C# mengiktiraf *double* sebagai jenis data. Bahasa pengaturcaraan Python tidak menyokong jenis data *double* tetapi menggunakan jenis data *float* bagi mewakili nombor-nombor yang mempunyai titik perpuluhan. Perbezaan jenis data *double* dan *float* ialah penggunaan memori bagi pemboleh ubah yang diisytiharkan sebagai *double* ataupun *float*. Apabila sesuatu pemboleh ubah diisytiharkan sebagai *double*, memori sebanyak 64 bit akan diperuntukkan. Rajah 3.5 menunjukkan pecahan bit bagi nilai *double*. Apabila sesuatu pemboleh ubah diisytiharkan sebagai *float*, memori sebanyak 32 bit akan diperuntukkan. Rajah 3.6 menunjukkan pecahan bit bagi nilai *float*.



Rajah 3.5 Pecahan bit bagi nilai *double*



Rajah 3.6 Pecahan bit bagi nilai *float*

Contoh 3.2

Kod arahan yang menggunakan bahasa pengaturcaraan C++ untuk melakukan pengiraan mudah dengan jenis data *double*.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     // your code goes here
6     double num1 = 3.5;
7     cout<<"Nilai double pertama:"<<num1;
8     double num2 = 4.9;
9     cout<<"\nNilai double kedua:"<<num2;
10    double num3;
11    num3 = num1 + num2;
12    cout<<"\nHasil tambah bagi kedua-dua nombor ialah:"<<num3;
13    return 0;
14 }
15
```

Pemboleh ubah num1 diisytihar dengan jenis data *double* dengan sintaks umum **double**.

Cout ialah fungsi output bagi bahasa pengaturcaraan C++.

Output

```
Nilai double pertama:3.5
Nilai double kedua:4.9
Hasil tambah bagi kedua-dua nombor ialah:8.4
```

Anda boleh mencuba kod arahan ini dengan menggunakan program pengaturcaraan C secara dalam talian melalui laman sesawang berikut:

<https://www.codechef.com/ide>

Contoh 3.3

Kod arahan yang menggunakan bahasa pengaturcaraan Python untuk melakukan pengiraan mudah dengan jenis data *float*.

```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900
32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> nom1 = float (3.333)
>>> nom2 = float (10.401)
>>> nom3 = nom1 + nom2
>>> print (nom3)
13.734
>>> round (nom3, 2)
13.73
>>>
```

Pemboleh ubah nom1 diisytihar dengan jenis data *float* dengan sintaks umum **float**.

Sintaks umum **float** digunakan untuk membuat pengisytiharan bagi nilai *float*. Jika diperhatikan, hasil tambah bagi nom1 dan nom2 juga akan memberikan nilai *float*. Dalam pengaturcaraan Python, fungsi **round()** digunakan untuk membundarkan nilai *float* kepada bilangan tempat perpuluhan yang dikehendaki.

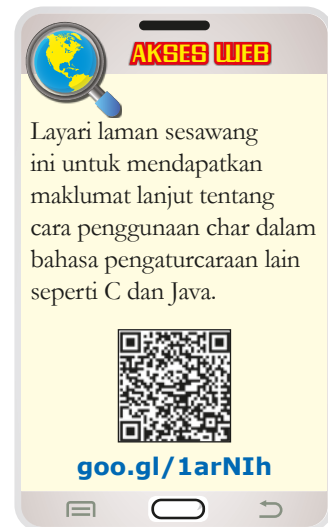
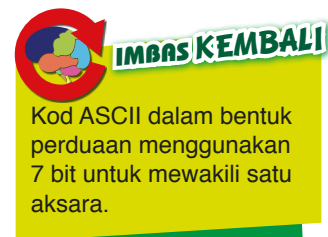
Char

Char ialah jenis data yang digunakan untuk mewakili aksara (*character*). Aksara terdiri daripada huruf abjad, angka dan simbol yang berdasarkan Kod Piawai Amerika untuk Pertukaran Maklumat ASCII (*American Standard Code for Information Interchange*). Jadual 3.2 menunjukkan jenis-jenis data yang diwakili oleh char.

Jadual 3.2 Pengelasan jenis aksara berserta contoh

Jenis aksara (char)	Contoh
Huruf	A hingga Z atau a hingga z
Digit	0, 1, 2, ...
Simbol	\$, @, % dan lain-lain

Bahasa-bahasa pengaturcaraan seperti C, C++, C# dan Java mengiktiraf jenis data char. Sintaks umum **char** digunakan semasa pengisytiharan sesuatu pemboleh ubah. Bahasa pengaturcaraan Python tidak mempunyai jenis data char akan tetapi semua aksara diwakili dalam jenis data lain yang disebut *string*.



Contoh 3.4

Kod arahan yang menggunakan bahasa pengaturcaraan C++ untuk memaparkan satu aksara 'a'.

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      // your code goes here
6      char huruf1 = 'a';
7      cout<< huruf1;
8      return 0;
9  }
10
```

Sintaks umum **char** digunakan untuk mengisytiharkan pemboleh ubah huruf1 dengan jenis data char.

Output

a

Anda boleh mencuba kod arahan ini dengan menggunakan program pengaturcaraan C++ secara dalam talian melalui laman sesawang berikut:

<https://www.codechef.com/ide>

TIP

Dalam satu ayat (*string*), ruang kosong (*empty space*) yang terdapat di dalamnya juga diambil kira sebagai satu aksara.

String

String ialah satu perkataan yang dibina daripada aksara. *String* terdiri daripada satu urutan aksara-aksara yang berjujukan yang membentuk suatu rentetan (*string*). Rajah 3.7 mengilustrasikan pembentukan satu perkataan daripada beberapa aksara.

N + E + G + A + R + A + K + U = NEGARAKU

Rajah 3.7 Pembentukan satu perkataan (*string*) daripada beberapa aksara (*char*)

Kebanyakan bahasa pengaturcaraan akan menyokong jenis data *string*, termasuk C++, Ruby dan Python. Semasa pengisytiharan jenis data *string* kepada sesuatu pemboleh ubah, sintaks umum **str** digunakan.

Semasa mengatur cara, nilai *string* yang diberikan kepada sesuatu pemboleh ubah perlu diletakkan tanda petikan tunggal ('...') atau tanda petikan berganda ("..."). Tanda petikan ini berfungsi untuk memberitahu permulaan dan pengakhiran satu ayat. Berikut menunjukkan cara mengisytiharkan pemboleh ubah lagu kepada nilai *string* Negaraku.

```
lagu = str ('Negaraku')  
      atau  
lagu = str ("Negaraku")
```

TIP

Bergantung kepada sesuatu kod arahan yang ingin dibangunkan, adakala *string* boleh dijadikan sebagai pemboleh ubah dan boleh juga digunakan untuk dijadikan pemalar.



Fakta Segera

Walaupun bahasa pengaturcaraan Python tidak mempunyai jenis data *char*, tetapi penggunaan fungsi lain membolehkan Python memanipulasikan aksara-aksara dalam *string*.

Contoh 3.5

Kod arahan bahasa pengaturcaraan Python yang menggunakan jenis data *string*.

```
Python 3.5.2 Shell  
File Edit Shell Debug Options Window Help  
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900  
32 bit (Intel)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>> ayat = str ("Selamat Datang")  
>>> print (ayat)  
Selamat Datang  
>>>
```

Sintaks umum yang digunakan dalam membuat pengisytiharan *string* ialah **str**.

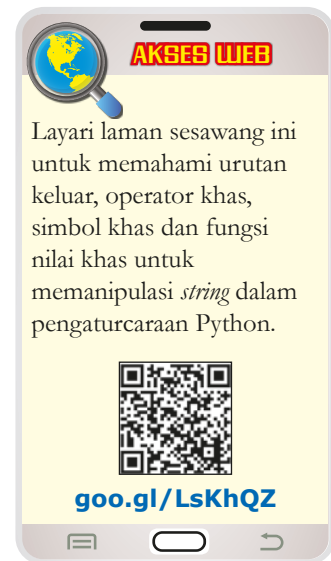
Ln: 6 Col: 4

Teks berwarna biru ialah output bagi fungsi **print()**. Dalam pengaturcaraan Python, mencetak suatu pemboleh ubah dengan fungsi **print()** tidak perlu menggunakan tanda petikan.

Dalam pengaturcaraan Python, jenis data *string* dapat digunakan bersama fungsi-fungsi Python yang lain untuk memanipulasi nilai *string* mengikut keperluan dan tujuan sesuatu program atau aplikasi yang ingin dibangunkan. Jadual 3.3 menunjukkan fungsi yang memanipulasi nilai *string* dan tujuannya.

Jadual 3.3 Fungsi yang memanipulasi nilai *string* dan tujuannya

Fungsi	Tujuan
Tanda [] atau [:]	Mencapai satu aksara atau lebih dalam <i>string</i> .
len ()	Menghitung bilangan aksara dalam <i>string</i> .
Simbol +	Menggabungkan dua <i>string</i> tanpa ruang kosong.
Simbol ,	Menggabungkan dua <i>string</i> dengan satu ruang kosong.
Simbol *	Mengulang <i>string</i> dengan bilangan yang dinyatakan.
join ()	Menambahkan aksara di antara setiap aksara.



Contoh 3.6 hingga Contoh 3.9 memperjelaskan lagi cara untuk menggunakan semua fungsi-fungsi dalam Jadual 3.3. Contoh-contoh ini akan membimbing murid memahami kegunaan fungsi-fungsi tersebut untuk membolehkan murid mengatur cara dengan menggunakan Python dalam Subtopik 3.2.

Contoh 3.6

Kod arahan bahasa pengaturcaraan Python untuk mencapai satu aksara atau lebih dalam *string*.

```

Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900
32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ayat = str ("Selamat datang")
>>> print (ayat[4])
m
>>> print (ayat[10:14])
tang
>>>
Ln: 8 Col: 4

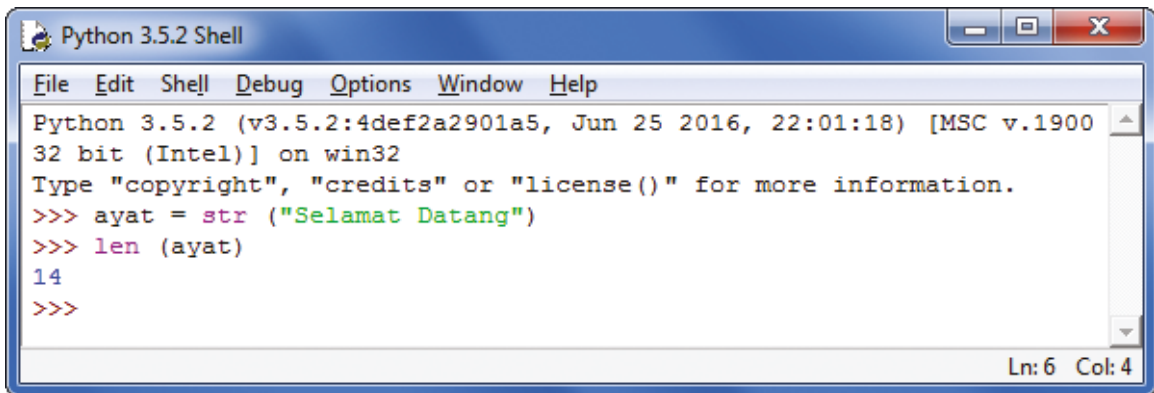
```

Arahan **print (ayat [4])** merujuk kepada aksara kelima dalam *string*, iaitu aksara "m". Nombor dalam tanda [] disebut nombor indeks.

Arahan **print (ayat [10 : 14])** merujuk kepada aksara dalam lingkungan ke-10 hingga ke-14 dalam *string*, iaitu "tang".

Contoh 3.7

Kod arahan bahasa pengaturcaraan Python untuk menghitung bilangan aksara dalam *string*.

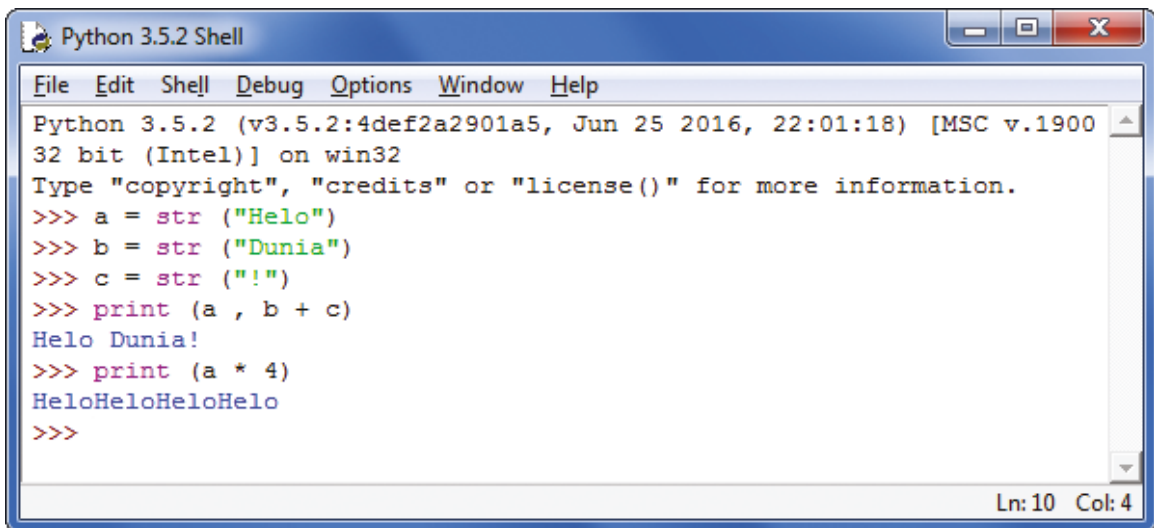


```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900
32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ayat = str ("Selamat Datang")
>>> len (ayat)
14
>>>
```

Ruang kosong dalam suatu *string* juga diambil kira sebagai satu aksara.

Contoh 3.8

Kod arahan bahasa pengaturcaraan Python untuk menggabungkan dua *string* dan mengulang *string*.



```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900
32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> a = str ("Helo")
>>> b = str ("Dunia")
>>> c = str ("!")
>>> print (a , b + c)
Helo Dunia!
>>> print (a * 4)
HeloHeloHeloHelo
>>>
```

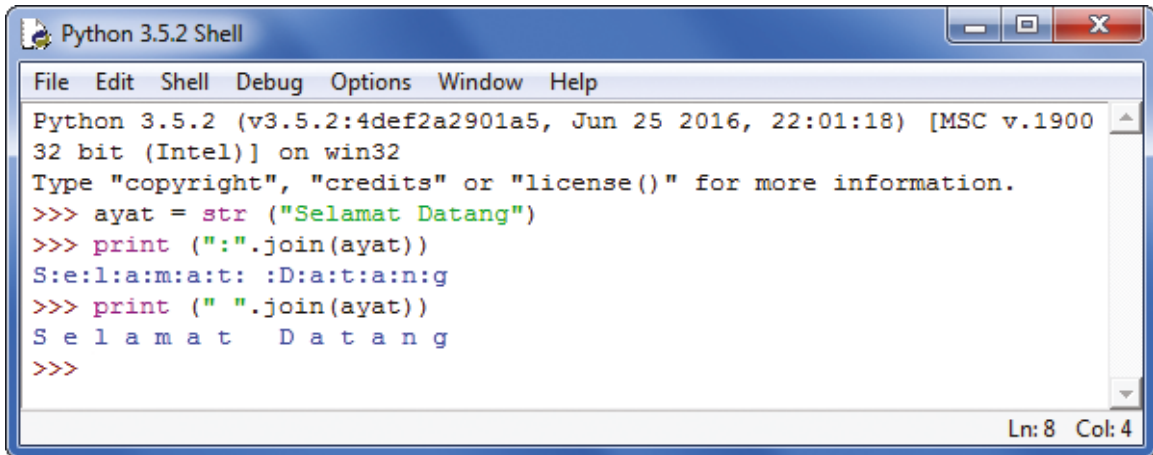
Simbol , menggabungkan dua *string* dengan satu ruang kosong.

Simbol + menggabungkan dua *string* tanpa sebarang ruang kosong.

Simbol * mengulang *string* dengan bilangan yang dinyatakan, iaitu 4.

Contoh 3.9

Kod arahan bahasa pengaturcaraan Python untuk menambahkan aksara di antara setiap aksara.



```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900
32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ayat = str ("Selamat Datang")
>>> print (":".join(ayat))
S:e:l:a:m:a:t: :D:a:t:a:n:g
>>> print (" ".join(ayat))
S e l a m a t   D a t a n g
>>>
```

Fungsi join () menambahkan aksara yang dinyatakan di antara setiap aksara dalam suatu *string*.

Boolean

Boolean berasal daripada nama George Boole yang merupakan seorang ahli matematik British. Beliau telah membangunkan satu sistem logik untuk memudahkan satu pernyataan yang kompleks kepada satu persamaan yang mudah. Jenis data *boolean* mempunyai hanya dua nilai, iaitu Benar dan Palsu. Dua nilai ini juga digelar nilai logik. Nilai logik *boolean* sering kali disebut 0 dan 1 di mana 0 mewakili Palsu dan 1 mewakili Benar.



Dalam bahasa pengaturcaraan Python, Palsu akan dipaparkan sebagai **False** dan Benar akan dipaparkan sebagai **True**.



Fakta Segera

Pernyataan *boolean* juga digunakan dalam pembangunan alatan elektronik yang lebih kompleks yang memerlukan satu keputusan yang lebih jitu. Contohnya alat sensor, suatu atur cara yang berkait rapat dengan operator logik dan kegunaan pernyataan *boolean* yang sesuai digunakan.

Nilai logik Benar atau Palsu amat penting dan digunakan sebagai pernyataan bersyarat dalam struktur kawalan pilihan dan struktur kawalan ulangan. Pernyataan bersyarat ini disebut ungkapan *boolean*. Ungkapan *boolean* ialah ungkapan perbandingan yang terdiri daripada operator-operator perbandingan seperti (>, >=, <, <=).

Contoh 3.10

Penggunaan *boolean* dalam segmen kod dengan menggunakan perisian aplikasi **Scratch 2.0**.



Blok yang berbentuk heksagon digunakan sebagai syarat dalam sesuatu struktur kawalan pilihan.

Dalam segmen kod di atas, pemboleh ubah *x* diumpukkan dengan nilai 14. Jika pengguna memasukkan nombor 14, syarat struktur kawalan pilihan menjadi benar, maka Benar dipaparkan. Jika pengguna memasukkan nombor selain 14, syarat struktur kawalan pilihan menjadi tidak benar, maka Palsu dipaparkan.

Contoh 3.11

Paparan logik *boolean*.

Teks **True** dan **False** yang berwarna biru ialah paparan yang diperoleh apabila ungkapan *boolean* ditaip masuk ke **IDLE Python 3.5.2**.

The screenshot shows the Python 3.5.2 Shell window. The command prompt shows the following interactions:

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900
32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> x = 13
>>> x > 3
True
>>> x < 0
False
>>>
```

The status bar at the bottom right indicates 'Ln: 8 Col: 4'.

Penerangan:

Simbol $=$ memberi maksud sama dengan. Oleh itu, $x = 13$ bermaksud nilai 13 diberi kepada x .

Simbol $>$ memberi maksud lebih besar. Oleh itu, $x > 3$ bermaksud x lebih besar daripada 3. Pernyataan ini adalah benar, maka **True** dipaparkan.

Simbol $<$ memberi maksud lebih kecil. Oleh itu, $x < 0$ bermaksud x lebih kecil daripada 0. Pernyataan ini adalah palsu, maka **False** dipaparkan.

Aktiviti 3.2

Kumpulan

Mengelaskan jenis data.

1. Pernyataan berikut menunjukkan data-data untuk profil seorang murid.

Nama: Nurul Aini
Kelas: 2 Aman
Jumlah subjek: 8
Purata markah: 75.8
Adakah purata markah lulus? Ya

2. Bincang bersama-sama ahli kumpulan anda tentang jenis data yang sesuai untuk data-data di atas.
3. Selain contoh profil murid di atas, berikan dua contoh lain yang serupa dengannya dan ulangi langkah 2.
4. Susunkan hasil perbincangan kumpulan anda dalam bentuk lembaran grafik yang kemas.

```
>>> x = 2
>>> 1 < x < 3
True
>>> 10 < x < 20
False
>>> 3 > x <= 2
True
```

Dengan adanya fungsi True dan False dalam sesuatu segmen kod, perbandingan dalam satu ungkapan persamaan dapat dilihat dengan mudah.



RUMUSAN JENIS DATA

Sintaks:
`int`

Contoh:

```
>>> int (10)
10
```

Penggunaan:

1. Menyelesaikan masalah matematik dan boleh digunakan bersama operator matematik
2. Pembilang bagi struktur kawalan

Definisi:
Nombor bulat tanpa titik perpuluhan. Termasuk nombor positif dan negatif.

Integer

JENIS DATA

Definisi:
Satu jujukan susunan aksara.

Sintaks:
`str, string`

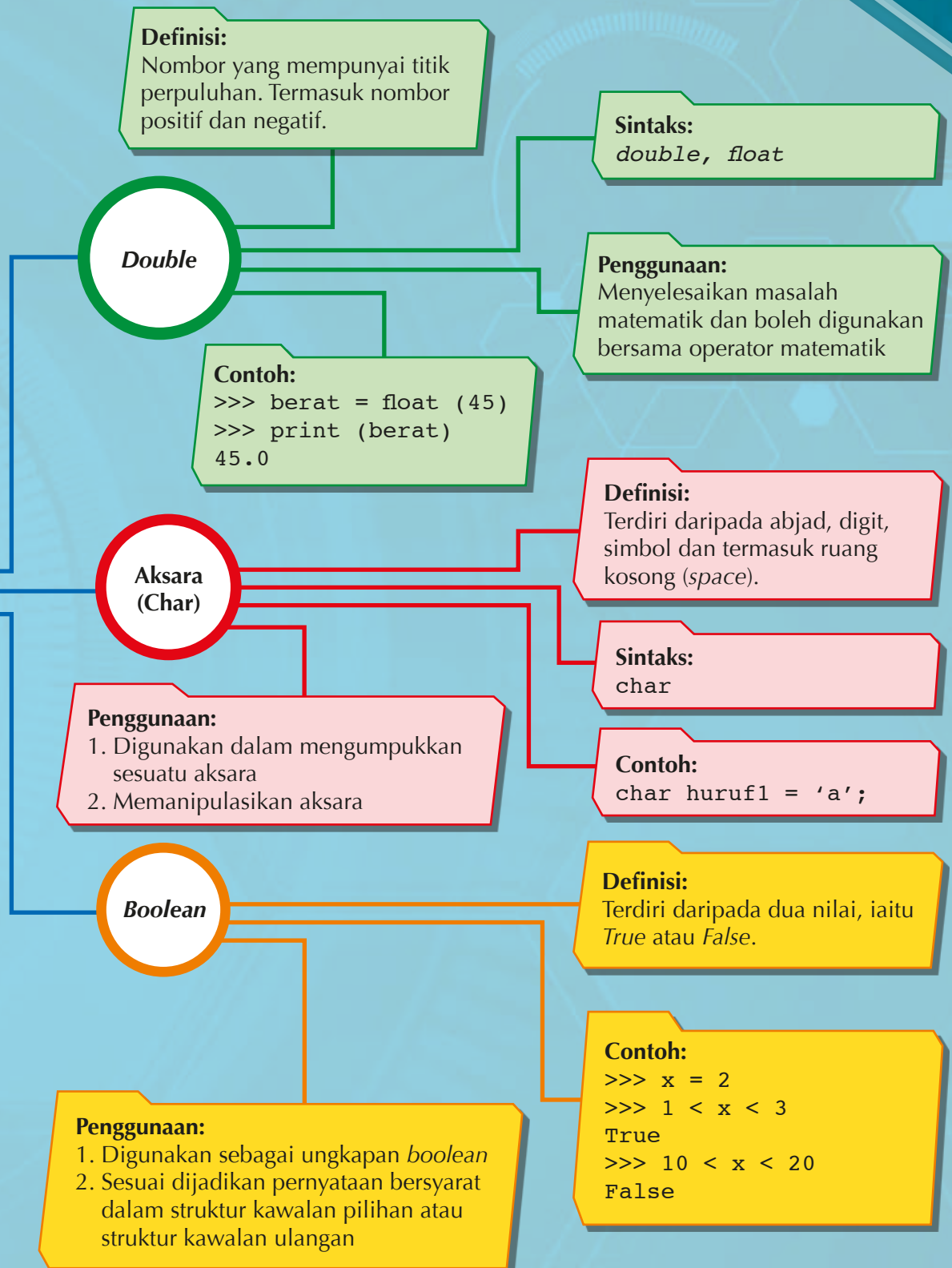
Contoh:

```
>>> print (str ("123abc"))
123abc
```

**Rentetan
(string)**

Penggunaan:

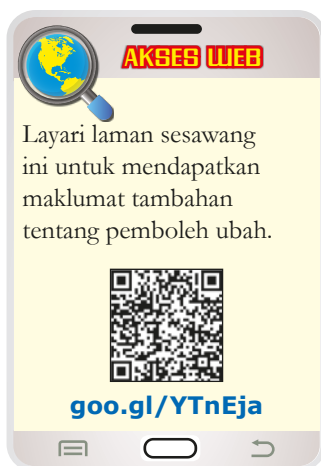
1. Aksara dalam *string* boleh dimanipulasi
2. Antara fungsi yang digunakan untuk memanipulasi *string* ialah:
 - tanda siku (nombor indeks)
 - `len ()`
 - `join ()`
 - simbol tambah (+), simbol koma (,) dan simbol berbentuk bintang (*)



3.1.2 Perbezaan antara pemboleh ubah dan pemalar dalam segmen kod

Standard Pembelajaran

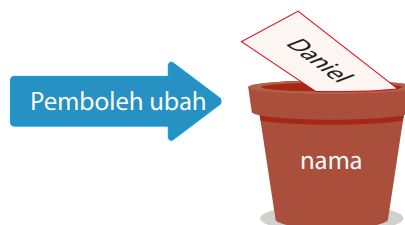
Murid boleh:
3.1.2 Membezakan antara pemboleh ubah dan pemalar dalam segmen kod.



Pemboleh ubah

Pemboleh ubah ialah storan ingatan dalam komputer yang digunakan oleh atur cara. Dalam konteks pengaturcaraan komputer, pemboleh ubah digunakan untuk mewakili sesuatu data yang perlu digunakan berulang kali oleh atur cara yang dibangunkan. Pengatur cara dapat memanipulasi data yang disimpan dalam pemboleh ubah mengikut kod arahan yang dibangunkan.

Semasa penulisan kod arahan, pemboleh ubah yang diisytiharkan akan diberi satu label yang disebut **pengecam** yang sesuai dan disertakan dengan **nilai**. Pengecam suatu pemboleh ubah perlu dilabelkan dengan nama yang bersesuaian bagi membolehkan pengecam dapat dirujuk semula dalam kod arahan dengan mudah. Rajah 3.8 mengilustrasikan pembentukan pemboleh ubah semasa diisytiharkan dalam segmen kod.



Rajah 3.8 Pembentukan pemboleh ubah semasa diisytiharkan dalam segmen kod

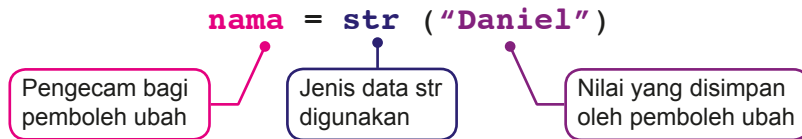
Pengecam 'nama' digunakan sebagai pemboleh ubah yang diisytiharkan dan nilai yang diberi ialah 'Daniel'. Pengecam yang diberikan kepada pemboleh ubah boleh diubah apabila perlu dalam kod arahan.

Semasa pengisytiharan pemboleh ubah, ruang ingatan dalam komputer ditempah berdasarkan jenis data yang diberi kepada pemboleh ubah. Pentafsir akan memperuntukkan tempat dan menetapkan saiz ingatan yang boleh digunakan dalam ingatan komputer yang telah dikhaskan.



Adakalanya pemboleh ubah yang sukar dikenal pasti digunakan bagi mengelakkan segmen kod mudah dibaca oleh penggodam.

Dalam bahasa pengaturcaraan Python, perisytiharan berlaku secara automatik apabila anda menetapkan nilai kepada pengecam suatu pemboleh ubah. Tanda sama dengan (=) digunakan untuk menetapkan nilai kepada pemboleh ubah.



Rajah 3.9 Pengisytiharan suatu pemboleh ubah dalam segmen kod

Syarat untuk membuat pengisytiharan pemboleh ubah adalah seperti berikut:

- ✓ Nama bagi pemboleh ubah hanya boleh mengandungi aksara (a hingga z atau A hingga Z) dan digit (0 hingga 9).
- ✓ Nama pemboleh ubah harus menggambarkan nilai yang dipegang oleh pemboleh ubah tersebut supaya kod-kod atur cara yang dibangunkan dapat difahami dengan mudah.
- ✗ Nama bagi pemboleh ubah tidak boleh dimulakan dengan nombor.
- ✗ Nama bagi pemboleh ubah tidak boleh ada ruang kosong.

Contoh 3.12

Pengisytiharan pemboleh ubah dengan menggunakan bahasa pengaturcaraan Python.

The screenshot shows a Python 3.5.2 Shell window. The code entered is:

```
>>> nama = "Daniel"
>>> print("Nama saya:", nama)
Nama saya: Daniel
>>>
```

A callout box points to the first line of code, stating: "Pemboleh ubah dengan pengecam 'nama' dan diberi nilai 'Daniel'". The status bar at the bottom right shows "Ln: 6 Col: 4".

Pemalar

Pemalar bertindak seakan-akan pemboleh ubah, iaitu pemalar menyimpan sesuatu nilai dan nilai ini disimpan dalam ingatan komputer. Perbezaan utama ialah nilai sesuatu pemalar tidak akan berubah menjadi nilai yang lain semasa pelaksanaan atur cara. Lazimnya, pemalar digunakan dalam pengaturcaraan yang melibatkan pengiraan matematik. Contohnya, semasa pengiraan keluasan sesuatu bulatan, nilai Pi (π) harus diisytiharkan sebagai pemalar supaya nilai ini kekal sama dalam atur cara yang dibangunkan.

Contoh 3.13

Penggunaan pemalar untuk mengira luas bulatan.

```

Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900
32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> pi = 3.142
>>> jejari_1 = 3
>>> jejari_2 = 5
>>> luas_bulatan_1 = pi * (jejari_1 * jejari_1)
>>> luas_bulatan_2 = pi * (jejari_2 * jejari_2)
>>> print(luas_bulatan_1)
28.278
>>> print(luas_bulatan_2)
78.55
>>>
Ln: 12 Col: 4

```

- 1 Pengisytiharan penggunaan nilai pemalar jenis pi.
- 2 Mengisytiharkan pemboleh ubah, iaitu ukuran jejari bulatan.
- 3 Formula luas bulatan = $\pi \times \text{jejari}^2$.
- 4 Cetak pemboleh ubah luas_bulatan_2.

Jadual 3.4 Perbezaan antara pemboleh ubah dan pemalar

Ciri-ciri	Pemboleh ubah	Pemalar
Perubahan nilai	Mempunyai nilai yang boleh berubah-ubah	Mempunyai nilai yang tetap dan tidak berubah
Nilai yang perlu diingat	Pengaturcara program perlu mengingat label dan nilai yang digunakan	Pengaturcara tidak perlu untuk mengingat label dan nilai yang digunakan
Standard penggunaan nilai	Tiada nilai standard	Nilai yang standard



Fakta Segera

Kebanyakan nilai pemalar untuk pengiraan telah disimpan dalam pustaka sokongan sesuatu bahasa pengaturcaraan. Nilai pemalar ini boleh digunakan setelah pustaka sokongan diaktifkan.

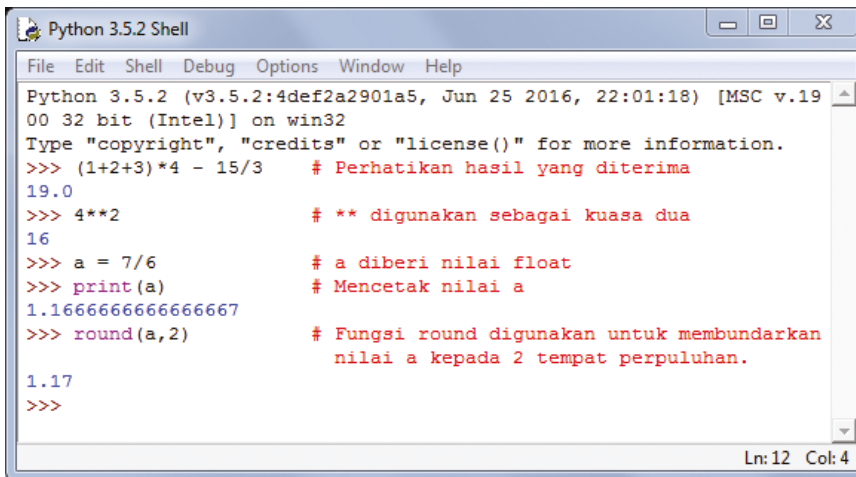
Amali Komputer 3.1 akan membimbing anda menulis segmen kod yang melibatkan penggunaan pelbagai jenis data, pemboleh ubah, pemalar dan operator matematik.



Amali Komputer 3.1

A Jenis data untuk nombor

1. Lancarkan perisian aplikasi **IDLE Python 3.5.2**.
2. Taip segmen kod yang ditunjukkan di bawah ke dalam tetingkap **IDLE Python 3.5.2** yang dipaparkan.



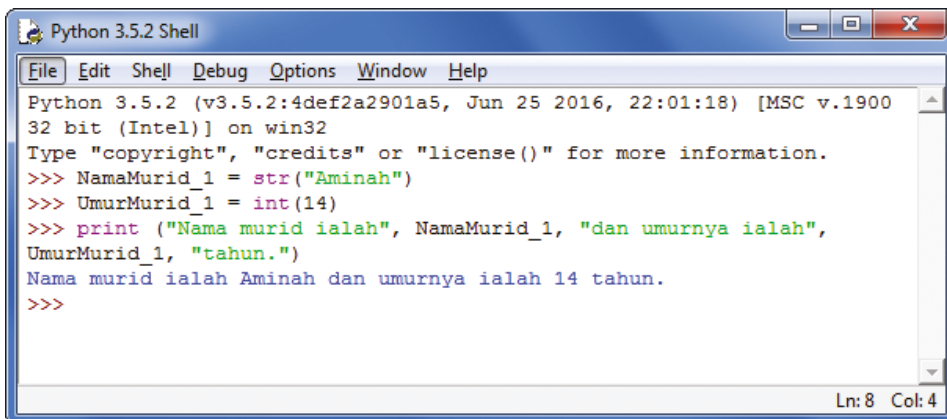
```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> (1+2+3)*4 - 15/3      # Perhatikan hasil yang diterima
19.0
>>> 4**2                  # ** digunakan sebagai kuasa dua
16
>>> a = 7/6                # a diberi nilai float
>>> print(a)               # Mencetak nilai a
1.1666666666666667
>>> round(a,2)            # Fungsi round digunakan untuk membundarkan
                           nilai a kepada 2 tempat perpuluhan.

1.17
>>>
```

3. Simpan fail sebagai **nombor.py** dan tutup aplikasi.

B Jenis data untuk teks

1. Lancarkan perisian aplikasi **IDLE Python 3.5.2**.
2. Taip segmen kod yang ditunjukkan di bawah ke dalam tetingkap **IDLE Python 3.5.2** yang dipaparkan.



```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> NamaMurid_1 = str("Aminah")
>>> UmurMurid_1 = int(14)
>>> print("Nama murid ialah", NamaMurid_1, "dan umurnya ialah",
UmurMurid_1, "tahun.")
Nama murid ialah Aminah dan umurnya ialah 14 tahun.
>>>
```

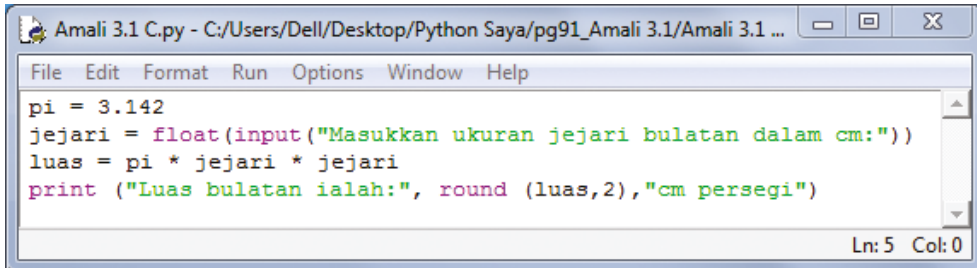
3. Simpan fail sebagai **teks.py** dan tutup aplikasi.

C Menyelesaikan masalah pengiraan luas bulatan

1. Lancarkan perisian aplikasi **IDLE Python 3.5.2** dan lancarkan satu dokumen baharu.

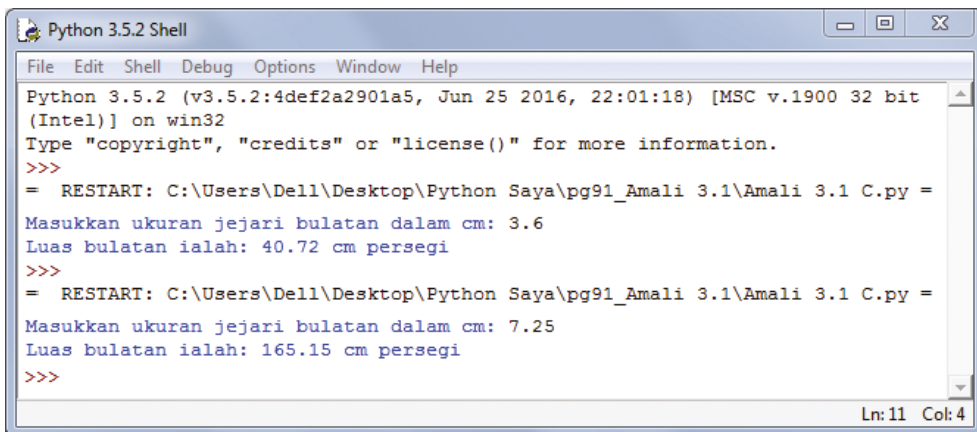
Windows Start Button > All Programs > IDLE (Python 3.5) > File > New File

2. Taip segmen kod yang ditunjukkan di bawah ke dalam tetingkap **IDLE Python 3.5.2** yang dipaparkan.



```
Amali 3.1 C.py - C:/Users/Dell/Desktop/Python Saya/pg91_Amali 3.1/Amali 3.1 ...
File Edit Format Run Options Window Help
pi = 3.142
jejari = float(input("Masukkan ukuran jejari bulatan dalam cm:"))
luas = pi * jejari * jejari
print ("Luas bulatan ialah:", round (luas,2),"cm persegi")
Ln: 5 Col: 0
```

3. Simpan fail sebagai **luas-bulatan.py**. Kemudian, klik menu **Run > Run Module**. Tetingkap seperti yang berikut akan dipaparkan.



```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit
(Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Dell\Desktop\Python Saya\pg91_Amali 3.1\Amali 3.1 C.py =
Masukkan ukuran jejari bulatan dalam cm: 3.6
Luas bulatan ialah: 40.72 cm persegi
>>>
= RESTART: C:\Users\Dell\Desktop\Python Saya\pg91_Amali 3.1\Amali 3.1 C.py =
Masukkan ukuran jejari bulatan dalam cm: 7.25
Luas bulatan ialah: 165.15 cm persegi
>>>
Ln: 11 Col: 4
```



Aktiviti 3.3

Individu

Menggunakan pemboleh ubah dan pemalar untuk menyelesaikan masalah matematik.

Berdasarkan Amali Komputer 3.1, tulis segmen kod untuk mengira isi padu sebuah sfera [formula = $\frac{3}{4}\pi j^3$]. Anda perlu menentukan pemboleh ubah bagi jejari, *j*. Nilai isi padu harus dibundarkan kepada empat tempat perpuluhan. Ayat berikut harus dipaparkan pada akhir atur cara anda.

Isi padu sfera yang berjejari __ ialah ____ sentimeter padu.

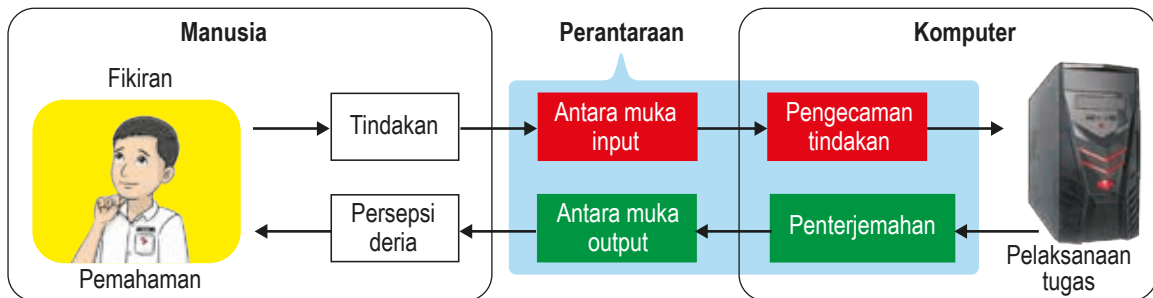
3.1.3 (i) Menghasilkan segmen kod menggunakan fungsi input dan fungsi output

Dalam contoh-contoh segmen kod yang lepas, terdapat fungsi yang memaparkan hasil yang dikehendaki pada skrin komputer. Bahasa pengaturcaraan memainkan peranan yang penting untuk mengawal dan memberi arahan kepada peranti-peranti yang lain. Dalam bahasa pengaturcaraan, fungsi input dan fungsi output merupakan kaedah interaktif yang digunakan dalam **Interaksi Manusia Komputer (IMK)**. Fungsi input dan fungsi output membolehkan mesej yang hendak disampaikan dan diterima dapat difahami oleh pengguna dan juga komputer.

Standard Pembelajaran

Murid boleh:

3.1.3 Menghasilkan segmen kod menggunakan (i) fungsi input dan output.



Rajah 3.10 Input dan output memainkan peranan penting dalam Interaksi Manusia Komputer (IMK)

IMK melibatkan cara manusia berkomunikasi dengan komputer dengan ciri-ciri perantaraan dalam aspek yang lebih luas. Faktor komputer adalah seperti teknik grafik komputer, sistem pengoperasian, bahasa pengaturcaraan, dan persekitaran pembangunan. Faktor manusia pula melibatkan teori komunikasi, bahasa, sains sosial, psikologi kognitif, psikologi sosial, dan perasaan manusia seperti kepuasan penggunaan komputer. Perantaraan merujuk kepada medium seperti antara muka suatu atur cara, warna, cara data dimasukkan dan reka bentuk grafik yang mesra pengguna supaya komunikasi antara manusia dengan komputer dapat berlaku dengan betul dan tepat.

Fungsi input

Fungsi input digunakan untuk mendapatkan data input daripada pengguna. Proses ini dapat dilakukan melalui segmen kod yang dibina. Apabila fungsi input diaktifkan, atur cara akan meminta pengguna untuk memasukkan data mengikut kesesuaian atur cara yang dibina.

Dalam proses menginput data, papan kekunci memainkan peranan penting bagi memastikan data yang dimasukkan adalah seperti yang terkandung dan dipatuhi dalam segmen kod. Sintaks bagi fungsi input juga berbeza mengikut bahasa pengaturcaraan. Jadual 3.5 menunjukkan sintaks umum bagi fungsi input untuk beberapa bahasa pengaturcaraan.

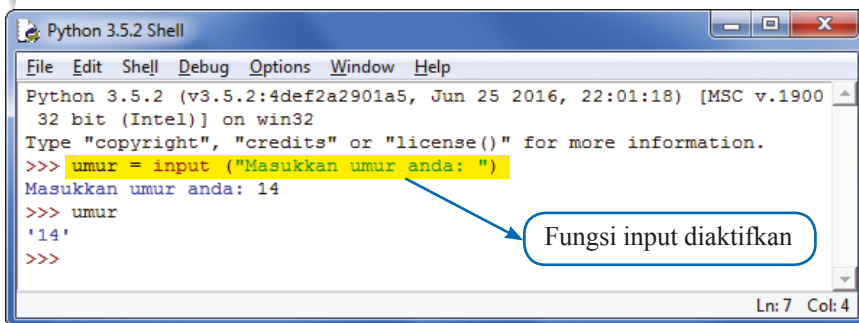


Jadual 3.5 Fungsi input dalam beberapa bahasa pengaturcaraan

Bahasa pengaturcaraan	Sintaks umum	Format segmen kod	Contoh
C++	cin >>	cin >> nama_ pemboleh_ubah;	int umur; cin >> umur;
C	scanf()	scanf (nama_ pemboleh_ubah);	int umur; scanf (umur);
Java	System.in	Scanner nama pemboleh ubah = objek (System.in);	Scanner umur = new Scanner(System.in);
Python	input ()	input (nama_ pemboleh_ubah)	>>> umur = input ("Umur: ")
Scratch			

Contoh 3.14

Penggunaan fungsi input dalam bahasa pengaturcaraan Python.



Apabila sintaks input () dibaca, atur cara akan menunggu input daripada pengguna.

Fungsi output

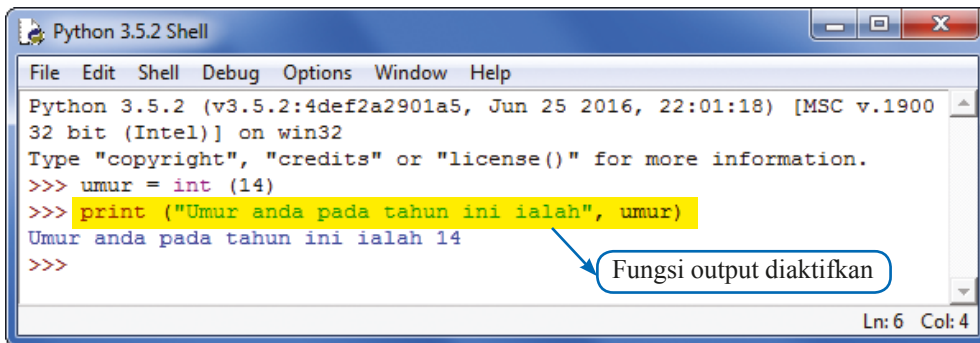
Fungsi output merujuk kepada maklumat yang dipaparkan pada skrin komputer. Maklumat yang dipaparkan ialah hasil daripada proses yang dilakukan oleh segmen kod ataupun data yang dimasukkan oleh pengguna melalui fungsi input. Fungsi output akan diaktifkan oleh sintaks umum mengikut bahasa pengaturcaraan yang digunakan.

Monitor ialah peranti output yang biasa digunakan. Maklumat yang dipaparkan pada skrin monitor ialah maklumat dalam bentuk mesej ataupun hasil daripada proses yang dilakukan oleh segmen kod. Jadual 3.6 menunjukkan sintaks umum bagi fungsi output untuk beberapa bahasa pengaturcaraan.

Jadual 3.6 Fungsi output dalam beberapa bahasa pengaturcaraan

Bahasa pengaturcaraan	Sintaks umum	Format segmen kod	Contoh
C++	cout<<	cout << nama_pemboleh_ubah;	cout << "Mesej"; atau cout << umur;
C	printf()	printf (nama_pemboleh_ubah);	printf ("Mesej"); atau printf (umur);
Java	System.out	System.out.println	System.out.println ("Mesej"); atau System.out.println(umur);
Python	print ()	print (nama_pemboleh_ubah)	print ("Umur: ") atau print (umur)
Scratch			

Contoh 3.15 Penggunaan fungsi output dalam bahasa pengaturcaraan Python.



```

Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900
32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> umur = int(14)
>>> print("Umur anda pada tahun ini ialah", umur)
Umur anda pada tahun ini ialah 14
>>>

```

Apabila sintaks `print ( )` dibaca, atur cara akan memaparkan teks yang berada dalam tanda petikan (" "). Tanda petikan tidak perlu untuk mencetak nilai suatu pembolehubah.

3.1.3 (ii) Operator perbandingan

Standard Pembelajaran

Murid boleh:

- 3.1.3 Menghasilkan segmen kod menggunakan (ii) operator perbandingan.

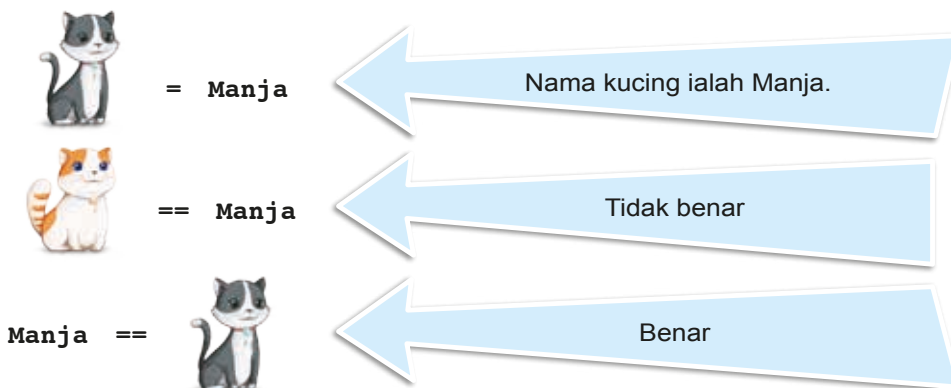
Operator perbandingan digunakan untuk membandingkan nilai di sebelah kiri operator dengan nilai di sebelah kanan operator. Hasil kepada hubungan perbandingan ialah nilai logik, iaitu Benar atau Palsu. Operator perbandingan digunakan dalam ungkapan *boolean* untuk pernyataan bersyarat bagi struktur kawalan pilihan dan struktur kawalan ulangan.

Operator perbandingan juga dikenali sebagai pengendali perhubungan. Terdapat enam jenis operator perbandingan. Jadual 3.7 menerangkan dan memberikan contoh kegunaan operator-operator perbandingan ini.

Jadual 3.7 Operator perbandingan

Operator	Penerangan	Contoh a = 5 dan b = 10
==	Jika nilai dua operan adalah sama, maka keadaan menjadi benar.	(a == b) adalah tidak benar. (a == 5) adalah benar.
!=	Jika nilai bagi dua operan tidak sama, maka keadaan menjadi benar.	(a != b) adalah benar. (a != 5) adalah tidak benar.
>	Jika nilai operan kiri adalah lebih besar daripada nilai operan kanan, maka keadaan menjadi benar.	(a > b) adalah tidak benar. (b > a) adalah benar.
<	Jika nilai operan kiri adalah kurang daripada nilai operan kanan, maka keadaan menjadi benar.	(a < b) adalah benar. (b < a) adalah tidak benar.
>=	Jika nilai operan kiri adalah lebih besar daripada atau sama dengan nilai operan kanan, maka keadaan menjadi benar.	(a >= b) adalah tidak benar. (b >= a) adalah benar.
<=	Jika nilai operan kiri adalah kurang daripada atau sama dengan nilai operan kanan, maka keadaan menjadi benar.	(a <= b) adalah benar. (b <= a) adalah tidak benar.

!= juga boleh ditulis sebagai <>, akan tetapi penggunaannya semakin kurang.



Contoh 3.16

Penggunaan operator perbandingan dalam segmen kod Python.

```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32
bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> a = 8
>>> b = 10
>>> a > b
False
>>> b = a      # Operator = bermaksud memberi nilai operan kanan kepada
                operan kiri
>>> a > b
False
>>> b == a
True
>>>
```



Jika ungkapan *boolean* adalah benar, **True** akan dipaparkan.
Jika ungkapan *boolean* adalah salah, **False** akan dipaparkan.

Hasil bagi setiap ungkapan yang menggunakan operator perbandingan bersamaan dengan nilai logik. Oleh yang demikian, operator perbandingan adalah penting dan sesuai digunakan dalam ungkapan *boolean* sebagai pernyataan bersyarat dalam struktur kawalan pilihan dan struktur kawalan ulangan.

3.13 (iii) Operator logik

Operator logik digunakan dalam sesuatu ungkapan algebra ringkas untuk mendapatkan nilai *boolean*, iaitu Benar atau Palsu yang juga disebut nilai logik. Dalam pengaturcaraan, segmen kod yang mengandungi ungkapan algebra bersama operator logik penting supayaatur cara dapat melakukan perbandingan yang mudah terutamanya sebagai pernyataan bersyarat dalam struktur kawalan pilihan dan struktur kawalan ulangan. Jadual 3.8 menunjukkan tiga operator logik, penerangan dan contoh penggunaannya.

Standard Pembelajaran

Murid boleh:
3.1.3 Menghasilkan segmen kod menggunakan (iii) operator logik.

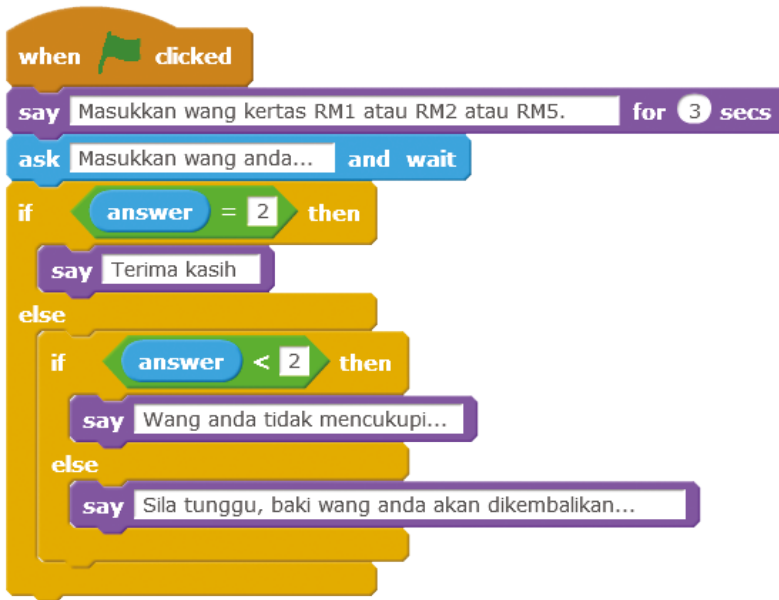
Jadual 3.8 Operator logik, penerangan dan contoh

Operator logik	Penerangan	Contoh (operan a = benar, operan b = palsu)
AND (logik DAN)	Keadaan menjadi benar jika kedua-dua operan adalah benar.	(a AND b) adalah palsu
OR (logik ATAU)	Keadaan menjadi benar jika salah satu operan adalah benar.	(a OR b) adalah benar
NOT (logik TIDAK)	Menyongsangkan seluruh keadaan logik operan.	NOT (a AND b) adalah benar

Contoh 3.17

Pembelian minuman ringan menggunakan mesin layan diri.

Aini diminta membangunkan segmen kod yang membolehkan mesin layan diri menerima wang kertas RM1, RM2 dan RM5 sahaja. Harga kesemua minuman ringan yang dijual adalah RM2. Jika wang kertas RM1 diterima, mesej “Wang anda tidak mencukupi...” akan dipaparkan. Jika wang kertas RM2 diterima, mesej “Terima kasih” akan dipaparkan. Jika wang kertas RM5 diterima, mesej “Sila tunggu, baki wang anda akan dikembalikan...” akan dipaparkan.



Adakalanya, atur cara perlu menjalankan sesuatu operasi matematik yang lebih kompleks untuk menentukan nilai baki, hasil bahagi dan kuasa sesuatu nombor. Maka, operator-operator tambahan bagi matematik seperti yang ditunjukkan dalam Jadual 3.9 diperlukan.

Jadual 3.9 Operator-operator tambahan bagi matematik

Operator	Nama operator	Penerangan	Contoh a = 10 dan b = 2
%	Modulus	Operan kiri dibahagi dengan operan kanan. Baki pembahagian dipaparkan.	Contoh 1: a % b = 0 Contoh 2: 11 % 2 = 1
**	Eksponen	Operan kiri dikuasakan mengikut nilai operan kanan.	Contoh 1: a ** b = 10 ² Contoh 2: 11 ** 2 = 11 ²
//	Floor Division	Operan kiri dibahagi dengan operan kanan. Hasil pembahagian tanpa nilai baki dipaparkan.	Contoh 1: a // b = 5 Contoh 2: 11 // 2 = 5



Aktiviti 3.4

Kumpulan

Membina segmen kod ujian soalan matematik yang interaktif secara sistematik.

1. Lihat hasil paparan atur cara yang berikut. Teks hitam ialah input pengguna. **False** dan **True** yang dipaparkan selepas input pengguna ialah semakan jawapan yang dilakukan oleh atur cara.

```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Dell\Desktop\Python Saya\soalan matematik.py =====
Jawab soalan-soalan matematik berikut.

Soalan 1
Tentukan sama ada 6 ialah faktor bagi 96. Jawab Ya atau Tidak.
Jawapan:Tidak
False

Soalan 2
Hitungkan 1 - 3 x 5/24. Beri jawapan dalam 3 titik perpuluhan.
Jawapan:0.375
True

Soalan 3
Adakah ungkapan berikut 'Betul' atau 'Salah'.
6+(-7) < 3-(-4)
Jawapan:Betul
True
>>>
```

2. Lakukan sumbang saran bersama-sama rakan untuk membincangkan perkara-perkara yang berikut.
 - (a) Aliran atur cara
 - (b) Tindakan atur cara apabila jawapan yang betul atau yang salah dimasukkan
 - (c) Cara semakan jawapan setiap soalan yang dipaparkan
 - (d) Jenis data yang digunakan
 - (e) Fungsi yang digunakan
 - (f) Pemboleh ubah dan pemalar yang digunakan
 - (g) Operator yang digunakan
3. Setiap kumpulan perlu menulis segmen kod yang bersesuaian untuk mendapatkan hasil pada langkah 1.
4. Uji segmen kod yang telah ditulis dengan perisian aplikasi **IDLE Python 3.5.2**. Baiki ralat yang dijumpai, jika ada.

Standard Pembelajaran

Murid boleh:

3.1.4 Mengesan dan membaiki ralat pada segmen kod yang dihasilkan dalam penyelesaian masalah.

3.1.4 Mengesan dan membaiki ralat pada segmen kod dalam penyelesaian masalah

Ralat ditakrifkan sebagai kesilapan atau kesalahan dalam proses pembangunan sesuatu atur cara atau segmen kod. Terdapat tiga jenis ralat, iaitu ralat sintaks, ralat masa larian dan ralat logik yang telah diterangkan dalam Bab 2. Penerangan dalam subtopik ini memberi penekanan kepada cara mengesan dan membaiki ralat pada segmen kod.



IMBAS KEMBALI

Penjelasan untuk jenis-jenis ralat telah diterangkan dalam Bab 2.

Ralat sintaks

Ralat sintaks lazimnya berlaku disebabkan oleh kesilapan dalam penggunaan bahasa pengaturcaraan. Antara ralat sintaks adalah seperti berikut:

- (a) Penggunaan sintaks yang tidak betul, (`printf` adalah tidak betul, kerana `printf` adalah penulisan untuk bahasa pengaturcaraan C++).
- (b) Perkataan sintaks dieja dengan tidak betul.
- (c) Tertinggal atau salah menggunakan simbol seperti tanda titik bertindih, tanda petikan dan tanda kurung.
- (d) Cara penggunaan sintaks yang tidak betul seperti meletakkan sintaks di tempat yang salah. Contohnya [`>>> y print`] adalah salah.
- (e) Penggunaan inden yang salah.

Contoh 3.18

Pengesanan dan membaiki ralat sintaks pada segmen kod yang dihasilkan untuk penyelesaian masalah.

Pernyataan masalah:

Rosli dikehendaki menghasilkan satu segmen kod yang dapat mengira purata ketinggian bagi setiap ahli kumpulan dalam unit meter. Setiap kumpulan terdiri daripada lima orang murid. Setiap murid perlu memasukkan ukuran ketinggian masing-masing. Kemudian, purata ketinggian bagi ahli-ahli kumpulan dikira. Segmen kod akan memaparkan ketinggian setiap ahli berserta dengan purata ketinggian yang dibundarkan kepada dua tempat perpuluhan.

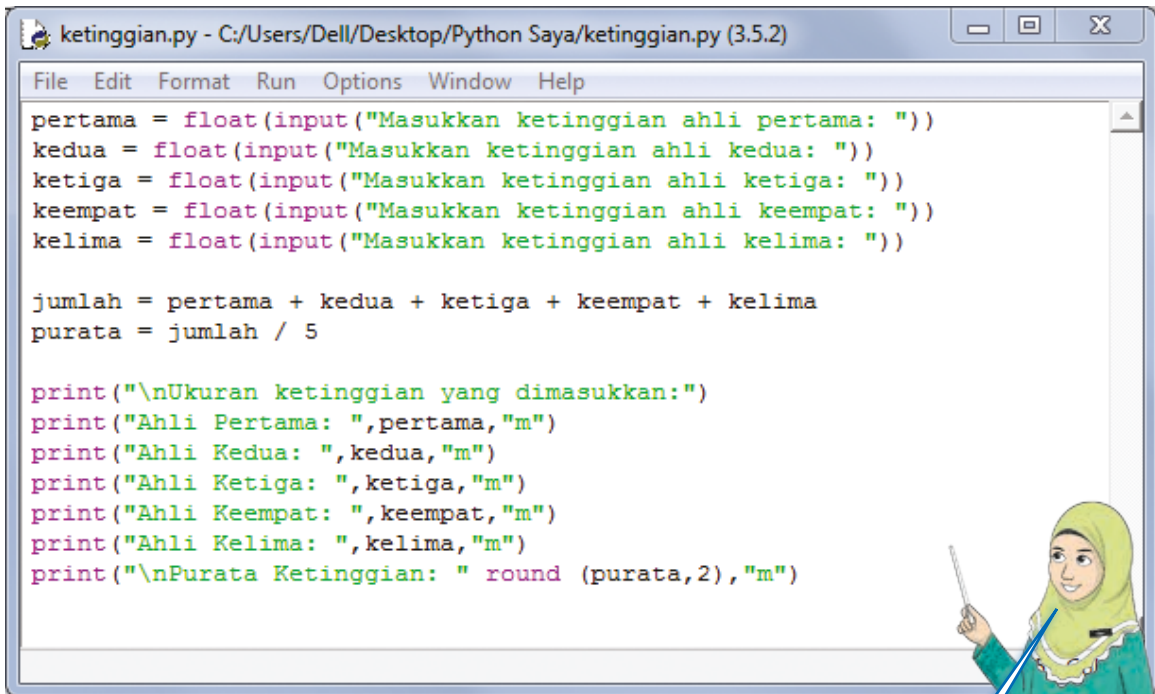
Output yang diinginkan:

Berdasarkan kepada pernyataan masalah tersebut, output yang diinginkan perlulah memaparkan ukuran ketinggian daripada setiap ahli kumpulan dan pengiraan purata ketinggian.

Contoh output:

```
Ukuran ketinggian yang dimasukkan:  
Ahli Pertama: 14.8 m  
Ahli Kedua: 15.6 m  
Ahli Ketiga: 16.3 m  
Ahli Keempat: 15.9 m  
Ahli Kelima: 16.8 m  
  
Purata Ketinggian: 15.88 m
```

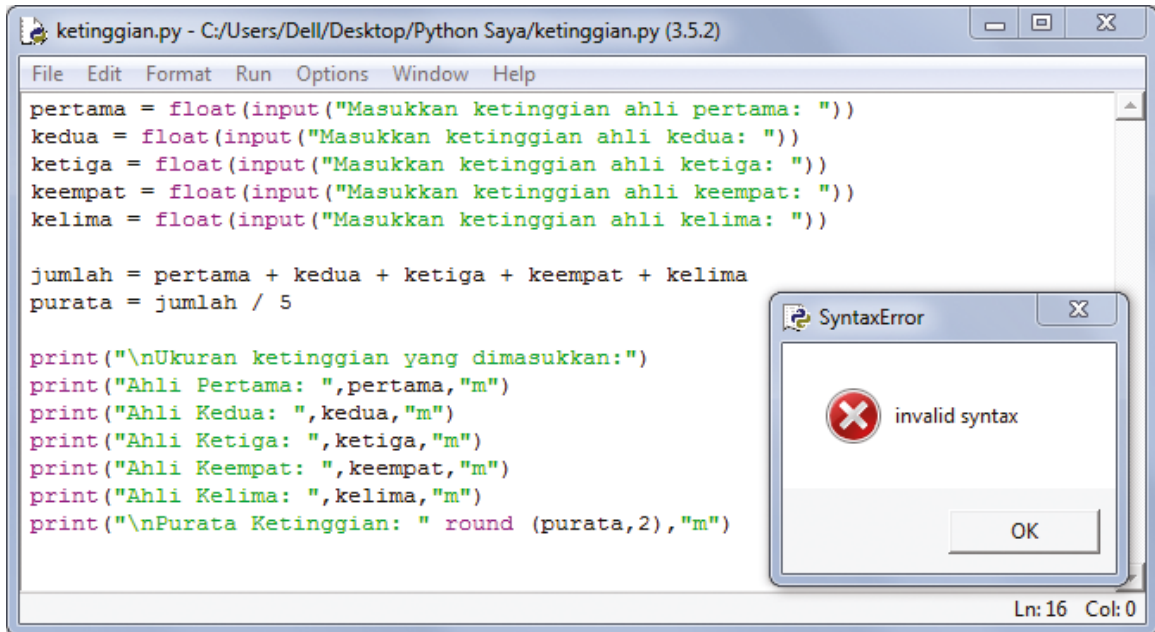
Segmen kod yang dihasilkan oleh Rosli:



```
File Edit Format Run Options Window Help  
pertama = float(input("Masukkan ketinggian ahli pertama: "))  
kedua = float(input("Masukkan ketinggian ahli kedua: "))  
ketiga = float(input("Masukkan ketinggian ahli ketiga: "))  
keempat = float(input("Masukkan ketinggian ahli keempat: "))  
kelima = float(input("Masukkan ketinggian ahli kelima: "))  
  
jumlah = pertama + kedua + ketiga + keempat + kelima  
purata = jumlah / 5  
  
print("\nUkuran ketinggian yang dimasukkan:")  
print("Ahli Pertama: ",pertama,"m")  
print("Ahli Kedua: ",kedua,"m")  
print("Ahli Ketiga: ",ketiga,"m")  
print("Ahli Keempat: ",keempat,"m")  
print("Ahli Kelima: ",kelima,"m")  
print("\nPurata Ketinggian: " round (purata,2), "m")
```

Klik menu **Run > Run Module** untuk melaksanakan segmen kod yang dihasilkan. Ralat sintaks akan dikesan semasa pelaksanaan segmen kod.

Paparan mesej ralat apabila segmen kod yang dibangunkan dilaksanakan:



Terdapat dua jenis teknik semakan kualiti, iaitu teknik semakan meja dan teknik langkah demi langkah yang telah dipelajari semasa Tingkatan 1.

Segmen kod yang dibangunkan oleh Rosli mempunyai ralat sintaks. Ini adalah berikutan daripada mesej ralat yang terpapar. Rosli perlu melakukan semakan ke atas segmen kod di atas.

Pilihan teknik langkah demi langkah:

Bagi membuat penyemakan ralat sintaks seperti di atas, teknik langkah demi langkah (*step through*) boleh digunakan. Semakan dilakukan sebaris demi sebaris dalam segmen kod untuk memastikan segmen kod bebas daripada ralat.

Mengesan ralat dan memperbaiki ralat:

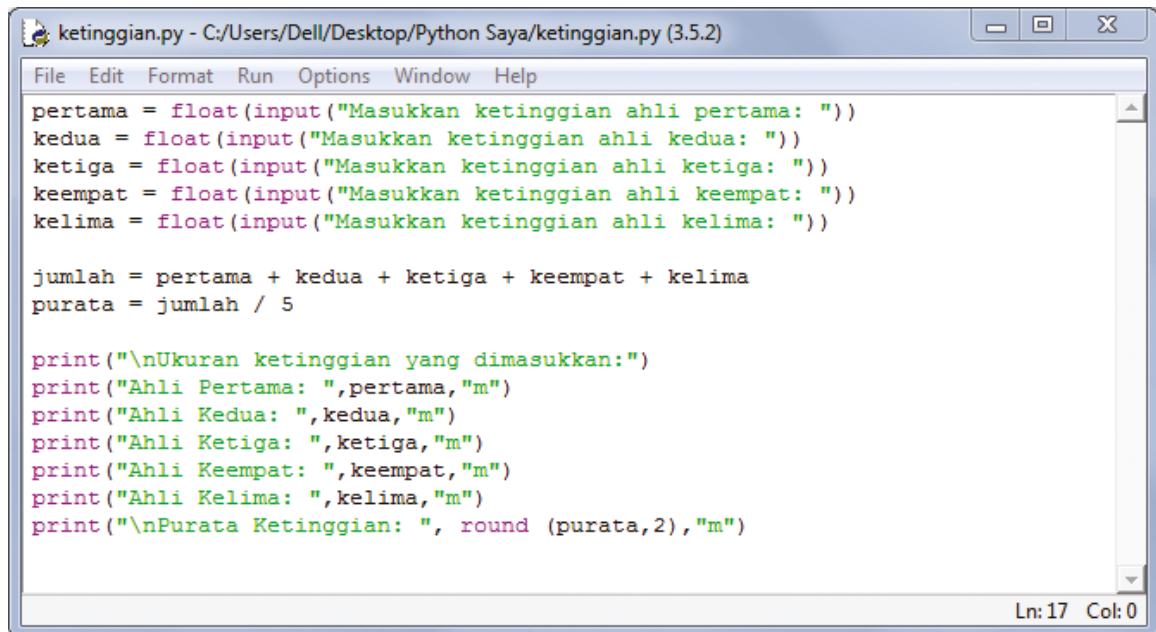
Hasil daripada teknik langkah demi langkah, didapati terdapat sebaris kod yang mengandungi ralat dan mengganggu pelaksanaan segmen kod yang telah dibangunkan. Berikut ialah kod baris yang mengandungi ralat yang perlu dibaiki.

```
print ("\nPurata Ketinggian: " round (purata,2), "m")
```

Ralat sintaks yang terkandung dalam segmen kod tersebut ialah, ketiadaan tanda koma (,) sebelum fungsi **round**. Berikut ialah kod baris yang telah dibaiki.

```
print ("\nPurata Ketinggian: ", round (purata,2), "m")
```

Segmen kod yang telah dibaiki:



```
ketinggian.py - C:/Users/Dell/Desktop/Python Saya/ketinggian.py (3.5.2)
File Edit Format Run Options Window Help

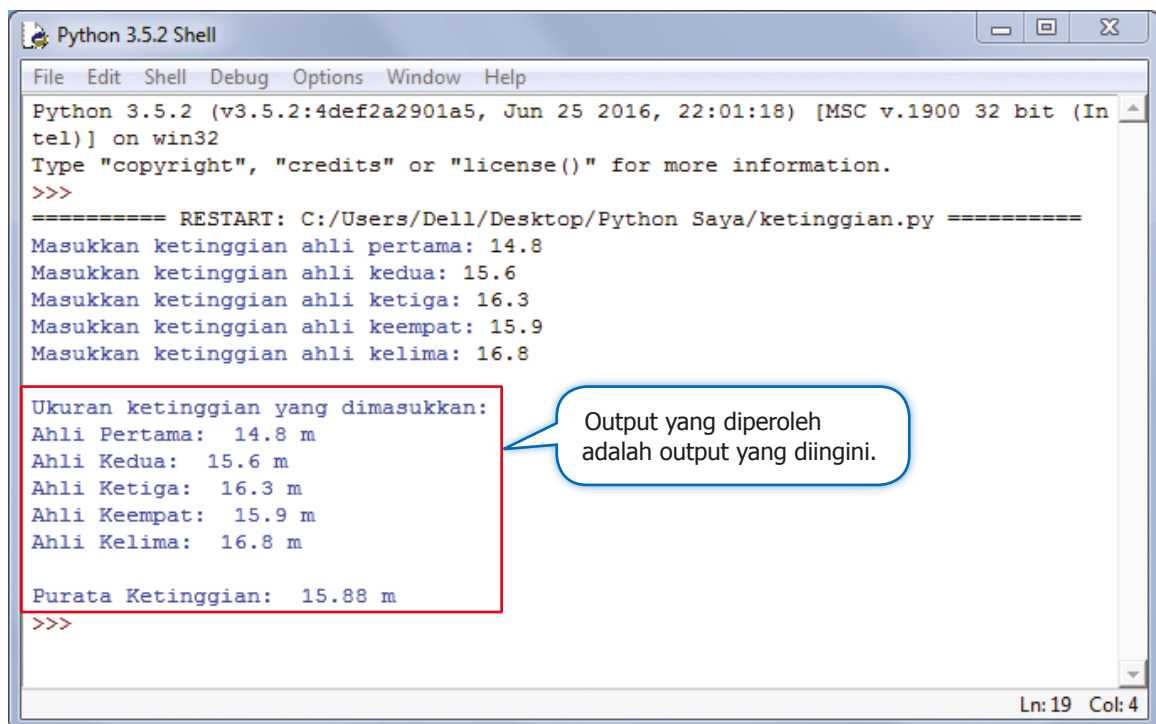
pertama = float(input("Masukkan ketinggian ahli pertama: "))
kedua = float(input("Masukkan ketinggian ahli kedua: "))
ketiga = float(input("Masukkan ketinggian ahli ketiga: "))
keempat = float(input("Masukkan ketinggian ahli keempat: "))
kelima = float(input("Masukkan ketinggian ahli kelima: "))

jumlah = pertama + kedua + ketiga + keempat + kelima
purata = jumlah / 5

print("\nUkuran ketinggian yang dimasukkan:")
print("Ahli Pertama: ", pertama, "m")
print("Ahli Kedua: ", kedua, "m")
print("Ahli Ketiga: ", ketiga, "m")
print("Ahli Keempat: ", keempat, "m")
print("Ahli Kelima: ", kelima, "m")
print("\nPurata Ketinggian: ", round(purata, 2), "m")

Ln: 17 Col: 0
```

Output yang diperoleh:



```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help

Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Dell/Desktop/Python Saya/ketinggian.py =====
Masukkan ketinggian ahli pertama: 14.8
Masukkan ketinggian ahli kedua: 15.6
Masukkan ketinggian ahli ketiga: 16.3
Masukkan ketinggian ahli keempat: 15.9
Masukkan ketinggian ahli kelima: 16.8

Ukuran ketinggian yang dimasukkan:
Ahli Pertama: 14.8 m
Ahli Kedua: 15.6 m
Ahli Ketiga: 16.3 m
Ahli Keempat: 15.9 m
Ahli Kelima: 16.8 m

Purata Ketinggian: 15.88 m
>>>

Ln: 19 Col: 4
```

Output yang diperoleh adalah output yang diinginkan.

Ralat masa larian

Kejadian ralat masa larian akan menyebabkan pelaksanaan segmen kod terhenti secara tiba-tiba tanpa diduga. Mesej ralat akan dipaparkan apabila ralat masa larian ditemui. Pernyataan-pernyataan berikut merupakan contoh kesilapan umum yang menyebabkan kejadian ralat masa larian.

- (a) Pembahagian dengan sifar.
- (b) Melaksanakan operasi untuk dua pemboleh ubah yang berlainan jenis data.
- (c) Menggunakan pemboleh ubah, fungsi, modul atau objek yang belum ditakrifkan.
- (d) Mengakses nilai pustaka atau objek atribut yang tidak wujud.
- (e) Mengakses fail yang tidak wujud.

Contoh 3.19

Pengesanan dan memperbaiki ralat masa larian pada segmen kod yang dihasilkan untuk penyelesaian masalah.

Pernyataan masalah:

Anda ditugaskan oleh guru anda untuk mengira jumlah perbelanjaan setiap peserta untuk aktiviti perkhemahan di sekolah pada bulan depan. Anda diberitahu bahawa, jumlah kos terdiri daripada dua jenis kos, iaitu kos tetap dan kos berubah. Item-item yang tergolong dalam kos tetap ialah kelengkapan peribadi (RM13.50), pakaian (RM105.90) dan sewaan khemah (RM12.00). Kos berubah terdiri daripada barangan keperluan memasak yang perlu dimasukkan oleh pengguna. Anda diminta menunjukkan jumlah kos tetap, kos berubah dan jumlah kos dalam senarai yang kemas kepada guru anda.

Output yang diingini:

Berdasarkan kepada pernyataan masalah tersebut, output yang diingini perlulah menjumlahkan kos yang tiada perubahan terlebih dahulu. Kemudian jumlah kos adalah daripada jumlah kos tetap berserta kos berubah yang dimasukkan.

Contoh output:

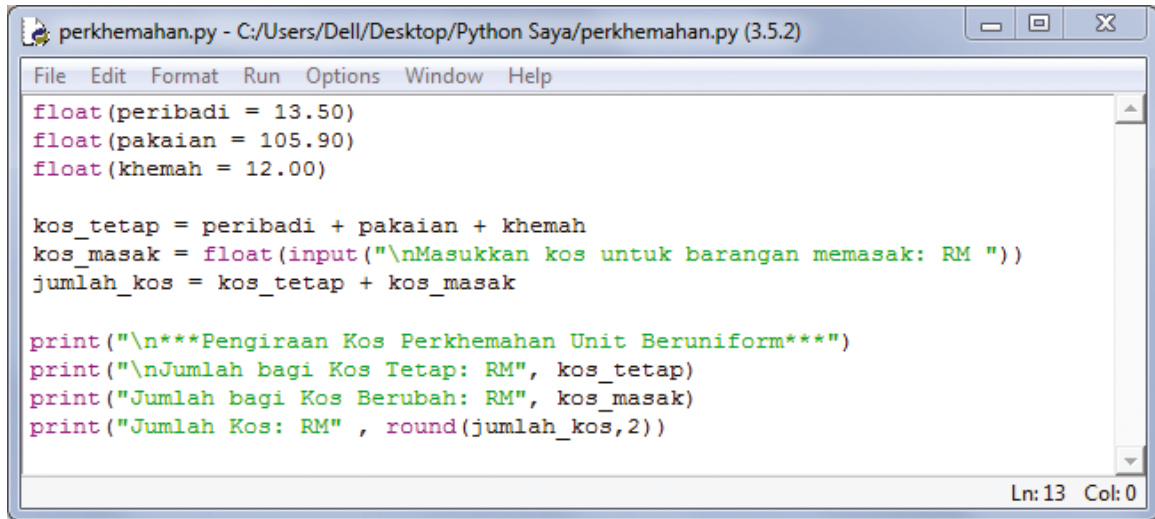
```
***Pengiraan Kos Perkhemahan Unit Beruniform***
```

```
Jumlah bagi Kos Tetap: RM131.4
```

```
Jumlah bagi Kos Berubah: RM30.0
```

```
Jumlah Kos: RM161.4
```

Segmen kod yang dihasilkan:



```
perkhemahan.py - C:/Users/Dell/Desktop/Python Saya/perkhemahan.py (3.5.2)
File Edit Format Run Options Window Help

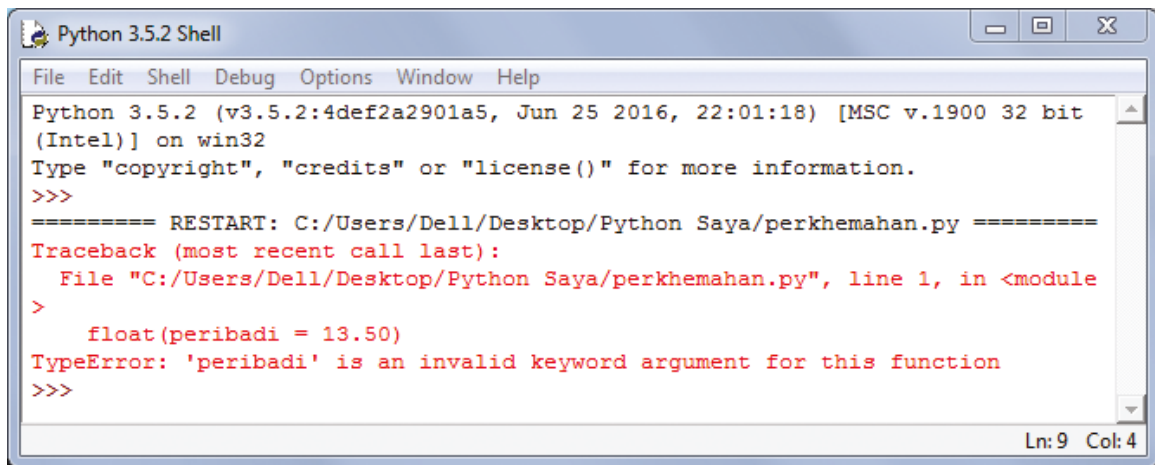
float(peribadi = 13.50)
float(pakaian = 105.90)
float(khemah = 12.00)

kos_tetap = peribadi + pakaian + khemah
kos_masak = float(input("\nMasukkan kos untuk barangan memasak: RM "))
jumlah_kos = kos_tetap + kos_masak

print("\n***Pengiraan Kos Perkhemahan Unit Beruniform***")
print("\nJumlah bagi Kos Tetap: RM", kos_tetap)
print("Jumlah bagi Kos Berubah: RM", kos_masak)
print("Jumlah Kos: RM" , round(jumlah_kos,2))

Ln:13 Col:0
```

Paparan mesej ralat apabila segmen kod yang dibina dilaksanakan:



```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help

Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit
(Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Dell/Desktop/Python Saya/perkhemahan.py =====
Traceback (most recent call last):
  File "C:/Users/Dell/Desktop/Python Saya/perkhemahan.py", line 1, in <module>
    >
      float(peribadi = 13.50)
TypeError: 'peribadi' is an invalid keyword argument for this function
>>>

Ln:9 Col:4
```

Pelaksanaan segmen kod terhenti secara tiba-tiba dan baris kod yang tidak dapat dibaca tersenarai dalam paparan mesej ralat.

Mengesan ralat dan memperbaiki ralat:

Pengesanan ralat bermula dengan memahami mesej yang dipaparkan:

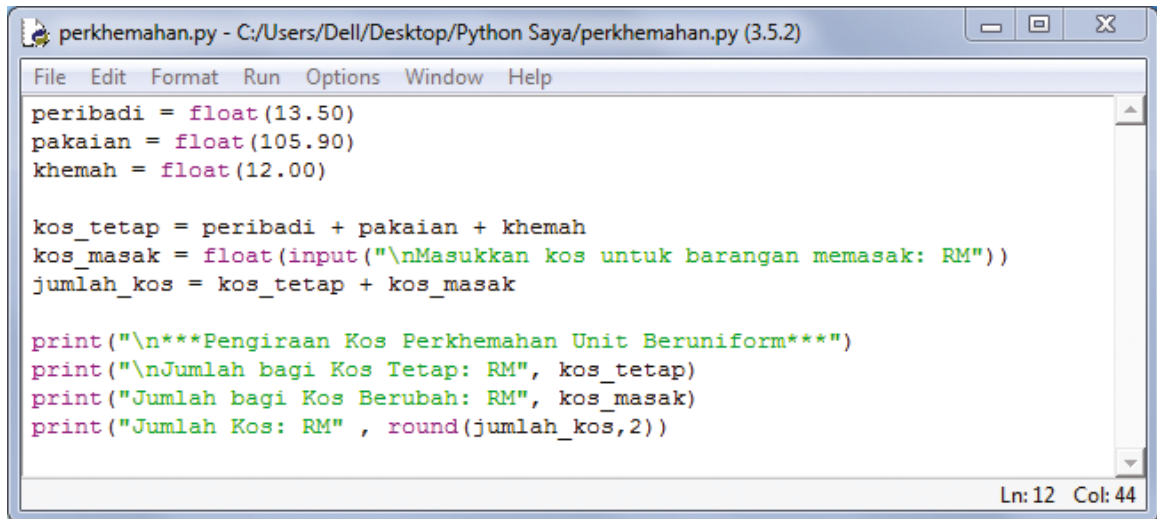
```
line 1, in <module>
    float(peribadi = 13.50)
TypeError: 'peribadi' is an invalid keyword argument for this function
```

Mesej di atas bermaksud, pengisytiharan pemboleh ubah dengan menggunakan jenis data **float** adalah tidak betul.

Ralat tersebut dapat dibaiki dengan mengisytiharkan pemboleh ubah dengan jenis data yang betul.

```
peribadi = float(13.50)
```

Segmen kod yang telah dibaiki:



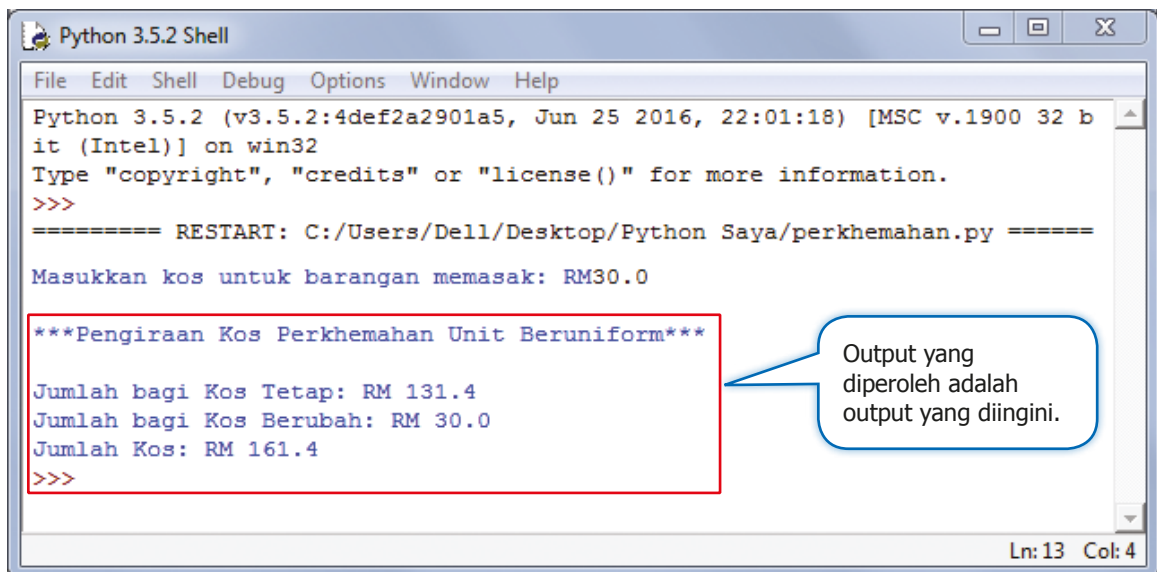
```
perkhemahan.py - C:/Users/Dell/Desktop/Python Saya/perkhemahan.py (3.5.2)
File Edit Format Run Options Window Help
peribadi = float(13.50)
pakaian = float(105.90)
khemah = float(12.00)

kos_tetap = peribadi + pakaian + khemah
kos_masak = float(input("\nMasukkan kos untuk barangan memasak: RM"))
jumlah_kos = kos_tetap + kos_masak

print("\n***Pengiraan Kos Perkhemahan Unit Beruniform***")
print("\nJumlah bagi Kos Tetap: RM", kos_tetap)
print("Jumlah bagi Kos Berubah: RM", kos_masak)
print("Jumlah Kos: RM" , round(jumlah_kos,2))

Ln: 12 Col: 44
```

Output yang diperoleh:



```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 b
it (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Dell/Desktop/Python Saya/perkhemahan.py =====
Masukkan kos untuk barangan memasak: RM30.0

***Pengiraan Kos Perkhemahan Unit Beruniform***
Jumlah bagi Kos Tetap: RM 131.4
Jumlah bagi Kos Berubah: RM 30.0
Jumlah Kos: RM 161.4
>>>

Ln: 13 Col: 4
```

Output yang diperoleh adalah output yang diinginkan.

Ralat logik

Ralat logik tidak akan mengganggu pelaksanaan segmen kod yang dibangunkan, maka tiada paparan mesej ralat. Lazimnya ralat logik disebabkan oleh kecuaiian pengatur cara. Pernyataan-pernyataan berikut ialah contoh kesilapan umum yang membawa kepada ralat logik.

- (a) Pemanggilan pemboleh ubah yang salah.
- (b) Penggunaan nombor bulat atau nombor perpuluhan yang tidak betul atau bukan di tempat yang sepatutnya.
- (c) Penggunaan operator pengendali yang salah.
- (d) Kesilapan dalam ungkapan *Boolean*.
- (e) Penggunaan ruang jarak yang salah.

Contoh 3.20

Pengesanan dan membaiki ralat logik pada segmen kod yang dihasilkan untuk penyelesaian masalah.

Pernyataan masalah:

Aziz telah membeli dua naskhah buku cerita, senaskhah majalah dan dua naskhah buku motivasi. Semasa membuat pembayaran, Aziz memberikan RM200 kepada juruwang dan menerima RM66 sebagai baki bayaran buku yang dibeli. Setelah tiba di rumah, ibu Aziz ingin mengetahui harga untuk setiap buku tersebut. Harga seunit buku cerita dan majalah adalah masing-masing RM39 dan RM16. Harga buku motivasi tidak diketahui. Aziz telah menulis satu segmen kod bagi mendapatkan harga senaskhah buku motivasi. Diketahui bahawa kedua-dua buku motivasi adalah sama harga.

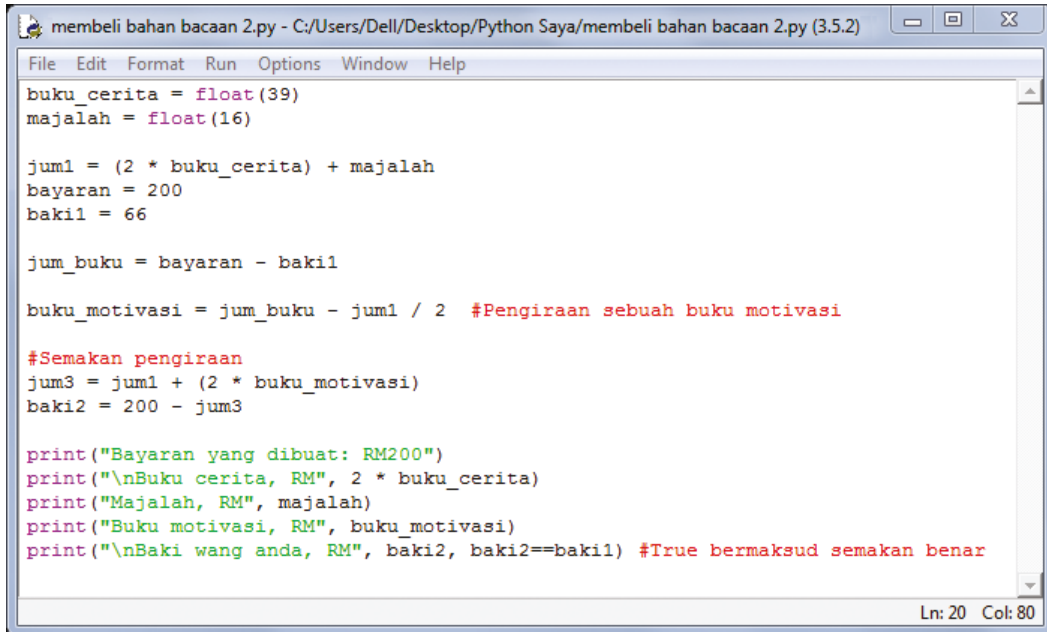
Output yang diinginkan:

Berdasarkan kepada pernyataan masalah tersebut, output yang diinginkan perlu mendapatkan harga bagi senaskhah buku motivasi yang dibeli oleh Aziz. Seterusnya, jumlah keseluruhan belian tersebut mestilah berbaki sama dengan yang diberi, iaitu RM66.

Contoh output:

```
Bayaran yang dibuat: RM200
Buku cerita, RM78
Majalah, RM16.0
Buku motivasi, RM20.0
Baki wang anda, RM66.0
```

Segmen kod yang dihasilkan:



```
membeli bahan bacaan 2.py - C:/Users/Dell/Desktop/Python Saya/membeli bahan bacaan 2.py (3.5.2)
File Edit Format Run Options Window Help

buku_cerita = float(39)
majalah = float(16)

jum1 = (2 * buku_cerita) + majalah
bayaran = 200
baki1 = 66

jum_buku = bayaran - baki1

buku_motivasi = jum_buku - jum1 / 2 #Pengiraan sebuah buku motivasi

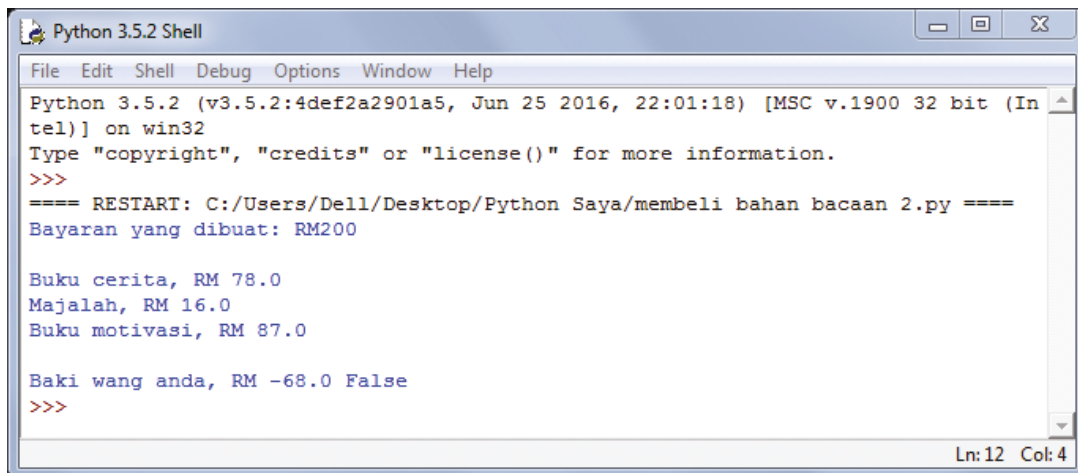
#Semakan pengiraan
jum3 = jum1 + (2 * buku_motivasi)
baki2 = 200 - jum3

print("Bayaran yang dibuat: RM200")
print("\nBuku cerita, RM", 2 * buku_cerita)
print("Majalah, RM", majalah)
print("Buku motivasi, RM", buku_motivasi)
print("\nBaki wang anda, RM", baki2, baki2==baki1) #True bermaksud semakan benar

Ln: 20 Col: 80
```

Output yang diperoleh:

Output segmen kod yang diperoleh adalah bukan output yang diinginkan. Maka, terdapat ralat logik dalam segmen kod ini. Bahagian semakan baki wang yang dikira adalah RM-68.



```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help

Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/Dell/Desktop/Python Saya/membeli bahan bacaan 2.py ====
Bayaran yang dibuat: RM200

Buku cerita, RM 78.0
Majalah, RM 16.0
Buku motivasi, RM 87.0

Baki wang anda, RM -68.0 False
>>>

Ln: 12 Col: 4
```

Mengesan ralat dan memperbaiki ralat:

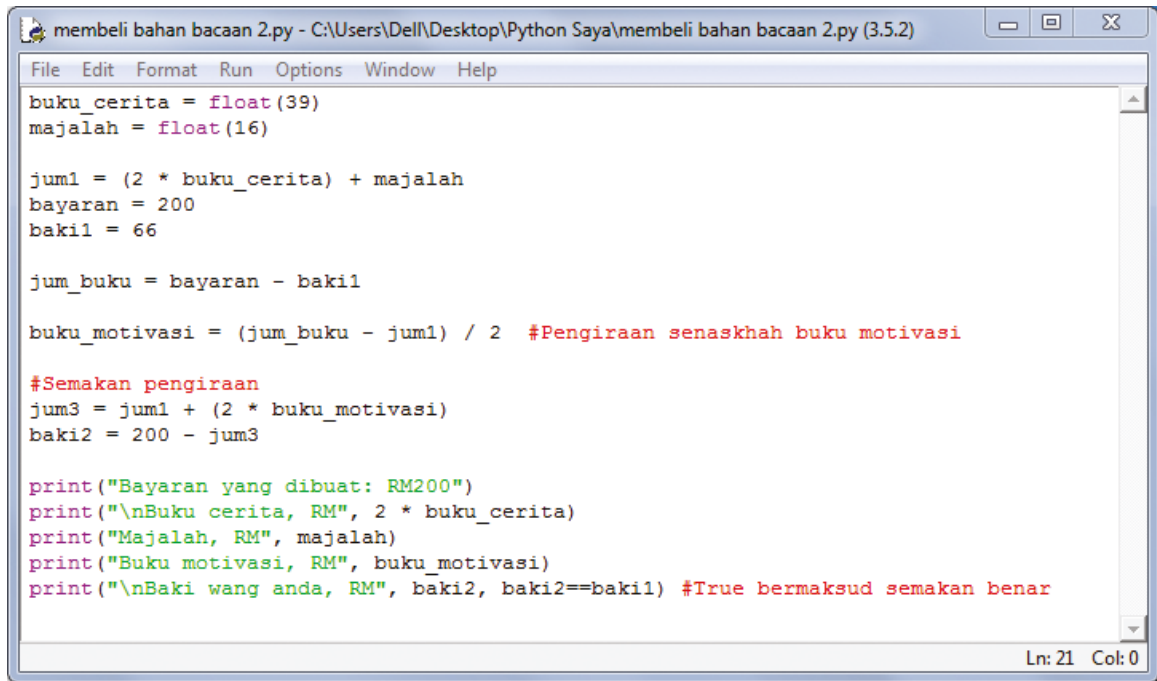
Teknik langkah demi langkah dilakukan sebaris demi sebaris pada segmen kod untuk mengesan kedudukan ralat. Dalam pelaksanaan operasi matematik, operasi yang melibatkan darab dan bahagi akan diutamakan dahulu sebelum melakukan operasi tambah dan tolak. Lihat sebaris kod di bawah, disebabkan tiada tanda kurungan, operasi bahagi telah dilakukan dahulu. Ini telah menyebabkan pengiraan menjadi tidak tepat.

```
buku_motivasi = jum_buku - jum1 / 2
```

Kod dibaiki dengan menambahkan kurungan untuk mengarah komputer melakukan operasi tolak dahulu sebelum membahagikan jumlah kepada 2.

```
buku_motivasi = (jum_buku - jum1) / 2
```

Segmen kod yang telah dibaiki:



```
membeli bahan bacaan 2.py - C:\Users\Dell\Desktop\Python Saya\membeli bahan bacaan 2.py (3.5.2)
File Edit Format Run Options Window Help
buku_cerita = float(39)
majalah = float(16)

jum1 = (2 * buku_cerita) + majalah
bayaran = 200
baki1 = 66

jum_buku = bayaran - baki1

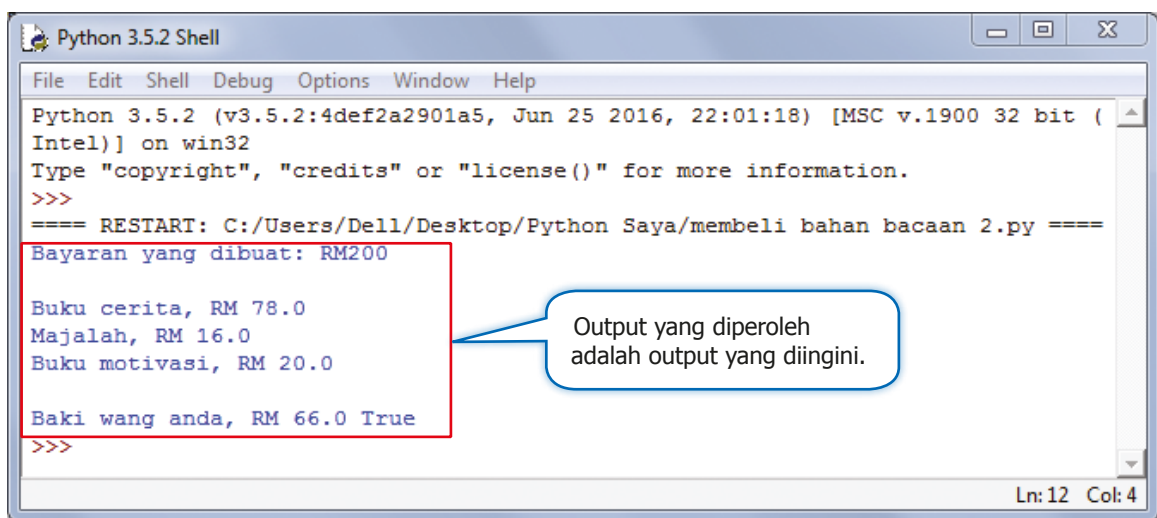
buku_motivasi = (jum_buku - jum1) / 2 #Pengiraan senaskhah buku motivasi

#Semakan pengiraan
jum3 = jum1 + (2 * buku_motivasi)
baki2 = 200 - jum3

print("Bayaran yang dibuat: RM200")
print("\nBuku cerita, RM", 2 * buku_cerita)
print("Majalah, RM", majalah)
print("Buku motivasi, RM", buku_motivasi)
print("\nBaki wang anda, RM", baki2, baki2==baki1) #True bermaksud semakan benar

Ln: 21 Col: 0
```

Output yang diperoleh:



```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/Dell/Desktop/Python Saya/membeli bahan bacaan 2.py ====
Bayaran yang dibuat: RM200

Buku cerita, RM 78.0
Majalah, RM 16.0
Buku motivasi, RM 20.0

Baki wang anda, RM 66.0 True
>>>

Ln: 12 Col: 4
```

Output yang diperoleh adalah output yang diinginkan.



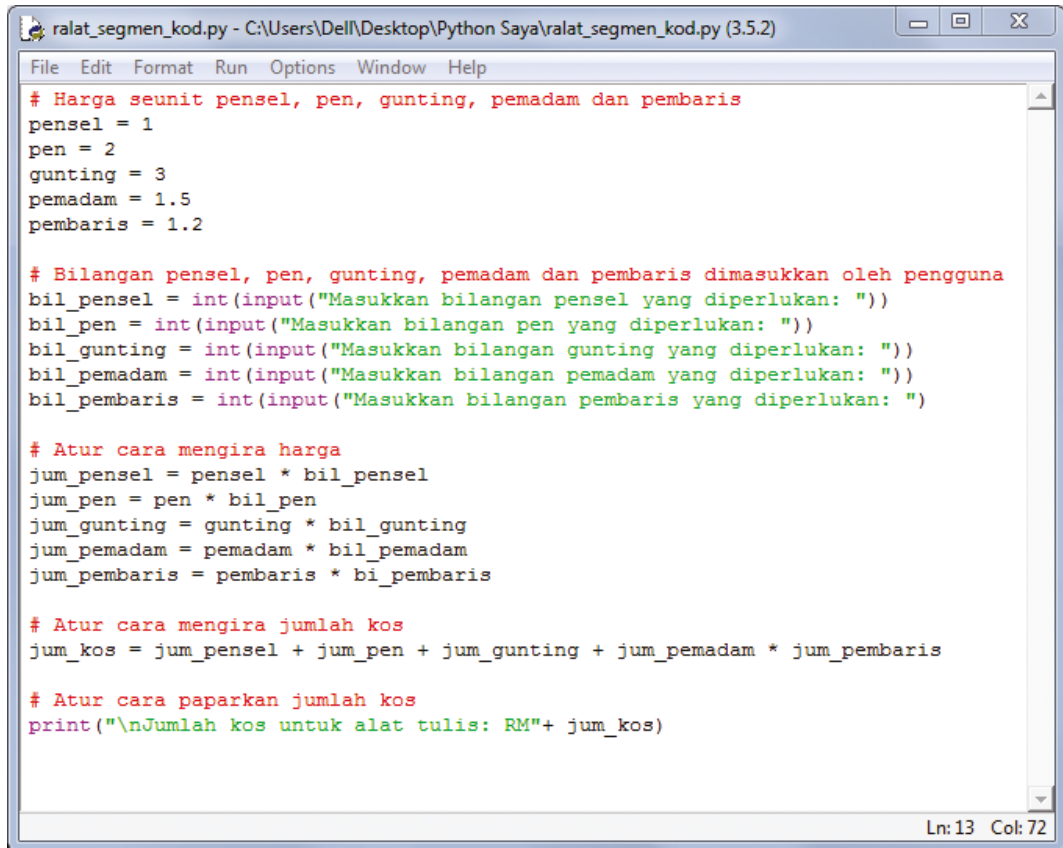
Amali Komputer 3.2

Mengesan ralat berdasarkan teknik yang sesuai dan memperbaiki ralat.

1. Gunakan perisian aplikasi Python 3.5.2 dan lancarkan satu dokumen baharu.

Windows Start Button > All Programs > IDLE (Python 3.5) > File > New File

2. Taipkan segmen kod seperti yang ditunjukkan di bawah.



```
ralat_segmen_kod.py - C:\Users\Del\\Desktop\Python Saya\ralat_segmen_kod.py (3.5.2)
File Edit Format Run Options Window Help
# Harga seunit pensel, pen, gunting, pemadam dan pembaris
pencil = 1
pen = 2
gunting = 3
pemadam = 1.5
pembaris = 1.2

# Bilangan pensel, pen, gunting, pemadam dan pembaris dimasukkan oleh pengguna
bil_pencil = int(input("Masukkan bilangan pensel yang diperlukan: "))
bil_pen = int(input("Masukkan bilangan pen yang diperlukan: "))
bil_gunting = int(input("Masukkan bilangan gunting yang diperlukan: "))
bil_pemadam = int(input("Masukkan bilangan pemadam yang diperlukan: "))
bil_pembaris = int(input("Masukkan bilangan pembaris yang diperlukan: "))

# Atur cara mengira harga
jum_pencil = pencil * bil_pencil
jum_pen = pen * bil_pen
jum_gunting = gunting * bil_gunting
jum_pemadam = pemadam * bil_pemadam
jum_pembaris = pembaris * bi_pembaris

# Atur cara mengira jumlah kos
jum_kos = jum_pencil + jum_pen + jum_gunting + jum_pemadam * jum_pembaris

# Atur cara paparkan jumlah kos
print("\nJumlah kos untuk alat tulis: RM"+ jum_kos)

Ln: 13 Col: 72
```

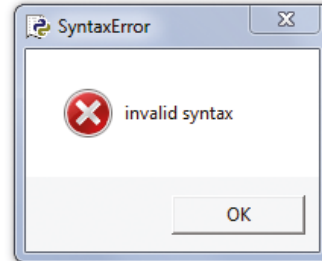
3. Pilih menu **File > Save As** untuk menyimpan fail anda dalam folder **Python Saya**. Namakan fail anda sebagai **ralat_segmen_kod.py**.
4. Kemudian pada tettingkap fail **ralat_segmen_kod.py** yang masih aktif, klik menu **Run > Run Module** untuk melaksanakan kod.
5. Tetingkap IDLE akan terpapar seperti yang berikut. Mesej ralat sintaks dipaparkan dan bahagian yang ditandakan merah menunjukkan baris di mana pelaksanaan kod terhenti.

```
# Bilangan pensel, pen, gunting, pemadam dan pembaris dimasukkan oleh pengguna
bil_pensel = int(input("Masukkan bilangan pensel yang diperlukan: "))
bil_pen = int(input("Masukkan bilangan pen yang diperlukan: "))
bil_gunting = int(input("Masukkan bilangan gunting yang diperlukan: "))
bil_pemadam = int(input("Masukkan bilangan pemadam yang diperlukan: "))
bil_pembaris = int(input("Masukkan bilangan pembaris yang diperlukan: "))
```

```
# Atur cara mengira harga
jum_pensel = pensel * bil_pensel
jum_pen = pen * bil_pen
jum_gunting = gunting * bil_gunting
jum_pemadam = pemadam * bil_pemadam
jum_pembaris = pembaris * bi_pembaris
```

```
# Atur cara mengira jumlah kos
jum_kos = jum_pensel + jum_pen + jum_gunting + jum_pemadam * jum_pembaris
```

```
# Atur cara paparkan jumlah kos
print("\nJumlah kos untuk alat tulis: RM"+ jum_kos)
```

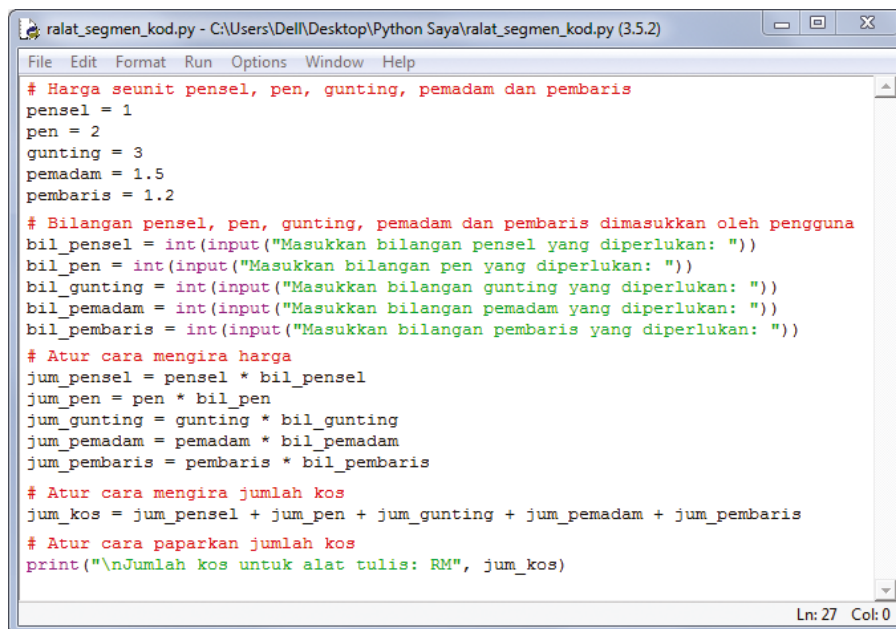


6. Lakukan semakan pada kod baris demi baris. Kesan ralat dan baiki ralat tersebut. Ulang langkah 4. Jika ada paparan mesej ralat, lakukan semakan pada segmen kod baris demi baris dan baiki ralat tersebut.
7. Setelah pelaksanaan segmen kod berjaya tanpa sebarang mesej ralat, cuba masukkan sampel data yang diminta dan semak sama ada output yang diperoleh adalah output yang diinginkan atau tidak. Jika tidak, semak baris kod yang melakukan pengiraan, iaitu kod yang ditunjukkan di bawah:

```
jum_kos = jum_pensel + jum_pen + jum_gunting + jum_pemadam * jum_pembaris
```
8. Lakukan teknik semakan meja, gunakan satu set sampel data untuk menguji kod di atas. Kesan ralat dan baiki ralat tersebut.
9. Ulang langkah 4.
10. Lengkapkan jadual di bawah untuk mencatatkan bilangan ralat yang dikesan.

Ralat	Jenis ralat	Paparan mesej ralat	Mempengaruhi output
Ralat 1	Ralat sintaks	Ada	Tidak
Ralat 2			
Ralat 3			
Ralat 4			

11. Segmen kod yang berikut ialah kod yang telah dibaiki kesemua ralat.



```

# Harganya seunit pensel, pen, gunting, pemadam dan pembaris
pencil = 1
pen = 2
gunting = 3
pemadam = 1.5
pembaris = 1.2

# Bilangan pensel, pen, gunting, pemadam dan pembaris dimasukkan oleh pengguna
bil_pencil = int(input("Masukkan bilangan pensel yang diperlukan: "))
bil_pen = int(input("Masukkan bilangan pen yang diperlukan: "))
bil_gunting = int(input("Masukkan bilangan gunting yang diperlukan: "))
bil_pemadam = int(input("Masukkan bilangan pemadam yang diperlukan: "))
bil_pembaris = int(input("Masukkan bilangan pembaris yang diperlukan: "))

# Atur cara mengira harga
jum_pencil = pencil * bil_pencil
jum_pen = pen * bil_pen
jum_gunting = gunting * bil_gunting
jum_pemadam = pemadam * bil_pemadam
jum_pembaris = pembaris * bil_pembaris

# Atur cara mengira jumlah kos
jum_kos = jum_pencil + jum_pen + jum_gunting + jum_pemadam + jum_pembaris

# Atur cara paparkan jumlah kos
print("\nJumlah kos untuk alat tulis: RM", jum_kos)

```

Aktiviti 3.5

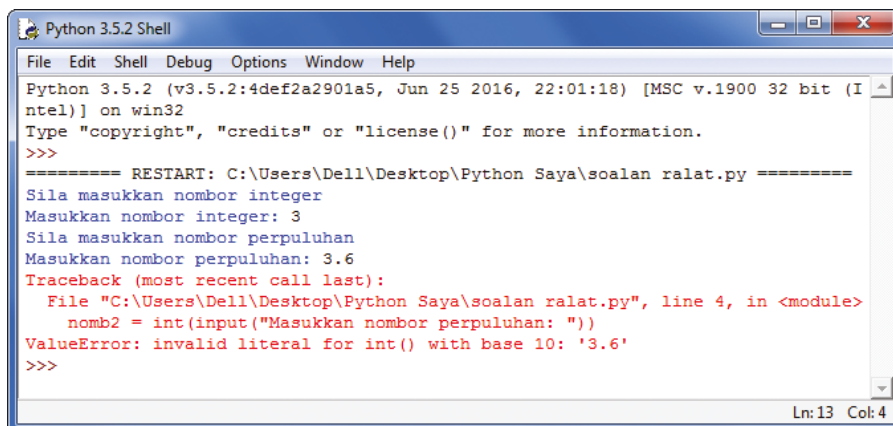
Individu

Kenal pasti jenis-jenis ralat.

1. Tuliskan jenis ralat yang terdapat dalam segmen kod di bawah danbaiki ralat tersebut.

```
>>> 3d = 2
[Mesej ralat yang terdapat]
```

2. Lihat paparan output Python di bawah yang menunjukkan mesej ralat apabila suatu segmen kod Python dilaksanakan. Kenal pasti ralat danbaiki ralat tersebut. Tulis semula segmen kod yang telah dibaiki.



```

Python 3.5.2 Shell
File Edit Shell Debug Options Window Help

Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Dell\Desktop\Python Saya\soalan ralat.py =====
Sila masukkan nombor integer
Masukkan nombor integer: 3
Sila masukkan nombor perpuluhan
Masukkan nombor perpuluhan: 3.6
Traceback (most recent call last):
  File "C:\Users\Dell\Desktop\Python Saya\soalan ralat.py", line 4, in <module>
    nomb2 = int(input("Masukkan nombor perpuluhan: "))
ValueError: invalid literal for int() with base 10: '3.6'
>>>

```

3.1.5 Menggunakan gabungan jenis data, pemboleh ubah, pemalar dan operator untuk menyelesaikan masalah

Kebanyakan aplikasi yang sedia ada menggunakan gabungan daripada beberapa jenis data yang berlainan. Ini bermaksud segmen-segmen kod yang dibangunkan terdiri daripada pelbagai jenis data. Gabungan beberapa jenis data diperlukan mengikut keperluan dan kegunaan aplikasi yang ingin dibangunkan. Jadual 3.10 menunjukkan beberapa jenis data yang digunakan dalam pembangunan aplikasi yang bermula daripada segmen kod.

Standard Pembelajaran

Murid boleh:
3.1.5 Menyelesaikan masalah yang melibatkan gabungan jenis data, pemboleh ubah, pemalar dan operator dalam segmen kod.

Jadual 3.10 Contoh aplikasi dan penggunaan gabungan jenis data

Contoh aplikasi	Penggunaan jenis data	Contoh penggunaan jenis data
Permainan perlumbaan kereta	Integer	Kedudukan pemenang
	Double atau float	Perkiraan masa
	Pemboleh ubah	Nama pemain
Simulasi fesyen pakaian	Double atau float	Ukuran badan atau baju
	Pemalar	Ukuran aksesori fesyen
	Pemboleh ubah	Nama pakaian
E-mel	Integer	Umur
	Pemboleh ubah	Nama

Contoh 3.21

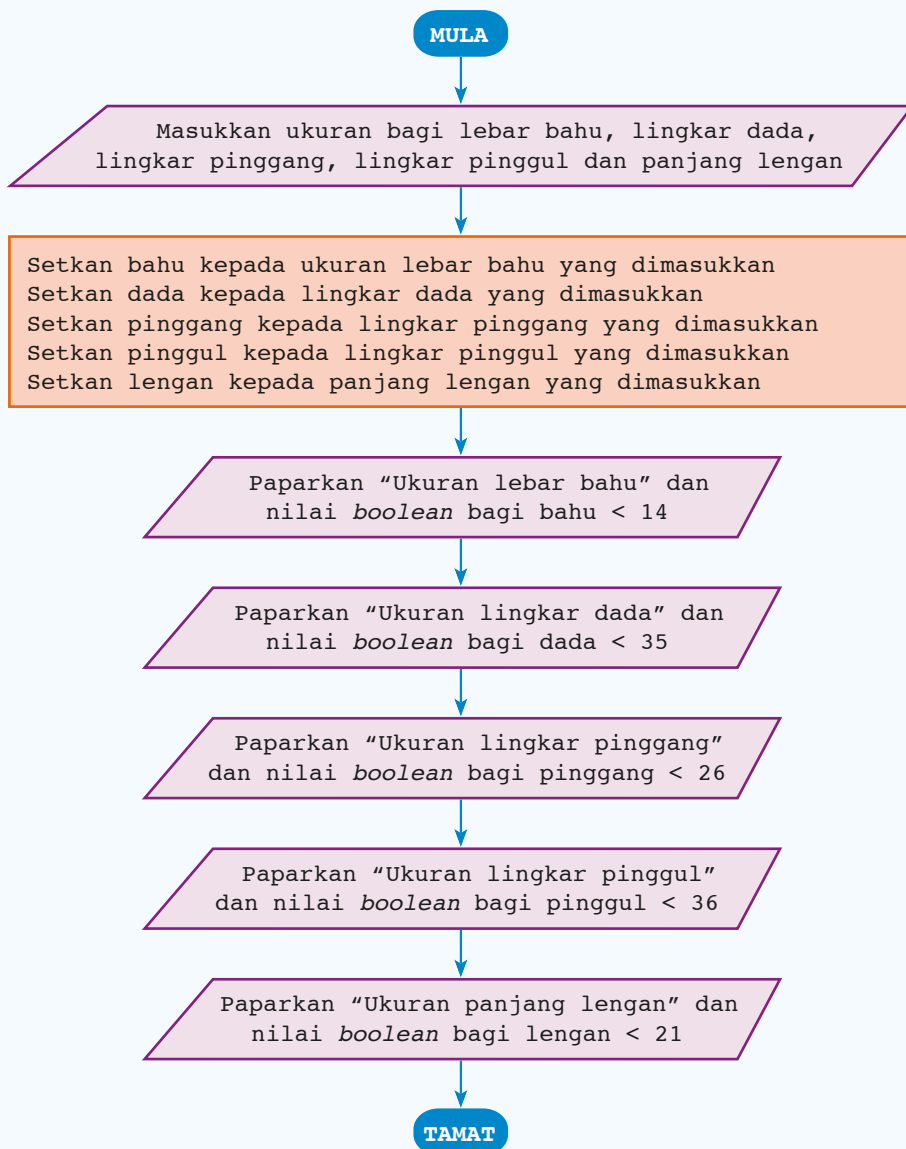
Penyelesaian masalah dengan menggunakan gabungan jenis data, pemboleh ubah, pemalar dan operator.

Langkah 1: Pernyataan masalah

Aini seorang usahawan pakaian tradisional Melayu. Dia ingin memudahkan pengguna laman sesawangnya dalam membuat pemilihan saiz pakaian berdasarkan ukuran badan yang dimasukkan oleh pengguna. Anda diminta membangunkan satu segmen kod yang berfungsi untuk menerima input bacaan ukuran bagi penentuan pakaian saiz S. Ukuran yang perlu dimasukkan oleh pengguna ialah lebar bahu, lingkaran dada, lingkaran pinggang, lingkaran pinggul dan panjang lengan. Segmen kod perlu menerima input daripada pengguna dan menyemak sama ada ukuran yang diberi menepati ukuran dalam jadual yang berikut.

Jenis ukuran	Ukuran (inci)
Lebar bahu	Tidak melebihi 14
Lingkar dada	Tidak melebihi 35
Lingkar pinggang	Tidak melebihi 26
Lingkar pinggul	Tidak melebihi 36
Panjang lengan	Tidak melebihi 21

Langkah 2: Melukis carta alir untuk penyelesaian



Langkah 3: Menulis segmen kod berdasarkan carta alir yang dilukis dengan menggunakan perisian aplikasi Scratch 2.0

Contoh output bagi segmen kod perisian aplikasi Scratch 2.0



Langkah 4: Menulis segmen kod berdasarkan carta alir yang dilukis dengan menggunakan perisian aplikasi Python 3.5.2

```

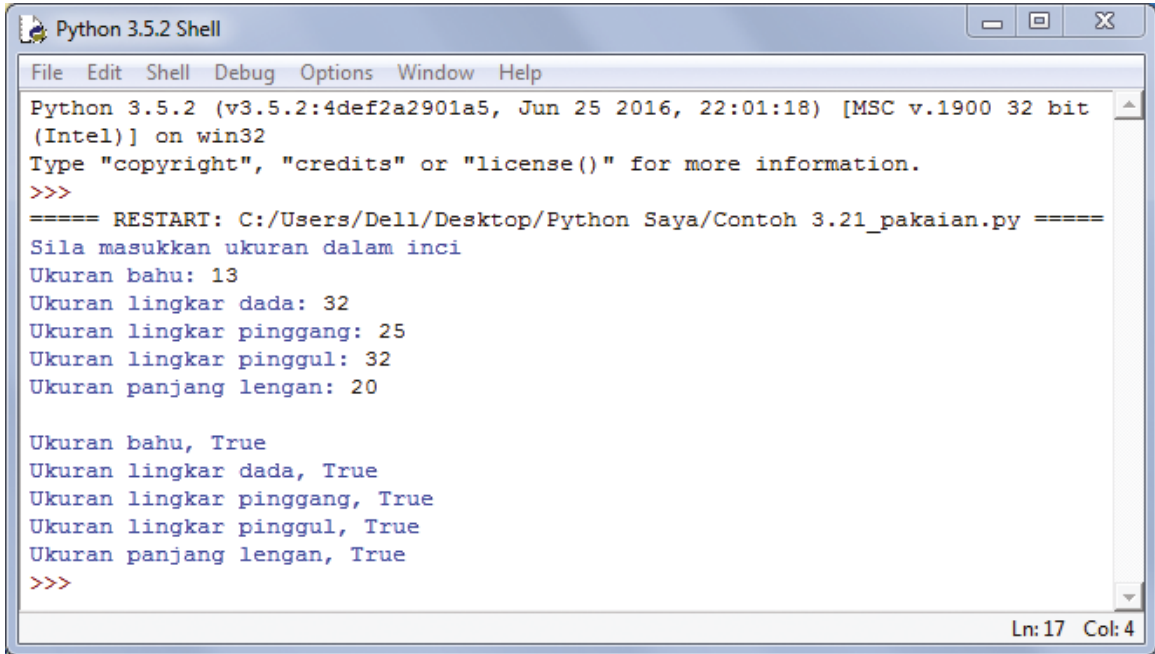
Contoh 3.21_pakaian.py - C:/Users/Dell/Desktop/Python Saya/Contoh 3.21_pakaia...
File Edit Format Run Options Window Help
print("Sila masukkan ukuran dalam inci")

bahu = float(input("Ukuran bahu: "))
dada = float(input("Ukuran lingkar dada: "))
pinggang = float(input("Ukuran lingkar pinggang: "))
pinggul = float(input("Ukuran lingkar pinggul: "))
lengan = float(input("Ukuran panjang lengan: "))

print ("\nUkuran bahu,",bahu < 14)
print ("Ukuran lingkar dada,",dada < 35)
print ("Ukuran lingkar pinggang,",pinggang < 26)
print ("Ukuran lingkar pinggul,",pinggul < 36)
print ("Ukuran panjang lengan,",lengan < 21)
Ln: 14 Col: 0

```

Contoh output bagi segmen kod perisian aplikasi Python 3.5.2



```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit
(Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Dell/Desktop/Python Saya/Contoh 3.21_pakaian.py =====
Sila masukkan ukuran dalam inci
Ukuran bahu: 13
Ukuran lingkar dada: 32
Ukuran lingkar pinggang: 25
Ukuran lingkar pinggul: 32
Ukuran panjang lengan: 20

Ukuran bahu, True
Ukuran lingkar dada, True
Ukuran lingkar pinggang, True
Ukuran lingkar pinggul, True
Ukuran panjang lengan, True
>>>
```



Aktiviti 3.6

Pasangan

Menyelesaikan masalah dengan menggunakan gabungan jenis data, pemboleh ubah, pemalar dan operator.

Guru kelas anda perlu menyediakan satu laporan ringkas mengenai Indeks Jisim Tubuh (*BMI*) untuk murid-murid dalam kelas. Anda dilantik untuk menyediakan satu segmen kod untuk mengambil input daripada pengguna dan memaparkan maklumat yang dikehendaki. Berikut menunjukkan contoh output yang diperlukan oleh guru anda.

Contoh output:

```
Nama: Aminah binti Ali
Umur: 14 tahun
Jantina: Perempuan
BMI anda ialah 22.3
```



Praktis 3.1

1. Jadual di bawah menunjukkan pengiraan markah bagi murid-murid Tingkatan 2.

Bil	Nama	Kelas	Peratusan markah (%)	Gred
1	Azril bin Mohammad	2B	68.8	C
2	Abdullah Ihsan bin Zainudin	2B	45	E
3	Anathrah binti Ahmed	2B	77.5	B
4	Anuratha a/p N.Ganesan	2B	60	C
5	Atheera binti Mazli	2B	85	A
6	Balbir Kaur a/p Ajeet Singh	2B	96.3	A
7	Chew Suan Ying	2B	47.5	E
8	Chong Chiu Cheik	2B	86.3	A
9	Farah Rafieza binti Saripan	2B	70	B
10	Fatimah binti Khalil	2B	53.8	D
11	Fazleen binti Md Rusli	2B	88.8	A
12	Foo Wei Hung	2B	75	B

Berdasarkan jadual di atas, isytiharkan **empat** pemboleh ubah bagi situasi pengiraan markah murid-murid yang ditunjukkan di atas.

2. Senaraikan **tiga** perbezaan bagi pemboleh ubah dan pemalar.
3. Kelas anda diberikan tugas membina sebuah replika bagi “Majlis Pelancaran Bijak Menabung.” Setelah melakukan perbincangan, keputusan telah diperoleh dengan membuat tabung berbentuk sfera yang dinamakan “Sfera Magnet Duit.” Sfera tersebut perlu dimasukkan ke dalam sebuah kotak semasa majlis pelancaran. Ukuran kotak adalah 75 cm panjang, 85 cm lebar dan 95 cm tinggi. Anda bersama-sama pasukan anda perlu membangunkan segmen kod dengan menggunakan jenis data yang bersesuaian dan mampu menerima input serta memaparkan isi padu sfera yang terbabit. Bundarkan jawapan anda kepada dua tempat perpuluhan.

4. Isikan tempat kosong pada segmen kod di bawah dengan operator logik yang sesuai. Kemudian uji segmen kod tersebut. Contoh output yang dihasilkan haruslah sama dengan yang diberikan.

Segmen kod:

```
soalan 4.py - C:\Users\Dell\Desktop\soalan 4.py (3.5.2)
File Edit Format Run Options Window Help
a = 6
b = 7
c = 42
print ("1", a == 6)
print ("2", a == 7)
print ("3", a == 6 ____ b == 7)
print ("4", a == 7 ____ b == 7)
print ("5", ____ a == 7 ____ b == 7)
print ("6", a == 7 ____ b == 7)
print ("7", a == 7 ____ b == 6)
print ("8", not (a == 7 ____ b == 6))
print ("9", ____ a == 7 ____ b == 6)
Ln: 13 Col: 0
```

Contoh output:

```
1 True
2 False
3 True
4 False
5 True
6 True
7 False
8 True
9 False
>>>
```

5. Dalam keadaan hari yang cerah, Aini perlu mengambil masa selama 12 minit untuk berjalan kaki ke sekolah. Jika hari hujan, tempoh perjalanan akan ditambahkan 4 minit. Berapakah jumlah masa yang digunakan oleh Aini untuk perjalanan ke sekolah dalam tempoh tiga hari yang cerah dan dua hari yang hujan? Paparkan jumlah masa yang digunakan. Aini telah membangunkan segmen kod berikut untuk mengira jumlah masa dalam lima hari tersebut.

```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900
32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> hujan = 12
>>> cerah = 16
>>> jumlah = 12 * 2 + 16 * 3
>>> print ("jumlah")
jumlah
>>>
Ln: 8 Col: 4
```

- (a) Kenal pasti ralat yang terdapat dalam segmen kod di atas.
(b) Baiki ralat tersebut mengikut output yang dikehendaki.



3.2 Struktur Kod Arahan



Gambar foto 3.1 Cef robot yang pertama di dunia

Gambar foto 3.1 menunjukkan cef robot yang pertama di dunia. Robot ini dapat menyimpan lebih daripada 2000 resipi dan memasak seperti manusia kerana robot ini telah merakam cara dan pergerakan seorang cef ketika memasak. Pengguna cuma perlu memilih resipi yang dikehendaki dan menekan butang mula bagi membolehkan cef robot beroperasi.

Atur cara di sebalik operasi cef robot ini adalah kompleks dan teratur. Sebuah atur cara yang kompleks lazimnya mengandungi gabungan pelbagai struktur kawalan supaya atur cara yang dibina adalah lebih interaktif dan mampu membuat pilihan bagi memberi keputusan yang tepat. Manakala, dalam setiap pilihan terkandung satu senarai langkah-langkah yang perlu dilakukan.

Penggunaan struktur kawalan diperlukan bagi penyelesaian masalah yang kompleks. Dalam Subtopik 3.2, anda akan mempelajari empat jenis struktur kawalan, iaitu struktur kawalan jujukan, struktur kawalan pilihan, struktur kawalan bersarang dan struktur kawalan ulangan untuk penyelesaian masalah. Pembelajaran Subtopik 3.2 akan diterangkan dengan merujuk kepada contoh penggunaan bahasa pengaturcaraan Python.



Mempelajari struktur kod arahan akan meningkatkan daya pemikiran secara kreatif dan inovatif.



Semasa penyelesaian masalah dengan kod arahan yang melibatkan struktur kawalan, anda telah mempraktikkan penggunaan pelbagai teknik dalam pemikiran komputasional.



Struktur kod arahan ialah paradigma pengaturcaraan yang juga bertindak sebagai model untuk menjelaskan atur cara yang hendak dibangunkan. Struktur kod arahan juga bertujuan untuk memberikan penjelasan, meningkatkan kualiti dan menjimatkan masa pembangunan sesuatu atur cara. Di samping itu, penggunaan struktur blok dalam beberapa struktur kawalan merupakan antara mekanisme yang terdapat dalam struktur kod arahan.

Kod arahan dibangunkan bagi mengarahkan komputer untuk menyelesaikan masalah dan memudahkan pengguna. Sebagai contoh, rekaan cef robot yang pertama di dunia yang dapat memasak seperti manusia memudahkan tugas manusia dan dapat menyelesaikan masalah bagi manusia yang tidak mahir memasak.

3.2.1 (i) Struktur kawalan jujukan dalam penyelesaian masalah

Standard Pembelajaran

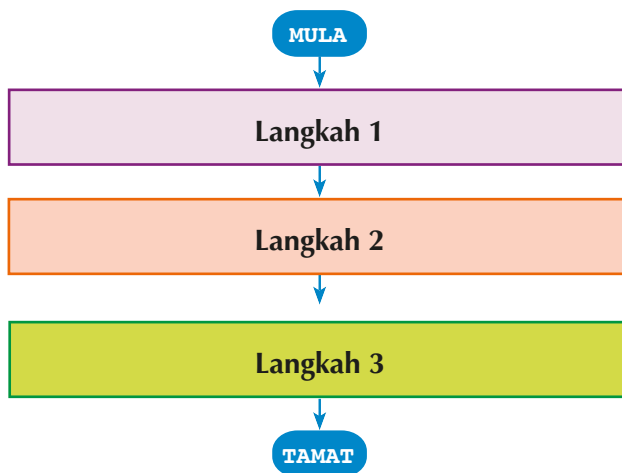
Murid boleh:

3.2.1 Menghasilkan atur cara yang melibatkan:

- (i) struktur kawalan jujukan dalam penyelesaian masalah.

Struktur kawalan jujukan ialah salah satu struktur kod arahan yang paling mudah untuk difahami. Struktur ini melaksanakan arahan baris demi baris mengikut susunan satu aliran sahaja, iaitu dari atas ke bawah secara tertib. Oleh itu, struktur kawalan jujukan melaksanakan kod atur cara secara linear, di mana setiap pernyataan dilaksanakan (*executed*) mengikut urutan yang telah ditetapkan.

Struktur kawalan jujukan digunakan untuk menyelesaikan sesuatu permasalahan yang mudah. Struktur kawalan jujukan dapat dikenal pasti dengan mudah melalui tertib penyelesaian masalah secara langkah demi langkah seperti dalam Rajah 3.11.



Rajah 3.11 Carta alir struktur kawalan jujukan

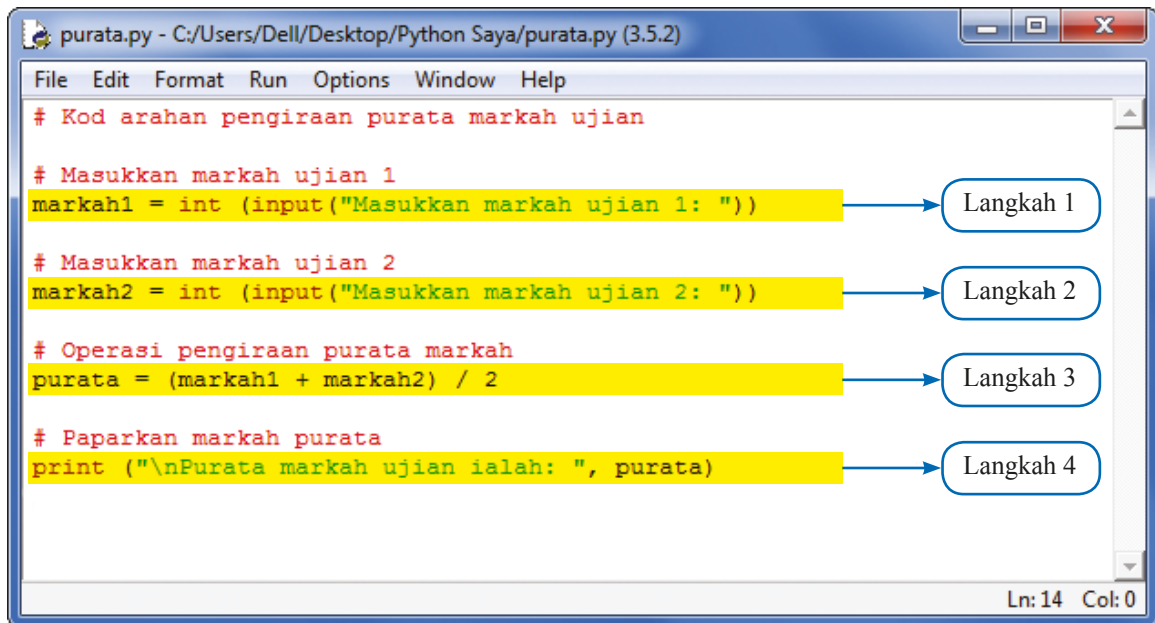
Contoh 3.23

Pengiraan purata markah bagi dua ujian bulanan.

Permasalahan:

- Input: Dua markah ujian bulanan
- Proses: Pengiraan purata
- Output: Hasil bagi nilai purata

Kod arahan pengiraan purata markah ujian:



```
purata.py - C:/Users/Dell/Desktop/Python Saya/purata.py (3.5.2)
File Edit Format Run Options Window Help
# Kod arahan pengiraan purata markah ujian

# Masukkan markah ujian 1
markah1 = int (input("Masukkan markah ujian 1: "))

# Masukkan markah ujian 2
markah2 = int (input("Masukkan markah ujian 2: "))

# Operasi pengiraan purata markah
purata = (markah1 + markah2) / 2

# Paparkan markah purata
print ("\nPurata markah ujian ialah: ", purata)
```

Langkah 1

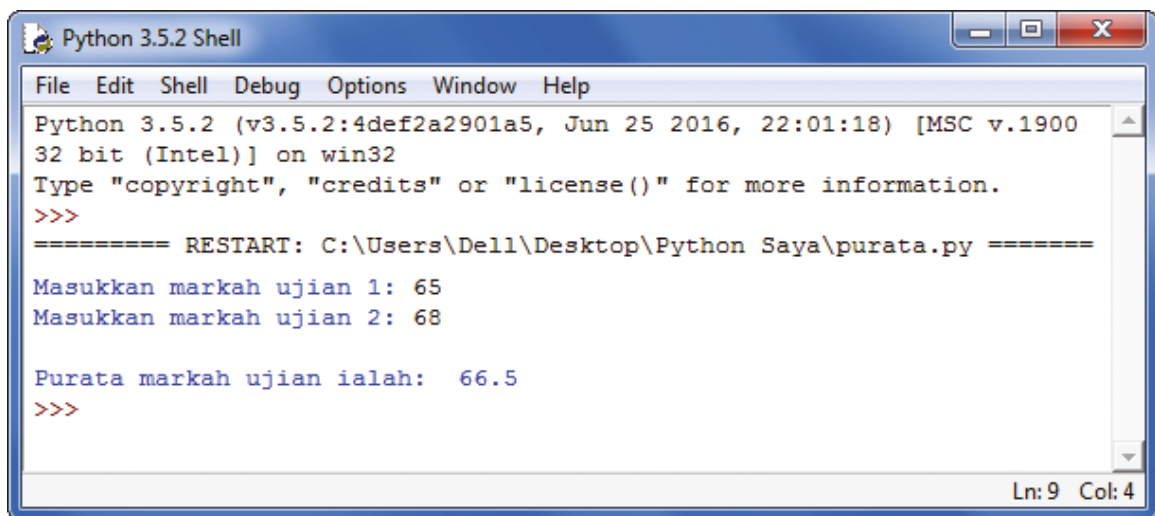
Langkah 2

Langkah 3

Langkah 4

Ln: 14 Col: 0

Output pengiraan purata markah ujian:



```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900
32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Dell\Desktop\Python Saya\purata.py =====
Masukkan markah ujian 1: 65
Masukkan markah ujian 2: 68

Purata markah ujian ialah:  66.5
>>>
```

Ln: 9 Col: 4

Contoh 3.24

Penyelesaian masalah menggunakan struktur kawalan jujukan.

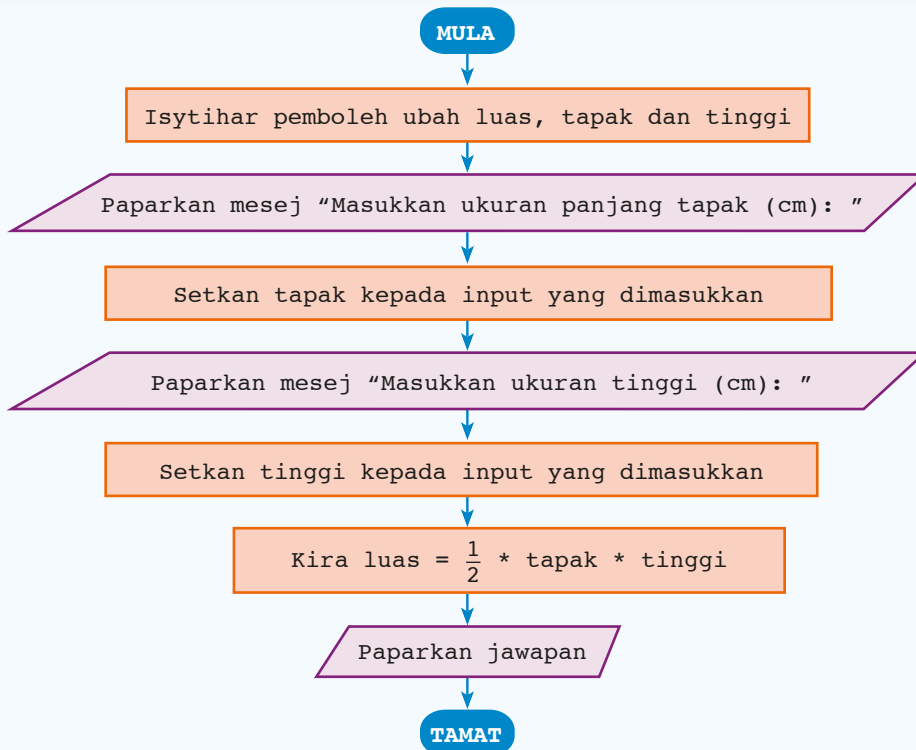
Jessica ingin membina satu atur cara yang mengira luas bagi segi tiga. Dalam langkah penyiasatan, Jessica perlu memikirkan input yang dikehendaki dan proses yang terlibat. Input yang dikehendaki ialah ukuran panjang tapak dan tinggi segi tiga. Rumus bagi mengira luas segi tiga ialah $\frac{1}{2} \times \text{panjang tapak} \times \text{tinggi}$.

Langkah 1: Pernyataan masalah

1. Mengenal pasti input yang diperlukan, iaitu panjang tapak dan tinggi dalam unit sentimeter.
2. Mengenal pasti proses dan struktur kawalan yang terlibat, iaitu struktur kawalan jujukan.
3. Mengenal pasti output yang diperlukan, iaitu luas segi tiga yang dikira.

Langkah 2: Menulis algoritma bagi penyelesaian masalah

1. Isytihar pemboleh ubah luas, tapak dan tinggi.
2. Masukkan ukuran panjang tapak.
3. Setkan tapak kepada input yang dimasukkan.
4. Masukkan ukuran tinggi.
5. Setkan tinggi kepada input yang dimasukkan.
6. Kira luas = $\frac{1}{2} \times \text{panjang tapak} \times \text{tinggi}$.
7. Paparkan jawapan.
8. Tamat

Langkah 3: Melukis carta alir

Langkah 4: Pengekodan dengan menggunakan

(a) perisian aplikasi Scratch 2.0



(b) perisian aplikasi Python 3.5.2

The screenshot shows a Python 3.5.2 script in a text editor window titled "jujukan_luas segi tiga.py - C:\Users\Dell\Desktop\Python Saya\3.2\jujukan_luas segi tiga.py (3.5.2)". The script contains the following code:

```
File Edit Format Run Options Window Help
# Pengguna masukkan ukuran panjang tapak
tapak = float(input("Masukkan ukuran panjang tapak (cm): "))

# Pengguna masukkan ukuran tinggi
tinggi = float(input("Masukkan ukuran tinggi (cm): "))

# Formula mengira luas segi tiga
luas = 0.5 * tapak * tinggi

# Mencetak output
print ("Luas segi tiga ialah", luas)
```

Ln: 10 Col: 18

Langkah 5: Contoh output bagi

(a) perisian aplikasi Scratch 2.0



(b) perisian aplikasi Python 3.5.2

```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:\Users\Dell\Desktop\Python Saya\3.2\jujukan_luas segi tiga.py ==
Masukkan ukuran panjang tapak (cm): 15
Masukkan ukuran tinggi (cm): 30
Luas segi tiga ialah 225.0
>>>
== RESTART: C:\Users\Dell\Desktop\Python Saya\3.2\jujukan_luas segi tiga.py ==
Masukkan ukuran panjang tapak (cm): 33
Masukkan ukuran tinggi (cm): 25.3
Luas segi tiga ialah 417.45
>>>
```

Ln: 13 Col: 4



Aktiviti 3.7

Individu

Menggunakan struktur kawalan jujukan untuk menyelesaikan masalah pengiraan.

Bagi tujuan keceriaan di dalam kelas, senarai nama rakan-rakan anda akan diletakkan dalam satu bulatan dan ukuran diameter bulatan tersebut tidak boleh melebihi 1.5 meter. Tulis satu atur cara bagi mengira luas permukaan bulatan yang diperlukan. Formula luas permukaan bulatan ialah, $Luas = \pi j^2$. Ukuran jejari (j) perlu dimasukkan bagi tujuan pengiraan.

3.2.1 (ii) Struktur kawalan pilihan dalam penyelesaian masalah

Semua bahasa pengaturcaraan menyediakan pelbagai struktur kawalan yang membenarkan laluan pelaksanaan yang lebih kompleks. Ini membolehkan masalah yang lebih rumit diselesaikan dengan atur cara yang interaktif.

Sebuah atur cara yang interaktif perlu berupaya menyemak syarat dan bertindak berdasarkan pilihan pengguna yang berlainan. Struktur kawalan pilihan yang bertindak mengikut syarat membolehkan atur cara yang lebih mesra pengguna dan bermanfaat dihasilkan.

Dalam satu struktur kawalan pilihan, pengatur cara akan menentukan syarat yang perlu dipatuhi bagi menghasilkan sesuatu keputusan yang tepat. Lazimnya, satu syarat akan menentukan satu keputusan melalui satu atau dua atau pelbagai cabang pilihan seperti yang ditunjukkan dalam Rajah 3.12.

Standard Pembelajaran

Murid boleh:

3.2.1 Menghasilkan atur cara yang melibatkan:

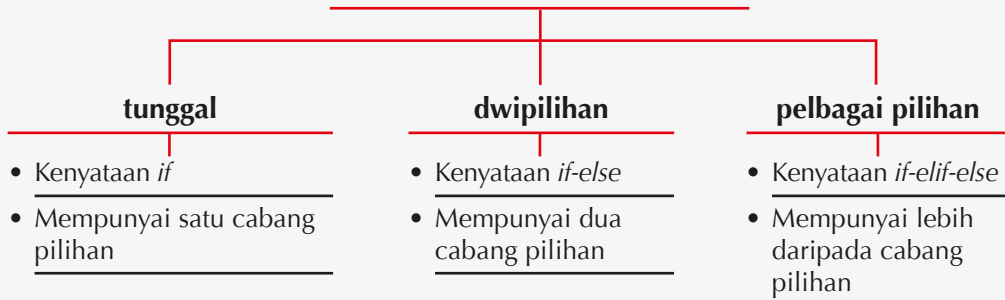
- (ii) struktur kawalan pilihan dalam penyelesaian masalah.



IMBAS KEMBALI

Struktur kawalan pilihan telah anda pelajari semasa Tingkatan 1.

STRUKTUR KAWALAN PILIHAN



Rajah 3.12 Jenis-jenis struktur kawalan pilihan

Contoh 3.25

Penyelesaian masalah menggunakan struktur kawalan pilihan.

Demi menjaga kesihatan dan mengatasi masalah obesiti dalam kalangan warga sekolah, anda diminta membangunkan satu atur cara yang akan memaparkan cadangan jumlah kalori yang dibenarkan untuk sehari berdasarkan jantina dan umur pengguna. Atur cara perlu mengikut jadual cadangan pengambilan kalori yang berikut.

Lelaki		Perempuan	
13 – 15 tahun	2690 kcal	13 – 15 tahun	2180 kcal
16 – 18 tahun	2840 kcal	16 – 18 tahun	2050 kcal
19 – 29 tahun	2440 kcal	19 – 29 tahun	2000 kcal
30 – 59 tahun	2460 kcal	30 – 59 tahun	2180 kcal
60 tahun dan ke atas	2010 kcal	60 tahun dan ke atas	1780 kcal

Langkah 1: Pernyataan masalah

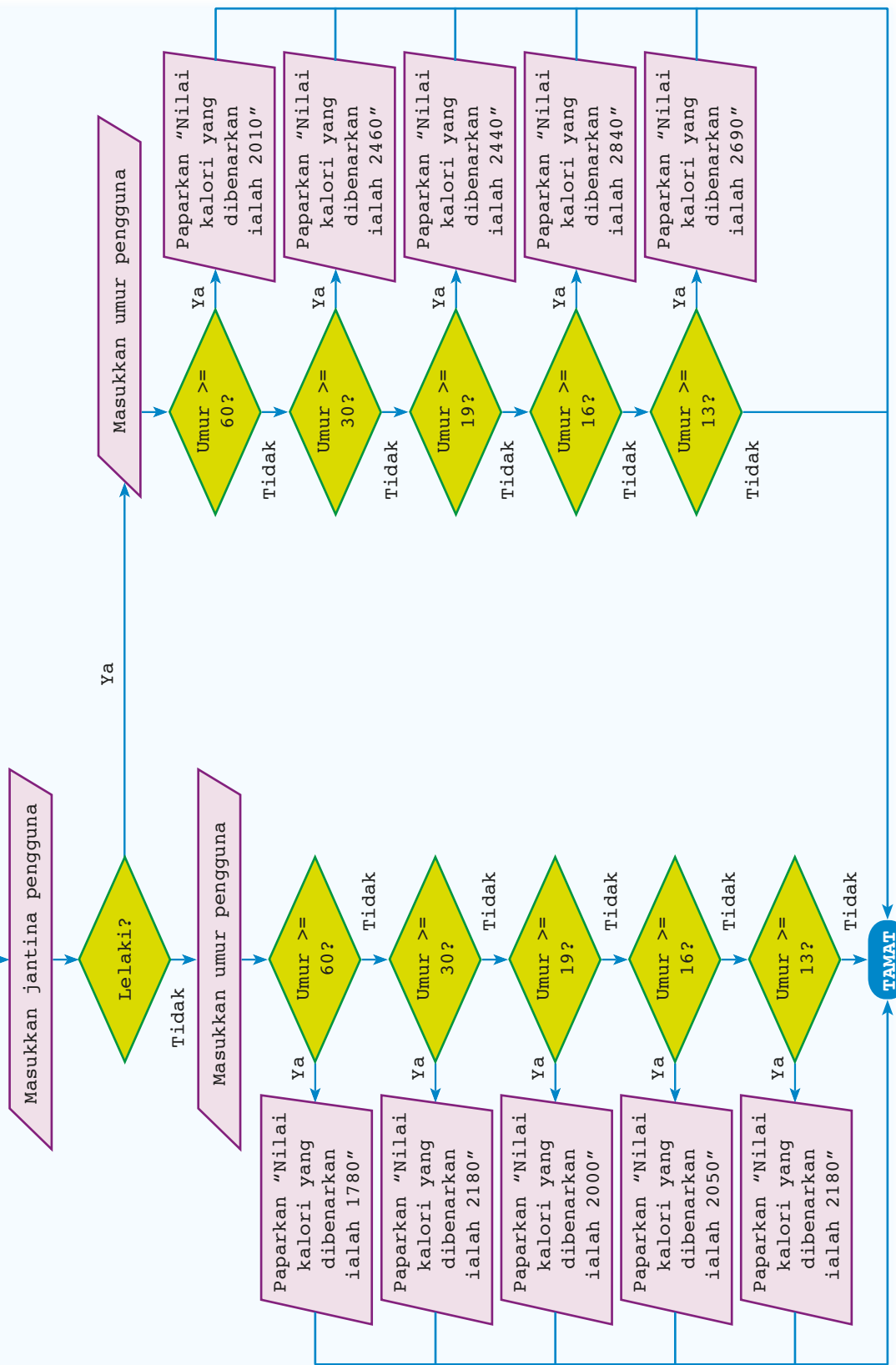
1. Mengetahui pasti input yang diperlukan, iaitu jantina dan umur pengguna.
2. Mengetahui pasti proses dan struktur kawalan yang terlibat, iaitu struktur kawalan dwipilihan dan pelbagai pilihan.
3. Mengetahui pasti output yang diperlukan, iaitu cadangan jumlah pengambilan kalori untuk sehari.

Langkah 2: Menulis algoritma bagi penyelesaian masalah

1. Pengguna diminta memasukkan jantina.
Jika lelaki, langkau ke Langkah 2.
Jika perempuan, langkau ke Langkah 3.
2. Pengguna diminta memasukkan umur.
Jika umur ≥ 60 , paparkan "Nilai kalori yang dibenarkan ialah 2010".
Jika umur ≥ 30 , paparkan "Nilai kalori yang dibenarkan ialah 2460".
Jika umur ≥ 19 , paparkan "Nilai kalori yang dibenarkan ialah 2440".
Jika umur ≥ 16 , paparkan "Nilai kalori yang dibenarkan ialah 2840".
Jika umur ≥ 13 , paparkan "Nilai kalori yang dibenarkan ialah 2690".
3. Pengguna diminta memasukkan umur.
Jika umur ≥ 60 , paparkan "Nilai kalori yang dibenarkan ialah 1780".
Jika umur ≥ 30 , paparkan "Nilai kalori yang dibenarkan ialah 2180".
Jika umur ≥ 19 , paparkan "Nilai kalori yang dibenarkan ialah 2000".
Jika umur ≥ 16 , paparkan "Nilai kalori yang dibenarkan ialah 2050".
Jika umur ≥ 13 , paparkan "Nilai kalori yang dibenarkan ialah 2180".

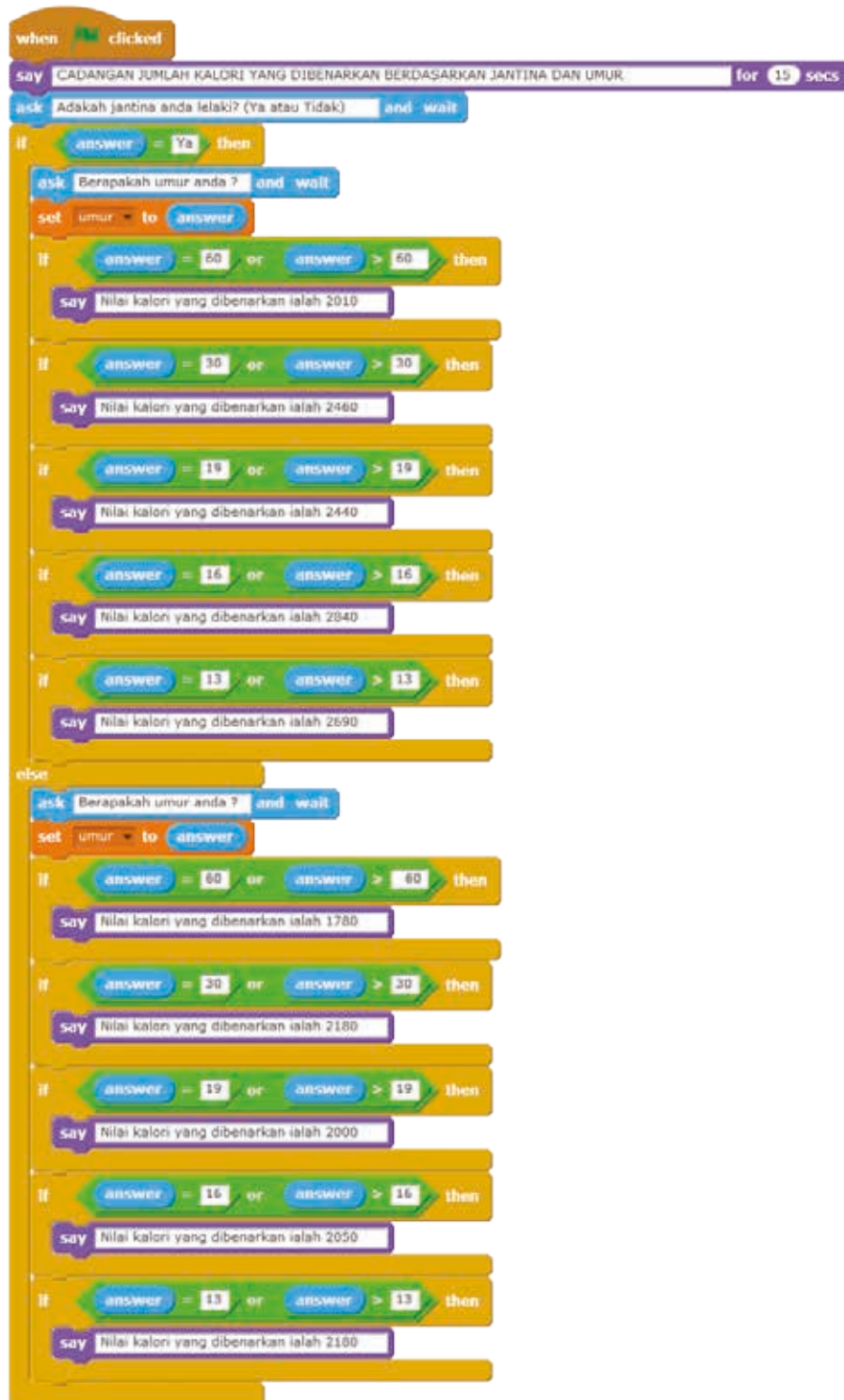
Langkah 3: Melukis carta alir

MULA

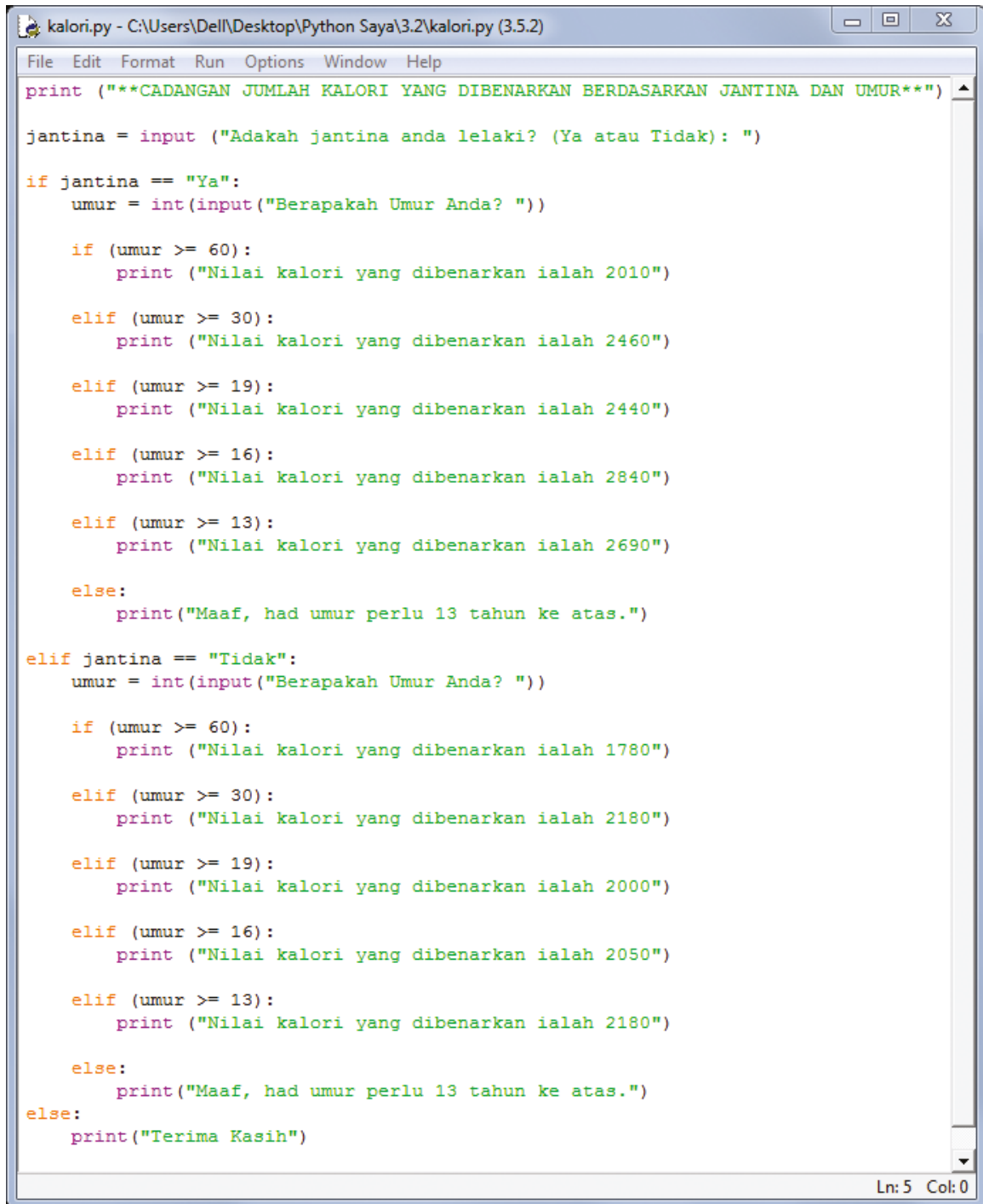


Langkah 4: Pengkodean dengan menggunakan

(a) perisian aplikasi Scratch 2.0



(b) perisian aplikasi Python 3.5.2



```
kalori.py - C:\Users\Dell\Desktop\Python Saya\3.2\kalori.py (3.5.2)
File Edit Format Run Options Window Help
print ("**CADANGAN JUMLAH KALORI YANG DIBENARKAN BERDASARKAN JANTINA DAN UMUR**")

jantina = input ("Adakah jantina anda lelaki? (Ya atau Tidak): ")

if jantina == "Ya":
    umur = int(input("Berapakah Umur Anda? "))

    if (umur >= 60):
        print ("Nilai kalori yang dibenarkan ialah 2010")

    elif (umur >= 30):
        print ("Nilai kalori yang dibenarkan ialah 2460")

    elif (umur >= 19):
        print ("Nilai kalori yang dibenarkan ialah 2440")

    elif (umur >= 16):
        print ("Nilai kalori yang dibenarkan ialah 2840")

    elif (umur >= 13):
        print ("Nilai kalori yang dibenarkan ialah 2690")

    else:
        print("Maaf, had umur perlu 13 tahun ke atas.")

elif jantina == "Tidak":
    umur = int(input("Berapakah Umur Anda? "))

    if (umur >= 60):
        print ("Nilai kalori yang dibenarkan ialah 1780")

    elif (umur >= 30):
        print ("Nilai kalori yang dibenarkan ialah 2180")

    elif (umur >= 19):
        print ("Nilai kalori yang dibenarkan ialah 2000")

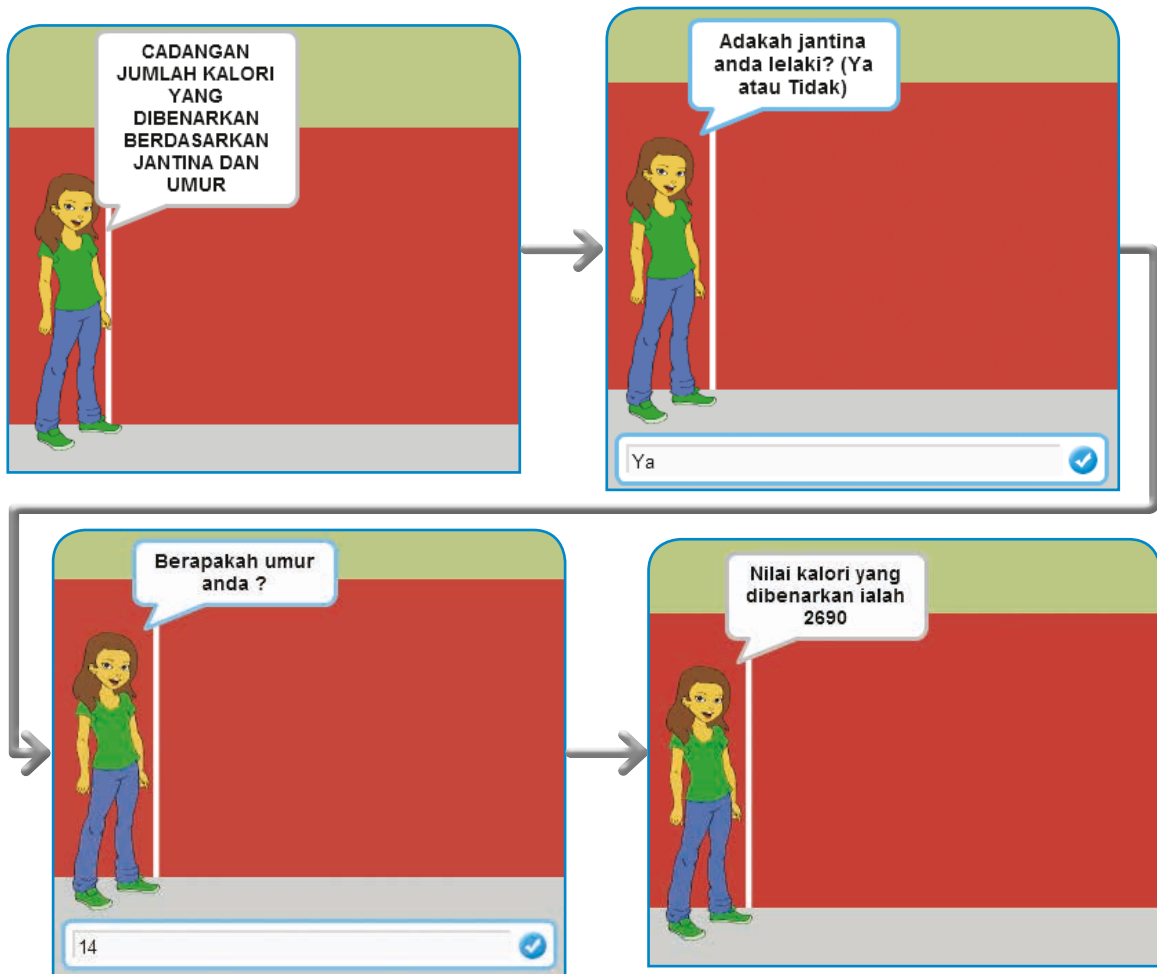
    elif (umur >= 16):
        print ("Nilai kalori yang dibenarkan ialah 2050")

    elif (umur >= 13):
        print ("Nilai kalori yang dibenarkan ialah 2180")

    else:
        print("Maaf, had umur perlu 13 tahun ke atas.")
else:
    print("Terima Kasih")

Ln: 5 Col: 0
```


Langkah 5: Contoh output bagi
(a) perisian aplikasi Scratch 2.0



(b) perisian aplikasi Python 3.5.2

The screenshot shows a Python 3.5.2 Shell window with the following content:

```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Dell\Desktop\Python Saya\3.2\kalori.py =====
***CADANGAN JUMLAH KALORI YANG DIBENARKAN BERDASARKAN JANTINA DAN UMUR***
Adakah jantina anda lelaki? (Ya atau Tidak): Ya
Berapakah Umur Anda? 14
Nilai kalori yang dibenarkan ialah 2690
>>>
```

The status bar at the bottom right indicates "Ln: 9 Col: 4".



Amali Komputer 3.3

Menggunakan struktur kawalan pilihan untuk menyelesaikan masalah

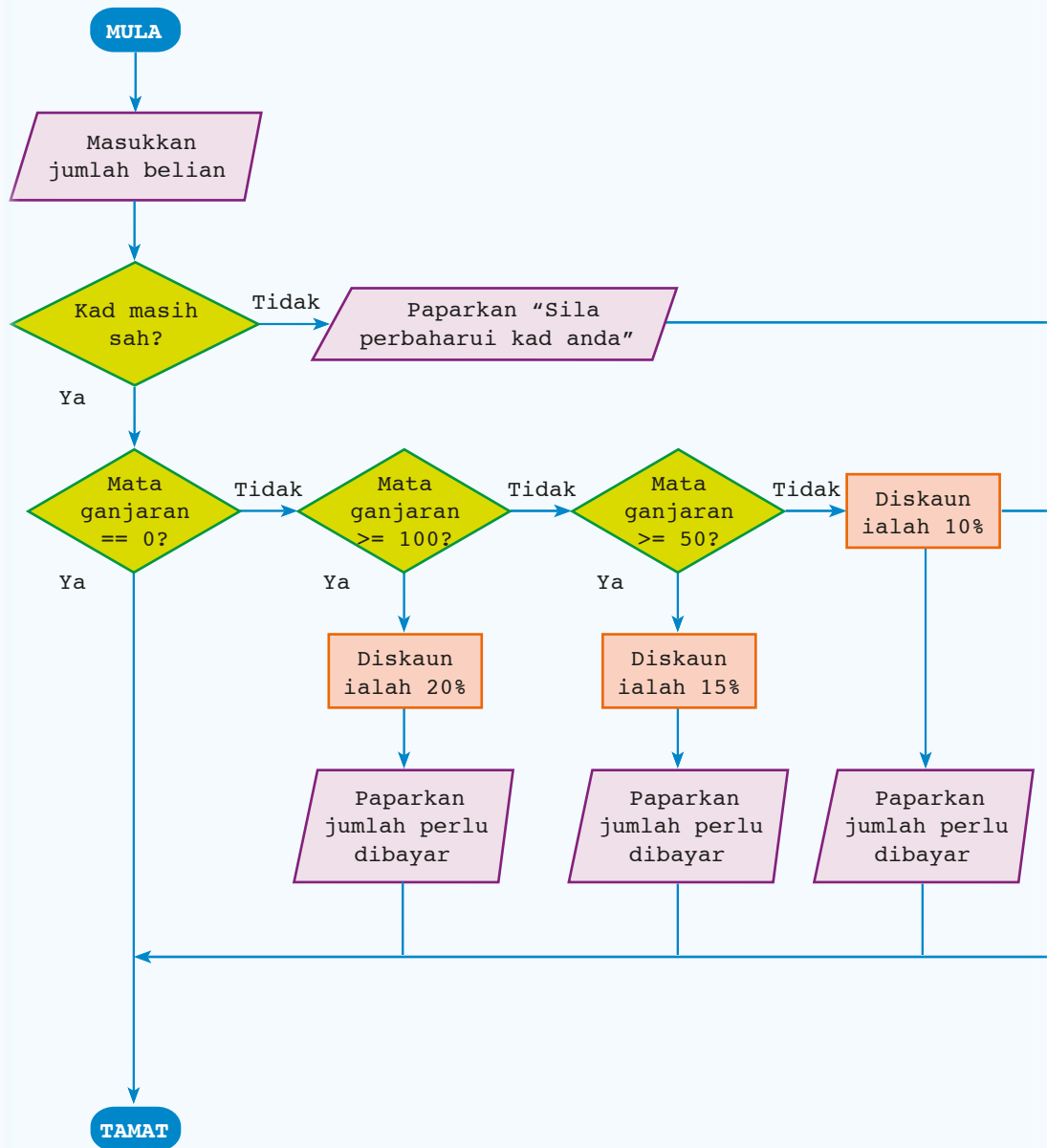
1. Kaji situasi berikut dan kenal pasti permasalahan.

Linda Palikat diberi tugas oleh gurunya untuk membina satu atur cara yang memberi diskaun berdasarkan mata ganjaran yang terkumpul dalam kad diskaun koperasi sekolah. Setiap murid menggunakan kad diskaun untuk membeli alatan sekolah di koperasi sekolah. Terdapat dua jenis kad yang dikeluarkan oleh koperasi sekolah, iaitu kad emas dan kad premium. Atur cara yang akan dibina oleh Linda adalah untuk memegang kad premium.

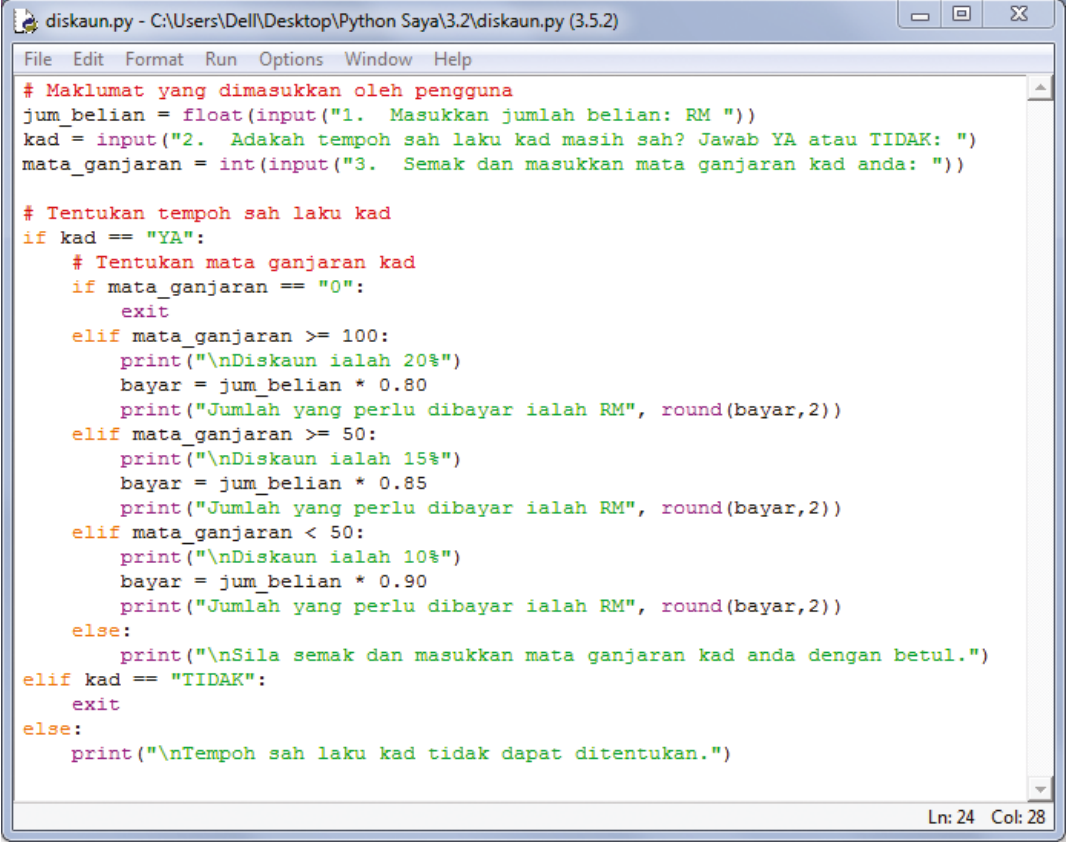
2. Bina algoritma berdasarkan keperluan yang diberi.

1. Mula
2. Masukkan jumlah belian.
3. Tentukan tempoh sah laku kad.
Jika sah, tentukan kadar diskaun.
Jika tidak, paparkan mesej "Sila perbaharui kad anda".
4. Tentukan kadar diskaun.
Jika mata ganjaran ialah 0
Tamat.
Jika mata ganjaran lebih besar atau sama dengan 100
Diskaun ialah 20%.
Paparkan jumlah yang perlu dibayar.
Jika mata ganjaran lebih besar atau sama dengan 50
Diskaun ialah 15%.
Paparkan jumlah yang perlu dibayar.
Jika tidak
Diskaun ialah 10%.
Paparkan jumlah yang perlu dibayar.
5. Tamat.

3. Melukis carta alir berdasarkan algoritma yang dibina.



4. Lancarkan perisian aplikasi Python 3.5.2 dan buka satu dokumen baharu. Taipkan segmen kod seperti yang ditunjukkan di bawah.



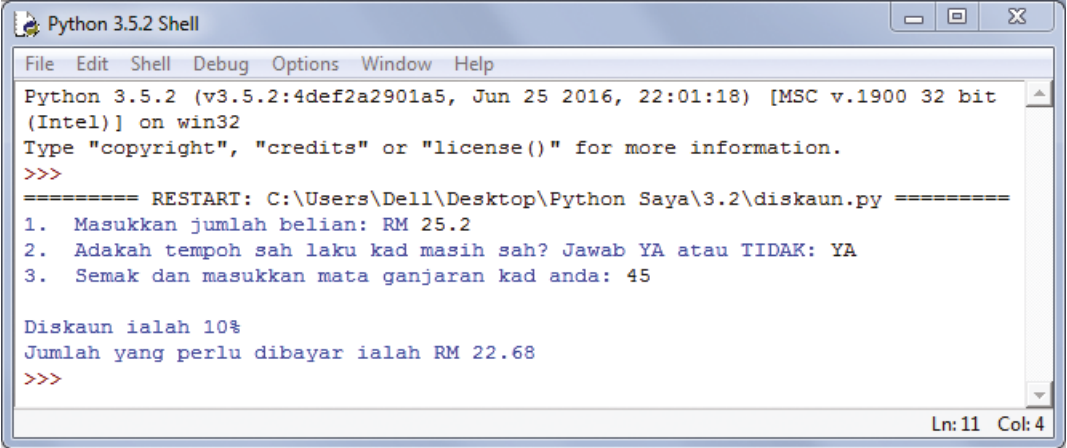
```
diskaun.py - C:\Users\Dell\Desktop\Python Saya\3.2\diskaun.py (3.5.2)
File Edit Format Run Options Window Help

# Maklumat yang dimasukkan oleh pengguna
jum_belian = float(input("1. Masukkan jumlah belian: RM "))
kad = input("2. Adakah tempoh sah laku kad masih sah? Jawab YA atau TIDAK: ")
mata_ganjaran = int(input("3. Semak dan masukkan mata ganjaran kad anda: "))

# Tentukan tempoh sah laku kad
if kad == "YA":
    # Tentukan mata ganjaran kad
    if mata_ganjaran == "0":
        exit
    elif mata_ganjaran >= 100:
        print("\nDiskaun ialah 20%")
        bayar = jum_belian * 0.80
        print("Jumlah yang perlu dibayar ialah RM", round(bayar,2))
    elif mata_ganjaran >= 50:
        print("\nDiskaun ialah 15%")
        bayar = jum_belian * 0.85
        print("Jumlah yang perlu dibayar ialah RM", round(bayar,2))
    elif mata_ganjaran < 50:
        print("\nDiskaun ialah 10%")
        bayar = jum_belian * 0.90
        print("Jumlah yang perlu dibayar ialah RM", round(bayar,2))
    else:
        print("\nSila semak dan masukkan mata ganjaran kad anda dengan betul.")
elif kad == "TIDAK":
    exit
else:
    print("\nTempoh sah laku kad tidak dapat ditentukan.")

Ln: 24 Col: 28
```

5. Simpan fail sebagai **diskaun.py**. Kemudian, klik **Run > Run Module**. Tetingkap IDLE yang menunjukkan output akan dipaparkan. Masukkan data seperti yang ditunjukkan di bawah.



```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help

Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit
(Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Dell\Desktop\Python Saya\3.2\diskaun.py =====
1. Masukkan jumlah belian: RM 25.2
2. Adakah tempoh sah laku kad masih sah? Jawab YA atau TIDAK: YA
3. Semak dan masukkan mata ganjaran kad anda: 45

Diskaun ialah 10%
Jumlah yang perlu dibayar ialah RM 22.68
>>>

Ln: 11 Col: 4
```



Aktiviti 3.8

Individu

1. Dengan menggunakan struktur kawalan pilihan yang sesuai, bina satu atur cara yang interaktif dan dapat merangkumi semua input pengguna. Jadual berikut ialah syarat-syarat yang perlu disemak oleh atur cara.

Umur yang dimasukkan	Paparan
>30	Anda ialah seorang dewasa.
>14	Anda ialah seorang belia.
>12	Anda ialah seorang remaja.
Lain-lain	Anda ialah seorang kanak-kanak.

- (a) Nyatakan struktur kawalan pilihan yang akan digunakan.
- (b) Lukiskan carta alir bagi atur cara yang perlu dibina.
- (c) Tuliskan kod arahan Python bagi atur cara anda.

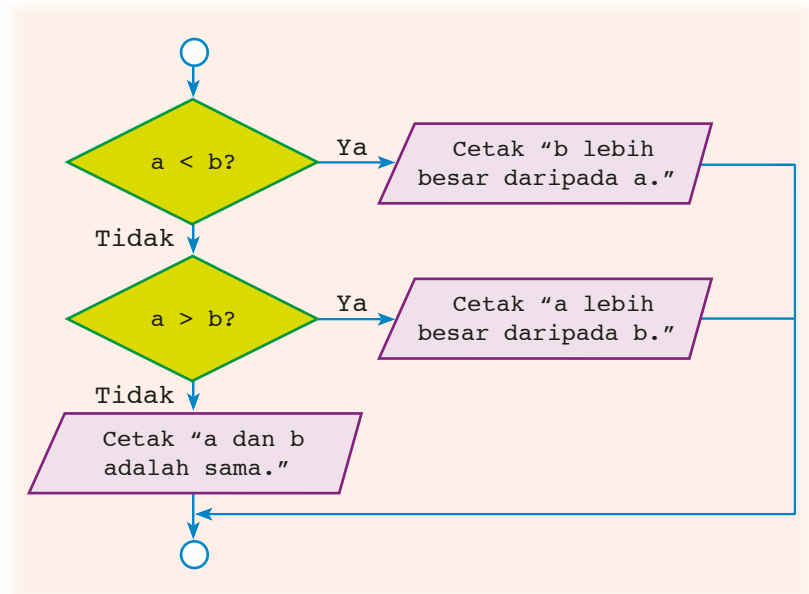
Standard Pembelajaran

Murid boleh:

- 3.2.1** Menghasilkan atur cara yang melibatkan:
- (iii) struktur kawalan pilihan bersarang dalam penyelesaian masalah.

3.2.1 (iii) Struktur kawalan pilihan bersarang dalam penyelesaian masalah

Struktur kawalan pilihan bersarang memberi maksud suatu struktur kawalan pilihan berada di dalam suatu struktur kawalan pilihan yang lain. Rajah 3.13 menunjukkan carta alir bagi struktur kawalan pilihan bersarang yang membandingkan nombor a dan b.



Rajah 3.13 Carta alir struktur kawalan pilihan bersarang yang membandingkan nombor a dan b

Berdasarkan Rajah 3.13, pernyataan bersyarat yang pertama mempunyai dua cabang. Dalam salah satu cabang, terdapat pernyataan bersyarat kedua yang mempunyai dua cabang lagi. Cabang daripada pernyataan bersyarat kedua boleh mengandungi pernyataan bersyarat yang seterusnya. Rangka struktur kawalan pilihan ini, yang berupa lapisan bersarang menjadikan namanya struktur kawalan pilihan bersarang.

Contoh 3.26

Penyelesaian masalah menggunakan struktur kawalan pilihan bersarang.

Kishen diberi tugas oleh gurunya untuk menulis satu atur cara yang meminta pengguna memasukkan satu nombor integer kecuali 0. Atur cara ini akan mengenal pasti dan memaparkan integer tersebut sebagai integer negatif atau integer positif. Jika integer positif, atur cara pula akan mengenal pasti dan memaparkan integer tersebut ialah nombor ganjil atau nombor genap.

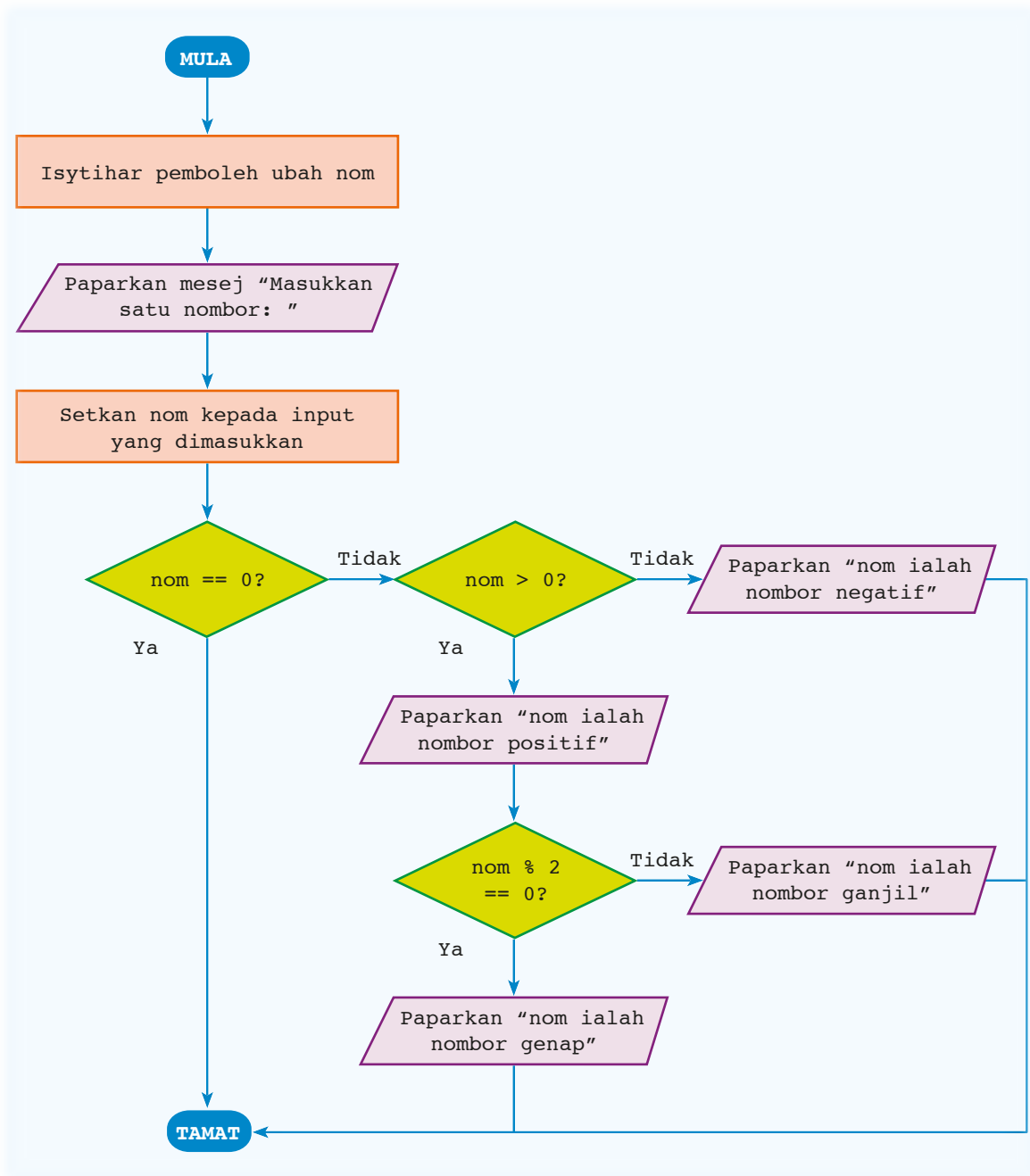
Langkah 1: Pernyataan masalah

1. Mengetahui pasti input yang diperlukan, iaitu satu nombor integer daripada pengguna selain sifar.
2. Mengetahui pasti proses yang terlibat.
 - (a) Proses mengetahui pasti integer yang dimasukkan bukan sifar.
 - (b) Proses mengetahui pasti integer yang dimasukkan adalah negatif atau positif.
 - (c) Proses mengetahui pasti integer positif ialah nombor genap atau nombor ganjil.
3. Mengetahui pasti output yang diperlukan, iaitu integer adalah positif atau negatif dan integer positif ialah nombor genap atau ganjil.

Langkah 2: Menulis algoritma bagi penyelesaian masalah

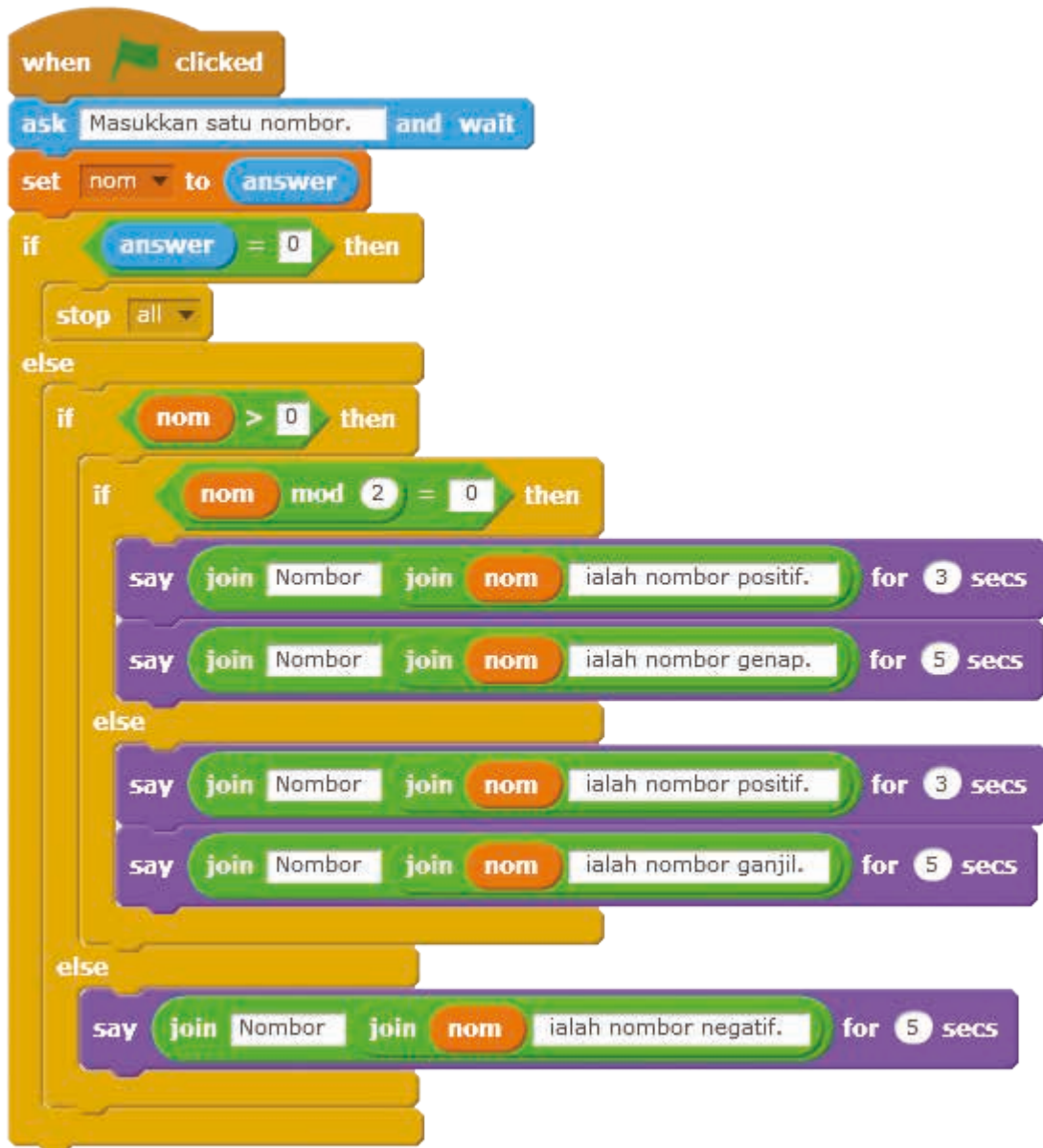
1. Isytihar pemboleh ubah nom.
2. Pengguna memasukkan satu nombor integer.
3. Setkan nom kepada input yang dimasukkan.
4. Semak nom ialah sifar.
Jika ya, atur cara tamat.
Jika tidak, langkau ke Langkah 5.
5. Semak nom lebih besar daripada sifar.
Jika ya, paparkan "nom ialah nombor positif".
Jika tidak, paparkan "nom ialah nombor negatif".
6. Semak hasil nom % 2 ialah sifar.
Jika ya, paparkan "nom ialah nombor genap".
Jika tidak, paparkan "nom ialah nombor ganjil".

Langkah 3: Melukis carta alir

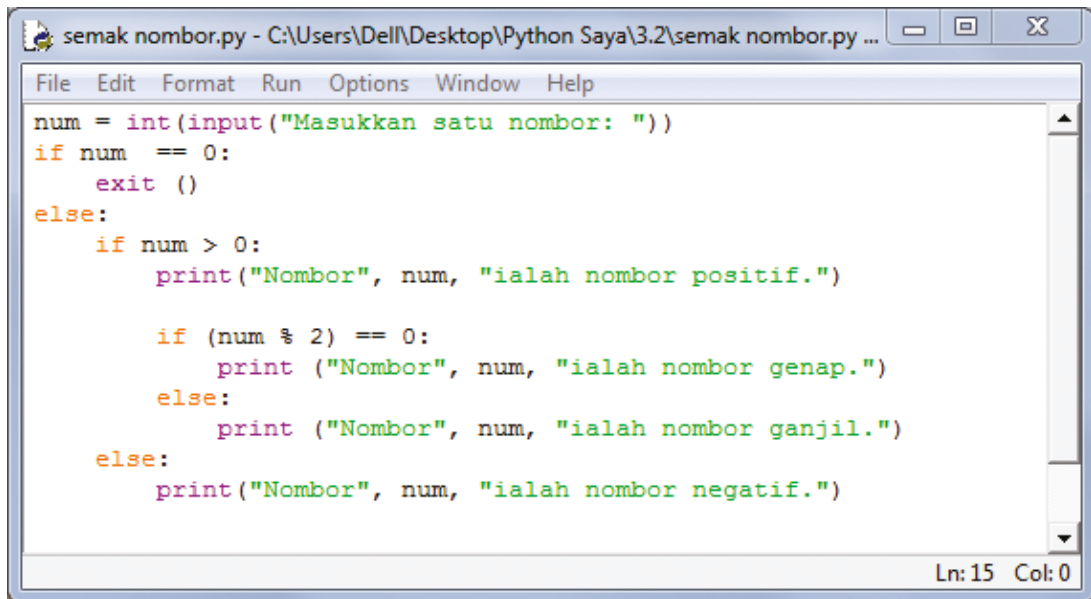


Langkah 4: Pengekodaan dengan menggunakan

(a) perisian aplikasi Scratch 2.0



(b) perisian aplikasi Python 3.5.2

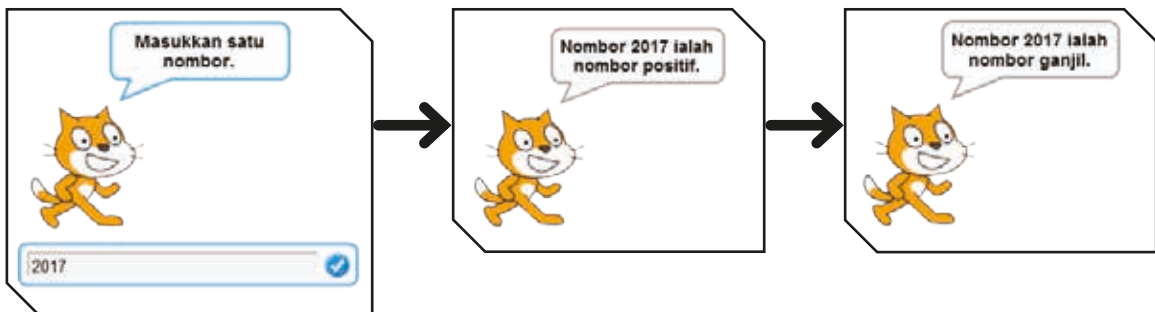


```
File Edit Format Run Options Window Help
num = int(input("Masukkan satu nombor: "))
if num == 0:
    exit ()
else:
    if num > 0:
        print("Nombor", num, "ialah nombor positif.")

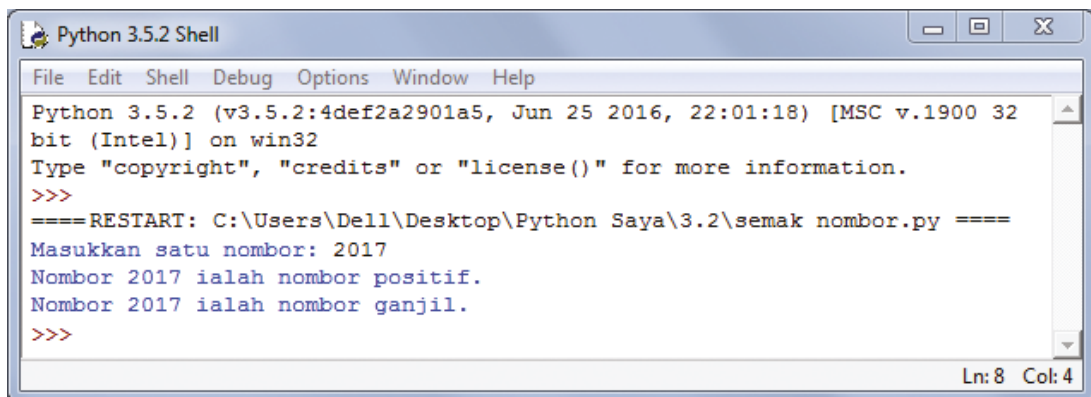
        if (num % 2) == 0:
            print ("Nombor", num, "ialah nombor genap.")
        else:
            print ("Nombor", num, "ialah nombor ganjil.")
    else:
        print("Nombor", num, "ialah nombor negatif.")
Ln: 15 Col: 0
```

Langkah 5: Contoh output bagi

(a) perisian aplikasi Scratch 2.0



(b) perisian aplikasi Python 3.5.2



```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32
bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
====RESTART: C:\Users\Dell\Desktop\Python Saya\3.2\semak nombor.py====
Masukkan satu nombor: 2017
Nombor 2017 ialah nombor positif.
Nombor 2017 ialah nombor ganjil.
>>>
Ln: 8 Col: 4
```

Lakukan **Amali Komputer 3.4** untuk mencuba struktur kawalan pilihan bersarang dengan menggunakan Python. Anda akan membina satu atur cara yang menyemak tahun lompat.



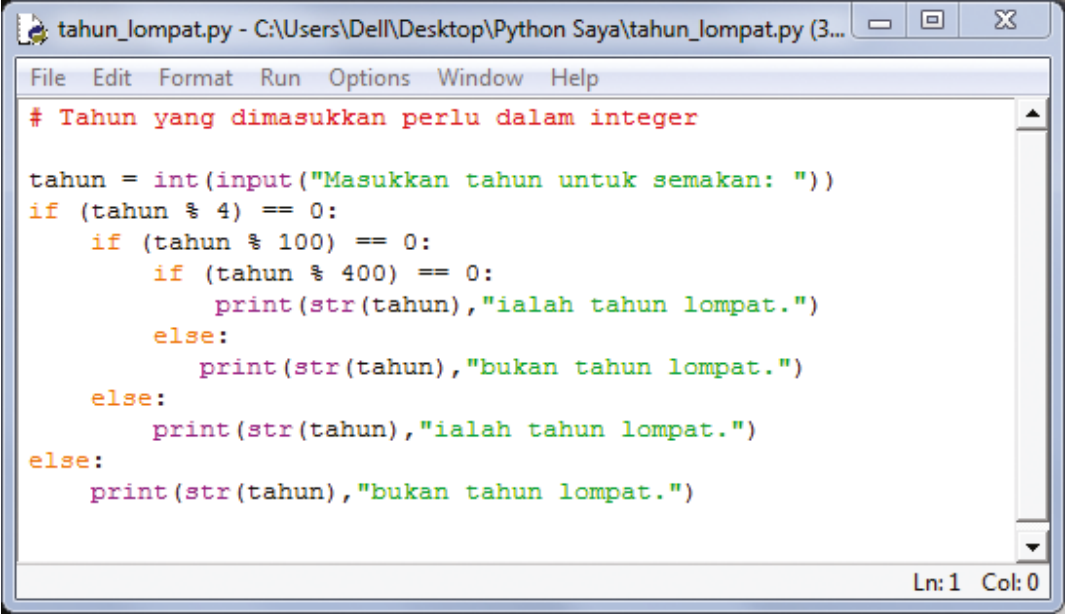
Amali Komputer 3.4

Membina satu atur cara yang menentukan tahun lompat.

1. Gunakan perisian aplikasi IDLE Python 3.5.2 dan lancarkan satu dokumen baharu.

Windows Start Button > All Programs > IDLE (Python 3.5) > File > New File

2. Taipkan segmen kod yang ditunjukkan di bawah ke dalam tetingkap dokumen baharu IDLE Python 3.5.2.

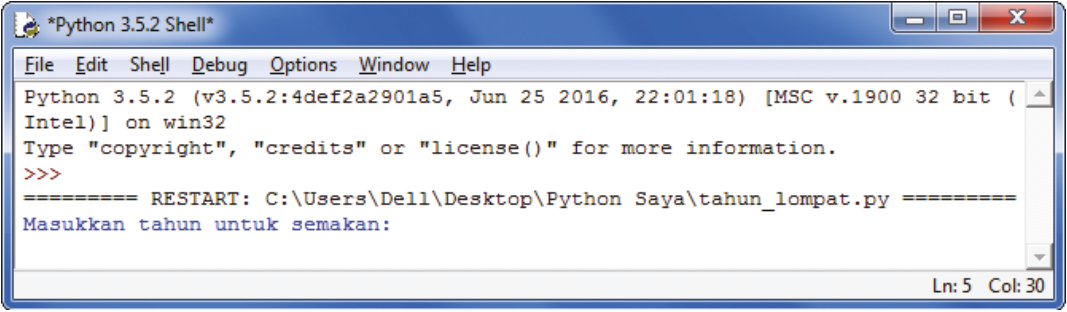


```
tahun_lompat.py - C:\Users\Dell\Desktop\Python Saya\tahun_lompat.py (3...
File Edit Format Run Options Window Help
# Tahun yang dimasukkan perlu dalam integer

tahun = int(input("Masukkan tahun untuk semakan: "))
if (tahun % 4) == 0:
    if (tahun % 100) == 0:
        if (tahun % 400) == 0:
            print(str(tahun), "ialah tahun lompat.")
        else:
            print(str(tahun), "bukan tahun lompat.")
    else:
        print(str(tahun), "ialah tahun lompat.")
else:
    print(str(tahun), "bukan tahun lompat.")

Ln: 1 Col: 0
```

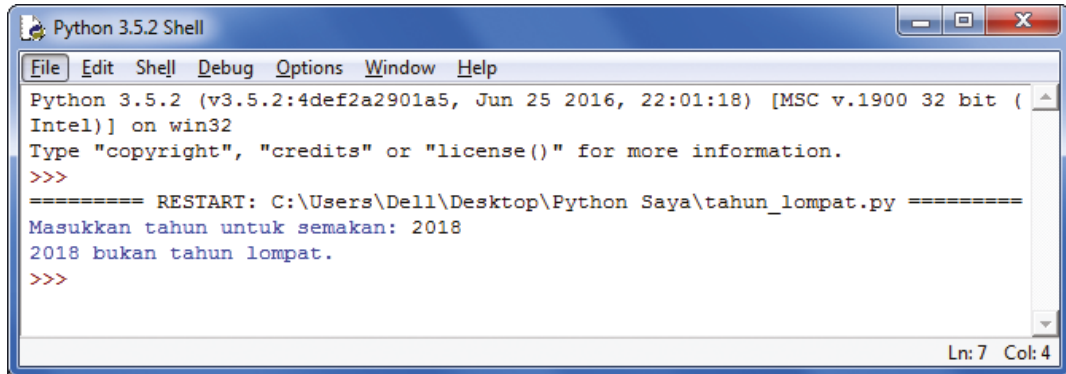
3. Pilih menu **File > Save As** untuk menyimpan fail anda sebagai **tahun_lompat.py** dalam folder **Python Saya**.
4. Kemudian pada tetingkap fail **tahun_lompat.py** yang masih aktif, klik menu **Run > Run Module**. Tetingkap baharu seperti yang ditunjukkan akan dipaparkan.



```
*Python 3.5.2 Shell*
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Dell\Desktop\Python Saya\tahun_lompat.py =====
Masukkan tahun untuk semakan:

Ln: 5 Col: 30
```

5. Jika terdapat paparan mesej ralat, baiki ralat tersebut.
6. Masukkan nombor 2018 dan tekan **Enter**. Paparan skrin yang berikut dapat diperhatikan.



```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Dell\Desktop\Python Saya\tahun_lompat.py =====
Masukkan tahun untuk semakan: 2018
2018 bukan tahun lompat.
>>>
```

7. Ulangi langkah 6 dengan nombor-nombor lain untuk mencuba atur cara anda.
8. Jika paparan yang diperoleh berlainan dengan yang dijangka, semak semula segmen kod anda. Baiki ralat yang dijumpai.



Fakta Segera

Sukan Olimpik Musim Panas yang diadakan setiap empat tahun sekali semuanya berlaku pada tahun lompat. Cuba masukkan tahun-tahun ini untuk mencuba atur cara anda.



EMK

Sains & Teknologi

Tahun lompat ialah tahun di mana satu hari ditambahkan ke bulan Februari, menjadikannya 29 hari dan bukannya 28 hari. Mengikut pengiraan yang tepat, setahun mengandungi 365.242 hari. Oleh yang demikian, penyelarasan perlu dilakukan setiap empat tahun ($0.242 \times 4 = \sim 1$ hari) dan 1 hari ini disebut hari lompat.



Aktiviti 3.9

Pasangan

Menambah baik atur cara dalam Amali Komputer 3.4.

1. Kod arahan bagi atur cara dalam **Amali Komputer 3.4** adalah panjang dan mempunyai ayat-ayat yang berulang.
2. Bincang bersama-sama rakan sekumpulan anda untuk memperbaiki kod arahan ini supaya menjadi lebih ringkas.
3. Output yang diperoleh mesti serupa dengan output yang anda peroleh dalam **Amali Komputer 3.4**.

3.2.1 (iv) Struktur kawalan ulangan dalam penyelesaian masalah

Struktur kawalan ulangan membolehkan pengatur cara mengarah komputer untuk melaksanakan langkah-langkah tertentu secara berulang kali. Struktur kawalan ulangan adalah penting supaya kod yang ditulis tidak perlu panjang dan memakan ruang. Selain itu, penambahbaikan kepada kod struktur kawalan ulangan adalah lebih mudah berbanding dengan kod atur cara yang tidak menggunakan gelung.

Semasa mengekodkan struktur kawalan ulangan, biasanya pengatur cara perlu menggunakan sintaks pemboleh ubah yang cekap untuk bertindak sebagai pembilang, terutamanya untuk struktur yang dikawal dengan bilangan.

Struktur kawalan ulangan boleh menerima arahan input untuk melakukan operasi matematik seperti penambahan dua senarai nombor, mengira purata beberapa nombor dan pelbagai operasi olahan nombor yang lain.

Bahagian kod yang mengandungi syarat dan langkah-langkah untuk berulang disebut **gelung** (*loop*). Manakala, setiap pusingan ulangan yang dilaksanakan disebut **lelaran** (*iteration*). Dua jenis struktur kawalan ulangan yang sering digunakan ialah **struktur kawalan UNTUK** (*for*) dan **struktur kawalan SELAGI** (*while*).

Standard Pembelajaran

Murid boleh:

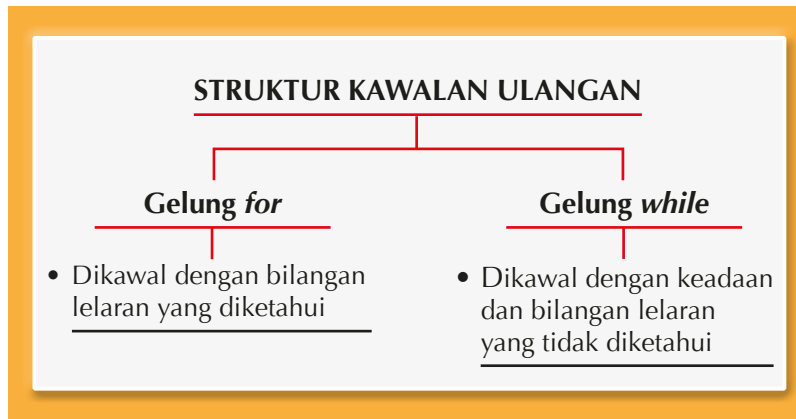
3.2.1 Menghasilkan atur cara yang melibatkan:

- (iv) struktur kawalan ulangan dalam penyelesaian masalah.



IMBAS KEMBALI

Struktur kawalan ulangan telah dipelajari dalam Bab 2 Algoritma.



Rajah 3.14 Pengelasan struktur kawalan ulangan

Struktur kawalan UNTUK dan struktur kawalan SELAGI sesuai digunakan sebagai struktur kawalan praujian (*pretest*) yang akan menguji syarat sebelum melaksanakan lelaran. Penyelesaian masalah dengan menggunakan struktur kawalan UNTUK dan struktur kawalan SELAGI akan diterangkan dengan menggunakan bahasa pengaturcaraan Python.

Struktur kawalan ulangan membolehkan sesuatu segmen kod yang panjang dapat diringkaskan.



Contoh 3.27

Penyelesaian masalah menggunakan struktur kawalan ulangan.

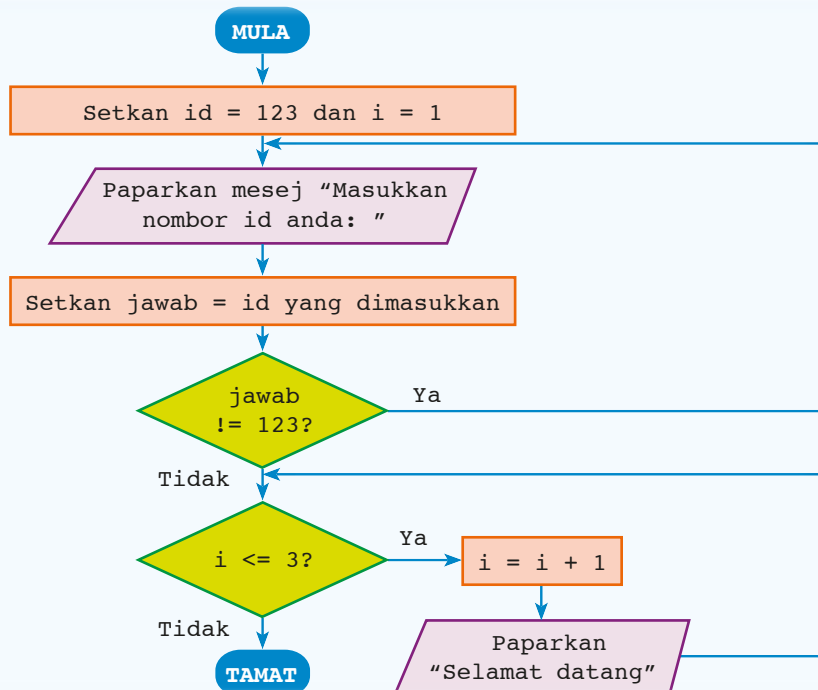
Permainan interaktif dalam telefon pintar menggunakan struktur kawalan ulangan bagi mengawal fungsi log masuk ke aplikasi. Balqis diminta oleh guru untuk membangunkan atur cara yang mengawal log masuk dengan menggunakan struktur kawalan ulangan. Jika pengguna berjaya log masuk, atur cara akan paparkan “Selamat datang” sebanyak tiga kali.

Langkah 1: Pernyataan masalah

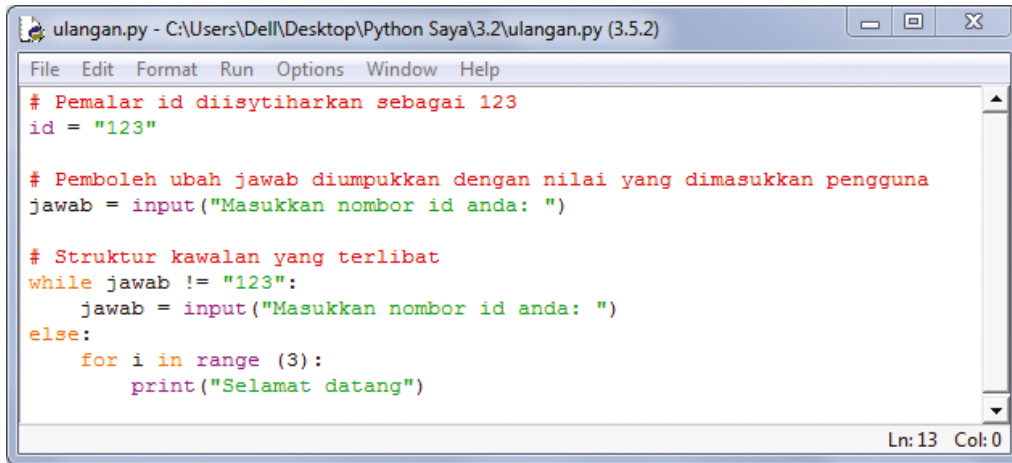
1. Menenal pasti input yang diperlukan, iaitu id untuk log masuk.
2. Menenal pasti proses dan struktur kawalan yang terlibat untuk log masuk.
3. Menenal pasti proses dan struktur kawalan yang terlibat untuk paparan “Selamat datang” sebanyak tiga kali.

Langkah 2: Menulis algoritma bagi penyelesaian masalah

1. Isytiharkan pemboleh ubah id dan i
2. Setkan id = 123 dan i = 1
3. Pengguna memasukkan id
4. Setkan jawab kepada id yang dimasukkan
5. while jawab != 123
 Ulang Langkah 3
6. for i <= 3
 Paparkan “Selamat datang”

Langkah 3: Melukis carta alir

Langkah 4: Pengekodaan dengan menggunakan perisian aplikasi Python 3.5.2



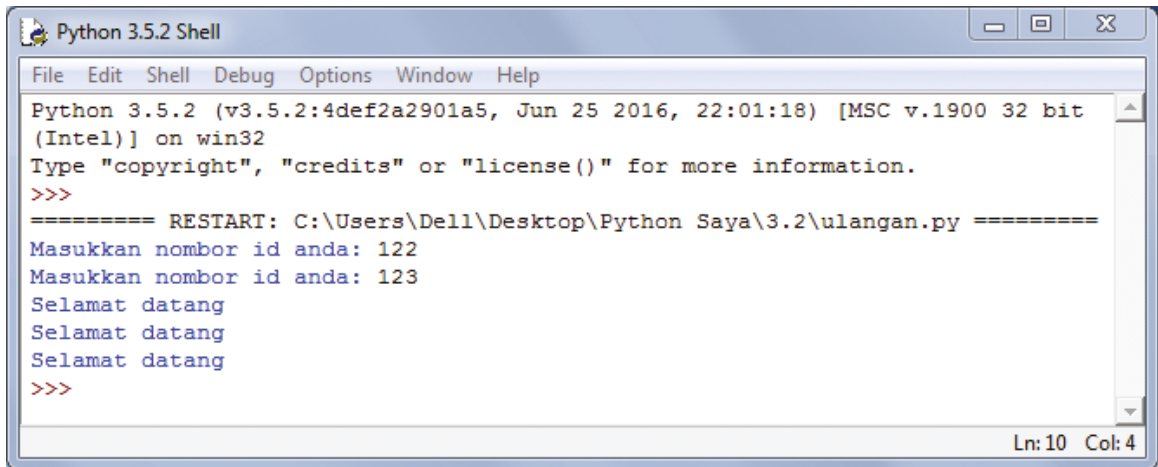
```
ulangan.py - C:\Users\Dell\Desktop\Python Saya\3.2\ulangan.py (3.5.2)
File Edit Format Run Options Window Help
# Pemalar id diisytiharkan sebagai 123
id = "123"

# Pemboleh ubah jawab diumpukkan dengan nilai yang dimasukkan pengguna
jawab = input("Masukkan nombor id anda: ")

# Struktur kawalan yang terlibat
while jawab != "123":
    jawab = input("Masukkan nombor id anda: ")
else:
    for i in range (3):
        print("Selamat datang")

Ln: 13 Col: 0
```

Langkah 5: Contoh output



```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit
(Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Dell\Desktop\Python Saya\3.2\ulangan.py =====
Masukkan nombor id anda: 122
Masukkan nombor id anda: 123
Selamat datang
Selamat datang
Selamat datang
>>>

Ln: 10 Col: 4
```

Dalam pengaturcaraan Python, sintaks **for** digunakan untuk struktur kawalan UNTUK. Berikut ialah rangka umum penggunaan sintaks **for**.

for pemboleh ubah gelung **in** [1, 2, 3, dll.]:
Pernyataan

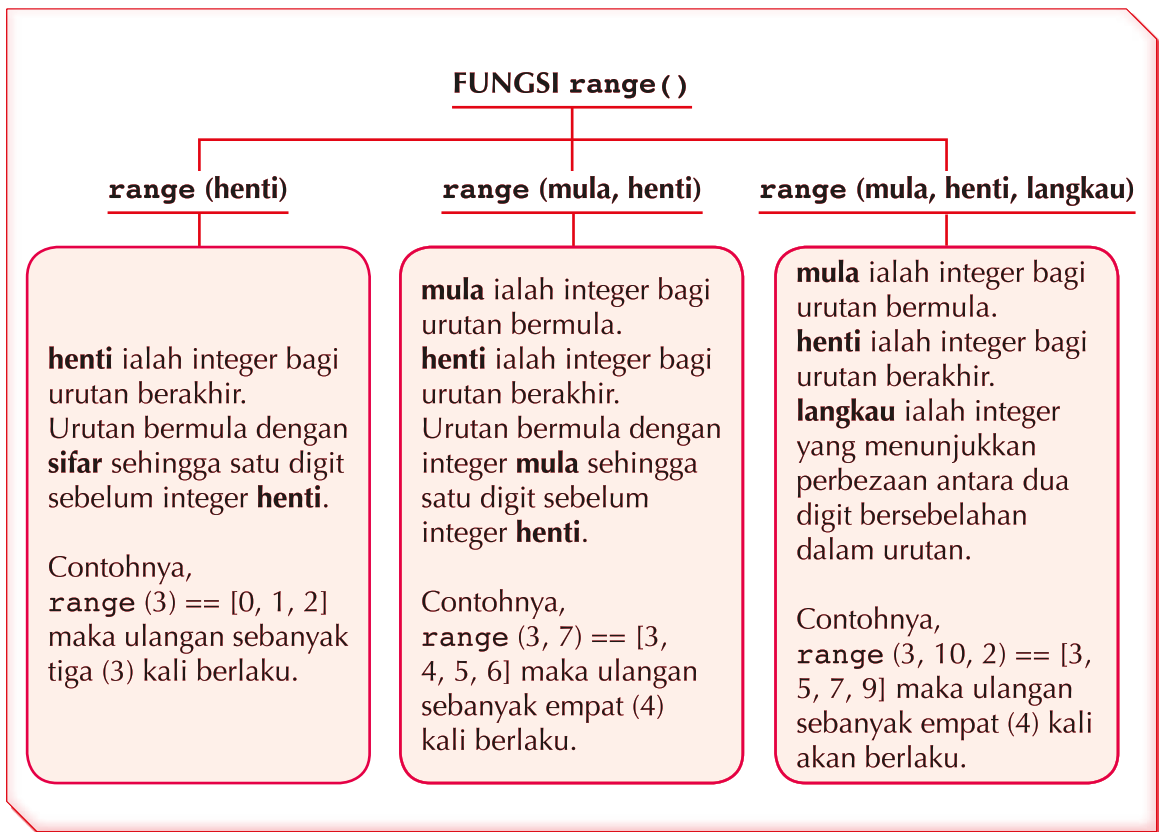
Pemboleh ubah gelung akan terbentuk sebaik sahaja sintaks **for** ditaip. Maka, pemboleh ubah gelung tidak perlu diisytiharkan terlebih dahulu. Lelaran bermula dengan pemboleh ubah gelung terumpuk dengan nilai pertama dalam data dan melaksanakan kod Pernyataan. Lelaran berterusan mengikut urutan dalam data dan berhenti apabila sampai ke nilai terakhir.





Pengatur cara memerlukan sebuah gelung yang berulang kali bagi bilangan lelaran yang ditetapkan. Contohnya, jika pengguna menjawab soalan dengan betul, atur cara akan menghasilkan output perkataan “Tahniah” sebanyak lima kali. Dalam situasi ini, penggunaan fungsi `range()` adalah amat berguna kerana fungsi `range()` menyatakan bilangan lelaran struktur kawalan *for* harus berulang.

Nombor yang digunakan dalam fungsi `range()` disebut parameter. Parameter yang digunakan dalam fungsi `range()` mestilah integer yang terdiri daripada nombor positif atau nombor negatif. Rajah 3.15 menunjukkan tiga cara penulisan parameter bagi fungsi `range()`.



Rajah 3.15 Tiga cara penulisan parameter bagi fungsi `range()`



Cabar Diri Anda

Senaraikan nombor-nombor bagi fungsi `range()`.

1. `range(10)`
2. `range(5, 16)`
3. `range(0, 30, 5)`
4. `range(5, -1, -1)`



Teknik
Pengitlakan

TIP

1. Inden untuk segmen kod yang berulang mestilah sama.
2. Mencetak pemboleh ubah dengan fungsi `print ( )` tidak memerlukan tanda petikan berganda (`" "`).

Dalam pengaturcaraan Python, sintaks *while* digunakan untuk struktur kawalan SELAGI. Berikut ialah rangka umum penggunaan sintaks *while*.

while Syarat dalam bentuk ungkapan *Boolean*:
Pernyataan

Sintaks untuk struktur kawalan *while* adalah serupa dengan rangka umum kenyataan *if*. Ungkapan *Boolean* ialah syarat (benar atau tidak benar) yang menentukan sama ada gelung akan berhenti atau terus berulang.



Amali Komputer 3.3

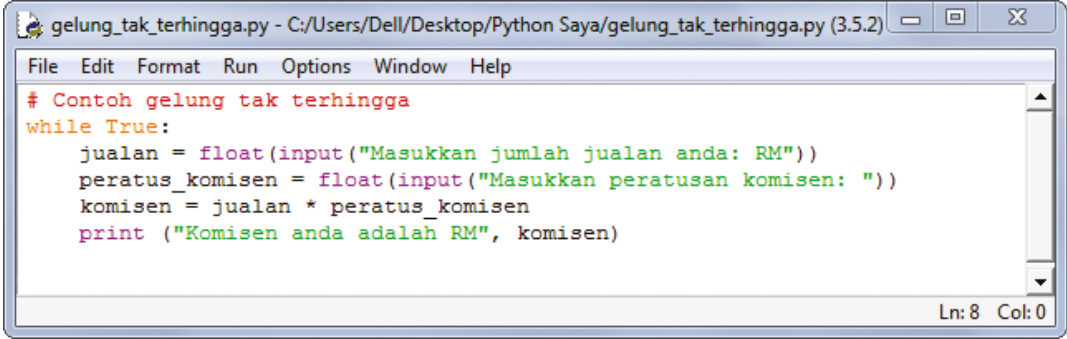
Membina satu atur cara dengan gelung *while*.

A Gelung tak terhingga

1. Gunakan perisian aplikasi **IDLE Python 3.5.2** dan lancarkan satu dokumen baharu.

Windows Start Button > All Programs > IDLE (Python 3.5) > File > New File

2. Taipkan kod segmen seperti yang ditunjukkan di bawah.

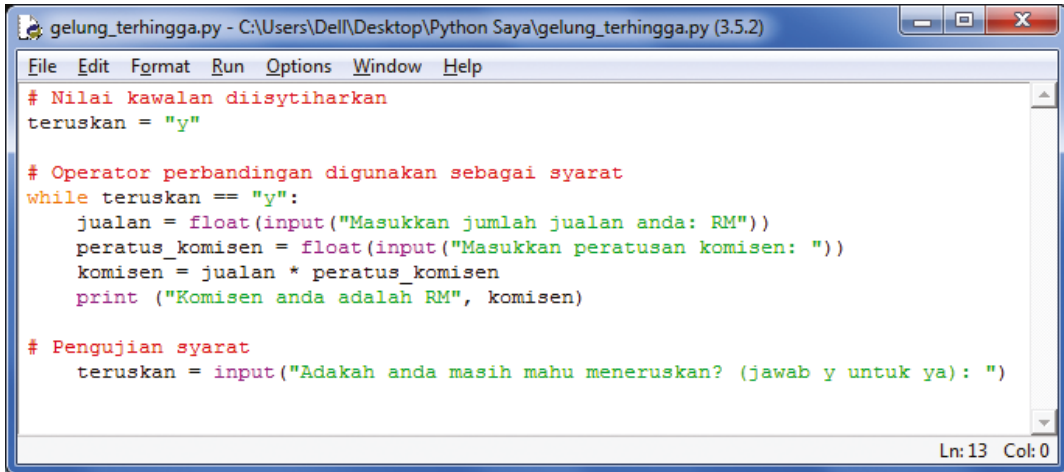


```
gelung_tak_terhingga.py - C:/Users/Dell/Desktop/Python Saya/gelung_tak_terhingga.py (3.5.2)
File Edit Format Run Options Window Help
# Contoh gelung tak terhingga
while True:
    jualan = float(input("Masukkan jumlah jualan anda: RM"))
    peratus_komisen = float(input("Masukkan peratusan komisen: "))
    komisen = jualan * peratus_komisen
    print ("Komisen anda adalah RM", komisen)
```

3. Pilih menu **File > Save As** untuk menyimpan fail anda dalam folder **Python Saya**. Namakan fail anda sebagai **gelung_tak_terhingga.py**.
4. Kemudian pada tettingkap fail **gelung_tak_terhingga.py** yang masih aktif, klik menu **Run > Run Module** untuk melaksanakan kod.
5. Jika terdapat paparan mesej ralat, baiki ralat dan ulangi langkah 4 sehingga tiada ralat lagi.
6. Langkah dalam badan gelung akan sentiasa berulang. Tekan **Ctrl + C** di papan kekunci untuk menghentikan gelung tak terhingga.

B Membetulkan satu gelung tak terhingga kepada gelung terhingga

1. Lancarkan satu dokumen baharu dengan memilih menu **File > New File**.
2. Taipkan segmen kod dalam berikut yang telah ditambah baik supaya gelung *while* yang dibina tidak menjadi gelung tak terhingga.



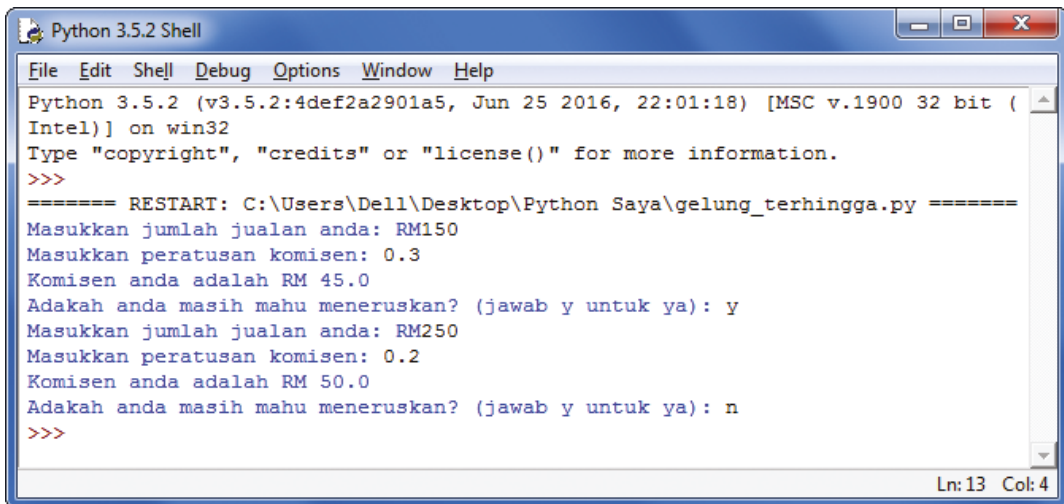
```
gelung_terhingga.py - C:\Users\Dell\Desktop\Python Saya\gelung_terhingga.py (3.5.2)
File Edit Format Run Options Window Help
# Nilai kawalan diisytiharkan
teruskan = "y"

# Operator perbandingan digunakan sebagai syarat
while teruskan == "y":
    jualan = float(input("Masukkan jumlah jualan anda: RM"))
    peratus_komisen = float(input("Masukkan peratusan komisen: "))
    komisen = jualan * peratus_komisen
    print ("Komisen anda adalah RM", komisen)

# Pengujian syarat
teruskan = input("Adakah anda masih mahu meneruskan? (jawab y untuk ya): ")

Ln: 13 Col: 0
```

3. Pilih menu **File > Save As** untuk menyimpan fail anda dalam folder **Python Saya**. Namakan fail anda sebagai **gelung_terhingga.py**.
4. Kemudian pada tettingkap fail **gelung_terhingga.py** yang masih aktif, klik menu **Run > Run Module** untuk melaksanakan segmen kod. Paparan seperti yang berikut akan diperhatikan. Jika terdapat paparan mesej ralat, baiki ralat dan cuba sekali lagi.



```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Dell\Desktop\Python Saya\gelung_terhingga.py =====
Masukkan jumlah jualan anda: RM150
Masukkan peratusan komisen: 0.3
Komisen anda adalah RM 45.0
Adakah anda masih mahu meneruskan? (jawab y untuk ya): y
Masukkan jumlah jualan anda: RM250
Masukkan peratusan komisen: 0.2
Komisen anda adalah RM 50.0
Adakah anda masih mahu meneruskan? (jawab y untuk ya): n
>>>

Ln: 13 Col: 4
```

5. Uji atur cara anda dengan menaipkan nilai-nilai ke dalam atur cara.
6. Bandingkan atur cara yang dibina dalam Bahagian A dan Bahagian B. Nyatakan perbezaan utama antara kedua-duanya.
7. Atur cara yang manakah lebih bagus? Terangkan sebab anda.



Aktiviti 3.10

Kumpulan

Menghasilkan satu atur cara yang menjana jadual sifir darab.

Gunakan struktur kawalan UNTUK bagi menghasilkan satu atur cara yang memaparkan sifir darab apabila pengguna memasukkan satu angka bulat. Paparan berikut ialah contoh output untuk atur cara anda.

```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:\Users\Dell\Desktop\Python Saya\sifir darab.py =====
***ATUR CARA INI AKAN MENJANA JADUAL SIFIR DARAB ***
Masukkan nombor Jadual Sifir Darab anda: 5
Jadual Sifir Darab 5 ialah:
1 x 5 = 5
2 x 5 = 10
3 x 5 = 15
4 x 5 = 20
5 x 5 = 25
6 x 5 = 30
7 x 5 = 35
8 x 5 = 40
9 x 5 = 45
10 x 5 = 50
11 x 5 = 55
12 x 5 = 60
>>>
```

3.2.2 Menguji atur cara dan membaiki ralat pada kod arahan

Menguji atur cara merupakan sebahagian daripada proses pembangunan perisian aplikasi komputer. Setiap kali baris-baris kod ditulis, diubah atau dipadamkan, baris-baris kod mesti melalui proses pengujian bagi memastikan atur cara tersebut masih bertindak sebagaimana yang diharapkan. Ini bermakna, pengujian atur cara adalah penting dan merupakan langkah yang tidak dapat dielakkan semasa pembangunan perisian aplikasi komputer.

Standard Pembelajaran

Murid boleh:
3.2.2 Menguji atur cara dan membaiki ralat pada kod arahan yang dihasilkan.



Bagi murid yang baru sahaja mempelajari pembangunan sesebuah atur cara lazimnya melakukan kesilapan menaip kod atur cara disebabkan kurang berpengalaman dalam bahasa pengaturcaraan tersebut. Kecuaian menaip kod atur cara juga menjadi salah satu faktor terjadinya ralat dalam kod atur cara.

TIP

Kejadian ralat yang sering dijumpai ialah:

- Lupa menaip penutup kurungan “)” semasa menggunakan fungsi `input()`.
- Inden kenyataan `else` yang salah.



Ralat atau kesilapan yang wujud dalam sesuatu atur cara sering dirujuk sebagai **pepijat**. Lazimnya, ralat berpunca daripada kesalahan ataupun kecuai pengatur cara. Proses mencari dan menghapuskan ralat dipanggil **penyahpepijat**. Ralat-ralat dalam atur cara boleh dikelaskan kepada tiga kumpulan mengikut ciri-cirinya yang tersendiri. Tiga kumpulan ini ialah **ralat sintaks** (*syntax error*), **ralat masa larian** (*runtime error*) dan **ralat logik** (*logical error*).



IMBAS KEMBALI

Maklumat lanjut jenis-jenis ralat telah diterangkan dalam Bab 2 dan Subtopik 3.1. Topik ini akan tertumpu kepada menguji atur cara dan membaiki ralat.

Contoh 3.28

Mengesan dan memperbaiki ralat pada kod arahan yang dihasilkan.

Anda telah dilantik menjadi ketua Program Jualan Amal di sekolah. Lima orang rakan anda turut membantu untuk menjayakan jualan amal tersebut. Pasukan anda hanya bertanggungjawab menjual tiga jenis barangan, iaitu buah-buahan, minuman dan biskut.

Bagi memberi semangat kepada semua ahli pasukan untuk mendapatkan jumlah jualan yang tinggi, guru telah membenarkan pemberian komisen. Kadar komisen 5% akan diberikan ke atas jumlah jualan yang melebihi RM500 dan kadar komisen 8% akan diberikan ke atas jumlah jualan yang melebihi RM800.

Anda perlu membangunkan satu atur cara yang membenarkan anda memasukkan jualan bagi tiga jenis barangan tersebut. Kemudian atur cara perlu mendapatkan jumlah jualan yang diperolehi dan paparkan kadar komisen yang didapati oleh pasukan anda. Akhirnya, atur cara akan mengira dan memaparkan jumlah komisen dan komisen individu untuk setiap ahli pasukan.

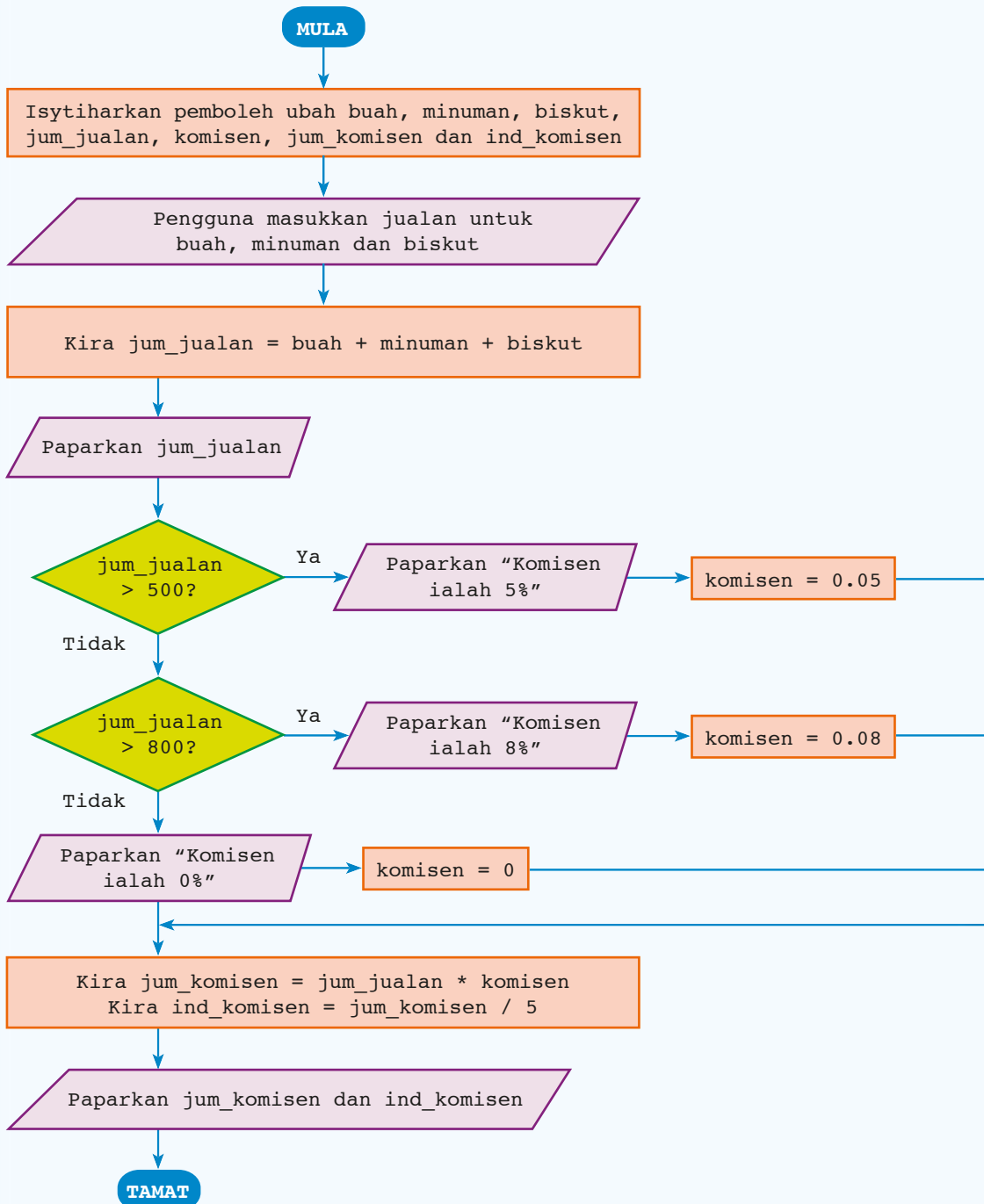
Pernyataan masalah:

1. Mengenal pasti input yang diperlukan, iaitu jualan untuk buah, minuman dan biskut.
2. Mengenal pasti proses yang terlibat, iaitu proses mengenal pasti kadar komisen yang diperolehi.
3. Mengenal pasti output yang diperlukan, iaitu jumlah jualan, kadar komisen yang dibenarkan dan komisen untuk setiap ahli.

Menulis algoritma bagi penyelesaian masalah:

1. Isytiharkan pemboleh ubah buah, minuman, biskut, jum_jualan, komisen, jum_komisen dan ind_komisen.
2. Pengguna memasukkan jualan untuk buah, minuman dan biskut.
3. Kira jumlah jualan.
4. Semak komisen yang dibenarkan.
Jika jumlah jualan $\leq$ RM500, paparkan mesej "Komisen ialah 0%".
Jika jumlah jualan $>$ RM500 tetapi $\leq$ RM800, paparkan mesej "Komisen ialah 5%".
Jika jumlah jualan $>$ RM800, paparkan mesej "Komisen ialah 8%".
5. Kira jumlah komisen dan paparkan.
6. Kira komisen individu dan paparkan.

Carta alir yang dilukis:



Apabila anda membuat semakan ke atas carta alir yang dihasilkan dengan menggunakan satu set sampel data, didapati hasil output tidak seperti yang diinginkan. Contoh output:

Output yang diinginkan

Jumlah jualan: RM 900
Komisen ialah 8%
Jumlah komisen: RM 72
Komisen untuk setiap ahli: RM 12

Output yang didapati

Jumlah jualan: RM 900
Komisen ialah 5%
Jumlah komisen: RM 45
Komisen untuk setiap ahli: RM 9

Anda perlu menggunakan kertas dan pensel untuk melakukan teknik semakan meja. Berikut menunjukkan nilai-nilai yang dimasukkan:

buah = 300
minuman = 300
biskut = 300

$\text{jum_jualan} = 300 + 300 + 300 = 900$

Apabila $\text{jum_jualan} = 900$, nilai ini digunakan untuk semakan syarat $\text{jum_jualan} > 500$,
 $900 > 500$ memberi nilai benar, maka "Komisen ialah 5%" dipaparkan dan komisen = 0.05.

$\text{jum_komisen} = 900 * 0.05 = 45$

$\text{ind_komisen} = 45 / 5 = 9$

Output yang diperoleh:

Jumlah jualan: 900
Komisen ialah 5 %
Jumlah komisen: 45
Komisen untuk setiap ahli: 9

Output ini berlainan dengan yang dikehendaki. Ralat logik ini berlaku kerana syarat bagi pilihan komisen tidak tepat. Syarat pilihan ($\text{jum_jualan} > 800$) harus disemak dahulu. Selepas memperbaiki ralat ini, semak semula dengan teknik yang sama.

Apabila $\text{jum_jualan} = 900$, nilai ini digunakan untuk semakan syarat $\text{jum_jualan} > 800$,
 $900 > 800$ memberi nilai benar, maka "Komisen ialah 8%" dipaparkan dan komisen = 0.08.

$\text{jum_komisen} = 900 * 0.08 = 72$

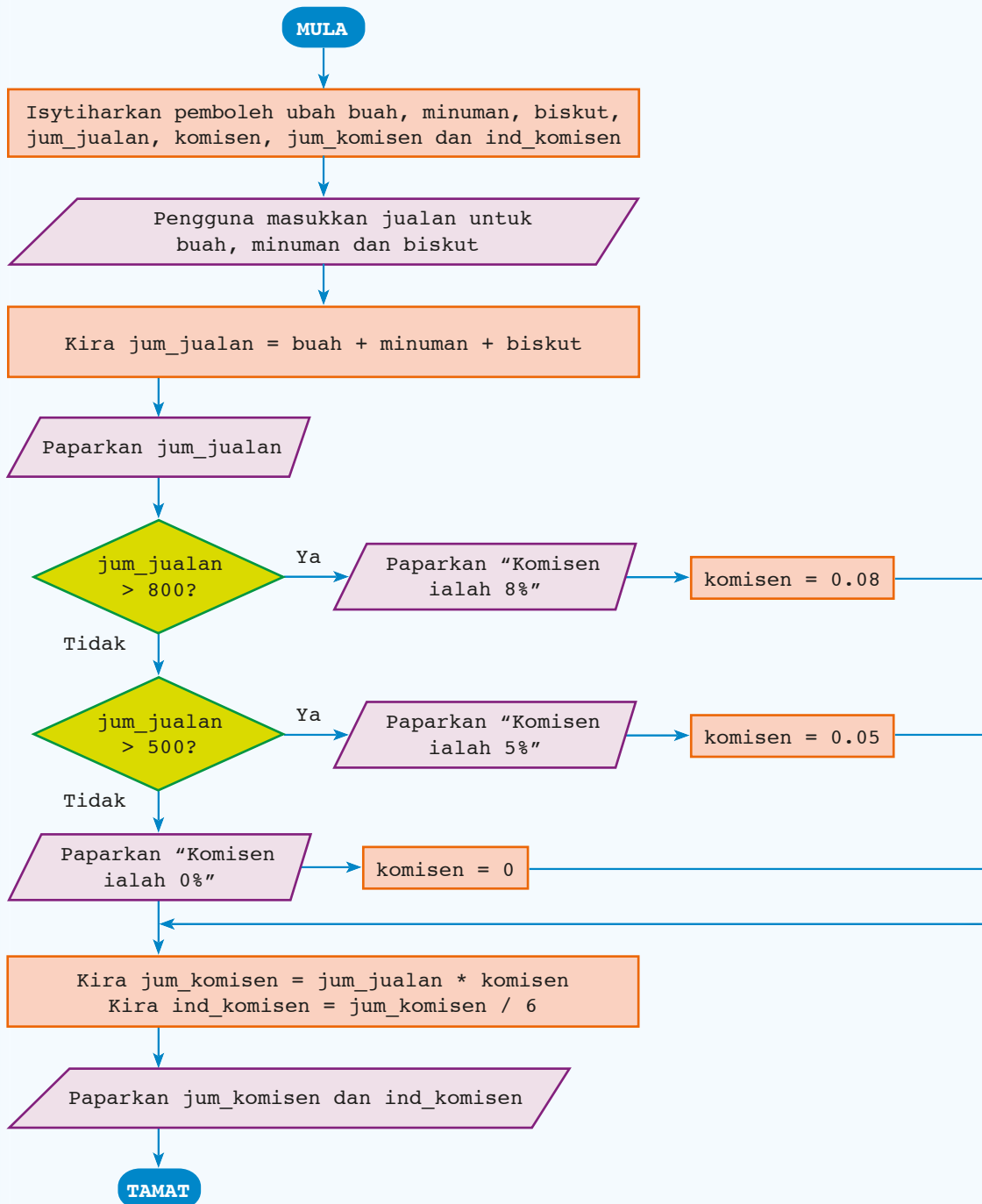
$\text{ind_komisen} = 72 / 5 = 14.4$

Output untuk ind_komisen adalah berlainan dengan yang diinginkan. Rumus berikut perlu disemak semula.

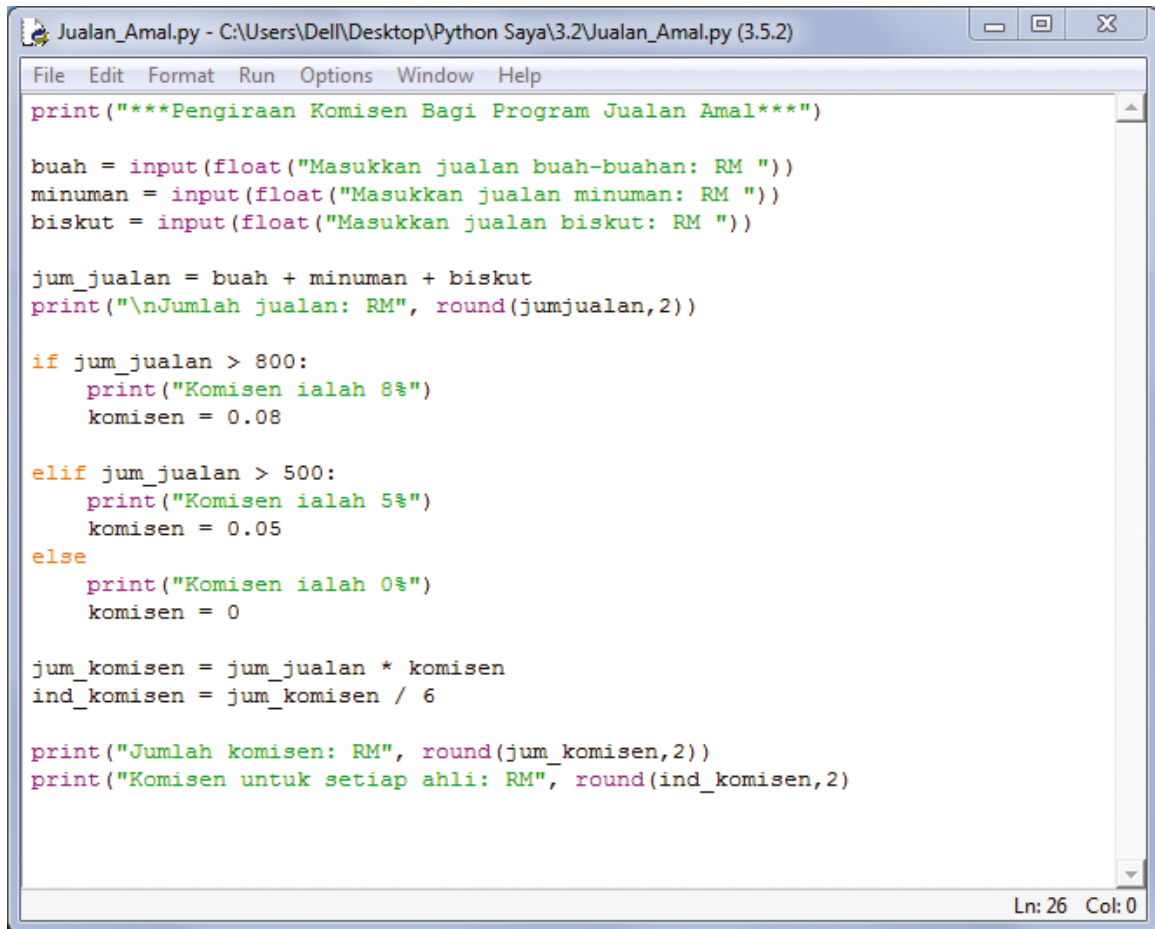
$$\text{ind_komisen} = \text{jum_komisen} / 5$$

Jumlah ahli pasukan termasuk ketua adalah enam orang dan bukan lima orang. Oleh itu, pembahagian komisen menjadi salah. Rumus perlu dibaiki dengan membahagi jum_komisen dengan enam. Carta alir di halaman 154 telah dibaiki ralat-ralat logik yang ditemui.

Carta alir yang telah dibaiki:



Kod atur cara yang dibangunkan:



```
print("***Pengiraan Komisen Bagi Program Jualan Amal***")

buah = input(float("Masukkan jualan buah-buahan: RM "))
minuman = input(float("Masukkan jualan minuman: RM "))
biskut = input(float("Masukkan jualan biskut: RM "))

jum_jualan = buah + minuman + biskut
print("\nJumlah jualan: RM", round(jum_jualan,2))

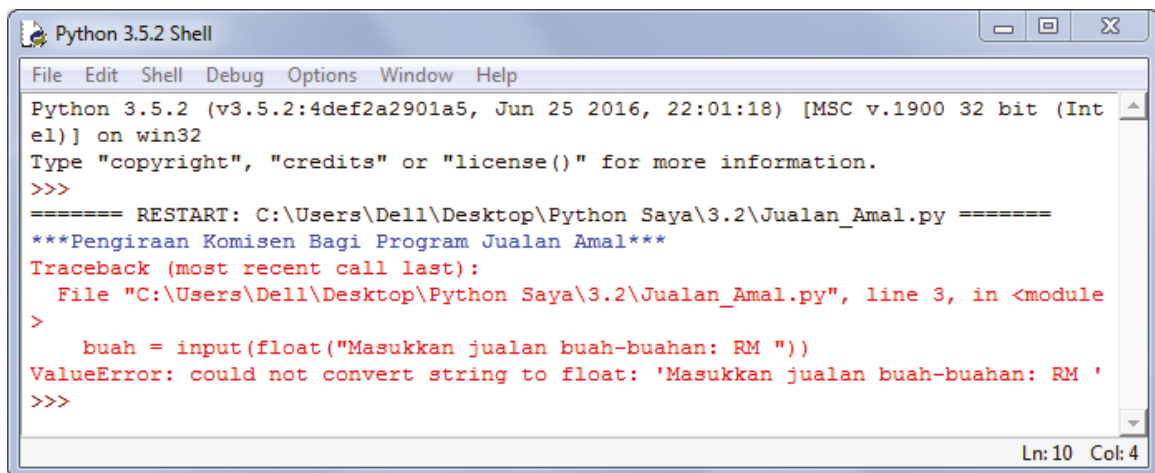
if jum_jualan > 800:
    print("Komisen ialah 8%")
    komisen = 0.08
elif jum_jualan > 500:
    print("Komisen ialah 5%")
    komisen = 0.05
else:
    print("Komisen ialah 0%")
    komisen = 0

jum_komisen = jum_jualan * komisen
ind_komisen = jum_komisen / 6

print("Jumlah komisen: RM", round(jum_komisen,2))
print("Komisen untuk setiap ahli: RM", round(ind_komisen,2))
```

Ln: 26 Col: 0

Paparan mesej ralat masa larian untuk pelaksanaan kali pertama:



```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Int
el)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Dell\Desktop\Python Saya\3.2\Jualan_Amal.py =====
***Pengiraan Komisen Bagi Program Jualan Amal***
Traceback (most recent call last):
  File "C:\Users\Dell\Desktop\Python Saya\3.2\Jualan_Amal.py", line 3, in <module>
    >
    buah = input(float("Masukkan jualan buah-buahan: RM "))
ValueError: could not convert string to float: 'Masukkan jualan buah-buahan: RM '
>>>
```

Ln: 10 Col: 4

Pengesanan ralat dapat dilakukan dengan memahami mesej ralat yang dipaparkan seperti yang berikut:

```
Traceback (most recent call last):
File "C:/Users/Dell\Desktop\Python Saya\3.2\Jualan_Amal", line
3, in <module>

    buah = input(float("Masukkan jualan buah-buahan: RM "))
ValueError: could not convert string to float: 'Masukkan jualan
buah-buahan: RM'
```

Mesej ralat tersebut memberitahu bahawa teks tidak boleh ditukarkan kepada jenis data *float*. Bagi memperbaiki ralat masa larian ini, penulisan kod untuk fungsi input bersama jenis data *float* perlu dibaiki. Semua barisan kod yang melibatkan ralat ini perlu dibaiki. Berikut menunjukkan penulisan kod yang betul:

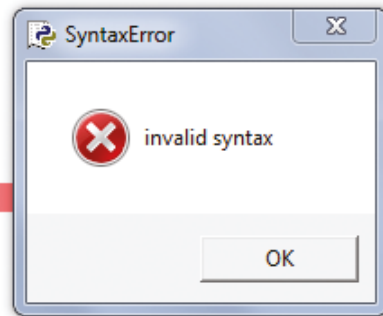
```
buah = float(input("Masukkan jualan buah-buahan: RM "))
```

Paparan mesej ralat sintaks untuk pelaksanaan kali kedua:

```
if jum_jualan > 800:
    print("Komisen ialah 8%")
    komisen = 0.08

elif jum_jualan > 500:
    print("Komisen ialah 5%")
    komisen = 0.05
else
    print("Komisen ialah 0%")
    komisen = 0

jum_komisen = jum_jualan * komisen
ind_komisen = jum_komisen / 6
```



Paparan mesej ralat di atas ialah mesej ralat sintaks. Mesej ralat sintaks akan dipaparkan bersama tanda merah pada baris kod yang bermasalah. Penulisan bagi sintaks **else** adalah tidak betul kerana tanpa tanda titik bertindih. Ralat ini dibaiki dengan menambahkan tanda **:** pada akhir sintaks **else**.

Paparan mesej ralat sintaks untuk pelaksanaan kali ketiga:

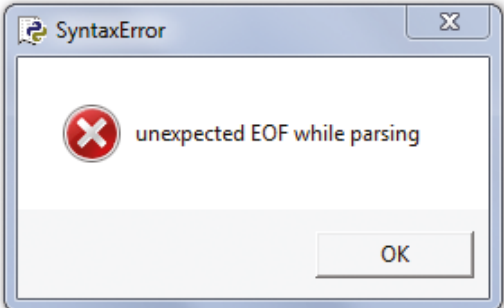
```
if jum_jualan > 800:
    print("Komisen ialah 8%")
    komisen = 0.08

elif jum_jualan > 500:
    print("Komisen ialah 5%")
    komisen = 0.05

else:
    print("Komisen ialah 0%")
    komisen = 0

jum_komisen = jum_jualan * komisen
ind_komisen = jum_komisen / 6

print("Jumlah komisen: RM", round(jum_komisen,2))
print("Komisen untuk setiap ahli: RM", round(ind_komisen,2))
```



Paparan mesej ralat sintaks dipaparkan dan kod untuk baris terakhir ditandakan merah. Lakukan semakan langkah demi langkah untuk baris terakhir ini. Didapati barisan kod ini mempunyai tanda kurungan yang tidak betul. Baiki kod ini dengan menambahkan kurungan tutup pada akhir kod.

Paparan mesej ralat sintaks untuk pelaksanaan kali keempat:

```
===== RESTART: C:\Users\Dell\Desktop\Python Saya\3.2\Jualan_Amal.py =====
***Pengiraan Komisen Bagi Program Jualan Amal***
Masukkan jualan buah-buahan: RM 300
Masukkan jualan minuman: RM 300
Masukkan jualan biskut: RM 300
Traceback (most recent call last):
  File "C:\Users\Dell\Desktop\Python Saya\3.2\Jualan_Amal.py", line 8, in <modul
e>
    print("\nJumlah jualan: RM", round(jumjualan,2))
NameError: name 'jumjualan' is not defined
>>>
```

Paparan mesej ralat di atas ialah mesej ralat masa larian. Atur cara terhenti secara tiba-tiba semasa di baris 8 disebabkan nama jumjualan tidak diisytiharkan. Ini bermakna, barisan kod yang berikut mempunyai kesalahan ejaan untuk pemboleh ubah jum_jualan.

```
print("\nJumlah jualan: RM", round(jumjualan,2))
```

Baiki ralat ini dengan menggunakan nama pemboleh ubah yang betul, iaitu jum_jualan.

Kod atur cara yang telah dibaiki semua ralat:

```
Jualan_Amal.py - C:\Users\Dell\Desktop\Python Saya\3.2\Jualan_Amal.py (3.5.2)
File Edit Format Run Options Window Help
print("***Pengiraan Komisen Bagi Program Jualan Amal***")

buah = float(input("Masukkan jualan buah-buahan: RM "))
minuman = float(input("Masukkan jualan minuman: RM "))
biskut = float(input("Masukkan jualan biskut: RM "))

jum_jualan = buah + minuman + biskut
print("\nJumlah jualan: RM", round(jum_jualan,2))

if jum_jualan > 800:
    print("Komisen ialah 8%")
    komisen = 0.08
elif jum_jualan > 500:
    print("Komisen ialah 5%")
    komisen = 0.05
else:
    print("Komisen ialah 0%")
    komisen = 0

jum_komisen = jum_jualan * komisen
ind_komisen = jum_komisen / 6

print("Jumlah komisen: RM", round(jum_komisen,2))
print("Komisen untuk setiap ahli: RM", round(ind_komisen,2))

Ln: 26 Col: 0
```

Output yang diperoleh:

```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit
(Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Dell\Desktop\Python Saya\3.2\Jualan_Amal.py =====
***Pengiraan Komisen Bagi Program Jualan Amal***
Masukkan jualan buah-buahan: RM 300
Masukkan jualan minuman: RM 300
Masukkan jualan biskut: RM 300

Jumlah jualan: RM 900.0
Komisen ialah 8%
Jumlah komisen: RM 72.0
Komisen untuk setiap ahli: RM 12.0
>>>

Ln: 14 Col: 4
```

Output yang diperoleh
adalah output yang diinginkan.

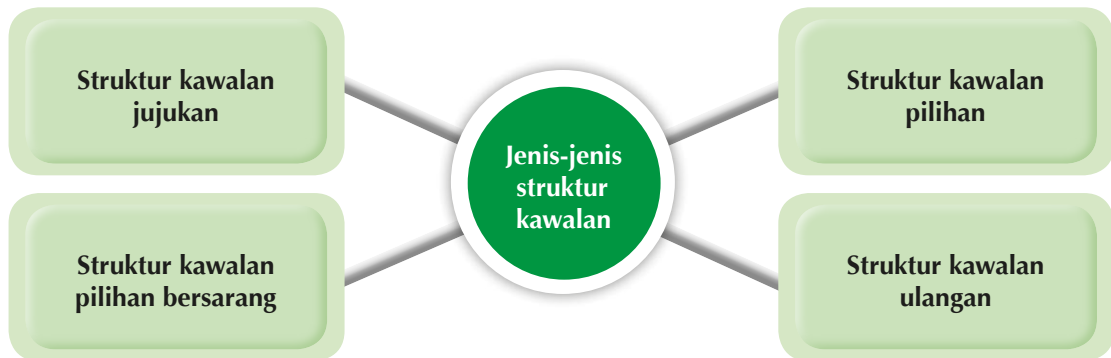
3.2.3 Atur cara yang melibatkan gabungan pelbagai struktur kawalan

Anda telah mempelajari tentang struktur kawalan yang melibatkan satu pilihan, dua pilihan dan penggunaan gelung yang membuat pengulangan dalam skop yang terhad. Namun begitu, dalam situasi dunia sebenar, sesebuah kod atur cara perlu lebih dinamik dan kompleks untuk menghasilkan atur cara yang mampu melakukan pelbagai fungsi seperti pengisihan, pemilihan, kawalan akses dan sebagainya.

Kod atur cara yang lebih dinamik dan kompleks dapat dihasilkan dengan menggabungkan pelbagai struktur kawalan yang anda telah pelajari seperti yang ditunjukkan dalam Rajah 3.16.

Standard Pembelajaran

Murid boleh:
3.2.3 Menghasilkan atur cara yang melibatkan gabungan pelbagai struktur kawalan.



Rajah 3.16 Atur cara yang mempunyai gabungan pelbagai struktur kawalan adalah lebih dinamik, kompleks dan interaktif

Tujuan utama penggabungan pelbagai struktur kawalan adalah untuk menghasilkan suatu atur cara yang boleh dan berupaya melakukan pelbagai fungsi, menarik dan seronok digunakan oleh pengguna. Suatu atur cara yang mempunyai fungsi yang terhad akan dianggap stereotaip dan besar kemungkinan pengguna tidak berminat untuk menggunakan atur cara sebegini.



Contoh 3.29

Menghasilkan atur cara yang melibatkan gabungan beberapa struktur kawalan.

Yoke Lian ingin menghasilkan satu permainan meneka nombor dengan menggunakan gabungan pelbagai struktur kawalan. Satu nombor rahsia yang perlu diteka akan dijadikan pemalar, di mana nilai ini tidak akan berubah. Pengguna perlu memasukkan nombor integer dan jika nombor yang dimasukkan tidak bersamaan dengan nombor rahsia, satu mesej akan dipaparkan. Mesej ini akan memberitahu pengguna sama ada nombor yang dimasukkan lebih besar atau lebih kecil daripada nombor rahsia. Setelah berjaya meneka, atur cara akan paparkan “Syabas” sebanyak lima kali.

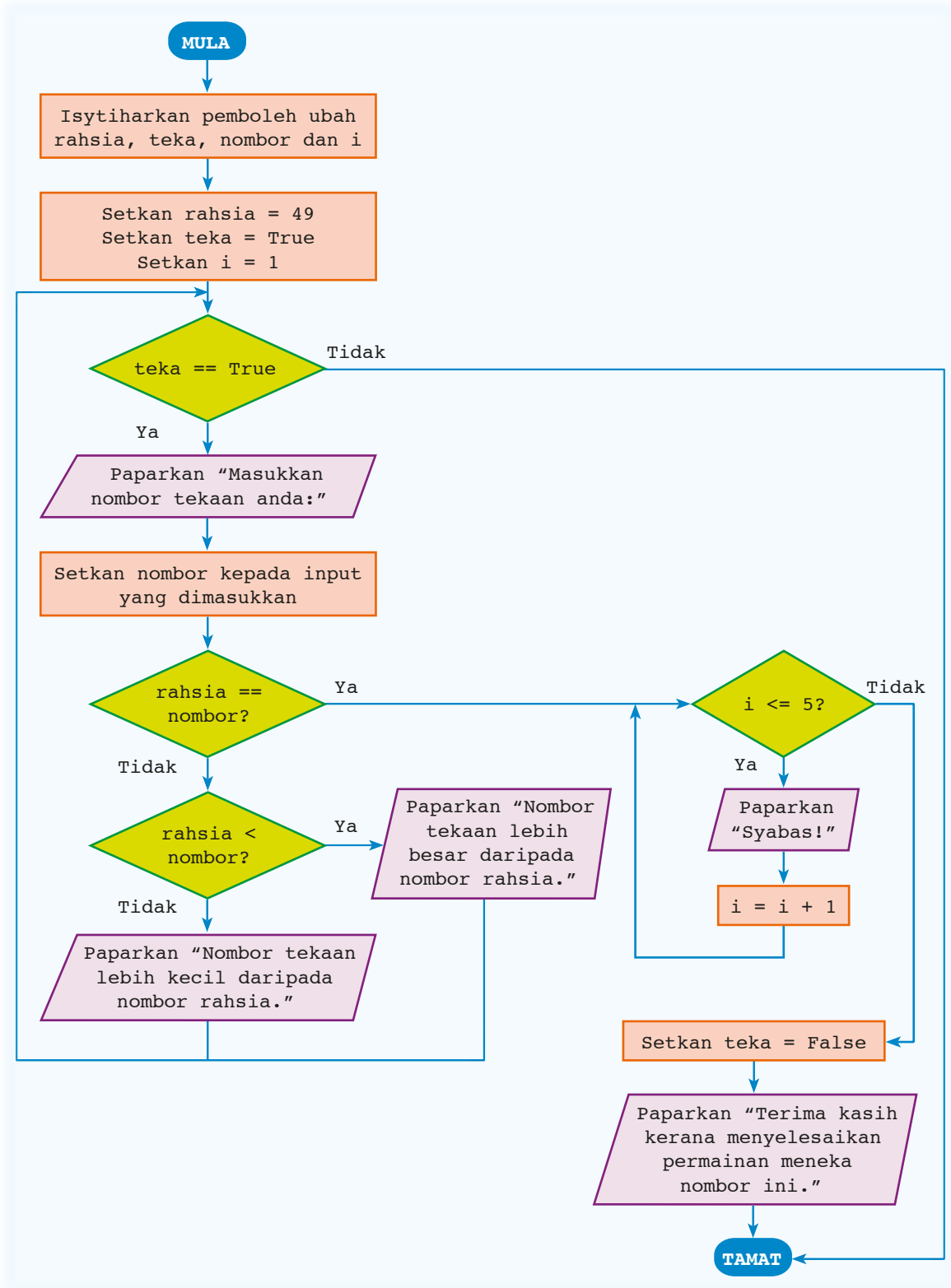
Langkah 1: Pernyataan masalah

1. Mengenal pasti input yang diperlukan, iaitu satu nombor integer daripada pengguna.
2. Mengenal pasti proses dan struktur kawalan yang terlibat.
 - (a) Proses meneka berterusan sehingga nombor yang betul dimasukkan. Struktur kawalan ulangan *while* digunakan.
 - (b) Proses membandingkan nombor yang dimasukkan dengan nombor rahsia. Struktur kawalan ulangan pilihan digunakan.
 - (c) Proses paparkan “Syabas” sebanyak lima kali. Struktur kawalan ulangan *for* digunakan.

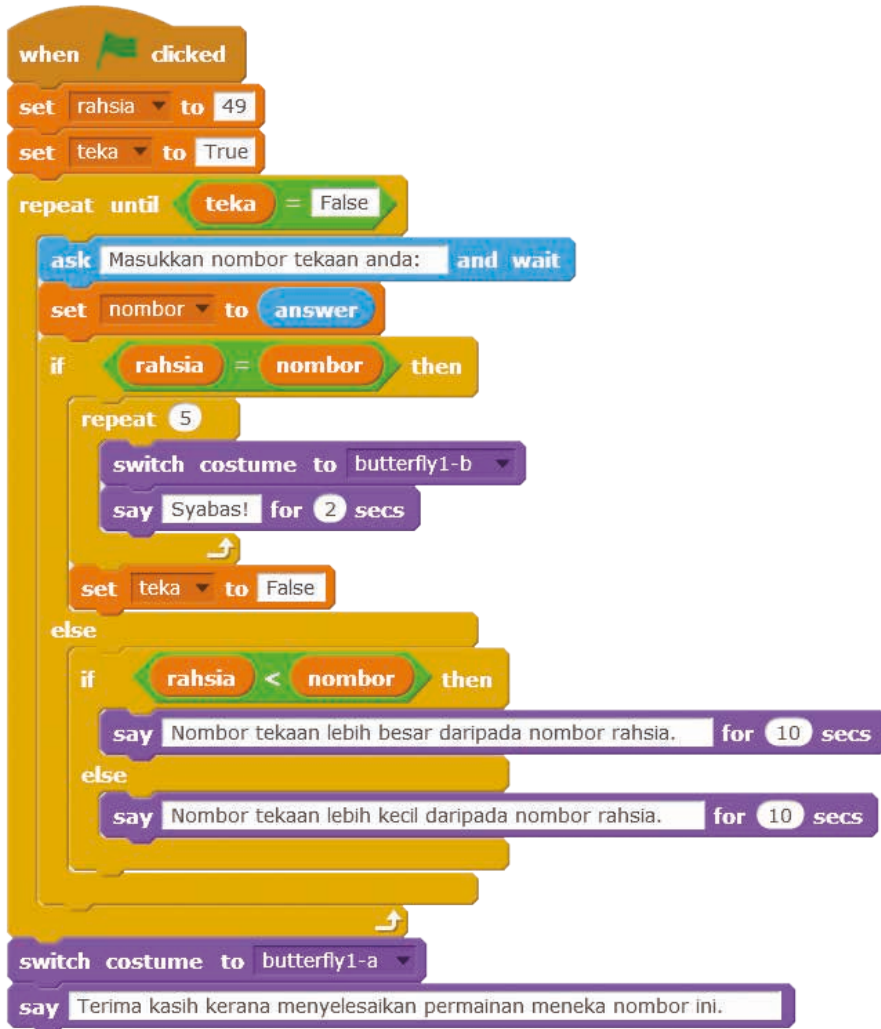
Langkah 2: Menulis algoritma bagi penyelesaian masalah

1. Isytiharkan pemboleh ubah rahsia, teka, nombor dan i.
2. Setkan rahsia kepada satu nombor integer, teka kepada True dan i kepada 1.
3. Selagi teka ialah True
Pengguna memasukkan satu nombor integer.
Setkan nombor kepada nombor integer yang dimasukkan.
4. Jika rahsia == nombor
5. Untuk i <= 5
Paparkan “Syabas”
i = i + 1
Setkan teka kepada False
Jika rahsia < nombor
Paparkan “Nombor tekaan lebih besar daripada nombor rahsia.”
Jika rahsia > nombor
Paparkan “Nombor tekaan lebih kecil daripada nombor rahsia.”
6. Paparkan “Terima kasih kerana menyelesaikan permainan meneka nombor ini.”

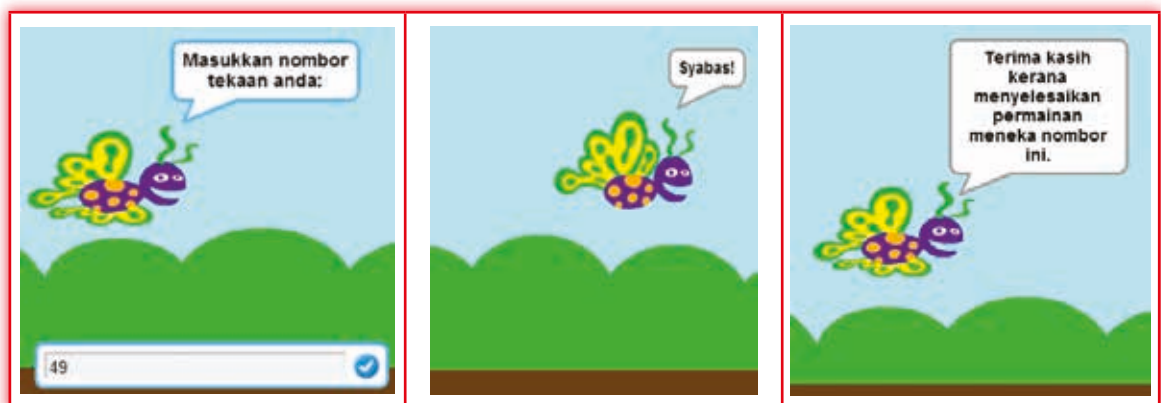
Langkah 3: Melukis carta alir



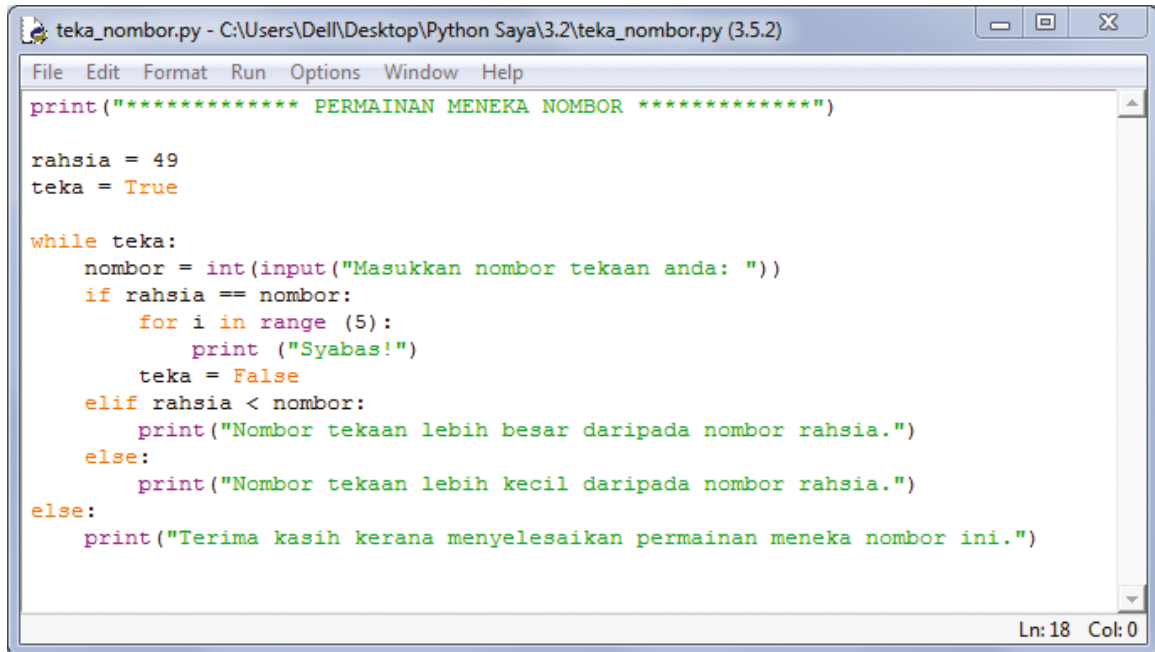
Langkah 4(a): Pengekoden dengan menggunakan Scratch 2.0



Contoh output:



Langkah 4(b): Pengekodan dengan menggunakan perisian aplikasi Python 3.5.2



```
teka_nombor.py - C:\Users\Dell\Desktop\Python Saya\3.2\teka_nombor.py (3.5.2)
File Edit Format Run Options Window Help

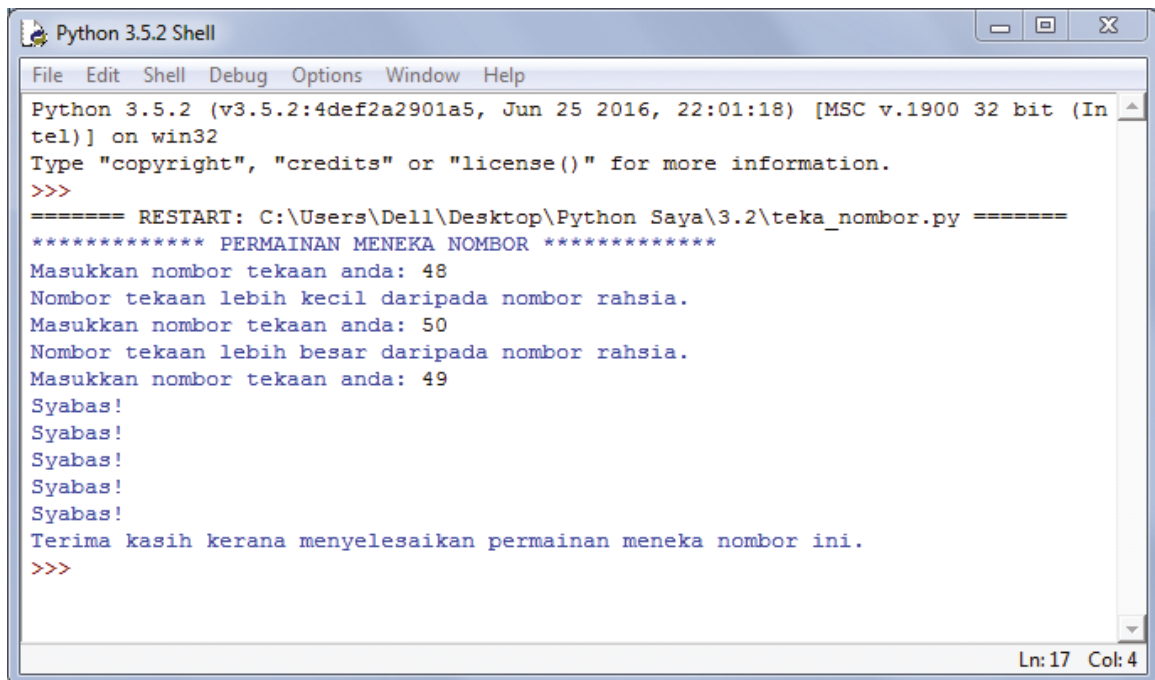
print("***** PERMAINAN MENEKA NOMBOR *****")

rahsia = 49
teka = True

while teka:
    nombor = int(input("Masukkan nombor tekaan anda: "))
    if rahsia == nombor:
        for i in range(5):
            print("Syabas!")
            teka = False
    elif rahsia < nombor:
        print("Nombor tekaan lebih besar daripada nombor rahsia.")
    else:
        print("Nombor tekaan lebih kecil daripada nombor rahsia.")
else:
    print("Terima kasih kerana menyelesaikan permainan meneka nombor ini.")

Ln:18 Col:0
```

Contoh output:



```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help

Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Dell\Desktop\Python Saya\3.2\teka_nombor.py =====
***** PERMAINAN MENEKA NOMBOR *****
Masukkan nombor tekaan anda: 48
Nombor tekaan lebih kecil daripada nombor rahsia.
Masukkan nombor tekaan anda: 50
Nombor tekaan lebih besar daripada nombor rahsia.
Masukkan nombor tekaan anda: 49
Syabas!
Syabas!
Syabas!
Syabas!
Syabas!
Terima kasih kerana menyelesaikan permainan meneka nombor ini.
>>>

Ln:17 Col:4
```

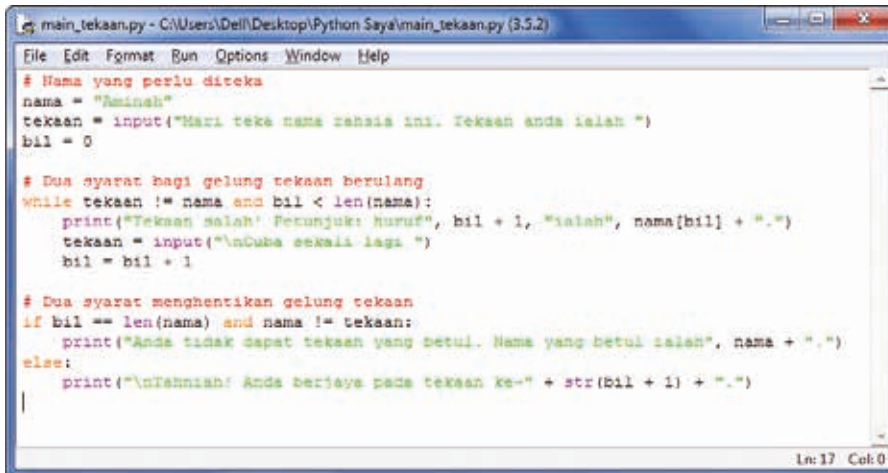
Amali Komputer 3.6 akan membimbing anda menggunakan struktur kawalan pilihan dwipilihan (*if-else*) dan struktur kawalan ulangan SELAGI (*while*) untuk membina satu permainan interaktif meneka nama.



Amali Komputer 3.6

Membina satu permainan interaktif meneka nama.

1. Gunakan perisian aplikasi **IDLE Python 3.5.2** dan lancarkan satu dokumen baharu. Taipkan kod atur cara seperti yang ditunjukkan di bawah.

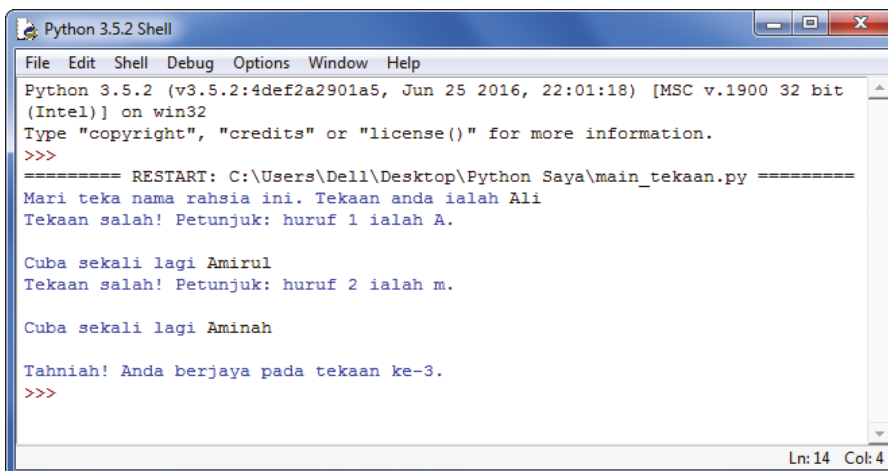


```
main_tekaan.py - C:\Users\Dell\Desktop\Python Saya\main_tekaan.py (3.5.2)
File Edit Format Run Options Window Help
# Nama yang perlu diteka
nama = "Aminah"
tekaan = input("Mari teka nama rahsia ini. Tekaan anda ialah ")
bil = 0

# Dua syarat bagi gelung tekaan berulang
while tekaan != nama and bil < len(nama):
    print("Tekaian salah! Petunjuk: huruf", bil + 1, "ialah", nama[bil] + ".")
    tekaan = input("\nCuba sekali lagi ")
    bil = bil + 1

# Dua syarat menghentikan gelung tekaan
if bil == len(nama) and nama != tekaan:
    print("Anda tidak dapat tekaan yang betul. Nama yang betul ialah", nama + ".")
else:
    print("\nTahniah! Anda berjaya pada tekaan ke-" + str(bil + 1) + ".")
|
```

2. Pilih menu **File > Save As** untuk menyimpan fail anda dalam folder **Python Saya**. Namakan fail anda sebagai **main_tekaan.py**.
3. Kemudian pada tettingkap fail **main_tekaan.py** yang masih aktif, klik menu **Run > Run Module** untuk melaksanakan segmen kod. Paparan seperti yang ditunjukkan di bawah akan diperhatikan. Jika terdapat paparan mesej ralat, baiki ralat tersebut.



```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit
(Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Dell\Desktop\Python Saya\main_tekaan.py =====
Mari teka nama rahsia ini. Tekaan anda ialah Ali
Tekaian salah! Petunjuk: huruf 1 ialah A.

Cuba sekali lagi Amirul
Tekaian salah! Petunjuk: huruf 2 ialah m.

Cuba sekali lagi Aminah

Tahniah! Anda berjaya pada tekaan ke-3.
>>>
```

4. Peluang untuk meneka nama ialah bilangan huruf dalam nama tersebut. Pada pendapat anda, adakah syarat ini bersesuaian? Terangkan.



Aktiviti 3.11

Kumpulan

Guru Asas Sains Komputer anda ingin membangunkan satu atur cara yang mengendalikan ujian kuiz bagi operator yang mempunyai lima soalan. Berikut ialah output yang diinginkan.

```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help

===== RESTART: C:/Users/Dell/Desktop/Python Saya/3.2/kuiz_operator.py =====
Jawab soalan-soalan berikut.
a = 12, b = 5, c = 2

1. a > b
2. a % b = 0
3. 12 // 5 = c
4. a > 6 > b
5. b < 2 < a
Jawapan soalan 1: True
Jawapan soalan 2: False
Jawapan soalan 3: True
Jawapan soalan 4: False
Jawapan soalan 5: True

Anda telah menjawab 3 soalan dengan betul dan 2 soalan dengan salah.
Sila buat ulang kaji.
>>>

===== RESTART: C:/Users/Dell/Desktop/Python Saya/3.2/kuiz_operator.py =====
Jawab soalan-soalan berikut.
a = 12, b = 5, c = 2

1. a > b
2. a % b = 0
3. 12 // 5 = c
4. a > 6 > b
5. b < 2 < a
Jawapan soalan 1: True
Jawapan soalan 2: False
Jawapan soalan 3: True
Jawapan soalan 4: True
Jawapan soalan 5: False

Tahniah, anda telah menjawab semua soalan dengan betul.
>>>
```

Anda dipilih sebagai ketua untuk membangunkan atur cara ini. Jawab soalan-soalan yang berikut.

- Nyatakan input, proses dan bilangan struktur kawalan yang terlibat.
- Lukis carta alir bagi atur cara ini.
- Pilih struktur kawalan yang sesuai untuk membangunkan atur cara ini dengan menggunakan perisian aplikasi Python 3.5.2.

TIP

Gunakan tag # untuk dokumentasi semasa membangunkan atur cara anda supaya ahli kumpulan yang lain dapat memahami atur cara dengan lebih mudah.



Praktis 3.2



1. Tuliskan satu atur cara ringkas yang dapat memeriksa huruf yang dimasukkan sama ada huruf vokal atau huruf konsonan. Atur cara perlu mengenal pasti lima huruf vokal, iaitu (A, E, I, O, U) atau (a, e, i, o, u). Gunakan struktur kawalan yang sesuai bagi atur cara tersebut. Atur cara akan memaparkan mesej contoh output yang ditunjukkan di bawah.

Contoh output:

```
>>>
Masukkan SATU huruf dari (A-Z) atau (a-z): K
K adalah huruf konsonan.
>>>
```



2. Dalam setahun terdapat 12 bulan. Setiap bulan mempunyai jumlah hari sama ada 30 hari, 31 hari atau 28/29 hari. Tuliskan satu atur cara yang dapat membaca nama bulan yang dimasukkan dan memaparkan bilangan hari bagi bulan tersebut. Gunakan struktur kawalan pilihan yang sesuai bagi atur cara tersebut.

Contoh output:

```
>>>
=====
Senarai bulan: Januari, Februari, Mac, April, Mei, Jun, Julai, Ogos, September,
Oktober, November, Disember
Masukkan nama bulan: February
Nama Bulan Yang Salah
>>>
=====
Senarai bulan: Januari, Februari, Mac, April, Mei, Jun, Julai, Ogos, September,
Oktober, November, Disember
Masukkan nama bulan: Februari
Bilangan Hari Dalam Bulan: 28/29 hari
>>>
```



3. Terdapat tiga jenis segi tiga berdasarkan ukuran sisi seperti yang ditunjukkan di bawah. Tuliskan satu atur cara berdasarkan input ukuran Sisi-x, Sisi-y dan Sisi-z yang dimasukkan. Atur cara akan memaparkan jenis segi tiga yang terbabit.



Segi tiga sama sisi



Segi tiga sama kaki



Segi tiga tidak sama sisi

Contoh output:

```
>>>
Masukkan ukuran segi tiga bagi setiap sisi dalam sentimeter.
sisi-x: 6
sisi-y: 6
sisi-z: 12
Segi tiga sama kaki.
>>>
```



4. Tuliskan satu atur cara yang dapat membina struktur nombor seperti contoh output yang ditunjukkan di bawah. Struktur tersebut perlu mencetak setiap nombor mengikut bilangan yang dikira. Gunakan struktur kawalan ulangan yang telah anda pelajari.

Contoh output:

```
>>>
1
22
333
4444
55555
666666
7777777
88888888
999999999
>>>
```



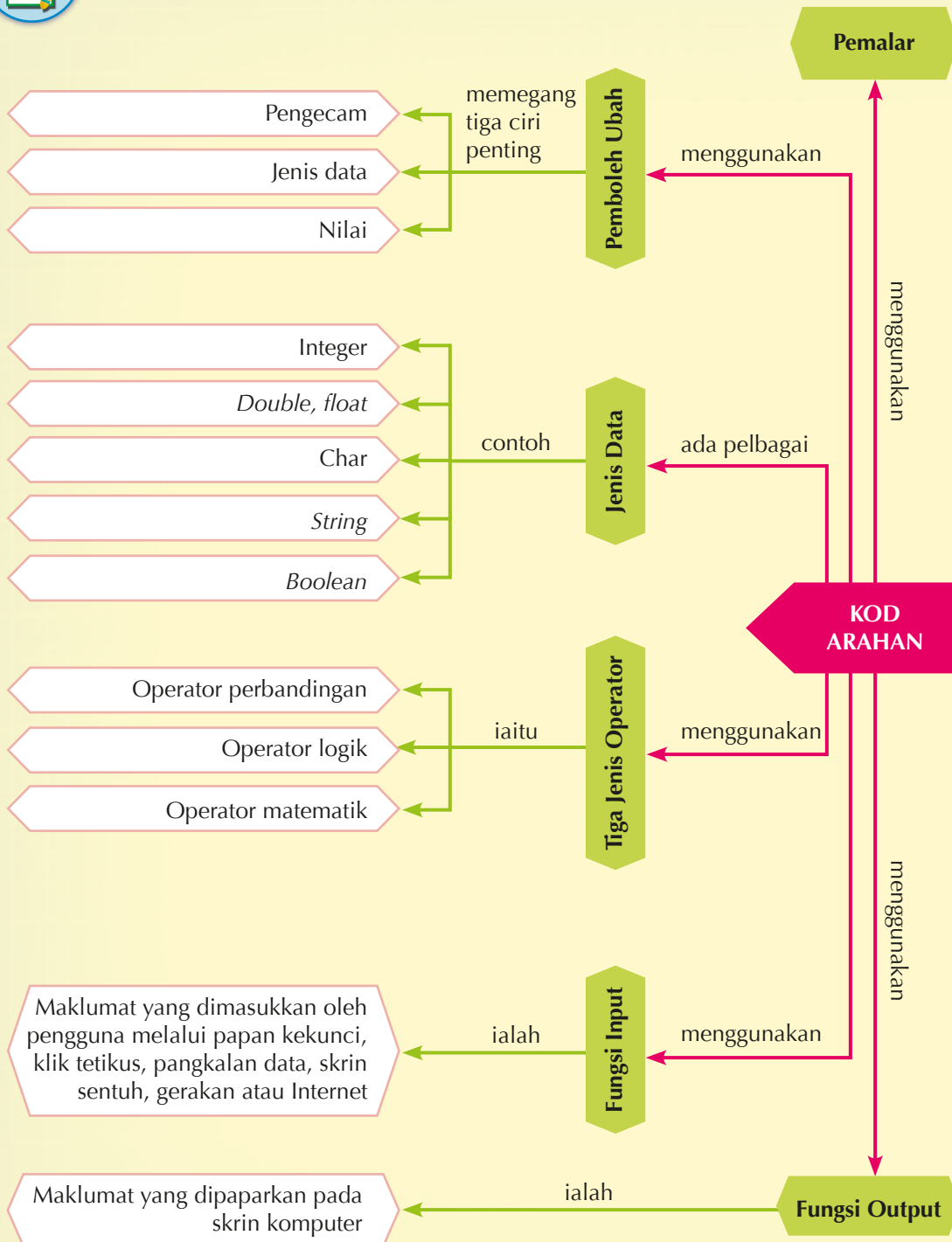
5. Tuliskan satu atur cara yang membolehkan pengguna memasukkan nombor-nombor integer sehingga nombor sifar dimasukkan. Apabila sifar dimasukkan, atur cara akan melakukan penambahan untuk semua nombor-nombor integer yang dimasukkan dan jumlahnya dipaparkan. Purata bagi nombor-nombor yang dimasukkan akan dikira dan dipaparkan. Gunakan struktur gelung ulangan dan struktur kawalan pilihan dalam atur cara ini. Berikut menunjukkan paparan contoh output.

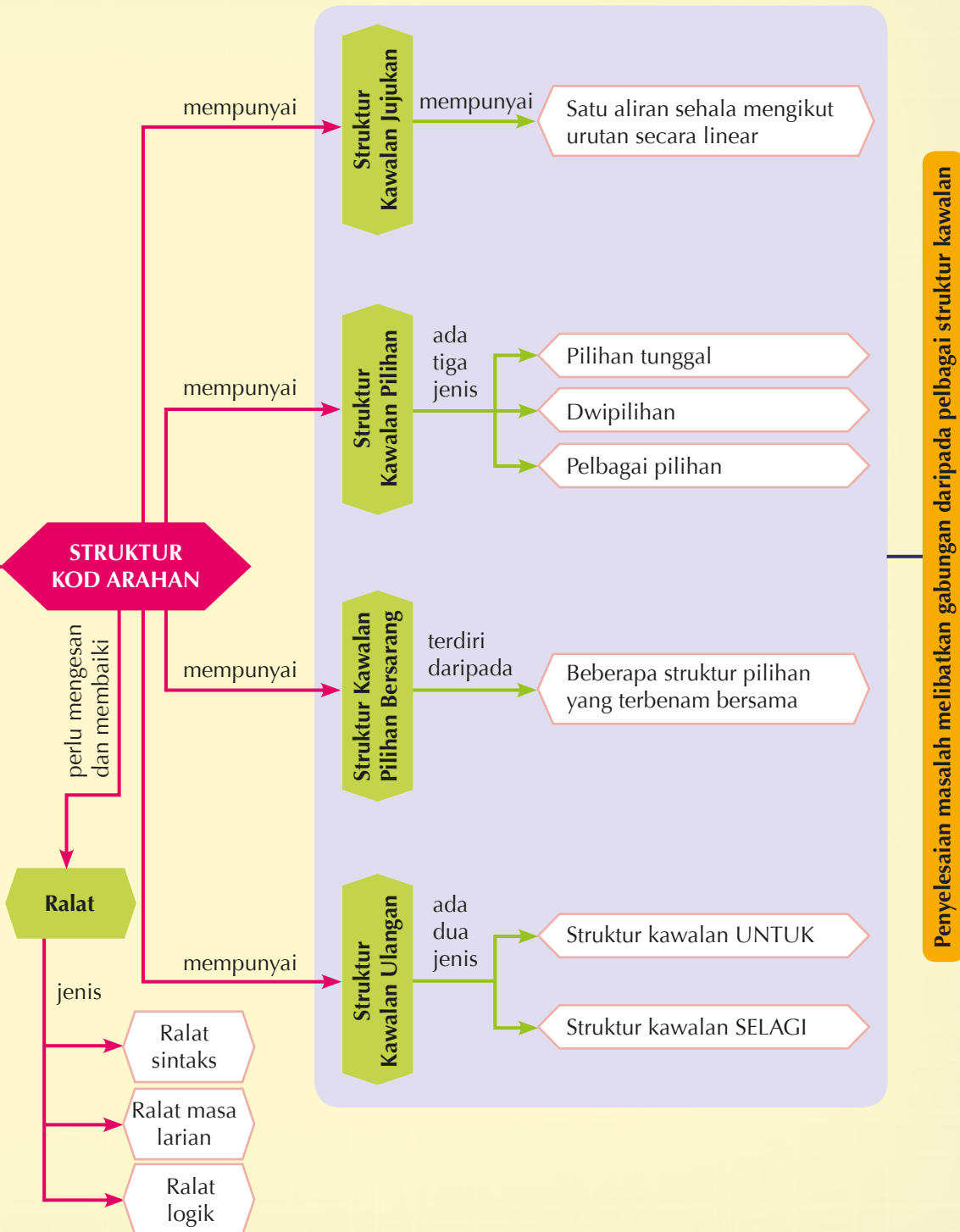
Contoh output:

```
>>>
Masukkan nombor-nombor integer yang perlu dijumlahkan.
Masukkan nombor SIFAR jika ingin berhenti memasukkan nombor.
23
66
78
0
Jumlah nombor: 167
Purata bagi nombor-nombor yang dimasukkan: 41.75
>>>
```




PETA KONSEP BAB 3







SENARAI SEMAK BAB 3

Pada akhir bab ini, saya dapat...

menghuraikan jenis-jenis data.

menerangkan jenis data *integer*.

menerangkan jenis data *double*.

menerangkan jenis data *char*.

menerangkan jenis data *string*.

menerangkan jenis data *boolean*.

menggunakan jenis data dalam segmen kod.

menghuraikan pemalar.

menerangkan pemboleh ubah.

menghubung kait pemboleh ubah dan jenis data.

membezakan pemboleh ubah dan pemalar.

menghuraikan fungsi input.

menggunakan fungsi input dalam segmen kod.

menghuraikan fungsi output.

menggunakan fungsi output dalam segmen kod.

menyatakan operator-operator yang digunakan dalam segmen kod.

menghuraikan operator perbandingan.

menghuraikan operator logik.

menggunakan operator perbandingan dalam segmen kod.

menggunakan operator logik dalam segmen kod.

mengesan ralat dalam segmen kod.

membaiki ralat dalam segmen kod.

menghasilkan segmen kod yang terdiri daripada gabungan jenis data, pemboleh ubah, pemalar dan operator.

Pada akhir bab ini, saya dapat...

menyelesaikan masalah secara sistematis dan kreatif dengan menghasilkan segmen kod yang terdiri daripada gabungan jenis data, pemboleh ubah, pemalar dan operator.	
mengenal pasti struktur kawalan jujukan.	
mengenal pasti tiga struktur kawalan pilihan, iaitu pilihan tunggal, dwipilihan dan pelbagai pilihan.	
mengenal pasti struktur kawalan pilihan bersarang.	
mengenal pasti struktur kawalan ulangan (<i>for</i> dan <i>while</i>).	
memberi contoh penggunaan struktur kawalan jujukan untuk menyelesaikan masalah.	
memberi contoh penggunaan struktur kawalan pilihan untuk menyelesaikan masalah.	
memberi contoh penggunaan struktur kawalan pilihan bersarang untuk menyelesaikan masalah.	
memberi contoh penggunaan struktur kawalan ulangan untuk menyelesaikan masalah.	
menghasilkan atur cara yang melibatkan struktur kawalan jujukan.	
menghasilkan atur cara yang melibatkan struktur kawalan pilihan.	
menghasilkan atur cara yang melibatkan struktur kawalan pilihan bersarang.	
menghasilkan atur cara yang melibatkan struktur kawalan ulangan.	
menguji atur cara untuk mengesan ralat.	
membaiki ralat yang dikesan dalam atur cara.	
menghasilkan atur cara yang melibatkan gabungan pelbagai struktur kawalan, iaitu jujukan, pilihan, pilihan bersarang dan ulangan.	
menyelesaikan masalah secara sistematis dan kreatif dengan mencipta atur cara yang melibatkan gabungan pelbagai struktur kawalan, iaitu jujukan, pilihan, pilihan bersarang dan ulangan.	



PENTAKSIRAN BAB 3

Soalan Objektif

- Apakah yang dikatakan atur cara komputer?
 - Satu carta alir yang dilukis bagi mengarahkan komputer menyelesaikan masalah.
 - Satu set arahan yang ditulis dalam bahasa mesin bagi mengarahkan komputer menyelesaikan masalah.
 - Satu set arahan yang ditulis dalam bahasa pengaturcaraan bagi mengarahkan komputer menyelesaikan masalah.
 - Satu pseudokod yang ditulis dalam bahasa tabii manusia bagi mengarahkan komputer menyelesaikan masalah.
- Antara berikut, yang manakah format bagi fail Python?
 - *.p
 - *.pt
 - *.py
 - *.pyt
- Jenis data yang manakah tidak disokong oleh Aplikasi Pengaturcaraan Python?
 - Integer
 - String
 - Double
 - Boolean
- Apakah output bagi kod berikut?


```
str = "Selamat Datang"
```

 - Selamat Datang
 - "Selamat Datang"
 - str
 - Mesej ralat
- Antara berikut, pengisytiharan integer kepada pemboleh ubah yang manakah benar?
 - abc = int(1,000,000)
 - a b c = int(100020003000)
 - a,b,c = int(1000,2000,3000)
 - a_b_c = int(1000000)
- Apakah kegunaan fungsi `output()` dalam pengaturcaraan Python?
 - Untuk memaparkan satu ayat pada skrin komputer.
 - Untuk memaparkan nilai pemboleh ubah pada skrin komputer.
 - Untuk memaparkan soalan dan menunggu pengguna memasukkan maklumat.
 - Untuk memaparkan mesej ralat.
 - I sahaja
 - III sahaja
 - I dan II
 - III dan IV
- Segmen kod Python berikut menunjukkan kegunaan operator perbandingan dan hasil yang diperoleh.

1	>>> a = 3
2	>>> b = 5
3	>>> a ___ b
4	False
5	>>> a ___ b
6	False
7	>>> a ___ b
8	False

Berdasarkan segmen kod di atas, apakah operator perbandingan di baris 3, 5 dan 7?

 - =, >, >=
 - =, <, <=
 - ==, >, >=
 - ==, <, <=

8. Berikut menunjukkan segmen kod Python.

```
markah1 = 80
markah2 = 90
purata = (markah1 + markah2)/2
print (purata)
```

Apakah output bagi nilai purata dalam segmen kod di atas?

- A 85 C 85.00
B 85.0 D 90.0

9. Operator logik yang manakah memberi output benar jika hanya salah satu keadaan adalah benar?

- A Logik DAN
B Logik ATAU
C Logik TIDAK
D Logik JIKA

10. Sintaks yang manakah digunakan untuk menambahkan pilihan dalam satu struktur kawalan?

- A *if* C *else*
B *elif* D *for*

11. Apakah kepentingan lelaran dalam suatu atur cara?

- A Membenarkan atur cara membuat pilihan berdasarkan syarat.
B Membenarkan atur cara melaksanakan arahan mengikut urutan.
C Membenarkan kod atur cara menjadi ringkas tanpa arahan yang berulang.
D Membenarkan atur cara boleh melalui pelbagai pilihan.

12. Struktur kawalan yang manakah paling lazim menggunakan pembilang dalam pernyataan bersyarat?

- A Struktur kawalan dwipilihan
B Struktur kawalan pilihan bersarang
C Struktur kawalan ulangan UNTUK
D Struktur kawalan ulangan SELAGI

13. Berikut menunjukkan segmen kod Python.

```
i = 0
while i < 5:
    print(i)
    i = i + 1
    if i == 3:
        break
    else:
        print(0)
```

Apakah output bagi segmen kod ini?

- A 0 C 0
1 0
1 1
0 0
2 2
B 1 D Gelung tak
2 terhingga
3

14. Segmen kod Python yang berikut menunjukkan mesej ralat.

```
>>> a=2
>>> b=3
>>> c=b
>>> d=b-c
>>> e=a/d
Traceback (most recent call last):
  File "<pyshell#19>", line 1,
    in <module>
      e=a/d
ZeroDivisionError: division by
zero
>>>
```

Apakah jenis ralat ini?

- A Ralat sintaks
B Ralat logik
C Ralat masa larian
D Ralat sifar

Soalan Berstruktur

1. (a) Apakah kegunaan operator dalam segmen kod?
(b) Apakah kepentingan pemboleh ubah dalam segmen kod?
(c) Namakan jenis-jenis data yang terkandung dalam segmen kod.
2. Apakah perbezaan antara pemboleh ubah dan pemalar?
3. Tulis segmen kod yang memberikan output seperti yang ditunjukkan di bawah. Jadikan <Nama_Anda> sebagai input pengguna.

```
Selamat Datang, <Nama_Anda>.
```

4. Terangkan struktur kawalan gelung UNTUK dan struktur kawalan gelung SELAGI. Apakah persamaan dan perbezaan antara dua struktur kawalan gelung ini?
5. Bina satu atur cara untuk mengira jumlah bilangan makanan di restoran yang perlu dibayar (termasuk 6% cukai perkhidmatan) dan memberitahu bayaran individu bagi setiap murid. Restoran tersebut cuma menghidangkan sejenis set makanan yang berharga RM36.60.

Contoh output:

```
Bilangan Set Makanan: 5  
Jumlah Perlu Dibayar: RM193.98  
Setiap murid perlu membayar RM38.8
```

6. (a) Teliti setiap baris segmen kod di bawah. Terangkan maksud kod bagi setiap baris.

1	a = 0 ; b = 10
2	while a < b:
3	print (a)
4	a = a + 1

(b) Tuliskan output bagi segmen kod di atas.

7. Tukar pseudokod berikut kepada segmen kod Python.

```
suhu = 115  
SELAGI suhu lebih tinggi atau sama dengan 80:  
    CETAK suhu  
    suhu - 1  
TAMAT SELAGI  
CETAK 'Sup Ini Telah Sedia Untuk Dihidangkan'
```



8. Teliti segmen kod Python di bawah. Kenal pasti empat ralat yang ada dalam segmen kod ini. Kemudian tulis semula segmen kod Python yang telah dibaiki.

```
nama = str(input("Masukkan nama anda: "))
umur = str(input("Masukkan umur anda: "))

if nama == "":
    nama = str(input("Sila masukkan nama anda: "))
else
    if umur >= 0:
        print("Umur anda mesti lebih daripada 0.")
        umur = int(input("Masukkan umur anda: "))
    else:
        print("Salam sejahtera", nama + ". Anda berumur",
              (umur), "tahun.")
```

9. Segmen kod berikut ialah soalan-soalan kuiz matematik yang digunakan semasa aktiviti Bulan Sains dan Matematik yang lepas dan dilabelkan sebagai Kuiz Matematik Versi 1.0.

```
File Edit Format Run Options Window Help
print("***** Kuiz Matematik Versi 1.0 *****")

print("Hitung hasil bagi 13 x 13.")
jawapan = int(input())
if jawapan == 169:
    print("Tahniah")
else:
    print("Maaf, jawapan ialah 169")

print("\nHitung hasil bagi 85 - (3 x 2).")
jawapan = int(input())
if jawapan == 79:
    print("Tahniah")
else:
    print("Maaf, jawapan ialah 79")

print("\nHitung hasil bagi (2 x 22 ÷ 11) + 56.")
jawapan = int(input())
if jawapan == 60:
    print("Tahniah")
else:
    print("Maaf, jawapan ialah 78")

Ln: 24 Col: 0
```

Bagi Kuiz Matematik Versi 1.1, anda ditugaskan untuk menambah baik atur cara supaya memenuhi syarat-syarat yang berikut:

- Nama murid yang melakukan kuiz diminta dan disimpan.
- Setiap soalan yang dijawab dengan betul diberikan 5 markah.
- Selepas menjawab semua soalan, nama dan jumlah markah yang terkumpul dipaparkan.
- Jika markah ialah 15, mesej "Tahniah, Penguasaan Memuaskan." akan dipaparkan.
- Jika markah kurang daripada 15, mesej "Usaha Lebih Gigih Lagi." akan dipaparkan.

LAMPIRAN A

Perwakilan kod ASCII dalam nombor perpuluhan, nombor perduaan, nombor perlapanan dan nombor perenambelasan.

Nombor perpuluhan	Nombor perduaan	Nombor perlapanan	Nombor perenambelasan	Aksara
0	00000000	000	0	NUL (NULL)
1	00000001	001	1	SOH (START OF HEADING)
2	00000010	002	2	STX (START OF TEXT)
3	00000011	003	3	ETX (END OF TEXT)
4	00000100	004	4	EOT (END OF TRANSMISSION)
5	00000101	005	5	ENQ (ENQUIRY)
6	00000110	006	6	ACK (ACKNOWLEDGE)
7	00000111	007	7	BEL (BELL)
8	00001000	010	8	BS (BACKSPACE)
9	00001001	011	9	TAB (HORIZONTAL TAB)
10	00001010	012	A	LF (LINE FEED)
11	00001011	013	B	VT (VERTICAL TAB)
12	00001100	014	C	FF (FORM FEED)
13	00001101	015	D	CR (CARRIAGE RETURN)
14	00001110	016	E	SO (SHIFT OUT)
15	00001111	017	F	SI (SHIFT IN)
16	00010000	020	10	DLE (DATA LINK ESCAPE)
17	00010001	021	11	DC1 (DEVICE CONTROL 1)
18	00010010	022	12	DC2 (DEVICE CONTROL 2)
19	00010011	023	13	DC3 (DEVICE CONTROL 3)
20	00010100	024	14	DC4 (DEVICE CONTROL 4)
21	00010101	025	15	NAK (NEGATIVE ACKNOWLEDGE)
22	00010110	026	16	SYN (SYNCHRONOUS IDLE)
23	00010111	027	17	ETB (END OF TRANS. BLOCK)
24	00011000	030	18	CAN (CANCEL)
25	00011001	031	18	EM (END OF MEDIUM)
26	00011010	032	1A	SUB (SUBSTITUTE)
27	00011011	033	1B	ESC (ESCAPE)
28	00011100	034	1C	FS (FILE SEPARATOR)
29	00011101	035	1D	GS (GROUP SEPARATOR)
30	00011110	036	1E	RS (RECORD SEPARATOR)
31	00011111	037	1F	US (UNIT SEPARATOR)

Nombor perpuluhan	Nombor perduaan	Nombor perlapanan	Nombor perenambelasan	Aksara
32	00100000	040	20	(ruang)
33	00100001	041	21	!
34	00100010	042	22	"
35	00100011	043	23	#
36	00100100	044	24	\$
37	00100101	045	25	%
38	00100110	046	26	&
39	00100111	047	27	'
40	00101000	050	28	(
41	00101001	051	29	)
42	00101010	052	2A	*
43	00101011	053	2B	+
44	00101100	054	2C	,
45	00101101	055	2D	-
46	00101110	056	2E	.
47	00101111	057	2F	/
48	00110000	060	30	0
49	00110001	061	31	1
50	00110010	062	32	2
51	00110011	063	33	3
52	00110100	064	34	4
53	00110101	065	35	5
54	00110110	066	36	6
55	00110111	067	37	7
56	00111000	070	38	8
57	00111001	071	39	9
58	00111010	072	3A	:
59	00111011	073	3B	;
60	00111100	074	3C	<
61	00111101	075	3D	=
62	00111110	076	3E	>
63	00111111	077	3F	?

LAMPIRAN A

Perwakilan kod ASCII dalam nombor perpuluhan, nombor perduaan, nombor perlapanan dan nombor perenambelasan.

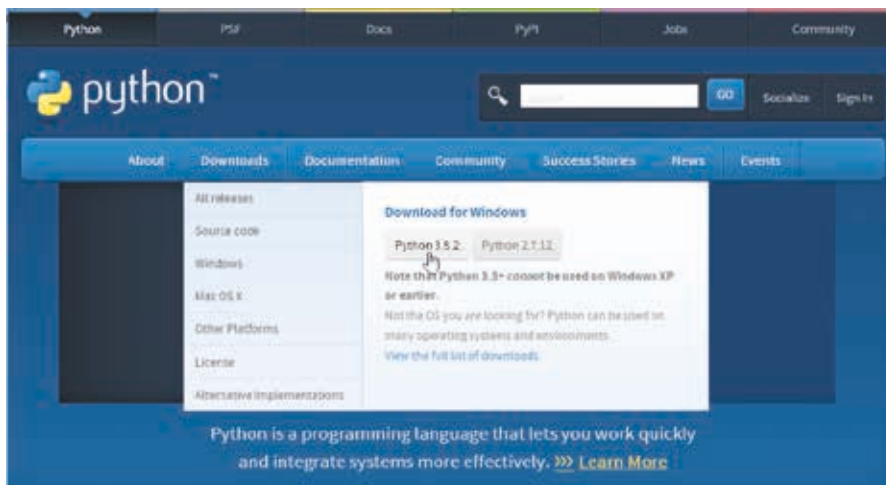
Nombor perpuluhan	Nombor perduaan	Nombor perlapanan	Nombor perenambelasan	Aksara
64	01000000	100	40	@
65	01000001	101	41	A
66	01000010	102	42	B
67	01000011	103	43	C
68	01000100	104	44	D
69	01000101	105	45	E
70	01000110	106	46	F
71	01000111	107	47	G
72	01001000	110	48	H
73	01001001	111	49	I
74	01001010	112	4A	J
75	01001011	113	4B	K
76	01001100	114	4C	L
77	01001101	115	4D	M
78	01001110	116	4E	N
79	01001111	117	4F	O
80	01010000	120	50	P
81	01010001	121	51	Q
82	01010010	122	52	R
83	01010011	123	53	S
84	01010100	124	54	T
85	01010101	125	55	U
86	01010110	126	56	V
87	01010111	127	57	W
88	01011000	130	58	X
89	01011001	131	59	Y
90	01011010	132	5A	Z
91	01011011	133	5B	[
92	01011100	134	5C	\
93	01011101	135	5D	]
94	01011110	136	5E	^
95	01011111	137	5F	_

Nombor perpuluhan	Nombor perduaan	Nombor perlapanan	Nombor perenambelasan	Aksara
96	01100000	140	60	`
97	01100001	141	61	a
98	01100010	142	62	b
99	01100011	143	63	c
100	01100100	144	64	d
101	01100101	145	65	e
102	01100110	146	66	f
103	01100111	147	67	g
104	01101000	150	68	h
105	01101001	151	69	i
106	01101010	152	6A	j
107	01101011	153	6B	k
108	01101100	154	6C	l
109	01101101	155	6D	m
110	01101110	156	6E	n
111	01101111	157	6F	o
112	01110000	160	70	p
113	01110001	161	71	q
114	01110010	162	72	r
115	01110011	163	73	s
116	01110100	164	74	t
117	01110101	165	75	u
118	01110110	166	76	v
119	01110111	167	77	w
120	01111000	170	78	x
121	01111001	171	79	y
122	01111010	172	7A	z
123	01111011	173	7B	{
124	01111100	174	7C	
125	01111101	175	7D	}
126	01111110	176	7E	~
127	01111111	177	7F	DEL

LAMPIRAN B

Memasang perisian aplikasi Python 3.5.2.

1. Layari laman sesawang <https://www.python.org>. Arahkan cursor pada **Downloads**, kemudian alihkan cursor kepada butang **Python 3.5.2** dan klik padanya. Rajah berikut menunjukkan paparan yang akan anda perhatikan.



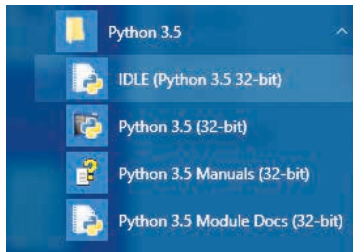
2. Setelah mengklik butang **Python 3.5.2**, paparan berikut akan diperhatikan. Klik **Run** untuk memulakan muat turun **perisian aplikasi Python 3.5.2**.



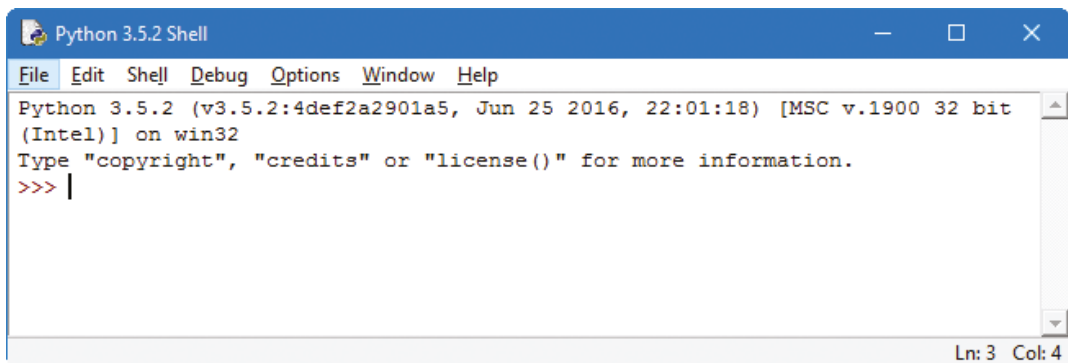
3. Setelah mengklik **Run**, paparan berikut akan diperhatikan. Pilih **Install Now**, kemudian ikut arahan yang diberikan.



- Setelah selesai memasang perisian aplikasi Python 3.5.2, klik pada butang **Start** dan cari ikon **Python 3.5**. Kemudian, pilih dan tekan pada **IDLE (Python 3.5 32-bit)**.



- Paparan tetingkap **IDLE Python 3.5.2** yang berikut akan dilancarkan. Simbol `>>>` dalam paparan menandakan permulaan setiap barisan kod arahan.



- Kod arahan Python boleh ditulis dengan dua kaedah. Kaedah pertama, gunakan paparan IDLE Python 3.5.2 di atas dan kod yang ditulis akan ditafsir oleh pentafsir Python dengan serta-merta selepas butang **Enter** ditekan.
- Kaedah kedua, suatu segmen kod arahan Python dapat ditulis dahulu tanpa ditafsir baris demi baris. Dalam paparan IDLE Python 3.5.2 yang ditunjukkan di atas, klik **File > New File**. Satu paparan tetingkap yang kosong akan dipaparkan. Di sini, segmen kod dapat ditaip tanpa ditafsir baris demi baris.
- Setelah siap menulis kod, simpan fail ini dalam format .py. Klik **Run > Run Module** untuk memulakan pentafsiran.



Pengekodan menggunakan Python bukan hanya boleh dilakukan daripada perisian aplikasi yang telah dipasang dalam komputer sahaja, malah boleh dilakukan secara dalam talian. Berikut ialah contoh laman sesawang untuk membangunkan atur cara Python secara dalam talian.

- https://www.tutorialspoint.com/ipython_terminal_online.php
- <https://www.tutorialspoint.com/python/>

JAWAPAN

BAB 1 PERWAKILAN DATA



Imbas QR code ini untuk mendapatkan jawapan bagi Aktiviti yang terpilih.

Praktis 1.1

- Sistem nombor perpuluhan, sistem nombor perduaan dan sistem nombor perlapanan.
- Senarai nombor: 0, 1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 14, 15, 16, 17, 20, 21, 22, 23, 24, 25, 26, 27, 30, 31, 32, 33, 34, 35, 36, 37, 40, 41, 42, 43, 44, 45, 46, 47
- (a) 123_8
(b) 1146_8
(c) 372_8
(d) 1037_8
- (a) 204_{10}
(b) 449_{10}
(c) 646_{10}
(d) 1062_{10}
- (a) ✓
(b)
(c) ✓
(d)
- (a) 651_8
(b) 406_8
(c) 3254_8
(d) 7310_8
- (a) 062_8
(b) 131_8
(c) 143_8
(d) 167_8
- E
- (a)

Aksara	Kod ASCII dalam nombor perduaan	Perwakilan nombor perduaan dalam nombor perlapanan
B	01000010	102
E	01000101	105
S	01010011	123
T	01010100	124
A	01000001	101
R	01010010	122
I	01001001	111

- (i) p@55word =
 $160_8 100_8 065_8 065_8 167_8 157_8 162_8 144_8$
(ii) komputer =
 $153_8 157_8 155_8 160_8 165_8 164_8 145_8 162_8$
(iii) Malaysia =
 $115_8 141_8 154_8 141_8 171_8 163_8 151_8 141_8$

10. Kelebihan Sistem Nombor Perlapanan:

- Nombor lebih ringkas iaitu lebih kurang satu per tiga nombor perduaan.
- Penukaran kepada nombor perduaan atau sebaliknya adalah mudah.

Kelemahan Sistem Nombor Perlapanan:

- Komputer tidak memahami nombor perlapanan.
- Perlu ditukarkan kepada nombor perduaan dahulu sebelum difahami oleh komputer.

Praktis 1.2

- (a) $19F_{16}$
(b) 232_{16}
(c) $14C4_{16}$
(d) $A9A_{16}$
- (a) 342_{10}
(b) 1191_{10}
(c) 4660_{10}
(d) 61711_{10}
- (a) 101001110010_2
(b) 1111111000_2
(c) 10101100010101_2
(d) 100000011100110_2
- (a) $D4_{16}$
(b) 83_{16}
(c) $6AC_{16}$
(d) $C38_{16}$

5. P

6.	Kod ASCII	0110	0010
	Nombor Perlapanan	$(1 \times 4) + (1 \times 2) = 6$	$(1 \times 2) = 2$

Jadi, nombor perenambelasan = 62

Aksara = b

- (a) Hello! = 48 65 6C 6C 6F 21
(b) BESTARI = 42 45 53 54 41 52 49
(c) Bunga Raya = 42 75 6E 67 61 20 52 61 79 61
(d) Hari Gawai = 48 61 72 69 20 47 61 77 61 69

8. Tahniah!

PENTAKSIRAN BAB 1

Soalan Objektif

- | | | | |
|------|-------|-------|-------|
| 1. D | 2. C | 3. A | 4. C |
| 5. B | 6. A | 7. C | 8. D |
| 9. D | 10. A | 11. B | 12. C |

Soalan Berstruktur

- Nombor perlapangan menggunakan sistem asas 8. Nombor ini menggunakan digit-digit 0, 1, 2, 3, 4, 5, 6 dan 7. Nombor perenambelasan menggunakan sistem asas 16. Nombor ini menggunakan digit-digit 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E dan F.
- Nombor perlapangan
 - 270₈
 - 352₈
 - 10644₈
 - 7363₈Nombor perenambelasan
 - B8₁₆
 - EA₁₆
 - 11A4₁₆
 - EF3₁₆
 - Kaedah bahagi dengan nombor dan gunakan bakinya
- 1010001₂
 - 100111110₂
 - 1010100101₂
 - 101011100011₂
 - 101000100011₂
 - 110111101001₂
 - 1101110100111₂
 - 1010010101101011₂
 - Persamaan:**
Penukaran dilakukan dengan memecahkan digit-digit dalam nombor perlapangan atau perenambelasan dan tukarkan setiap digit kepada nombor perdua. Akhirnya, cantumkan semua nombor perdua.
Perbezaan:
Penukaran nombor perlapangan kepada nombor perdua dilakukan dengan menukar setiap digit dalam nombor perlapangan kepada nilai tiga digit nombor perdua yang mewakilinya. Manakala penukaran nombor perenambelasan kepada nombor perdua dilakukan dengan menukar setiap digit dalam nombor perenambelasan kepada nilai empat digit nombor perdua yang mewakilinya.
- Nombor perlapangan
 - 7572₈
 - 4135₈Nombor perenambelasan
 - FTA₁₆
 - 85D₁₆
 - 141451₈
 - 45436₈
 - C329₁₆
 - 4B1E₁₆

- Kaedah pengumpulan tiga digit dan empat digit nombor perdua dilakukan dahulu semasa penukaran kepada nombor perlapangan dan nombor perenambelasan masing-masing.
- Pengekodan ASCII merupakan salah satu sistem pengekodan yang paling luas digunakan dalam dunia pengkomputeran. ASCII ialah singkatan daripada *American Standard Code for Information Interchange*. Kod ASCII digunakan untuk mewakili teks dalam komputer dan alatan komunikasi lain.
 - Komputer menggunakan nombor perdua iaitu 0 dan 1 yang biasa disebut bit untuk perwakilan data. Kod ASCII ialah salah satu skema pengekodan yang menggunakan gabungan lapan nombor 1 dan 0 (8 bit) untuk mewakili 256 aksara yang berbeza. Kod ASCII mampu memberi rujukan yang spesifik kepada susunan bit digital dan juga *glyph* untuk sebuah bahasa penulisan. Ini membolehkan peralatan digital berkomunikasi antara satu sama lain untuk memproses, menyimpan dan berkongsi maklumat berdasarkan simbol. Bahasa mesin atau kod mesin merupakan satu set arahan serta data yang boleh dilaksanakan secara langsung oleh pemproses komputer. Bahasa mesin diwakili oleh nombor perdua.
 - Sistem nombor perlapangan dan sistem nombor perenambelasan digunakan untuk mewakili jujukan nombor perdua yang panjang.
 - Sistem nombor perlapangan dan sistem nombor perenambelasan digunakan untuk memudahkan perwakilan data pada pandangan mata kasar manusia, iaitu tidak melibatkan banyak 0 dan 1. Dalam manual komputer selalunya arahan mesin dinyatakan dalam perwakilan perlapangan ataupun perwakilan perenambelasan.
 - Anda telah berjaya!
 - Saya suka subjek ASK.

BAB 2 ALGORITMA



Imbas QR code ini untuk mendapatkan jawapan bagi Aktiviti yang terpilih.

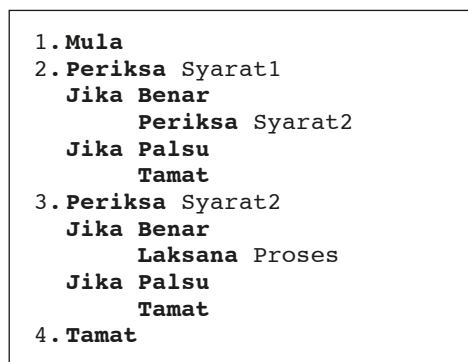
Praktis 2.1

- Algoritma ialah satu siri langkah untuk menyelesaikan sesuatu masalah atau tugas dalam program komputer. Algoritma merupakan tatacara atau prosedur langkah demi langkah yang tersusun untuk menghasilkan output yang diperlukan.

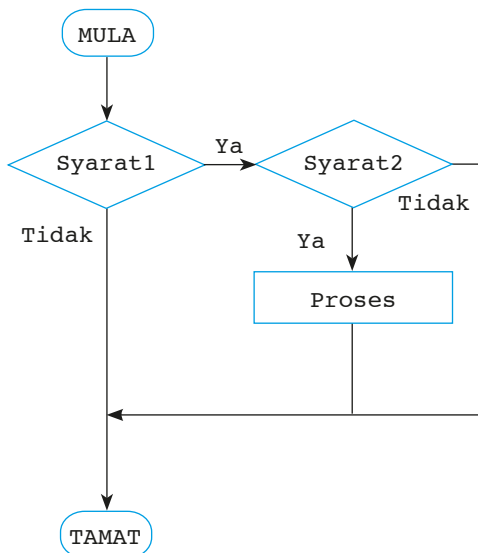
2. Algoritma terdiri daripada satu siri langkah teratur yang membantu proses meleraikan satu masalah yang besar kepada masalah-masalah yang kecil. Setiap masalah kecil dapat diselesaikan dengan lebih mudah.

3. (a) Struktur kawalan pilihan bersarang

Pseudokod:

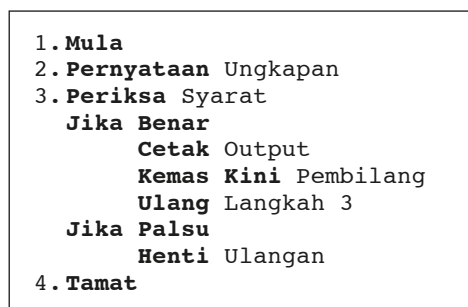


Carta alir:

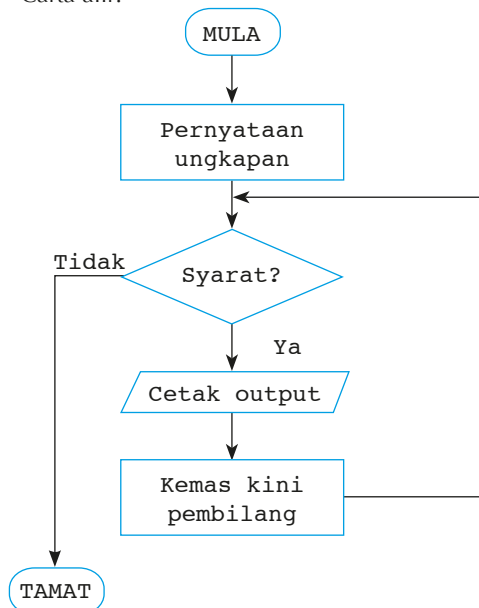


- (b) Struktur kawalan ulangan for

Pseudokod:

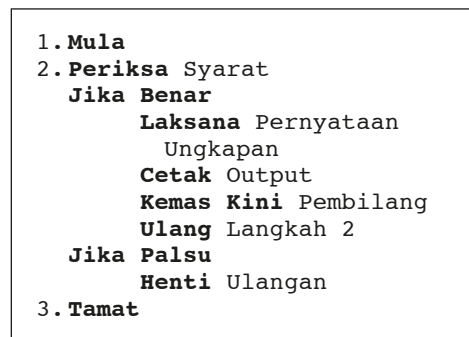


Carta alir:

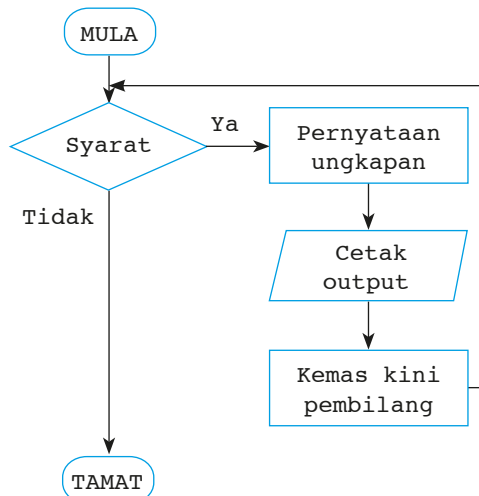


- (c) Struktur kawalan ulangan while

Pseudokod:



Carta alir:



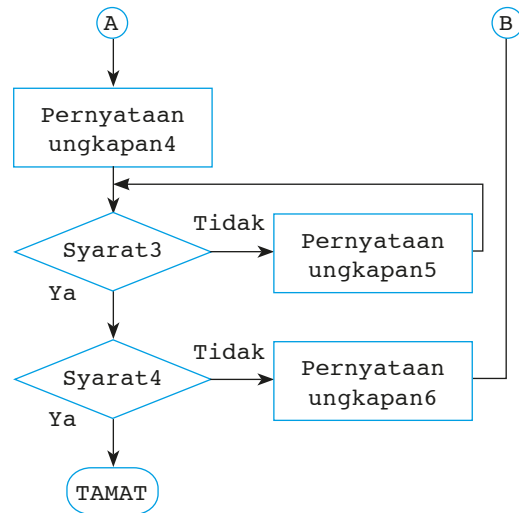
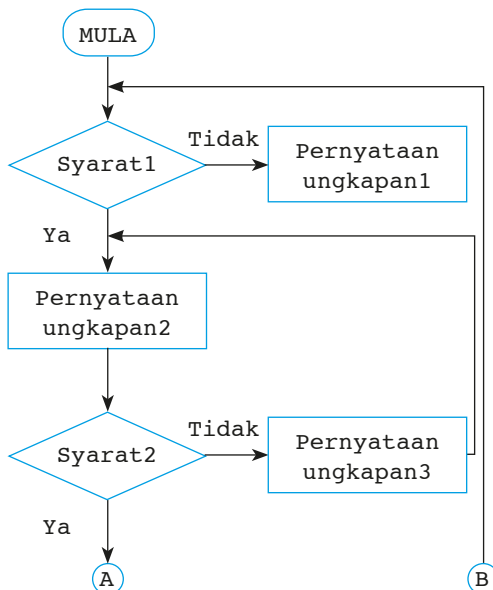
- (d) Gabungan struktur kawalan
Pseudokod:

```

1. Mula
2. Periksa syarat1
   Jika palsu
     Laksana pernyataan ungkapan1
     Ulang langkah 2
   Jika benar
     Laksana pernyataan ungkapan2
3. Periksa Syarat2
   Jika palsu
     Laksana pernyataan ungkapan3
     Ulang laksana pernyataan ungkapan2
   Jika benar
     Laksana pernyataan ungkapan4
4. Periksa syarat3
   Jika palsu
     Laksana pernyataan ungkapan5
     Ulang periksa syarat3
   Jika benar
     Periksa syarat4
5. Periksa syarat4
   Jika palsu
     Laksana pernyataan ungkapan6
     Ulang dari mula
   Jika benar
     Henti ulangan
6. Tamat

```

Carta alir:



4. **Ralat sintaks** berlaku semasa dalam proses menulis kod, contohnya tertinggal tanda kata atau tanda kurung. Lazimnya, ralat sintaks akan dikesan oleh penterjemah atur cara dan mesej ralat akan dipaparkan.

Ralat logik biasanya disebabkan oleh pengatur cara yang memasukkan nilai yang salah ataupun urutan sesuatu tindakan adalah salah. Penterjemah tidak akan dapat mengesan ralat-ralat logik.

Ralat masa larian merujuk kepada kesilapan yang menyebabkan atur cara terhenti secara tiba-tiba tanpa diduga semasa jalanan ujian (*test run*).

PENTAKSIRAN BAB 2

Soalan Objektif

- | | | | |
|------|-------|-------|------|
| 1. B | 2. C | 3. D | 4. D |
| 5. C | 6. C | 7. C | 8. B |
| 9. B | 10. B | 11. B | |

Soalan Berstruktur

1. Sebelum menyelesaikan masalah, masalah harus dikenal pasti dahulu. Selepas masalah dikenal pasti, baharulah bermula langkah pertama membangunkan algoritma.

Langkah pertama ialah langkah penyiasatan untuk mengenal pasti proses-proses yang terlibat, input yang diperlukan, penglibatan struktur kawalan dan pemboleh ubah.

Langkah kedua ialah langkah mendraf algoritma dan melakukan semakan logik. Dalam langkah ini, bina algoritma beraras tinggi seakan-akan bahasa pengaturcaraan. Kemudian, lakukan semakan logik ke atas algoritma yang direka. Semakan logik boleh membantu mengesan ralat dan membuat penambahbaikan ke atas algoritma tersebut.

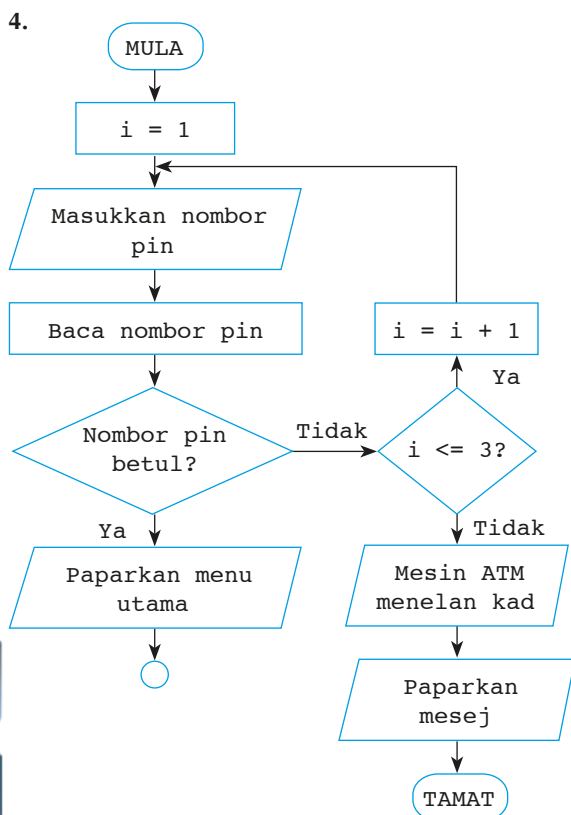
Langkah ketiga ialah memurnikan langkah-langkah algoritma dengan membuat penambahbaikan ke atas langkah-langkah algoritma. Dalam langkah ini juga semua proses dan pemboleh ubah yang terlibat dikumpulkan untuk dimurnikan. Kemudian uji larian ke atas algoritma yang telah dibina dengan meneliti langkah-langkahnya.

- Struktur kawalan pilihan bersarang adalah suatu struktur kawalan yang dibenamkan ke dalam suatu struktur kawalan yang lain bagi membentuk satu struktur kawalan.

Contohnya, apabila kad seorang pelanggan setia dimasukkan ke dalam mesin, mesin akan memeriksa kad tersebut sah dan tidak luput tarikhnya. Jika kad tidak sah, mesin akan memberhentikan tindakan selanjutnya. Jika kad sah, mesin akan melaksanakan tindakan-tindakan selanjutnya.

- Satu contoh aktiviti harian yang melibatkan langkah berulang ialah mengayuh basikal. Pengayuh basikal harus memastikan roda basikal sentiasa bergerak supaya basikalnya tidak terbalik atau terjatuh.

Contoh lain yang melibatkan langkah berulang ialah membasuh kain. Kain dibilas dalam baldi yang berisi air sebanyak tiga hingga empat kali sehingga kain menjadi bersih dan tidak bersabun.

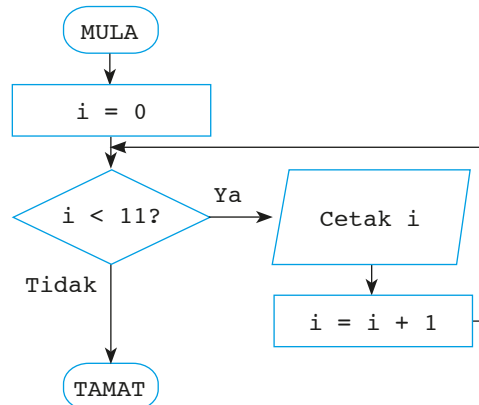


5.

```

Mula
Isytihar pembilang i = 1
for pembilang <= 3
    Cetak "Saya bernama Surendren!"
    pembilang = pembilang + 1
Tamat
  
```

6.



BAB 3 KOD ARAHAN



Imbas QR code ini untuk mendapatkan jawapan bagi Aktiviti yang terpilih.

Praktis 3.1

- str nama
str kelas
float peratus_marah
char gred

2.

Pemboleh ubah	Pemalar
Mempunyai nilai yang boleh berubah-ubah	Mempunyai nilai yang tetap dan tidak berubah
Perlu diisytiharkan	Fungsi perlu diaktifkan
Tidak perlu pengaktifan pustaka sokongan	Perlu pengaktifan pustaka sokongan

3.

```

# Jenis data yang digunakan ialah float
# Isytihar pemboleh ubah
jejari = 37.5
# Isytihar pemalar
pi = 3.142

isipadu = 4/3 * pi *
(jejari*jejari)
print ("Isi padu sfera magnet duit
ialah", round (isipadu,2))
  
```

```

4. a = 6
   b = 7
   c = 42
   print ("1", a == 6)
   print ("2", a == 7)
   print ("3", a == 6 and b == 7)
   print ("4", a == 7 and b == 7)
   print ("5", not a == 7 and b == 7)
   print ("6", a == 7 or b == 7)
   print ("7", a == 7 or b == 6)
   print ("8", not (a == 7 and b == 6))
   print ("9", not a == 7 and b == 6)

```

5.

```

>>> hujan = 12
>>> cerah = 16
>>> jumlah = 12 * 2 + 16 * 3
>>> print ("jumlah")
jumlah
>>>

```

Tambah kurungan untuk operasi darab seperti berikut:
jumlah = (12 * 2) + (16 * 3)

Tidak perlu letak tanda " " untuk cetak pemboleh ubah. Baikinya dengan:
print (jumlah)

Praktis 3.2

1.

```

huruf = input("Masukkan SATU huruf dari (A-Z) atau (a-z): ")
if huruf=="A" or huruf=="a":
    print(huruf , "adalah huruf vokal.")
elif huruf=="E" or huruf=="e":
    print(huruf , "adalah huruf vokal.")
elif huruf=="I" or huruf=="i":
    print(huruf , "adalah huruf vokal.")
elif huruf=="O" or huruf=="o":
    print(huruf , "adalah huruf vokal.")
elif huruf=="U" or huruf=="u":
    print(huruf , "adalah huruf vokal.")
else:
    print(huruf , "adalah huruf konsonan.")

```

2.

```

print("Senarai bulan: Januari, Februari, Mac, April, Mei, Jun, Julai, Ogos, September, Oktober, November, Disember")

nama_bulan = input("Masukkan nama bulan: ")

if nama_bulan == "Januari" or nama_bulan == "Mac" or nama_bulan == "Mei" or nama_bulan == "Julai" or nama_bulan == "Ogos" or nama_bulan == "Oktober" or nama_bulan == "Disember":
    print("Bilangan Hari Dalam Bulan: 31 hari")

```

```

elif nama_bulan == "Februari":
    print("Bilangan Hari Dalam Bulan: 28/29 hari")

```

```

elif nama_bulan == "April" or nama_bulan == "Jun" or nama_bulan == "September" or nama_bulan == "November":
    print("Bilangan Hari Dalam Bulan: 30 hari")

```

```

else:
    print("Nama Bulan Yang Salah")

```

3.

```

print("Masukkan ukuran segi tiga bagi setiap sisi dalam sentimeter.")
x = float(input("Sisi-x: "))
y = float(input("Sisi-y: "))
z = float(input("Sisi-z: "))
if x == y == z:
    print("Segi tiga sama sisi.")
elif x != y != z:
    print("Segi tiga tidak sama sisi.")
else:
    print("Segi tiga sama kaki.")

```

4.

```

for i in range(10):
    print (str (i) * i)

```

5.

```

print("Masukkan nombor-nombor integer yang perlu dijumlahkan.")
print("Masukkan nombor SIFAR jika ingin berhenti memasukkan nombor.")

```

```

bilang = 0
jumlah = 0
nombor = 1

```

```

while nombor != 0:
    nombor = int(input(""))
    jumlah = jumlah + nombor
    bilang = bilang + 1

```

```

if bilang == 0:
    print ("Masukkan beberapa nombor")
else:
    print ("Jumlah nombor:", jumlah)

```

```

print ("Purata bagi nombor-nombor yang dimasukkan:", jumlah/bilang)

```

PENTAKSIRAN BAB 3

Soalan Objektif

- | | | | | |
|-------|-------|-------|-------|-------|
| 1. C | 2. C | 3. C | 4. D | 5. D |
| 6. C | 7. B | 8. B | 9. B | 10. B |
| 11. C | 12. C | 13. C | 14. C | |

Soalan Berstruktur

- (a) Operator yang digunakan dalam kod segmen merupakan simbol untuk melakukan operasi matematik, menyelesaikan sesuatu hubungan yang berkaitan dengan logik dan seterusnya dapat menghasilkan suatu keputusan.
(b) Kepentingan pemboleh ubah dalam kod segmen ialah ia dapat menyimpan data yang akan digunakan dalam kod segmen dan juga sebagai kaedah menerima data daripada pengguna.
(c) Jenis-jenis data seperti integer, *boolean*, *double*, *float*, *char* dan *string*.

- Atur cara menggunakan pemboleh ubah untuk menyimpan maklumat dalam ingatan komputer yang boleh digunakan semula oleh atur cara. Maklumat dalam pemboleh ubah boleh berubah mengikut kod arahan.

Atur cara menggunakan pemalar untuk menyimpan nilai dalam ingatan komputer. Nilai yang disimpan oleh pemalar tidak boleh berubah.

- ```
nama = input('Nama Anda:')
print('Selamat Datang', nama)
```

- Persamaan dan perbezaan struktur kawalan gelung UNTUK dan SELAGI

### Persamaan

Kedua-dua gelung sesuai digunakan sebagai struktur kawalan praujian (*pretest*) yang akan menguji syarat sebelum melaksanakan lelaran.

### Perbezaan

Dikawal dengan bilangan lelaran yang diketahui.

Dikawal dengan keadaan dan bilangan lelaran tidak diketahui.

- ```
bil = int(input("Bilangan Set Makanan:"))
jumlah = round((bil * 36.60) * 1.06,2)
per_set = round(jumlah / bil,2)
print("Jumlah Perlu Dibayar: RM" + str(jumlah))
print("Setiap murid perlu membayar RM" + str(per_set))
```

- (a) Baris 1: Memberi nilai pemboleh ubah bagi a dan b
Baris 2: Syarat bagi struktur kawalan ulangan *while* ialah a mesti kurang daripada nilai b
Baris 3: Mencetak nilai a
Baris 4: Kemas kini nilai a dengan menambahkan 1 kepada nilai a

- 0
- 1
- 2
- 3
- 4
- 5
- 6

7
8
9

- ```
suhu = 115
while suhu >= 80:
 print(suhu)
 suhu = suhu - 1
print("'Sup Ini Telah Sedia Untuk Dihidangkan'")
```

- ```
nama = str(input("Masukkan nama anda: "))
umur = int(input("Masukkan umur anda: "))

if nama == "":
    nama = str(input("Sila masukkan nama anda: "))
else:
    if umur <= 0:
        print("Umur anda mesti lebih daripada 0.")
        umur = int(input("Masukkan umur anda: "))
    else:
        print("Salam sejahtera", nama + ". Anda berumur", str(umur), "tahun.")
```

Ralat 1

Ralat 2

Ralat 3

Ralat 4

- ```
print("** Kuiz Matematik Versi 1.1 **")
nama = input("Sila Masukkan Nama Anda: ")
skor = 0
print("Hitung hasil bagi 13 x 13.")
jawapan = int(input())
if jawapan == 169:
 print("Tahniah")
 skor = skor + 5
else:
 print("Maaf, jawapan adalah 169")
print("\nHitung hasil bagi 85 - (3 x 2).")
jawapan = int(input())
if jawapan == 79:
 print("Tahniah")
 skor = skor + 5
else:
 print("Maaf, jawapan adalah 79")
print("\nHitung hasil bagi (2 x 22 + 11) + 56.")
jawapan = int(input())
if jawapan == 60:
 print("Tahniah")
 skor = skor + 5
else:
 print("Maaf, jawapan adalah 78")
print("\nJumlah Markah Terkumpul:", skor)
if skor == 15:
 print(nama, ". Tahniah, Penguasaan Memuaskan.")
else:
 print(nama, ". Usaha Lebih Gigih Lagi.")
```

# GLOSARI

|                                          |                                                                                                                             |
|------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| <b>Aksara</b>                            | Terdiri daripada abjad, digit nombor atau simbol khas seperti titik perpuluhan atau tanda koma.                             |
| <b>Algoritma</b>                         | Satu set langkah atau tindakan untuk menyelesaikan masalah.                                                                 |
| <b>Atur cara</b>                         | Satu set kod arahan yang ditulis dalam bahasa pengaturcaraan bagi mengarah komputer untuk menyelesaikan masalah.            |
| <b>Bahasa mesin</b>                      | Bahasa asas yang difahami oleh komputer dan dinamakan juga sebagai kod mesin.                                               |
| <b>Boolean</b>                           | Jenis data yang mempunyai hanya dua nilai logik, iaitu "True" dan "False".                                                  |
| <b>Char</b>                              | Jenis data yang digunakan untuk mengisytiharkan aksara ( <i>character</i> ).                                                |
| <b>Double</b>                            | Jenis data yang digunakan untuk mengisytiharkan nombor yang bertitik perpuluhan.                                            |
| <b>Float</b>                             | Jenis data dalam pengaturcaraan Python yang digunakan untuk mengisytiharkan nombor yang bertitik perpuluhan.                |
| <b>Gelung (Loop)</b>                     | Bahagian kod yang perlu berulang dalam struktur kawalan ulangan.                                                            |
| <b>Gelung tak terhingga</b>              | Struktur kawalan ulangan yang mempunyai syarat sentiasa benar menyebabkan atur cara berulang tanpa henti.                   |
| <b>Input</b>                             | Data yang dimasukkan ke dalam komputer.                                                                                     |
| <b>Integer</b>                           | Jenis data yang digunakan untuk mengisytiharkan nombor bulat yang tiada titik perpuluhan yang positif atau negatif.         |
| <b>Jenis data</b>                        | Digunakan semasa pengisytiharan suatu pemboleh ubah untuk disimpan dan digunakan semula dengan lebih berkesan.              |
| <b>Kitar Hayat Pembangunan Atur Cara</b> | Terdiri daripada fasa-fasa yang perlu dilakukan mengikut urutan bagi memastikan atur cara yang dibangunkan berjaya.         |
| <b>Kod arahan</b>                        | Satu set langkah yang mengarahkan komputer melakukan sesuatu tugas.                                                         |
| <b>Kod ASCII perduaan</b>                | Sistem kod yang menggunakan nombor perduaan untuk mewakili aksara-aksara dalam komputer.                                    |
| <b>Kod ASCII perpuluhan</b>              | Sistem kod yang menggunakan nombor perpuluhan untuk mewakili aksara-aksara dalam komputer.                                  |
| <b>Lelaran (Iteration)</b>               | Setiap pusingan gelung dalam suatu struktur kawalan ulangan.                                                                |
| <b>Operator logik</b>                    | Operator yang digunakan dalam ungkapan algebra ringkas untuk mendapatkan nilai <i>Boolean</i> , iaitu "Benar" atau "Palsu". |
| <b>Operator perbandingan</b>             | Operator yang digunakan untuk membandingkan nilai di sebelah kiri operator dengan nilai di sebelah kanan operator.          |
| <b>Output</b>                            | Data yang dipaparkan pada skrin komputer.                                                                                   |
| <b>Pemalar</b>                           | Satu storan ingatan untuk menyimpan nilai secara tetap dan tidak akan berubah dengan atur cara yang dijalankan.             |
| <b>Pemboleh ubah</b>                     | Satu storan ingatan dalam komputer yang digunakan oleh atur cara untuk menyimpan nilai bagi kegunaan kemudian.              |
| <b>Penghimpun</b>                        | Sejenis penterjemah yang mengalihkan kod bahasa penghimpun kepada bahasa mesin untuk melaksanakan atur cara.                |

|                                              |                                                                                                                                                                       |
|----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Pengisytiharan</b>                        | Satu proses yang memberi nama dan jenis data kepada satu pemboleh ubah. Nilai kepada pemboleh ubah boleh ditetapkan oleh pengatur cara atau dimasukkan oleh pengguna. |
| <b>Pengkompil</b>                            | Sejenis penterjemah yang mengalihkan kod arahan kepada bahasa mesin kemudian melaksanakan atur cara.                                                                  |
| <b>Pentafsir</b>                             | Sejenis penterjemah yang terus melaksanakan atur cara dari kod arahan tanpa mengalihkan kepada bahasa mesin.                                                          |
| <b>Penterjemah</b>                           | Proses mengalihkan kod arahan kepada bahasa mesin yang boleh difahami oleh komputer.                                                                                  |
| <b>Perwakilan data</b>                       | Proses yang menggunakan sistem nombor bagi mewakili aksara-aksara dalam bentuk pernomboran untuk disimpan dalam komputer.                                             |
| <b>Pseudokod</b>                             | Perihalan tentang langkah-langkah dalam sesuatu algoritma dengan menggunakan ayat-ayat yang ringkas dan mudah difahami. Biasanya ditulis dalam bahasa tabii.          |
| <b>Ralat logik</b>                           | Ralat yang menghasilkan output yang tidak diinginkan.                                                                                                                 |
| <b>Ralat masa larian</b>                     | Ralat yang berlaku semasa pelaksanaan kod.                                                                                                                            |
| <b>Ralat sintaks</b>                         | Ralat yang berlaku disebabkan oleh kesalahan struktur atau simbol, tanda baca, kenyataan tidak sah dan pemindahan tidak sah.                                          |
| <b>RGB</b>                                   | Model warna yang digunakan untuk paparan alatan digital seperti komputer.                                                                                             |
| <b>Sintaks</b>                               | Peraturan yang menguasai struktur sesuatu bahasa pengaturcaraan.                                                                                                      |
| <b>Sistem nombor perenambelasan</b>          | Sistem nombor yang hanya menggunakan sepuluh digit dan enam abjad, iaitu 0–9 dan A–F.                                                                                 |
| <b>Sistem nombor perlapangan</b>             | Sistem nombor yang hanya menggunakan lapan pilihan digit, iaitu 0–7.                                                                                                  |
| <b>Skema pengekodan</b>                      | Sistem kod yang menggunakan nombor untuk mewakili aksara.                                                                                                             |
| <b>String</b>                                | Jenis data yang digunakan semasa pengisytiharan pemboleh ubah yang nilainya terdiri daripada rentetan aksara sama ada abjad atau nombor.                              |
| <b>Struktur kawalan jujukan</b>              | Struktur kawalan yang mempunyai satu aliran yang dilaksanakan satu per satu mengikut urutan, iaitu pelaksanaan kod secara linear.                                     |
| <b>Struktur kawalan pilihan</b>              | Struktur kawalan yang mempunyai pilihan sama ada untuk melangkau atau melaksanakan satu langkah atau satu set langkah.                                                |
| <b>Struktur kawalan ulangan</b>              | Struktur kawalan yang mengarahkan komputer melaksanakan langkah-langkah tertentu secara berulang kali.                                                                |
| <b>Struktur kawalan ulangan <i>for</i></b>   | Struktur kawalan ulangan <i>for</i> melakukan ulangan bagi sesuatu pernyataan untuk bilangan ulangan yang ditetapkan.                                                 |
| <b>Struktur kawalan ulangan <i>while</i></b> | Dalam struktur kawalan ulangan <i>while</i> , pernyataan berulang dilaksanakan selagi syarat yang diuji adalah benar.                                                 |
| <b>UNICODE</b>                               | Skema pengekodan 16 bit yang boleh menampung 65,536 aksara. Skema pengekodan ini sesuai digunakan mewakili simbol tulisan seperti Jawi, Jepun, Korea dan Cina.        |



# INDEKS

## A

Aksara 12–15, 20, 28–29, 31–33, 35–36, 77, 81–85, 89, 101  
Algoritma 40–47, 58, 62–71

## B

Bahasa mesin 2, 13, 18, 30, 76  
Bahasa pengaturcaraan 13, 40, 57, 76–82, 91, 95–97, 102  
Bait 12, 31  
Bit 9, 31, 79  
Boolean 85, 86, 89, 98, 99

## C

Carta alir 40–56, 63, 66–71  
Char 77, 81–82, 89

## D

Double 77, 79, 88, 112

## F

Float 79, 80, 115

## G

Gelung (*Loop*) 47, 143  
Gelung tak terhingga 148

## I

Input 14, 95, 96  
Integer 71–79, 88, 115

## J

Jenis data 75–83, 86, 115

## K

Kod arahan 76–82, 90, 120–124  
Kod ASCII nombor perdua 12, 13, 17, 176–179  
Kod ASCII nombor perpuluhan 12, 13, 176–179

## L

Lelaran (*Iteration*) 47, 143, 145

## M

Mesej ralat 18, 104, 106, 107, 109

## O

Operan 98–100  
Operator logik 99  
Operator matematik 78, 79, 88  
Operator perbandingan 98, 99  
Output 14, 95, 96, 97

## P

Pemalar 90–94, 115  
Pemboleh ubah 77–79, 90–92  
Pengisytiharan 77, 78, 80, 82, 90  
Pengkompil 76, 106  
Pentafsir 76, 90, 102, 106, 109  
Penterjemah 76, 95  
Pseudokod 41, 43–45, 47, 50, 53, 54, 56  
Python 13, 76, 180–181

## R

Ralat logik 60, 109, 150–158  
Ralat masa larian 60, 106, 150–158  
Ralat sintaks 60, 102, 150–158  
RGB 18, 30

## S

Sistem nombor perdua 2, 5, 14, 18  
Sistem nombor perenambelasan 18–20, 31, 32, 37  
Sistem nombor perlapanan 2–4, 7, 9, 14, 16, 17  
Sistem nombor perpuluhan 2–4, 19, 20  
Skema pengekodan 12, 28  
*String* 77, 81, 82–85, 88, 112  
Struktur kawalan dwipilihan 127, 133  
Struktur kawalan jujukan 120, 123, 162, 171  
Struktur kawalan pelbagai pilihan 41, 69, 70, 129  
Struktur kawalan pilihan bersarang 41, 43, 44  
Struktur kawalan ulangan  
    *for* 52, 53, 143, 145  
    *while* 47, 51, 56, 143

## T

Teknik langkah demi langkah 104, 110  
Teknik semakan meja 104, 113, 153



# SENARAI RUJUKAN

- Chris Roffey, 2013. *Coding Club Python Basics*. Cambridge: Cambridge University Press.
- Chris Roffey, 2013. *Coding Club Python: Next Steps*. Cambridge: Cambridge University Press.
- Cormen, Thomas, H. and Leisers, Charles, E., July 31, 2009. *Introduction to Algorithms 3<sup>rd</sup> Edition*. MIT Press.
- Dasgupta, Sanjoy and Papadimitriou, Christos, Sept. 13, 2006. *Algorithms*. McGraw Hill.
- Dorling, M. dan Rouse, G., 2014. *Compute-IT 1: Computing for KS3*. London: Hodder Education.
- Dorling, M. dan Rouse, G., 2014. *Compute-IT 2: Computing for KS3*. London: Hodder Education.
- Engel, A., 1998. *Problem - Solving Strategies*. Springer Verlag.
- Gelca, R. and Andreescu, T., 2007. *Putnam and Beyond*. Springer.
- Klamkin, M. S., J. M. Anthony (ed.), 1994. *Mathematical Creativity in Problem Solving II, In Eves' Circles*. MAA.
- Polya, G., 1981. *Mathematical Discovery*. John Wiley & Sons.
- "Python for Beginners," dicapai pada 28 Disember 2016, <https://www.python.org/about/gettingstarted/>
- Sedgewick, Robert & Wayne, Kevin, 2011. *Algorithms 4<sup>th</sup> Edition*. Pearson Education Inc.
- "The Python Tutorial," dicapai pada 21 November 2016, <https://docs.python.org/3.5/tutorial/index.html>
- Zeitz, P., 1999. *The Art and Craft of Problem Solving*. John Wiley & Sons.



**RM 10.30**

ISBN 978-967-14509-3-2



9 789671 450932 >

**FT302001**