



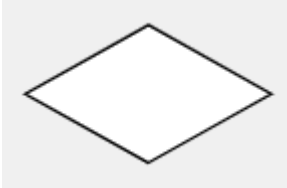



NOTA SAINS KOMPUTER

FORM 4		
BAB	PERKARA	PERNYATAAN/HURAIAN
1.1.1	<i>Kepentingan strategi dalam penyelesaian masalah</i>	Meningkatkan kemahiran berfikir
		Membantu pengembangan sesuatu konsep
		Menggalakkan pembelajaran sendiri
		Mewujudkan komunikasi dua hala
	<i>Teknik pemikiran komputasional dalam penyelesaian masalah</i>	Leraian (Decomposition) Memecahkan masalah kepada bahagian yang lebih kecil
		Pengecaman corak (Pattern recognition) Mencari persamaan antara masalah dan dalam masalah
		Peniskalaan (Abstraction) Menjana penyelesaian tepat kepada masalah yang dihadapi
		Algoritma (Algorithms) Membangunkan penyelesaian langkah demi langkah terhadap masalah yang dihadapi atau mematuhi peraturan untuk menyelesaikan masalah
1.1.2	<i>Ciri-ciri penyelesaian masalah berkesan</i>	Kos
		Masa
		Sumber
1.1.3	<i>Proses penyelesaian masalah</i>	Kumpulkan dan analisis data (<i>Compile and analyze data</i>)
		Tentukan masalah (<i>Determine the problem</i>)
		Jana idea, hapus halangan (<i>Generate ideas, eliminate obstacles</i>)
		Jana penyelesaian (<i>Generate solution</i>)
		Tentukan tindakan (<i>Determine action</i>)
		Laksanakan penyelesaian (<i>Execute solution</i>)
		Buat penilaian (<i>Conduct assessment/evaluation</i>)
		Buat penambahbaikan (<i>Make improvements</i>)
1.2.1	<i>Ciri-ciri algoritma</i>	Butiran jelas
		Boleh dilaksanakan
		Mempunyai batasan

	IPO	Input	Proses	Output
	<i>Perwakilan algoritma</i>	Pseudokod Senarai struktur kawalan komputer yang ditulis dalam bahasa pertuturan manusia dan mempunyai nombor turutan		
		Carta alir Alternatif kepada pseudokod, menggunakan simbol grafik untuk mewakili arahan-arahan penyelesaian		
	<i>Carta alir</i>		Terminal MULA/TAMAT Permulaan atau penamat algoritma dalam carta alir. Bergantung pada perkataan yang ditulis dalam simbol.	
			Input/Output Membaca input atau memaparkan output ke skrin.	
			Proses Arahan untuk memproses input dalam bentuk ungkapan, memproses fail, dan sebagainya.	
			Penghubung Titik sambungan untuk menyambungkan carta alir yang terpisah.	
			Syarat Menguji syarat. Menggunakan pernyataan YA atau TIDAK.	
			Aliran aktiviti Menghubungkan nod-nod (simbol-simbol) untuk menunjukkan aliran proses.	
1.2.2	<i>Struktur kawalan dalam pengaturcaraan</i>	Struktur kawalan urutan	Straightforward; tiada syarat.	
		Struktur kawalan pilihan	Mempunyai syarat yang menentukan satu daripada dua cabang yang dipilih.	
			IF-ELSE-IF (jika, jika tidak)	

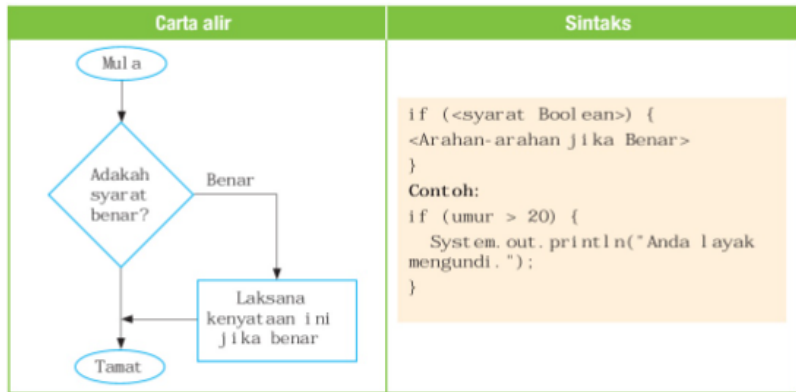
			SWITCH-CASE
		Struktur kawalan ulangan	Mengulang arahan-arahan sehingga menerima syarat berhenti atau mencapai bilangan yang sudah ditetapkan.
			FOR (berasaskan pembilang)
			WHILE (berasaskan syarat)
			DO-WHILE (berasaskan syarat)
	<i>Jenis penyataan</i>	UMPUKAN Diisytiharkan dalam aturcara. Merujuk "sama dengan" atau simbol "=" dan memberi nilai kepada pemboleh ubah. Contoh: int x = y	
		ARITMETIK Terdiri daripada satu atau lebih operasi aritmetik seperti penambahan, penolakan, pendaraban dan pembahagian. Contoh: int bmi = berat*tinggi*tinggi;	
1.2.3	<i>Ciri-ciri algoritma yang telah diuji</i>	Mudah difahami , mudah ditulis sebagai aturcara komputer	
		Lengkap , memenuhi semua keperluan penyelesaian masalah	
		Efisien , algoritma berfungsi dengan pantas dan tidak memakan banyak ruang memori	
		Memenuhi kriteria reka bentuk perisian yang ingin dibangunkan	
	<i>Ralat algoritma</i>	Ralat sintaks (syntax)	<ul style="list-style-type: none"> - Tidak wujud dalam algoritma - Bahasa pengaturcaraan salah - Biasanya ditemui secara automatik oleh perisian kompiler bahasa pengaturcaraan - Ralat algoritma tidak menyebabkan ralat sintaks
		Ralat logik (logic)	<ul style="list-style-type: none"> - Perisian yang dihasilkan tidak menjalankan fungsi-fungsi yang sepatutnya atau tidak lengkap - Output tidak tepat
		Ralat masa larian (runtime)	<ul style="list-style-type: none"> - Atur cara yang tidak dapat dimulakan, menjadi sangat perlahan atau tidak responsif

			<ul style="list-style-type: none"> - Dikenal pasti daripada kegagalan output dan paparan amaran dalam atur cara - Struktur kawalan urutan tidak betul - Pemboleh ubah tidak mengandungi nilai atau objek - Pembahagian dengan sifar - Struktur kawalan pengulangan yang tidak boleh berhenti kerana logik syarat yang salah 	
	<i>Langkah-langkah pengujian algoritma</i>	1. Kenal pasti "Output dijangka"		
		2. Kenal pasti "Output diperolehi"		
		3. Bandingkan "Output diperolehi" dan "Output dijangka"		
		4. Analisis dan baiki algoritma		
1.2.4 1.2.5 1.2.6	-	-		
1.3.0	<i>Pemboleh ubah</i>	Ruang simpanan sementara untuk nombor, teks dan objek.		
	<i>Pemalar</i>	Nilai tetap dan tidak akan berubah. Contoh: nilai pi.		
	<i>Jenis data</i>	<i>Jenis data</i>	<i>Contoh nilai</i>	<i>Kapasiti ingatan komputer</i>
		Integer	Min. -2147483648 Max. 2147483647	4 bait
		<i>float</i>	Min. -3.4e38 Max. 3.4e38	4 bait
		<i>double</i>	Min. -1.7e308 Max. 3.4e38	8 bait
		<i>char</i>	Satu karakter sahaja	2 bait
		<i>String</i>	Bermula dari 0 hingga tiada had	>10 bait
<i>Boolean</i>		Benar (<i>true</i>) Palsu (<i>false</i>)	1 bit	
1.3.1	-	-		
1.3.2	<i>Jenis data</i>	<i>Integer</i> int	Semua nombor yang tidak mempunyai pecahan atau	

			perpuluhan	
		Nombor nyata float, double	Semua nombor yang mempunyai pecahan atau perpuluhan	
		Aksara/rentetan char/String	Data dalam bentuk ruang kosong (space), teks, perkataan atau nilai yang mengandungi susunan aksara atau simbol	
		Boolean	Data dalam bentuk pilihan; benar atau palsu, <i>true or false</i>	
1.3.3	<i>Pemboleh ubah</i>	Sejagat (<i>Global</i>)	Berfungsi dalam atur cara sahaja.	
		Setempat (<i>Local</i>)	Berfungsi dalam subatur cara yang diisytiharkan.	
	<i>Ciri-ciri pemboleh ubah sejagat dan setempat</i>	Ciri-ciri	Sejagat (<i>Global</i>)	Setempat (<i>Local</i>)
		Pengisytiharan	Dilakukan di luar mana-mana fungsi	Diisytiharkan dalam sebuah fungsi
		Akses	Boleh diakses di mana-mana fungsi	Tidak boleh diakses di luar fungsi itu
Fungsi		Boleh digunakan hingga akhir program	Boleh digunakan untuk fungsi yang diisi sahaja	
<i>Contoh pemboleh ubah sejagat dan setempat</i>	<pre> public class A{ int data = 50; static int m = 100; //global variable ← Pemboleh ubah sejagat public static void main(String[] args) {; int n=90; //local variable ← Pemboleh ubah setempat } } //end of class </pre> <p>Rajah 1.21 Contoh atur cara yang menggunakan pemboleh ubah sejagat dan pemboleh ubah setempat</p>			
1.3.4 1.3.5 1.3.6	-	-		

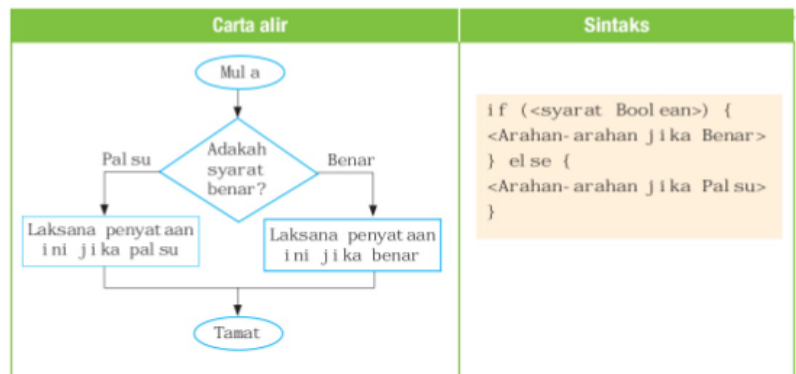
1.4.1

Struktur kawalan pilihan IF



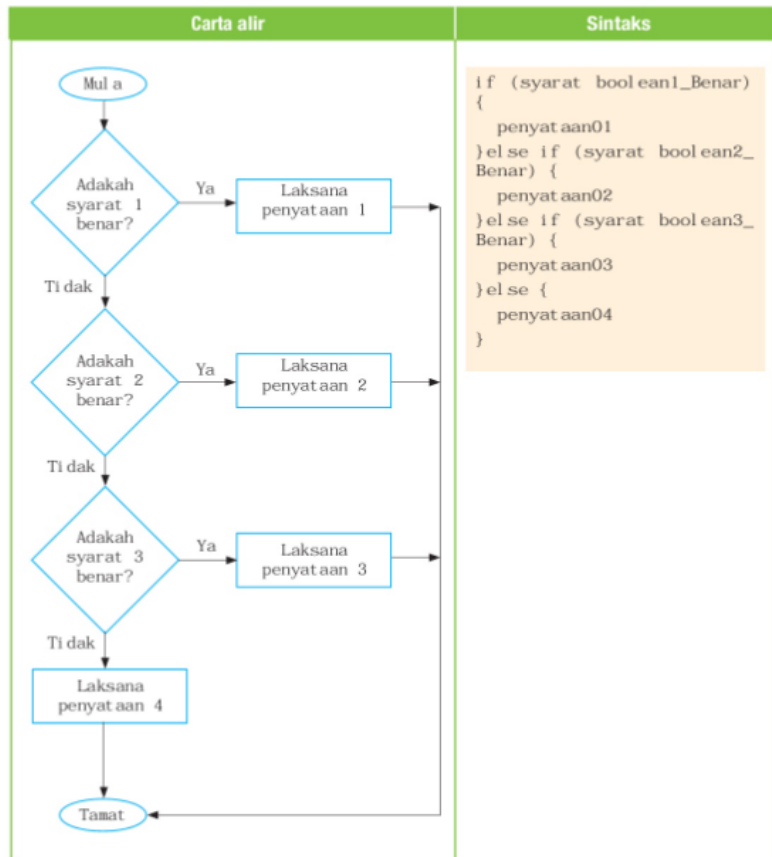
Rajah 1.26 Carta alir dan sintaks if

Struktur kawalan pilihan IF-ELSE

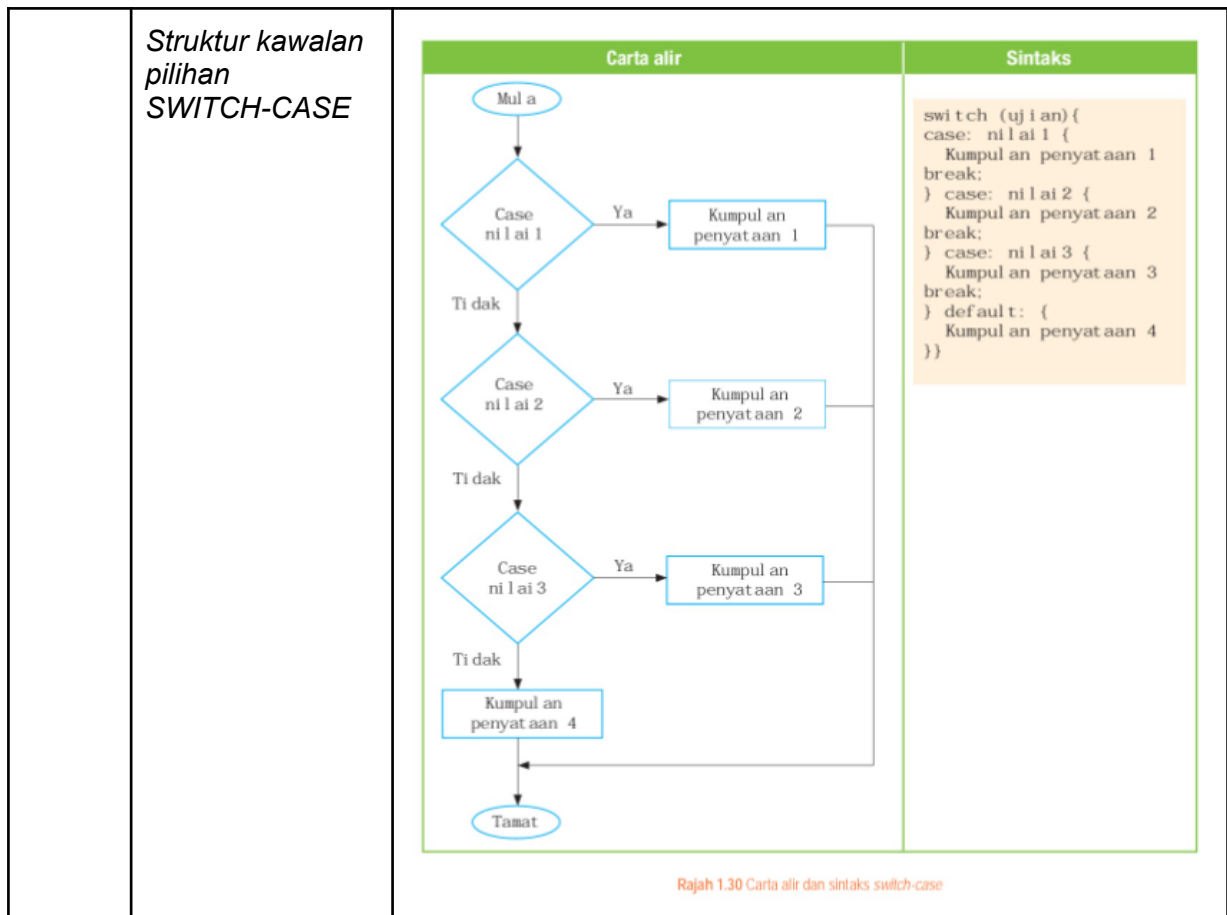


Rajah 1.27 Carta alir dan sintaks if-else

Struktur kawalan pilihan
IF-ELSE-IF



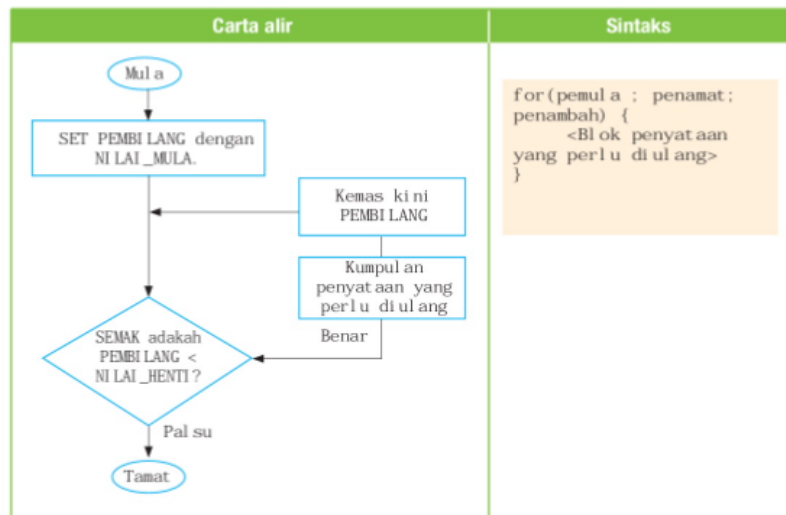
Rajah 1.28 Carta alir dan sintaks if-else-if



1.4.2	Operator hubungan	Operator hubungan	Penerangan
		==	Sama dengan
		!=	Tidak sama dengan
		>	Lebih besar daripada
		>=	Lebih besar daripada atau sama dengan
		<	Kurang daripada
		<=	Kurang daripada atau sama dengan
	Operator logikal	AND	A && B (logik AND)
			A & B (logik Boolean AND)
		OR	A B (logik OR)
		A B (logik Boolean OR)	
NOT		!A (logik NOT)	

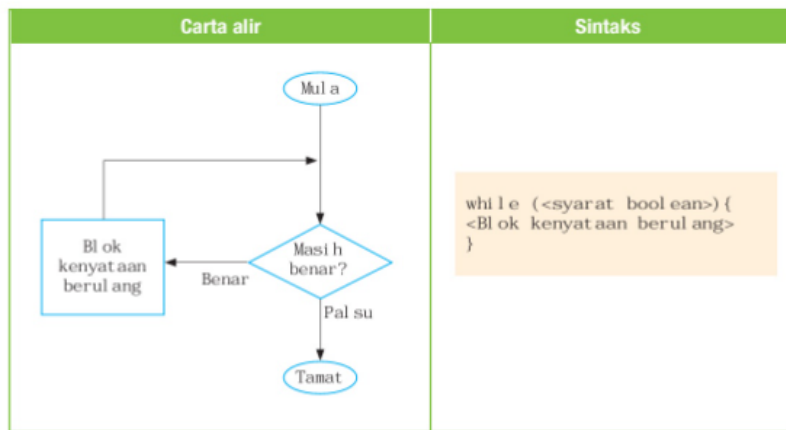
1.4.3

Struktur kawalan ulangan FOR



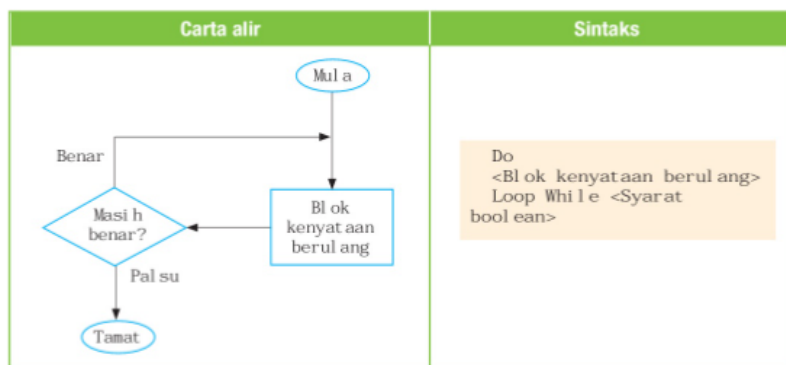
Rajah 1.35 Carta alir dan sintaks untuk kawalan ulangan for

Struktur kawalan ulangan WHILE



Rajah 1.36 Carta alir dan sintaks untuk kawalan ulangan while

Struktur kawalan ulangan DO-WHILE



Rajah 1.37 Carta alir dan sintaks untuk kawalan ulangan do-while

1.4.4

Operator increment (++)

Jadual 1.13 Ungkapan tokok

Ungkapan	Makna	Contoh
<code>i += 1</code>	<code>i = i + 1</code>	<pre>while (<syarat boolean>){ <Bl ok kenyat aan berul ang> <kemas ki ni nilai dalam syarat> }</pre> <p>Katakan i mengandungi 9. <code>i += 1</code> akan menambah 1 kepada 9. Jadi, nilai baharu i ialah 10.</p>
<code>i += 2</code>	<code>i = i + 2</code>	<pre>while (<syarat boolean>){ <Bl ok kenyat aan berul ang> <kemas ki ni nilai dalam syarat> }</pre> <p>Katakan i mengandungi 5. <code>i += 2</code> akan menambah 2 kepada 5. Jadi, nilai baharu i ialah 7.</p>

Ungkapan	Makna	Contoh
<code>i += 3</code>	<code>i = i + 3</code>	<pre>while (<syarat boolean>){ <Bl ok kenyat aan berul ang> <kemas ki ni nilai dalam syarat> }</pre> <p>Katakan i mengandungi 2. <code>i += 3</code> akan menambah 3 kepada 2. Jadi, nilai baharu i ialah 5.</p>

Operator decrement (--)

Jadual 1.14 Ungkapan susut

Ungkapan	Makna	Contoh
<code>i -= 1</code>	<code>i = i - 1</code>	<pre>while (<syarat boolean>){ <Bl ok kenyat aan berul ang> <kemas ki ni nilai dalam syarat> }</pre> <p>Katakan i mengandungi 9. <code>i -= 1</code> akan berkurang 1 daripada 9. Jadi, nilai baharu i ialah 8.</p>
<code>i -= 2</code>	<code>i = i - 2</code>	<pre>while (<syarat boolean>){ <Bl ok kenyat aan berul ang> <kemas ki ni nilai dalam syarat> }</pre> <p>Katakan i mengandungi 5. <code>i -= 2</code> akan berkurang 2 daripada 5. Jadi, nilai baharu i ialah 3.</p>
<code>i -= 3</code>	<code>i = i - 3</code>	<pre>while (<syarat boolean>){ <Bl ok kenyat aan berul ang> <kemas ki ni nilai dalam syarat> })</pre> <p>Katakan i mengandungi 2. <code>i -= 3</code> akan berkurang 3 daripada 2. Jadi, nilai baharu i ialah -1.</p>

Math.random()

Subatur cara Java untuk menjana nombor secara rambang.

1.5.0

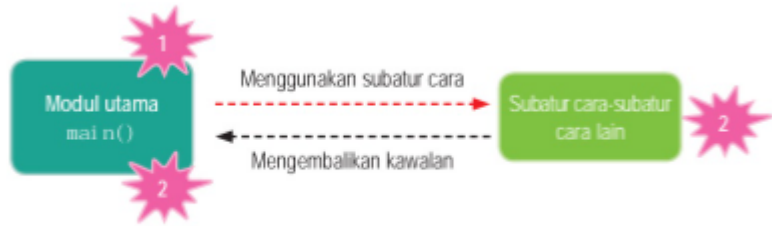
Amalan terbaik pengaturcaraan

Penggunaan inden yang konsisten

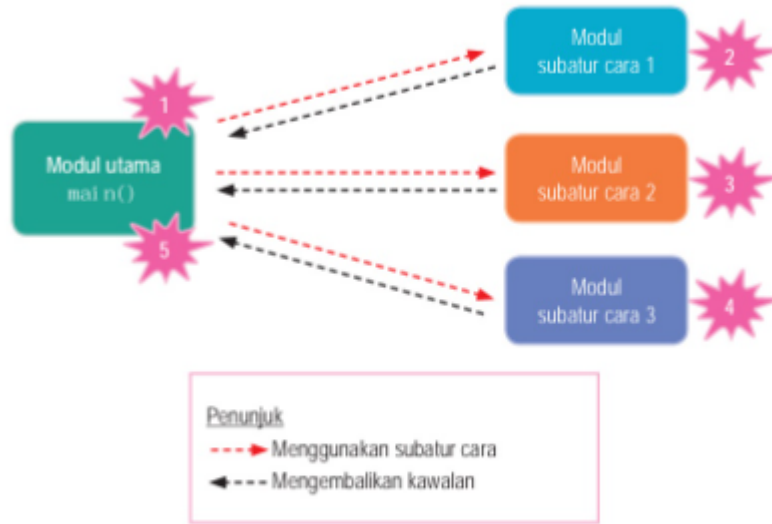
Jenis data yang bersesuaian

		Pemboleh ubah yang bermakna	
		Komen untuk menerangkan fungsi kod	
1.5.1	<i>Jenis ralat dalam aturcara</i>	RALAT SINTAKS	Kesalahan tatabahasa <i>Contoh: ejaan, tatatanda</i>
			Penggunaan objek atau aksara yang tidak dikenali
		RALAT LOGIK	Atur cara tidak berfungsi seperti yang diingini
		RALAT MASA LARIAN	Pengiraan data bukan berangka (<i>non-numerical</i>)
			Pembahagian dengan digit 0
			Mencari punca kuasa dua bagi nombor negatif
1.5.2 1.5.3	-	-	
1.5.4	<i>Menghasilkan atur cara yang baik dan mudah difahami</i>	Meletakkan komen pada setiap fungsi yang dibuat	
		Menggunakan nama pemboleh ubah yang bermakna	
		Menggunakan inden yang selesa untuk dibaca	
	<i>Jenis komen</i>	// komen	
		/* komen */	
		/** komen */	
1.6.1	<i>Tatasusunan (array)</i>	Pemboleh ubah yang membolehkan koleksi beberapa nilai data (<i>elemen</i>) dalam satu-satu masa dengan menyimpan setiap elemen dalam ruang memori berindeks.	
	<i>Pemboleh ubah</i>	Slot memori yang telah dikhaskan untuk menyimpan data	
1.6.2	<i>Kelebihan menggunakan struktur modul atau subatur cara</i>	Lebih mudah untuk digunakan semula	
		Lebih mudah untuk menangani projek komputer	
		Lebih mudah untuk diuji, dinyah pijat dan dibaiki	
		Projek kompleks menjadi lebih ringkas	
		Membolehkan tugas pengaturcaraan dibahagikan kepada ahli kumpulan yang berbeza	

Memanggil subatur daripada main ()



Rajah 1.60(a) Perhubungan antara subatur cara main() dengan subatur cara lain

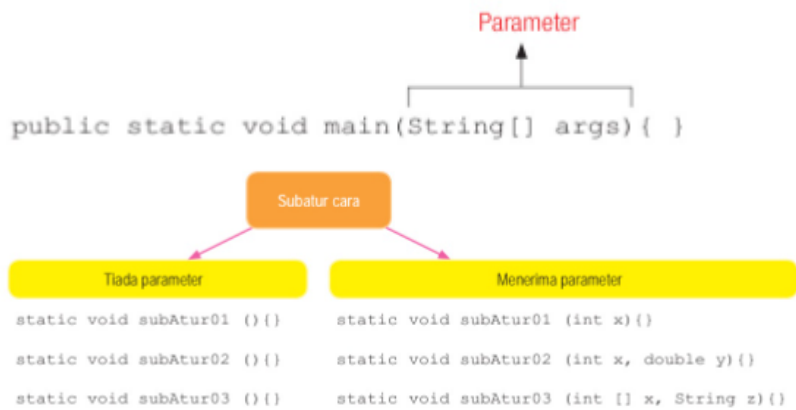


Rajah 1.60(b) Subatur cara main() sebagai modul utama

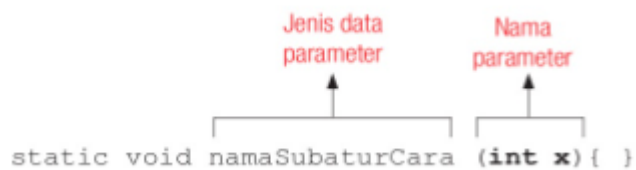
Parameter

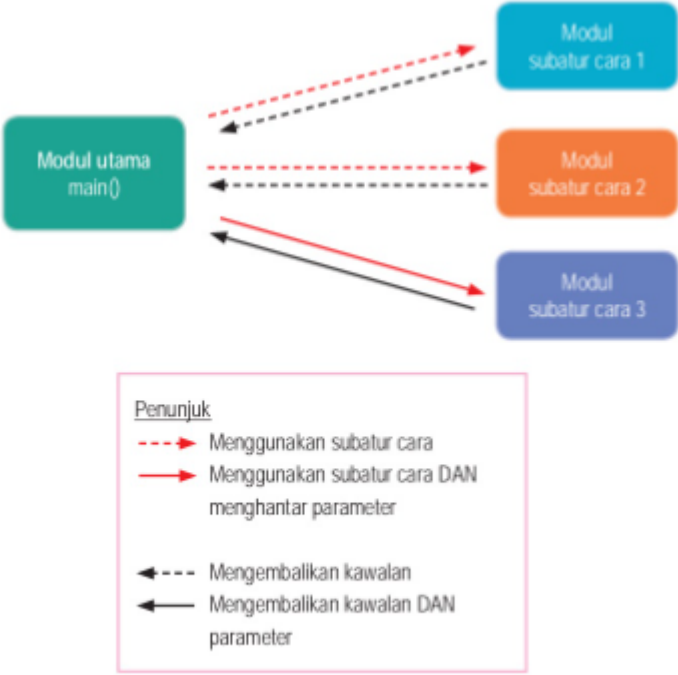
Parameter/argumen

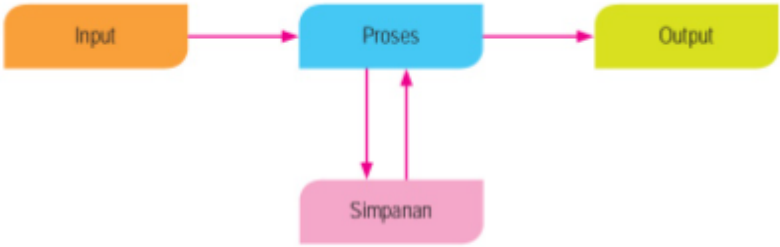
Pemboleh ubah yang membolehkan subatur cara menerima nilai daripada pemanggil

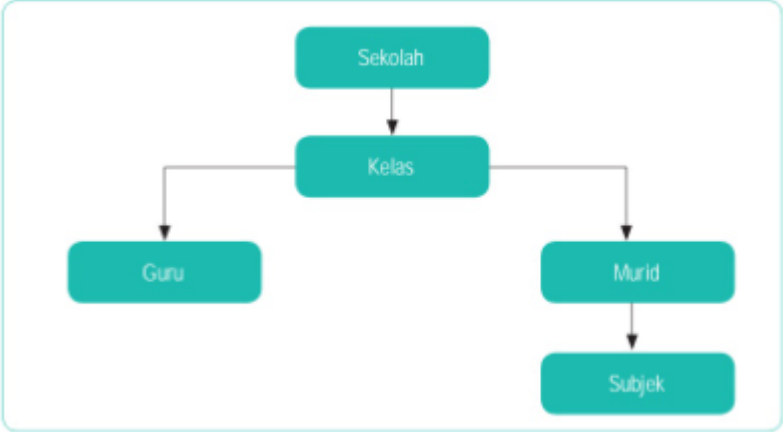
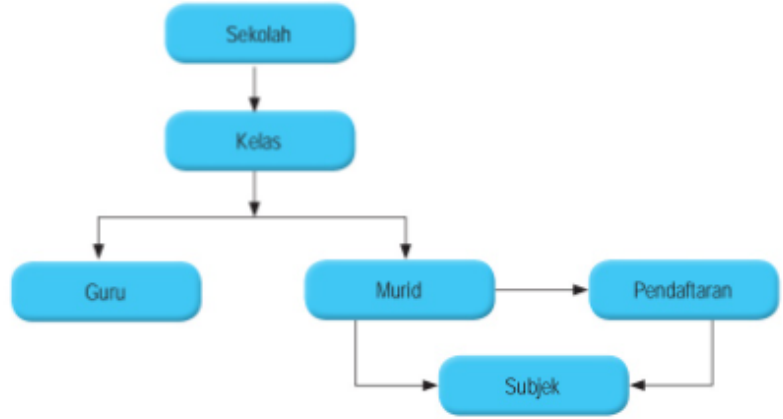




Rajah 1.63 Seselengah subatur cara boleh menerima parameter

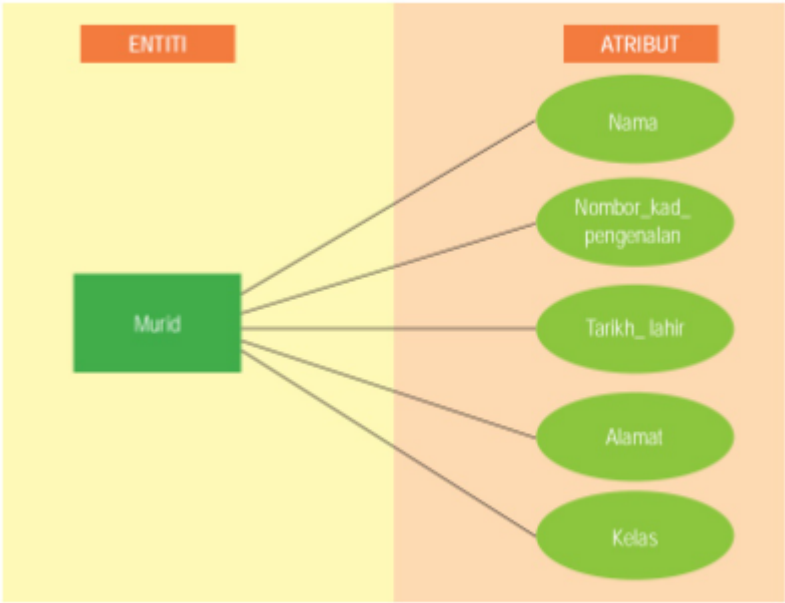












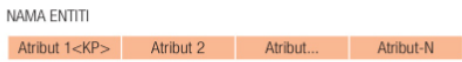
	Mengembalikan data	 <p>Rajah 1.64 Jenis-jenis parameter dan pulangan daripada subatur cara dipanggil</p>		
	Definisi subatur cara dan penjelasannya	<pre>static void subAturCara () { }</pre>	Tidak memulangkan data, Hanya kawalan dipulangkan	
		<pre>static int subAturCara () { }</pre>	Memulangkan data jenis integer	
		<pre>static double subAturCara () { }</pre>	Memulangkan data jenis double	
		<pre>static String subAturCara () { }</pre>	Memulangkan data jenis String	
1.6.3	Perbezaan antara Fungsi (Function) dan Prosedur (Procedure) pada subatur cara		Prosedur	Fungsi
		Persamaan	Mengembalikan kawalan.	
		Perbezaan	Tidak mengembalikan data.	Mengembalikan data.
1.6.4	-	-		
1.7.0	Kitaran Hayat Pembangunan Sistem (Software Development Life	Analisis masalah	Dapatkan data, analisis masalah, pernyataan masalah.	
		Reka bentuk	Disediakan daripada analisis	

	Cycle - SDLC)	penyelesaian	masalah. Spesifikasi reka bentuk sistem digunakan sebagai input untuk fasa seterusnya.
		Laksanakan penyelesaian	Kerja dibahagikan dalam sub modul atau unit dan pengekodan bermula. Fasa paling lama.
		Uji dan nyah ralat	Pengujian, menyah ralat, membaiki ralat, dan penambahbaikan dijalankan.
		Dokumentasi	Disediakan di setiap fasa. Contoh: carta alir, kod atur cara dan carta IPO
1.7.1.1 1.7.1.2 1.7.1.3 1.7.1.4 1.7.1.5	-	Ringkasan di atas.	
2.1.0	Data	Himpunan fakta mentah mengenai sesuatu. Boleh terdiri daripada perkataan, angka atau gambar.	
		Boleh menjadi sesuatu fakta yang bermakna atau tidak bermakna sehingga fakta-fakta tersebut diproses.	
	Maklumat	Hasil daripada pengumpulan, pemprosesan dan penganalisan data yang boleh digunakan untuk membuat sesuatu keputusan.	
		Menjadi lebih bermakna dan mudah untuk difahami.	
	Evolusi pengurusan data	Sistem pemprosesan manual	
		Sistem pemprosesan fail	
		Sistem pangkalan data	
Blok sistem komputer	 <pre> graph LR Input[Input] --> Proses[Proses] Proses --> Output[Output] Proses <--> Simpanan[Simpanan] </pre> <p>Rajah 2.5 Gambar rajah blok sistem komputer</p>		
2.1.1	Kualiti pangkalan data	Integriti data (<i>integrity/validity</i>)	
		Ketekalan data (<i>consistency</i>)	
		Kelewahan data (<i>redundancy</i>)	

2.1.2	Model-model pangkalan data	Hierarki (<i>Hierarchy</i>)																																
		Rangkaian (<i>Network</i>)																																
		Hubungan (<i>Relational</i>)																																
		Berorientasikan objek (<i>Object oriented</i>)																																
	Model pangkalan data hierarki	 <pre> graph TD Sekolah --> Kelas Kelas --> Guru Kelas --> Murid Murid --> Subjek </pre> <p>Rajah 2.9 Model Pangkalan Data Hierarki</p>																																
	Model pangkalan data rangkaian	 <pre> graph TD Sekolah --> Kelas Kelas --> Guru Kelas --> Murid Murid --> Pendaftaran Murid --> Subjek Pendaftaran --> Subjek </pre> <p>Rajah 2.13 Model pangkalan data rangkaian</p>																																
	Model pangkalan data hubungan	<table border="1" data-bbox="609 1550 1394 1657"> <thead> <tr> <th colspan="3">Murid</th> <th colspan="2">PENDAFTARAN</th> <th colspan="3">SUBJEK</th> </tr> <tr> <th>Kelas</th> <th>Nama</th> <th>ID murid</th> <th>ID murid</th> <th>Kod Subjek</th> <th>Kod Subjek</th> <th>Subjek</th> <th>Aliran</th> </tr> </thead> <tbody> <tr> <td>5A</td> <td>Izali bi Ahmad</td> <td>123/16</td> <td>123/16</td> <td>FZ1</td> <td>FZ1</td> <td>Fizik</td> <td>Sains Tulen</td> </tr> <tr> <td>5B</td> <td>Chong Wei</td> <td>124/16</td> <td>128/16</td> <td>KIM</td> <td>SC1</td> <td>Sains</td> <td>Sastera</td> </tr> </tbody> </table> <p>Rajah 2.15 Contoh model pangkalan data hubungan</p>	Murid			PENDAFTARAN		SUBJEK			Kelas	Nama	ID murid	ID murid	Kod Subjek	Kod Subjek	Subjek	Aliran	5A	Izali bi Ahmad	123/16	123/16	FZ1	FZ1	Fizik	Sains Tulen	5B	Chong Wei	124/16	128/16	KIM	SC1	Sains	Sastera
Murid			PENDAFTARAN		SUBJEK																													
Kelas	Nama	ID murid	ID murid	Kod Subjek	Kod Subjek	Subjek	Aliran																											
5A	Izali bi Ahmad	123/16	123/16	FZ1	FZ1	Fizik	Sains Tulen																											
5B	Chong Wei	124/16	128/16	KIM	SC1	Sains	Sastera																											


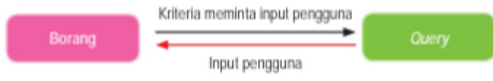


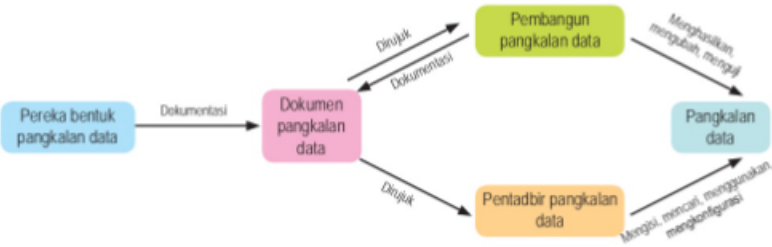
	<p><i>Model pangkalan data berorientasikan objek</i></p>	<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 5px; width: 45%;"> <p style="text-align: center;">Definisi kelas</p> <div style="background-color: #f4a460; padding: 5px; border: 1px solid black; margin-bottom: 5px;"> <p style="text-align: center; background-color: #e67e22; color: white; padding: 2px;">Pensyarah</p> <p style="text-align: center;">Atribut</p> <p>Nama Staff_No Fakulti Jantina No. Telefon</p> </div> <div style="background-color: #f4a460; padding: 5px; border: 1px solid black;"> <p style="text-align: center;">Kaedah</p> <p>KemaskiniProfil() DapatkanKodPenyelidikan() PaparKod()</p> </div> </div> <div style="width: 45%;"> <p style="text-align: center;">Kejadian kelas</p> <div style="background-color: #add8e6; padding: 5px; border: 1px solid black; margin-bottom: 5px;"> <p>Nama = Ahmad Naseer Staff_No = 0421 Fakulti = Pusat Asasi Jantina = Lelaki No. Telefon = 0123457891</p> </div> <div style="background-color: #ffff00; padding: 5px; border: 1px solid black;"> <p>Nama = Siti Alia Mansor Staff_No = 0500 Fakulti = FSTM Jantina = Perempuan No. Telefon = 0129876543</p> </div> </div> </div> <p style="text-align: center; color: #e67e22;">Rajah 2.16 Perwakilan kelas model pangkalan data berorientasikan objek</p>												
<p>2.1.3</p>	<p><i>Komponen dalam pangkalan data</i></p>	<div style="display: flex; align-items: center;"> <div style="text-align: center; margin-right: 20px;">  <p>Pangkalan data</p> <p>Jadual hubungan/entiti</p> </div> <div style="margin-right: 20px;">  </div> <div style="border: 1px solid black; padding: 5px; width: 300px;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #2e86c1; color: white;"> <th style="width: 33%;">Nama</th> <th style="width: 33%;">ID</th> <th style="width: 33%;">Markah</th> </tr> </thead> <tbody> <tr style="background-color: #fff9c4;"> <td>Brenda</td> <td>10359</td> <td>80</td> </tr> <tr style="background-color: #fff9c4;"> <td>Izzati</td> <td>10337</td> <td>92</td> </tr> <tr style="background-color: #fff9c4;"> <td>Faiz</td> <td>10555</td> <td>90</td> </tr> </tbody> </table> </div> <div style="margin-left: 20px;"> <p>Atribut</p> <p>Rekod</p> <p>Nilai atribut</p> </div> </div> <p style="text-align: center; color: #e67e22;">Rajah 2.17 Komponen dalam pangkalan data.</p>	Nama	ID	Markah	Brenda	10359	80	Izzati	10337	92	Faiz	10555	90
Nama	ID	Markah												
Brenda	10359	80												
Izzati	10337	92												
Faiz	10555	90												
	<p><i>Persekitaran</i></p>	<p>Organisasi di mana pangkalan data direka bentuk, dibangunkan dan digunakan.</p>												
	<p><i>Entiti</i></p>	<p>Suatu objek unik dan boleh dikenal pasti dalam sesuatu persekitaran seperti orang, tempat atau benda. Mempunyai data deskriptif yang boleh dikutip dan disimpan.</p>												
	<p><i>Atribut</i></p>	<p>Merupakan data-data deskriptif entiti. Menerangkan ciri-ciri sesuatu entiti.</p>												

	<p><i>Hubungan antara entiti dan atribut</i></p>	 <p style="text-align: center;">Rajah 2.20 Hubungan antara entiti dan atribut</p>
	<p><i>Hubungan</i></p>	<p>Perkaitan antara entiti.</p>
	<p><i>Set hubungan</i></p>	 <p style="text-align: center;">Rajah 2.21 Set hubungan yang menunjukkan hubungan di antara dua entiti</p>
	<p><i>Kekardinalan (Cardinality)</i></p>	<p>1:1 (<i>satu-ke-satu</i>)</p> <p>1:M (<i>satu-ke-banyak</i>)</p> <p>M:N (<i>banyak-ke-banyak</i>)</p>
<p>2.2.0</p>	<p><i>Penormalan</i></p>	<p>Proses memecahkan jadual yang lebar kepada beberapa jadual hubungan yang lebih kecil. Digunakan untuk mencegah kelewahan dan kehilangan ketekalan data.</p>
<p>2.2.1</p>	<p><i>Kunci hubungan</i></p> <p><i>Kunci primer</i></p> <p><i>Kunci asing</i></p>	<p>Nama atribut khusus dalam jadual yang digunakan untuk mengekalkan integriti data hubungan.</p> <p>Mempunyai nilai unik</p> <p>Nilai tetap dan tidak berubah</p> <p>Tidak boleh dibiarkan kosong (null)</p> <p>Tidak boleh mempunyai nilai sama secara berulang</p> <p>Atribut yang mengandungi nilai kunci primer daripada jadual sasaran.</p>

2.2.2	ERD	Entity Relationship Diagram (Gambar Rajah Perhubungan Entiti)								
		ERD konseptual								
		ERD logikal								
		ERD fizikal								
	Langkah-langkah melukis ERD	Kenal pasti data yang diperlukan oleh sistem dari persekitaran pangkalan data								
		Kenal pasti kumpulan untuk data yang berkait secara logikal (dinamakan sebagai entiti)								
		Kenal pasti perkaitan antara dua entiti untuk semua entiti								
Kekardinalan dalam ERD	<p style="text-align: center;">Jadual 2.7 Jenis-jenis kekardinalan</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #f4a460;">Kekardinalan</th> <th style="background-color: #f4a460;">Penerangan</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1:1 (satu-ke-satu)</td> <td> Hubungan: "Seorang warganegara mempunyai satu kad pengenalan sahaja."  </td> </tr> <tr> <td style="text-align: center;">1:M (satu-ke-banyak)</td> <td> Hubungan: "Seorang murid boleh menyertai lebih daripada satu kelab di sekolah."  </td> </tr> <tr> <td style="text-align: center;">M:N (banyak-ke-banyak)</td> <td> Hubungan: "Banyak pelanggan pasar raya membeli pelbagai jenis barangan."  </td> </tr> </tbody> </table>		Kekardinalan	Penerangan	1:1 (satu-ke-satu)	Hubungan: "Seorang warganegara mempunyai satu kad pengenalan sahaja." 	1:M (satu-ke-banyak)	Hubungan: "Seorang murid boleh menyertai lebih daripada satu kelab di sekolah." 	M:N (banyak-ke-banyak)	Hubungan: "Banyak pelanggan pasar raya membeli pelbagai jenis barangan." 
Kekardinalan	Penerangan									
1:1 (satu-ke-satu)	Hubungan: "Seorang warganegara mempunyai satu kad pengenalan sahaja." 									
1:M (satu-ke-banyak)	Hubungan: "Seorang murid boleh menyertai lebih daripada satu kelab di sekolah." 									
M:N (banyak-ke-banyak)	Hubungan: "Banyak pelanggan pasar raya membeli pelbagai jenis barangan." 									
2.2.3	Pertimbangan dalam reka bentuk pangkalan data	Hendaklah mengandungi data yang lengkap dan diperlukan sahaja								
		Hendaklah memudahkan penyimpanan, pencarian dan pengemaskinian								
		Mesti menggunakan model yang sesuai Misalnya, model pangkalan data hubungan yang popular								
Jenis entiti	Entiti teguh	Tidak bergantung pada entiti lain								
	Entiti lemah	Bergantung pada entiti sama jenis yang lain								
	Entiti bersama	Terjadi bersama dengan satu atau banyak jenis entiti yang lain								
Format set hubungan	Penyataan teks	NAMA ENTITI(Atribut 1 <KP>, Atribut 2, ... Atribut N)								
	Perwakilan grafik	NAMA ENTITI 								

2.2.4	<i>Kebergantungan fungsi sepenuh</i>	Berlaku apabila atribut-atribut bergantung sepenuhnya kepada kesemua atribut kunci dalam jadual.
	<i>Kebergantungan fungsi separa</i>	Berlaku apabila atribut-atribut bergantung kepada salah satu daripada atribut kunci dalam jadual.
	<i>Kebergantungan fungsi transitif</i>	Berlaku apabila atribut-atribut bergantung kepada atribut biasa yang lain dalam jadual.
2.2.5	<i>Penormalan</i>	Kaedah menganalisis jadual-jadual berasaskan atribut kunci dan kebergantungan fungsi dengan tujuan mengurangkan duplikasi data dalam pangkalan data.
	<i>Objektif 0NF - 1NF</i>	<ul style="list-style-type: none"> - Pastikan lajur adalah atomik - Kenal pasti kunci primer dalam jadual
	<i>Isu 0NF - 1NF</i>	<ul style="list-style-type: none"> - Jadual mempunyai kedua-dua kebergantungan fungsi sepenuh dan kebergantungan fungsi separa
	<i>Objektif 1NF - 2NF</i>	<ul style="list-style-type: none"> - Jadual tiada kebergantungan fungsi separa - Semua jadual hubungan mempunyai kebergantungan fungsi sepenuh sahaja - Setiap satu jadual hubungan mempunyai atribut kunci primer
	<i>Isu 1NF - 2NF</i>	<ul style="list-style-type: none"> - Mungkin terdapat kebergantungan fungsi transitif antara atribut-atribut yang belum dikenal pasti
	<i>Objektif 2NF - 3NF</i>	<ul style="list-style-type: none"> - Hapuskan kebergantungan fungsi transitif dalam jadual jika ada
	<i>Isu 2NF - 3NF</i>	<ul style="list-style-type: none"> - Tiada
2.3.0	<i>Aktiviti-aktiviti pembangunan pangkalan data hubungan</i>	Membina pangkalan data
		Membina jadual (<i>table</i>)
		Mengisi jadual
		Mewujudkan hubungan (<i>relationship</i>)
		Membina dan mengisi borang
		Menyunting rekod
		Membina carian (<i>query</i>)
		Membina laporan (<i>report</i>)
2.3.1	<i>Bina jadual berpandukan skema hubungan</i>	Cipta jadual baharu
		Buka "Design View"
		Set jenis data

		Set kunci primer
		Menambah jadual-jadual lain
		Membina hubungan antara jadual-jadual
2.3.2	<i>Tujuan penciptaan borang berkaitan kandungan jadual</i>	Memaparkan data daripada setiap rekod dalam jadual pangkalan data melalui susun atur yang lebih mudah difahami oleh pengguna
		Mengutip data untuk disimpan sebagai rekod dalam susun atur yang lebih mudah difahami oleh pengguna
		Mengesahkan data melalui ciri-ciri borang
2.3.2	<i>Split form</i>	Mempamerkan kedua-dua borang dan jadual dalam "Design View"
	<i>Multiple items</i>	Mempamerkan beberapa rekod jadual dalam satu borang
	<i>Form design</i>	Membolehkan borang direka cipta khusus dengan menentukan rekod-rekod yang ingin dipamerkan daripada satu atau lebih jadual dalam pangkalan data
2.3.4	-	-
2.3.5	<i>Kepentingan penjaanan laporan berdasarkan hasil Query</i>	Menghasilkan laporan untuk mesyuarat
		Membantu organisasi membuat keputusan
		Membantu dalam penerbitan
2.3.6	<i>Form name</i>	Nama borang terpilih dalam Access
	<i>View</i>	Mod untuk melihat borang: <i>Design, Datasheet, Print Preview, Layout, Pivotal Chart, Pivotal Layout</i> ataupun <i>Form</i>
	<i>Data Mode</i>	Sama ada dalam bentuk <i>Read-Only, Edit</i> ataupun <i>Add</i>
	<i>Window Mode</i>	Normal

	<p><i>Menggabungkan makro dengan borang yang menghasilkan laporan</i></p>	<ul style="list-style-type: none"> • menghasilkan kotak kombo dalam borang yang mempamerkan nilai-nilai atribut daripada jadual.  <ul style="list-style-type: none"> • menghasilkan <i>query</i> yang menggunakan kriteria daripada borang input pengguna.  <ul style="list-style-type: none"> • menghasilkan templat laporan <i>query</i> yang menggunakan <i>query</i> dengan kriteria.  <ul style="list-style-type: none"> • memanggil makro daripada butang <i>command</i> dalam borang untuk menghasilkan laporan <i>query</i> secara automatik berdasarkan input pengguna. 
	<p><i>Switchboard</i></p>	<p>Sistem antara muka bagi <i>Microsoft Access</i></p>
	<p><i>Aspek yang perlu dipertimbangkan dalam penghasilan sistem antara muka</i></p>	<p>Setelah memasuki pangkalan data, pengguna boleh mininjau baris-baris data dalam jadual pangkalan data melalui borang yang bersesuaian. Jika perlu, pengguna boleh mengemaskinikan, membetulkan, menambahkan atau menghapuskan rekod data.</p> <p>Sistem antara muka membantu pengguna menghasilkan laporan yang bersesuaian</p> <p>Pengguna dapat keluar daripada sistem selepas perubahan yang dibuat telah disimpan. Ciri-ciri seperti butang memudahkan pengguna bernavigasi dan melaksanakan tugas berkenaan pangkalan data.</p>
<p>2.3.7</p>	<p><i>Mendokumentasikan hasil kerja</i></p>	 <p>Rajah 2.97 Mendokumentasikan kerja memudahkan pembangunan, pentadbiran dan perubahan pangkalan data</p>
	<p><i>Dokumentasi</i></p>	<p>Nama pangkalan data</p> <p>Objektif pangkalan data</p> <p>Rajah alir data (<i>Data flow diagram</i>)</p> <p><i>Contoh rajah alir data:</i></p>

		<p style="text-align: center;">Rajah 2.98 Rajah alir data (Data flow diagram)</p>
		Kamus data (<i>data dictionary</i>)
		Spesifikasi entiti dan atribut
		Model konsep
		Reka bentuk logitkal (struktur jadual)
		Jadual hubungan ternormal dan sampel data
2.4.0	<i>Fasa-fasa pembangunan pangkalan data untuk sistem maklumat</i>	Merancang and menganalisis
		Mereka bentuk
		Membina
		Menyelenggara
2.4.1	<i>Penghasilan sebuah pangkalan data yang ternormal</i>	Merancang and menganalisis
		Mereka bentuk
		Membina
		Menyelenggara
2.4.2	-	-
3.1.0	<i>Reka bentuk interaksi</i>	Reka bentuk Susun atur teks, gambar, butang dan menu dalam sesuatu produk Interaksi Tindakan manusia semasa menggunakan produk tersebut
	<i>Prinsip reka bentuk interaksi</i>	Konsistensi (consistency)
		Kebolehan membuat pemerhatian (perceivability)
		Boleh dipelajari (learnability)
		Kebolehan untuk menjangka (predictability)
		Maklum balas (feedback)

3.1.1	<i>Keperluan interaksi antara manusia dengan komputer</i>	Mendapat permintaan dalam pasaran
		Meningkatkan produktiviti
		Mengurangkan kos selepas jualan
		Mengurangkan kos pembangunan
		Mengembangkan aktiviti dan menambahkan pengalaman manusia
		Perkomputeran sosial
3.1.2	<i>Skala Likert (1932)</i>	Bertujuan untuk melihat sejauh mana persetujuan pengguna terhadap produk yang dibangunkan.
	<i>Skala Guttman (1944)</i>	Bertujuan mendapatkan data yang jelas, tegas dan konsisten.
3.2.0	<i>Proses reka bentuk interaksi</i>	Mengenal pasti keperluan interaksi
		Membangunkan reka bentuk alternatif
		Membina prototaip interaksi
		Membuat penilaian reka bentuk interaksi produk
3.2.1 3.2.2 3.2.3	-	-

FORM 5

1.1.0	<i>Jenayah siber</i>	Serangan virus
		Penipuan
		Pornografi
		Penggodaman (<i>Hacking</i>)
		Pelanggaran hak cipta
		Ugutan
1.1.1	<i>Kesan inovasi dalam pengkomputeran</i>	Teknologi rangkaian
		Teknologi storan
		Menyumbang kepada kewujudan sistem dalam talian
		Sistem pengoperasian
		Jenayah siber
	<i>Jenis-jenis jenayah siber</i>	Dari perspektif pengguna persendirian dan institusi
		Dari perspektif keselamatan negara
1.1.2	<i>Etika dalam pengkomputeran</i>	Dilarang menggunakan komputer dengan niat untuk membahayakan orang lain
		Dilarang mengganggu urusan privasi atau tugas komputer orang lain
		Dilarang mencero boh ke dalam fail komputer pengguna yang lain
		Dilarang menggunakan komputer untuk mencuri
		Dilarang menggunakan komputer untuk bertindak sebagai saksi palsu
		Dilarang menggunakan atau menyalin perisian orang lain tanpa kebenaran
		Dilarang menggunakan harta intelek orang lain sebagai hak milik
		Tidak menggunakan sumber komputer pengguna lain tanpa kebenaran
		Mempertimbangkan kesan sosial daripada program dan sistem yang dibangunkan
		Disaran menggunakan komputer dengan menunjukkan pertimbangan dan menghormati pengguna lain



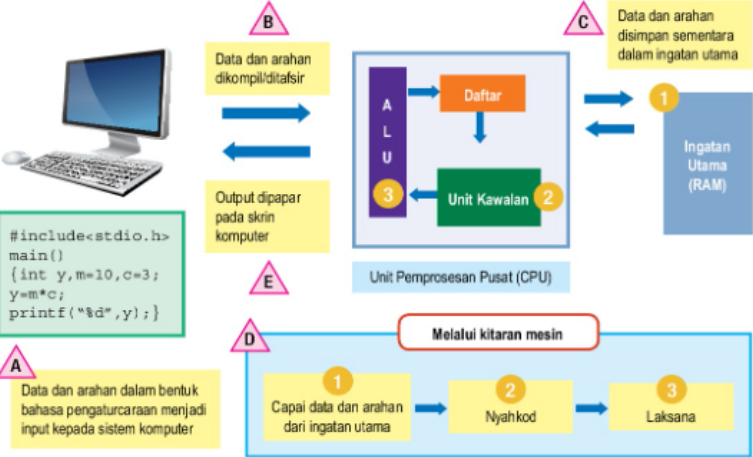
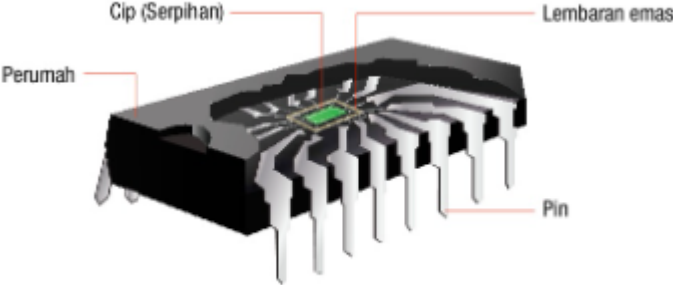
1.1.3	Langkah-langkah untuk memastikan keselamatan data	Penyulitan e-mel (<i>Encrypted email</i>)	
		Dua pengesahan (<i>Double verification</i>)	
		Kata laluan (<i>Password</i>)	
1.1.4	Penyulitan (<i>encryption</i>) data	Proses pengekodan teks biasa (<i>plaintext</i>) kepada bentuk teks sifer (<i>ciphertext</i>) yang tidak dapat difahami	
	Nyahsulit (<i>decryption</i>) data	Proses pengekodan teks sifer (<i>ciphertext</i>) kepada bentuk teks biasa (<i>plaintext</i>)	
	Jenis penyulitan	Penyulitan simetri (<i>Symmetric encryption</i>)	
		Penyulitan tidak simetri (<i>Asymmetric encryption</i>)	
Caesar Cipher	$E_n(x) = (x+n) \text{ mod } 26$ $n = \text{bilangan anjakan}$ $\text{mod } 26 = \text{jumlah huruf A - Z}$		
1.1.5	Implikasi hasil daripada aktiviti penggodaman dan cetak rompak	Cetak rompak mengancam industri kreatif dan perisian negara	
		Pengeluaran produk asli berkurang	
		Hilangnya keyakinan pelabur kepada negara	
		Wujud pengangguran dalam industri kreatif dan perisian	
		Menjejaskan kestabilan ekonomi negara	
1.1.6	Akta di bawah undang-undang siber	<p>Rajah 1.1.7 Akta-akta di bawah undang-undang siber</p>	
Penubuhan undang-undang siber antara negara	Amerika Syarikat	Akta Jenayah Komputer 1986, <i>Department of Justice</i>	
	China	1997 <i>Cybersecurity Law</i> oleh <i>Public Security Bureau (PSB)</i>	
	Arab Saudi	<i>Anti-Cyber Crime Law by Royal Decree No. M/17 of March 26, 2007</i>	
	Malaysia	Akta Jenayah Komputer 1997, SKMM, <i>Cyber Security Malaysia</i>	

1.1.7	<i>Kerjaya dalam bidang pengkomputeran</i>	Sistem rangkaian dan keselamatan rangkaian	Mengendalikan semua aspek keselamatan rangkaian dan membangunkan dasar keselamatan etc.
		Sokongan maklumat dan perkhidmatan	Mengintegrasikan pengurusan perniagaan dan analisis data, menyediakan sokongan pengkomputeran dan teknikal etc.
		Pengaturcaraan dan pembangunan perisian	Merancang, mereka bentuk, mengemaskinikan dan menguruskan perisian komputer dan sistem melalui pengaturcaraan etc.
		Web dan komunikasi digital	Membangunkan aplikasi berasaskan web, menulis kod, menyediakan bantuan teknikal etc.
<i>Elemen mutakhir dalam pengkomputeran yang mempengaruhi pembangunan bidang kerjaya baharu</i>	Mudah alih		
	Keselamatan data		
	Data raya (<i>Big data</i>) dan kecerdasan perniagaan dan analitik (<i>business intelligence and analytics</i>)		
1.2.0	<i>Komponen utama dalam seni bina komputer</i>	Unit pemprosesan pusat	
		Unit ingatan	
		Unit peranti input dan peranti output	
	<i>Unit pemprosesan pusat (CPU)</i>	Central Processing Unit - Unit aritmetik dan logik (ALU) - Unit kawalan - Daftar (<i>Register</i>)	
	<i>Unit ingatan</i>	Menyimpan data dan arahan semasa komputer digunakan	
	<i>Peranti input dan output</i>	Memasukkan data dan memaparkan hasil daripada sistem komputer	

<p>1.2.1</p>	<p><i>Kitaran mesin (Capai-Nyahkod-Laksana)</i></p>	<div data-bbox="612 224 1289 600" data-label="Diagram"> <p>The diagram illustrates the machine cycle. On the left is a blue box labeled 'INGATAN UTAMA (RAM)' with a yellow circle '1' above it. Two blue arrows point from RAM to the CPU. The CPU is a grey box containing three components: 'DAFTAR' (orange), 'UNIT KAWALAN' (green), and 'ALU' (purple). A yellow circle '2' is above the control unit, and a yellow circle '3' is above the ALU. Blue arrows show a cycle: from DAFTAR to UNIT KAWALAN, from UNIT KAWALAN to ALU, from ALU back to DAFTAR, and from DAFTAR back to UNIT KAWALAN. Below the CPU components is a light blue box labeled 'Unit Pemrosesan Pusat (CPU)'.</p> </div> <p>Rajah 1.2.5 Proses-proses dalam kitaran mesin (capai-nyahkod-laksana)</p> <p>Capai (fetch) Unit kawalan memberikan isyarat supaya pemproses mencapai data dan arahan dari ingatan utama dan disimpan dalam daftar sementara menunggu untuk diproses.</p> <p>Nyahkod (decode) Unit kawalan menyahkod atau menterjemah arahan bagi menentukan jenis operasi yang akan dilaksanakan oleh ALU (unit aritmetik dan logik).</p> <p>Laksana (execute) Pemproses melaksanakan operasi aritmetik dan logik terhadap data seperti yang dinyatakan oleh arahan.</p>
<p>1.2.1</p>	<p><i>Von Neuman</i></p>	<p>[1945] John Von Neuman mencipta kaedah untuk menyimpan atur cara komputer dalam sistem komputer. Atur cara dan data disimpan dalam ingatan dan digerakkan dari unit ingatan ke pemproses. Konsep ini dirujuk sebagai seni bina Von Neuman.</p>
<p>1.2.2</p>	<p><i>Komponen utama dalam seni bina Von Neuman</i></p>	<div data-bbox="612 1366 1380 1590" data-label="Diagram"> <p>The diagram shows the Von Neuman architecture. On the left is a computer monitor and keyboard labeled 'UNIT INPUT DAN OUTPUT'. In the center is a light orange box labeled 'Unit Pemrosesan Pusat (CPU)' containing 'DAFTAR', 'UNIT KAWALAN', and 'UNIT ARITMETIK DAN LOGIK'. On the right is a blue box labeled 'INGATAN UTAMA (RAM)'. A legend on the left titled 'Petunjuk:' shows three bus types: 'bas data' (purple double-headed arrow), 'bas alamat' (red double-headed arrow), and 'bas kawalan' (orange double-headed arrow). Arrows indicate the flow of these buses between the I/O unit, CPU, and RAM.</p> </div> <p>Rajah 1.2.6 Seni bina Von Neuman dan sistem bas</p> <p>Sistem bas</p> <p>Bas dalaman Menghubungkan komponen-komponen dalaman pemproses dan membenarkan komponen tersebut bertukar-tukar maklumat.</p> <p>Bas luaran Menghubungkan pemproses dengan komponen luaran seperti ingatan dan peranti input dan output.</p> <p>Jenis Bas</p> <p>Bas data</p> <ul style="list-style-type: none"> - Memindahkan data antara pemproses, ingatan dan peranti

			input/output - Memindahkan data antara daftar, unit kawalan dan ALU
		Bas alamat	- Membawa alamat lokasi yang menempatkan data dan arahan dalam ingatan utama - Mengandungi alamat peranti input/output
		Bas kawalan	- Isyarat kawalan yang digunakan bagi mengawal operasi keseluruhan sistem
	Jenis daftar	PC (<i>Program Counter</i> /Pembilang atur cara)	
		ACC (<i>Accumulator</i> /Daftar Pengumpul)	
		MAR (<i>Memory Address Register</i> /Daftar Alamat Ingatan)	
		MDR (<i>Memory Data Register</i> /Daftar Data Ingatan)	
		IR (<i>Instruction Register</i> /Daftar Arahan)	
	Sistem Jam (clock system)	Lebih tinggi kelajuan hertz, lebih laju komputer tersebut. <i>Contoh: Intel Core i7 995X 3.6GHz, Intel Core Duo 2.0GHz</i>	
1.2.3	Pemprosesan linear	Pemprosesan bersiri. Memproses data dan arahan secara berjuran, mengikut kitaran mesin, iaitu <i>capai(IF)-nyahkod(ID)-laksana(EX)</i> dan diulangi sehingga selesai diproses.	
	Pemprosesan semasa	Pemprosesan tidak linear atau pemprosesan selari. Boleh melakukan pelbagai tugas dalam satu masa (<i>multitasking</i>) dengan melaksanakan beberapa arahan secara serentak menggunakan teknik talian paip (<i>pipelining</i>). Mempertingkatkan prestasi pemprosesan semasa. Membenarkan pemprosesan arahan secara bertindih (<i>overlapping</i>).	
	Beza linear dan semasa	Pemprosesan Linear	- Kurang efektif dalam pemprosesan data dan arahan panjang dan kompleks - Hanya boleh melakukan satu tugas pada satu masa - Lebih perlahan
		Pemprosesan Semasa	- Lebih efektif dalam pemprosesan data dan arahan panjang dan kompleks - Boleh melakukan banyak tugas pada satu masa - Lebih cepat
1.2.4	Bahasa	Bahasa aras	Berasaskan objek dan visual

pengaturcaraan	tinggi	Bahasa berstruktur																
	Bahasa aras rendah	Bahasa himpunan (<i>Assembly Language</i>)																
		Bahasa mesin																
Bahasa aras tinggi	Menggunakan pernyataan yang mudah difahami oleh pengatur cara kerana bahasanya hampir sama dengan bahasa tabii manusia.																	
Contoh bahasa aras tinggi	JAVA, Visual Basic, PHP, Python - objek dan visual C, PASCAL, COBOL, FORTRAN - berstruktur																	
Bahasa aras rendah	Pernyataan yang digunakan untuk membina arahan bagi komputer, dalam bentuk simbol yang ringkas.																	
Penggunaan simbol dalam bahasa himpunan	<p style="text-align: center;">Jadual 1.2.4 Penggunaan simbol dalam bahasa himpunan</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr style="background-color: #f4a460;"> <th>Simbol Arahan</th> <th>Maksud</th> </tr> </thead> <tbody> <tr> <td>ADD</td> <td>Tambah</td> </tr> <tr> <td>MUL atau MPY</td> <td>Darab</td> </tr> <tr> <td>SUB</td> <td>Tolak</td> </tr> <tr> <td>DIV</td> <td>Bahagi</td> </tr> <tr> <td>OR, NOT, AND</td> <td>Fungsi logik dalam Boolean</td> </tr> <tr> <td>LOAD</td> <td>Mencapai data dari ingatan</td> </tr> <tr> <td>STORE</td> <td>Menyimpan hasil ke ingatan</td> </tr> </tbody> </table>		Simbol Arahan	Maksud	ADD	Tambah	MUL atau MPY	Darab	SUB	Tolak	DIV	Bahagi	OR, NOT, AND	Fungsi logik dalam Boolean	LOAD	Mencapai data dari ingatan	STORE	Menyimpan hasil ke ingatan
Simbol Arahan	Maksud																	
ADD	Tambah																	
MUL atau MPY	Darab																	
SUB	Tolak																	
DIV	Bahagi																	
OR, NOT, AND	Fungsi logik dalam Boolean																	
LOAD	Mencapai data dari ingatan																	
STORE	Menyimpan hasil ke ingatan																	
Penterjemah (translator) bahasa pengaturcaraan	Menukarkan arahan dalam bentuk yang difahami manusia kepada bentuk yang difahami mesin, daripada bahasa pengaturcaraan aras tinggi dan rendah kepada bahasa mesin sebelum atur cara dilaksanakan. Dilakukan oleh pengkompil (<i>compilers</i>) atau pentafsir (<i>interpreter</i>).																	
Pengkompil (Compiler)	Menukarkan seluruh kod sumber atur cara ke kod mesin sebelum dilaksanakan.																	
Pentafsir (Interpreter)	Menukar kod sumber atur cara mengikut baris demi baris dengan satu baris dalam satu masa. Lebih interaktif berbanding pengkompil.																	
Penggunaan pentafsir dan pengkompil	<div style="text-align: center;"> <p>Dikompil (<i>Compile</i>) Ditafsir (<i>Interpret</i>)</p> <pre> graph LR A[Bahasa pengaturcaraan Java] -- "Dikompil (Compile)" --> B[bytecode] B -- "Ditafsir (Interpret)" --> C[Kod mesin] </pre> </div> <p>Rajah 1.2.11 Penggunaan pentafsir (<i>interpreter</i>) dan pengkompil (<i>compiler</i>)</p>																	

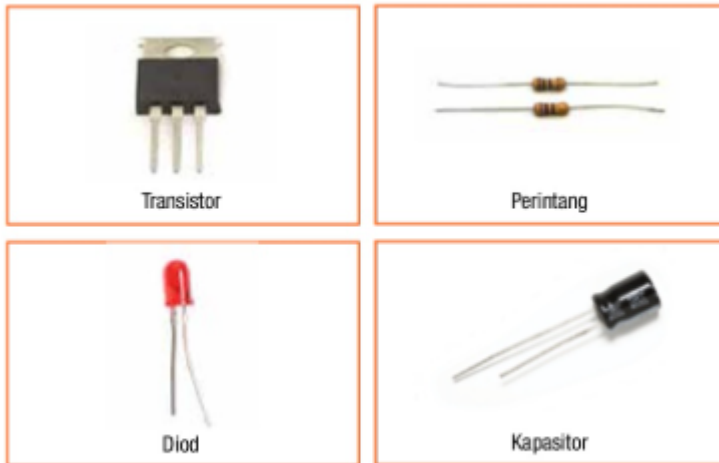
		<p style="text-align: center;">Dikompil (<i>Compile</i>)</p>  <p style="text-align: center;">Rajah 1.2.12 Penggunaan pengkompil (<i>compiler</i>)</p> <p style="text-align: center;">Ditafsir (<i>Interpret</i>)</p>  <p style="text-align: center;">Rajah 1.2.13 Penggunaan pentafsir (<i>interpreter</i>)</p>
1.2.5	Rumusan	 <p style="text-align: center;">Rajah 1.2.14 Rumusan pelaksanaan pemprosesan arahan dalam sistem komputer</p>
1.3.0	Get logik	Blok binaan asas litar bersepadu
1.3.1	Litar bersepadu (<i>integrated circuit, IC</i>)	Litar elektronik yang terdiri daripada gabungan komponen elektronik (transistor, perintang, diod dan kapasitor) yang perlu untuk melaksanakan pelbagai fungsi get logik.
	Gambaran binaan litar bersepadu	 <p style="text-align: center;">Rajah 1.3.1 Binaan litar bersepadu</p>

Evolusi litar bersepadu (IC)

Jadual 1.3.1 Evolusi litar bersepadu (IC)

Jenis	Tahun	Jumlah get logik per cip	Kegunaan
SSI: Penyepaduan skala kecil	1961 – 1966	<12	Get-get logik (DAN, ATAU, TAK DAN, TAK ATAU)
MSI: Penyepaduan skala pertengahan	1967 – 1971	12 – 99	Flip-flops Adder/counter Multiplexer & Demultiplexer
LSI: Penyepaduan skala besar	1972 – 1980	100 – 9999	Ingatan komputer Pemproses mikro 8 bit
VLSI: Penyepaduan skala sangat besar	1981 – 1990	10,000 – 99,999	Pemproses mikro 16 bit dan 32 bit
ULSI: Penyepaduan skala ultra besar	1991 – 1999	100,000 – 999,999	Pemproses mikro untuk pemprosesan grafik
GSI: Penyepaduan skala giga	2000 – kini	>1,000,000	Pemproses <i>Pentium IV</i>

Komponen dalam litar bersepadu



Rajah 1.3.2 Transistor, perintang, diod dan kapasitor

1.3.2

Get logik asas

Jadual 1.3.2 Nama dan simbol get logik asas

Nama get logik	Simbol get logik
DAN	
ATAU	
TAK	

Kombinasi Get Logik

Jadual 1.3.3 Nama dan simbol get logik bergabungan

Nama get logik	Kombinasi get logik	Simbol get logik
TAK DAN	Get DAN + Get TAK 	
TAK ATAU	Get ATAU + Get TAK 	
XATAU (Disebut sebagai eksklusif ATAU)	Get DAN + Get TAK + Get ATAU 	
XTAKATAU (Disebut sebagai eksklusif TAK-ATAU)	Get DAN + Get TAK + Get TAK ATAU 	

1.3.3

Jadual kebenaran

Untuk menyemak output yang dihasilkan daripada get-get logik atau daripada litar logik. Juga digunakan untuk buktikan teorem algebra Boolean.



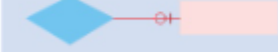



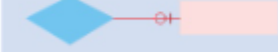



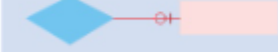

Rumusan get logik dan ungkapan Boolean serta jadual kebenaran

Jadual 1.3.5 Rumusan get-get logik beserta ungkapan Boolean dan jadual kebenaran

Nama get	Simbol get	Ungkapan Boolean	Jadual Kebenaran																		
DAN		$F = A \cdot B$ Dibaca: A DAN B $A \times B$	<table border="1"> <thead> <tr> <th colspan="2">INPUT</th> <th>OUTPUT</th> </tr> <tr> <th>A</th> <th>B</th> <th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	INPUT		OUTPUT	A	B	F	0	0	0	0	1	0	1	0	0	1	1	1
INPUT		OUTPUT																			
A	B	F																			
0	0	0																			
0	1	0																			
1	0	0																			
1	1	1																			
ATAU		$F = A + B$ Dibaca: A ATAU B	<table border="1"> <thead> <tr> <th colspan="2">INPUT</th> <th>OUTPUT</th> </tr> <tr> <th>A</th> <th>B</th> <th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	INPUT		OUTPUT	A	B	F	0	0	0	0	1	1	1	0	1	1	1	1
INPUT		OUTPUT																			
A	B	F																			
0	0	0																			
0	1	1																			
1	0	1																			
1	1	1																			


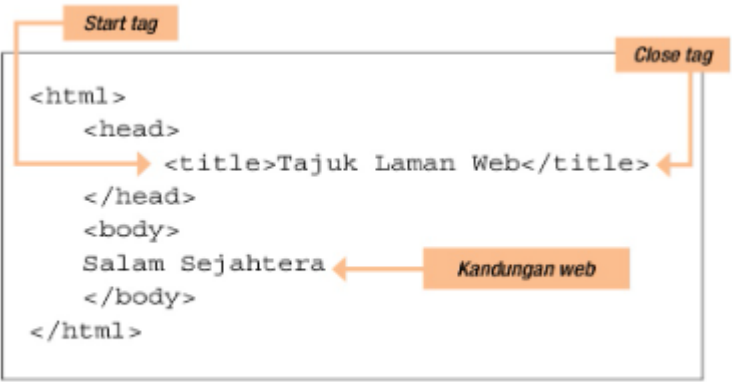

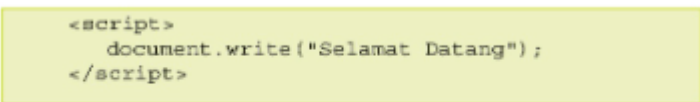
Nama get	Simbol get	Ungkapan Boolean	Jadual Kebenaran																								
TAK (penyongsang)		$F = \bar{A}$ F mempunyai nilai yang berlawanan dengan A	<table border="1"> <thead> <tr> <th colspan="2">INPUT</th> <th>OUTPUT</th> </tr> <tr> <th>A</th> <th>$F = \bar{A}$</th> </tr> </thead> <tbody> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </tbody> </table>	INPUT		OUTPUT	A	$F = \bar{A}$	0	1	1	0															
INPUT		OUTPUT																									
A	$F = \bar{A}$																										
0	1																										
1	0																										
TAK DAN		$F = \overline{A \cdot B}$ Dibaca: A DAN B bar	<table border="1"> <thead> <tr> <th colspan="2">INPUT</th> <th colspan="2">OUTPUT</th> </tr> <tr> <th>A</th> <th>B</th> <th>$A \cdot B$</th> <th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	INPUT		OUTPUT		A	B	$A \cdot B$	F	0	0	0	1	0	1	0	1	1	0	0	1	1	1	1	0
INPUT		OUTPUT																									
A	B	$A \cdot B$	F																								
0	0	0	1																								
0	1	0	1																								
1	0	0	1																								
1	1	1	0																								
TAK ATAU		$F = \overline{A + B}$ Dibaca: A ATAU B bar	<table border="1"> <thead> <tr> <th colspan="2">INPUT</th> <th colspan="2">OUTPUT</th> </tr> <tr> <th>A</th> <th>B</th> <th>$A + B$</th> <th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	INPUT		OUTPUT		A	B	$A + B$	F	0	0	0	1	0	1	1	0	1	0	1	0	1	1	1	0
INPUT		OUTPUT																									
A	B	$A + B$	F																								
0	0	0	1																								
0	1	1	0																								
1	0	1	0																								
1	1	1	0																								
XATAU Dibaca eksklusif ATAU		$F = A \oplus B$ Jika kedua-dua input bernilai sama, maka akan dihasilkan output '0'	<table border="1"> <thead> <tr> <th colspan="2">INPUT</th> <th>OUTPUT</th> </tr> <tr> <th>A</th> <th>B</th> <th>$F = A \oplus B$</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	INPUT		OUTPUT	A	B	$F = A \oplus B$	0	0	0	0	1	1	1	0	1	1	1	0						
INPUT		OUTPUT																									
A	B	$F = A \oplus B$																									
0	0	0																									
0	1	1																									
1	0	1																									
1	1	0																									
XTAKATAU Dibaca eksklusif TAK ATAU		$F = \overline{A \oplus B}$ Output dari XATAU diterbalikkan kerana tanda bar	<table border="1"> <thead> <tr> <th colspan="2">INPUT</th> <th colspan="2">OUTPUT</th> </tr> <tr> <th>A</th> <th>B</th> <th>$A \oplus B$</th> <th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> </tbody> </table>	INPUT		OUTPUT		A	B	$A \oplus B$	F	0	0	0	1	0	1	1	0	1	0	1	0	1	1	0	1
INPUT		OUTPUT																									
A	B	$A \oplus B$	F																								
0	0	0	1																								
0	1	1	0																								
1	0	1	0																								
1	1	0	1																								

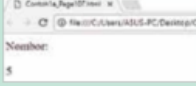

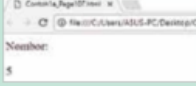

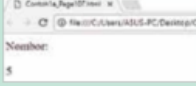


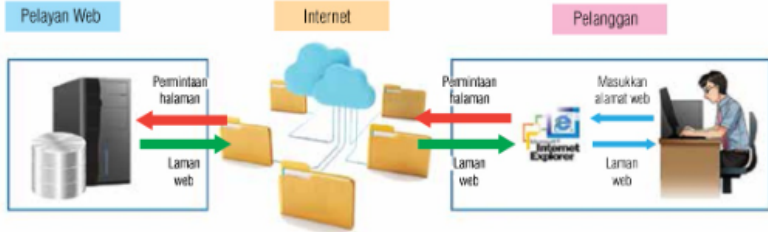
1.3.4 1.3.5 1.3.6 1.3.7	-	-	
2.1.1	<i>Pangkalan data</i>	Himpunan data berkaitan yang boleh dikongsi bersama oleh pelbagai kategori pengguna.	
	<i>Struktur rajah hubungan entiti</i>	Entiti	Objek atau konsep yang unik dan dapat dikenal pasti dalam persekitaran tertentu
		Entiti kuat	Boleh wujud tanpa bersandar pada entiti lain
		Entiti lemah	Kewujudan entiti lemah bersandar pada entiti kuat
		Atribut	Ciri-ciri yang menghuraikan sesuatu entiti; sifat yang memerihalkan sesuatu set entiti
		Hubungan	Penyatuan antara entiti
		Kunci primer	Atribut unik yang membezakan rekod dalam sebuah pangkalan data
		Kekardinalan	Kekardinalan minimum Kekardinalan maksimum
	<i>Atribut</i>	Atribut mudah	Tidak boleh dibahagikan kepada beberapa bahagian
		Atribut nilai gandaan	Mempunyai lebih daripada satu nilai
		Atribut terbitan	Terhasil berasaskan atribut lain
		Atribut kunci	Unik, membezakan setiap rekod dalam pangkalan data
		Atribut komposit	Atribut yang boleh dipecahkan kepada beberapa bahagian
Atribut bernilai tunggal		Mempunyai satu nilai sahaja	

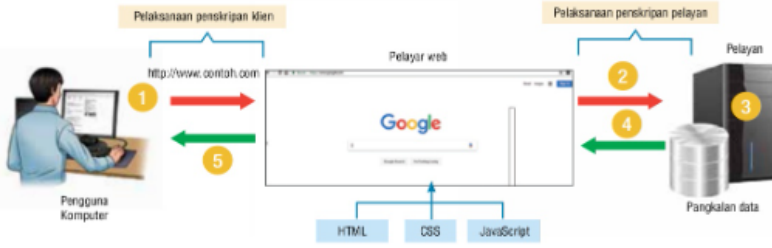
	<i>Kekardinalan</i>	<p style="text-align: center;">Jadual 2.1.3 Hubungan kekardinalan</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #0056b3; color: white;">Simbol</th> <th style="background-color: #0056b3; color: white;">Keterangan</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;"> Mandatori Satu  </td> <td>Minimum dan maksimum data adalah satu.</td> </tr> <tr> <td style="text-align: center;"> Mandatori Banyak  </td> <td>Minimum data adalah satu dan maksimum data adalah banyak.</td> </tr> <tr> <td style="text-align: center;"> Opsyenal Satu  </td> <td>Minimum data adalah kosong dan maksimum data adalah satu.</td> </tr> <tr> <td style="text-align: center;"> Opsyenal Banyak  </td> <td>Minimum data adalah kosong dan maksimum data adalah banyak.</td> </tr> </tbody> </table>		Simbol	Keterangan	Mandatori Satu 	Minimum dan maksimum data adalah satu.	Mandatori Banyak 	Minimum data adalah satu dan maksimum data adalah banyak.	Opsyenal Satu 	Minimum data adalah kosong dan maksimum data adalah satu.	Opsyenal Banyak 	Minimum data adalah kosong dan maksimum data adalah banyak.
Simbol	Keterangan												
Mandatori Satu 	Minimum dan maksimum data adalah satu.												
Mandatori Banyak 	Minimum data adalah satu dan maksimum data adalah banyak.												
Opsyenal Satu 	Minimum data adalah kosong dan maksimum data adalah satu.												
Opsyenal Banyak 	Minimum data adalah kosong dan maksimum data adalah banyak.												
2.1.2	<i>Model hubungan data</i>	Diperkenalkan oleh E. F. Codd pada tahun 1970											
	<i>Ciri-ciri model hubungan data</i>	Setiap hubungan (jadual) mempunyai nama yang berbeza daripada nama hubungan lain											
		Setiap data dalam hubungan mempunyai satu nilai sahaja											
		Setiap atribut (medan) mempunyai nama berbeza											
		Setiap rekod berbeza											
		Susunan atribut (medan) tidak mempunyai kepentingan											
Susunan rekod tidak mempunyai kepentingan													
<i>Skema hubungan</i>	<div style="display: flex; align-items: center; justify-content: center; gap: 20px;"> <div style="background-color: #fff9c4; padding: 5px; border: 1px solid #ccc;">Entiti ditukarkan menjadi nama hubungan</div> <div style="color: red; font-size: 24px;">→</div> <div style="background-color: #c8e6c9; padding: 5px; border: 1px solid #ccc;">Atribut akan menjadi nama medan</div> <div style="color: red; font-size: 24px;">→</div> <div style="background-color: #ffcdd2; padding: 5px; border: 1px solid #ccc;">Jenis hubungan dalam ERD akan menentukan pemilihan kunci asing</div> </div> <p style="text-align: center; color: red; font-weight: bold; font-size: small;">Rajah 2.1.9 Langkah-langkah untuk memetakan ERD kepada skema hubungan</p>												
<i>Jenis hubungan</i>	Satu-ke-satu (1:1)												
	Satu-ke-banyak (1:M)												
	Banyak-ke-banyak (M:N)												
2.1.3	<i>Bahasa arahan SQL</i>	<i>Data Definition Language (DDL) / Bahasa Takrifan Data</i>	Mencipta dan mengubah struktur pangkalan data - CREATE, ALTER, DROP										

	<p><i>Data Manipulation Language (DML) / Bahasa Pengolahan Data</i></p> <p>Memanipulasi data dalam jadual</p> <ul style="list-style-type: none"> - SELECT, INSERT, UPDATE, DELETE 																						
	<p><i>Data Control Language (DCL) / Bahasa Kawalan Data</i></p> <p>Mengawal capaian data dalam pangkalan data</p> <ul style="list-style-type: none"> - GRANT, REVOKE 																						
Jenis data dalam MySQL	<p style="text-align: center;">Jadual 2.1.11 Jenis-jenis data dalam MySQL</p> <table border="1"> <thead> <tr> <th>Jenis Data</th> <th>Keterangan</th> </tr> </thead> <tbody> <tr> <td>CHAR</td> <td>String yang tetap antara 0 hingga 255 patah perkataan. CHAR digunakan untuk lajur yang mengandungi data yang tetap dan seragam. Contoh: Jantina</td> </tr> <tr> <td>VARCHAR</td> <td>String yang boleh diubah antara 0 hingga 255 patah perkataan. VARCHAR digunakan untuk lajur yang mengandungi data yang tidak seragam. Contoh: Nama, Alamat</td> </tr> <tr> <td>TINYTEXT</td> <td>String dengan panjang maksimum 255 patah perkataan.</td> </tr> <tr> <td>TEXT</td> <td>String dengan panjang maksimum 65,535 patah perkataan.</td> </tr> <tr> <td>INT</td> <td>Nombor dalam julat -2,147,483,648 hingga 2,147,483,647 atau nombor positif 0 hingga 4,294,967,295.</td> </tr> <tr> <td>FLOAT</td> <td>Nombor kecil yang mempunyai titik perpuluhan.</td> </tr> <tr> <td>DOUBLE</td> <td>Nombor besar yang mempunyai titik perpuluhan.</td> </tr> <tr> <td>DATE</td> <td>Tarikh dengan format YYYY-MM-DD.</td> </tr> <tr> <td>DATETIME</td> <td>Tarikh dan waktu dengan format YYYY-MM-DD/HH:MM:SS.</td> </tr> <tr> <td>TIME</td> <td>Waktu dengan format HH:MM:SS</td> </tr> </tbody> </table>	Jenis Data	Keterangan	CHAR	String yang tetap antara 0 hingga 255 patah perkataan. CHAR digunakan untuk lajur yang mengandungi data yang tetap dan seragam. Contoh: Jantina	VARCHAR	String yang boleh diubah antara 0 hingga 255 patah perkataan. VARCHAR digunakan untuk lajur yang mengandungi data yang tidak seragam. Contoh: Nama, Alamat	TINYTEXT	String dengan panjang maksimum 255 patah perkataan.	TEXT	String dengan panjang maksimum 65,535 patah perkataan.	INT	Nombor dalam julat -2,147,483,648 hingga 2,147,483,647 atau nombor positif 0 hingga 4,294,967,295.	FLOAT	Nombor kecil yang mempunyai titik perpuluhan.	DOUBLE	Nombor besar yang mempunyai titik perpuluhan.	DATE	Tarikh dengan format YYYY-MM-DD.	DATETIME	Tarikh dan waktu dengan format YYYY-MM-DD/HH:MM:SS.	TIME	Waktu dengan format HH:MM:SS
Jenis Data	Keterangan																						
CHAR	String yang tetap antara 0 hingga 255 patah perkataan. CHAR digunakan untuk lajur yang mengandungi data yang tetap dan seragam. Contoh: Jantina																						
VARCHAR	String yang boleh diubah antara 0 hingga 255 patah perkataan. VARCHAR digunakan untuk lajur yang mengandungi data yang tidak seragam. Contoh: Nama, Alamat																						
TINYTEXT	String dengan panjang maksimum 255 patah perkataan.																						
TEXT	String dengan panjang maksimum 65,535 patah perkataan.																						
INT	Nombor dalam julat -2,147,483,648 hingga 2,147,483,647 atau nombor positif 0 hingga 4,294,967,295.																						
FLOAT	Nombor kecil yang mempunyai titik perpuluhan.																						
DOUBLE	Nombor besar yang mempunyai titik perpuluhan.																						
DATE	Tarikh dengan format YYYY-MM-DD.																						
DATETIME	Tarikh dan waktu dengan format YYYY-MM-DD/HH:MM:SS.																						
TIME	Waktu dengan format HH:MM:SS																						
CREATE table	<p>CREATE TABLE nama_jadual { Nama_medan1 jenis data (saiz) PRIMARY KEY Nama_medan2 jenis data (saiz) NOT NULL Nama_medan3 jenis data (saiz), ... }</p>																						
ALTER table	<p>ALTER TABLE nama_jadual Pilihan perubahan;</p> <p>ALTER TABLE nama_jadual ADD Nama_medan jenis data (saiz);</p> <p>ALTER TABLE nama_jadual MODIFY Nama_medan Jenis_data_baharu (saiz baharu);</p> <p>ALTER TABLE nama_jadual CHANGE Nama_medan Nama_medan_baharu Jenis_data_baharu (saiz baharu);</p> <p>ALTER TABLE nama_jadual DROP Nama_medan;</p>																						
Hapus jadual	<p>DROP TABLE nama_jadual</p>																						

2.1.4	Sisip/Tambah	INSERT INTO nama_jadual (medan1, medan2, ... medanN) VALUES (nilai1, nilai2, ... nilaiN);													
	Padam	DELETE FROM nama_jadual WHERE nama_medan = nilai;													
	Kemaskini	UPDATE nama_jadual SET nama_medan = nilai_baharu WHERE nama_medan = nilai_tertentu;													
2.1.5	Struktur asas pernyataan SQL	SELECT	Senaraikan nama medan yang ingin dipaparkan												
		FROM	Nama jadual yang hendak dicapai data												
		WHERE	Menentukan rekod tertentu yang ingin dipaparkan												
	Format	SELECT nama_medan FROM nama_jadual WHERE syarat1 AND syarat2 AND syarat...													
	Group by	SELECT nama_medan FROM nama_jadual GROUP BY nama_medan;													
	Order by	SELECT nama_medan FROM nama_jadual ORDER BY nama_medan;													
2.1.6	Fungsi Agregat	<p style="text-align: center;">Jadual 2.1.14 Fungsi Agregat</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #0070C0; color: white;">Fungsi Agregat</th> <th style="background-color: #0070C0; color: white;">Keterangan</th> </tr> </thead> <tbody> <tr> <td>SUM</td> <td>Mengembalikan jumlah nilai medan tertentu</td> </tr> <tr> <td>MIN</td> <td>Mengembalikan nilai terendah medan tertentu</td> </tr> <tr> <td>MAX</td> <td>Mengembalikan nilai tertinggi medan tertentu</td> </tr> <tr> <td>AVG</td> <td>Mengembalikan nilai purata medan tertentu</td> </tr> <tr> <td>COUNT</td> <td>Mengembalikan jumlah bilangan rekod</td> </tr> </tbody> </table>		Fungsi Agregat	Keterangan	SUM	Mengembalikan jumlah nilai medan tertentu	MIN	Mengembalikan nilai terendah medan tertentu	MAX	Mengembalikan nilai tertinggi medan tertentu	AVG	Mengembalikan nilai purata medan tertentu	COUNT	Mengembalikan jumlah bilangan rekod
	Fungsi Agregat	Keterangan													
	SUM	Mengembalikan jumlah nilai medan tertentu													
	MIN	Mengembalikan nilai terendah medan tertentu													
	MAX	Mengembalikan nilai tertinggi medan tertentu													
	AVG	Mengembalikan nilai purata medan tertentu													
COUNT	Mengembalikan jumlah bilangan rekod														
SUM	SELECT SUM (nama_medan) FROM nama_jadual;														
MIN	SELECT MIN (nama_medan) FROM nama_jadual;														
MAX	SELECT MAX (nama_medan) FROM nama_jadual;														
AVG	SELECT AVG (nama_medan) FROM nama_jadual;														
COUNT	SELECT COUNT (nama_medan) FROM nama_jadual;														

	<i>HAVING</i>	SELECT nama_medan, fungsi1, fungsi2, ... fungsiN FROM nama_jadual GROUP BY nama_medan HAVING (fungsi);
2.1.7	-	-
3.1.1	<i>HTML</i>	<p><i>Hypertext Markup Language</i>; bahasa <i>markup standard</i> yang digunakan oleh pereka web professional bagi mereka bentuk laman web.</p>  <p style="text-align: center;">Rajah 3.1.2 Contoh atribut nama dan atribut nilai</p>  <p style="text-align: center;">Rajah 3.1.1(a) Contoh tag HTML berpasangan</p>
	<i>CSS</i>	<p>Cascading Style Sheet; tambahan (extension) kepada HTML. Membenarkan perubahan kepada gaya sesuatu halaman web dan menerangkan persembahan bagi sesuatu dokumen HTML.</p>  <p style="text-align: center;">Rajah 3.1.4 Contoh HTML dengan CSS dan output</p>
	<i>JavaScript</i>	<p>Menyediakan keupayaan pengaturcaraan untuk memproses sesuatu tugas berdasarkan permohonan yang diterima daripada klien</p>  <p style="text-align: center;">Rajah 3.1.5 Tag <i>JavaScript</i></p>

	<p><i>Explicit embedding</i></p>	<table border="1"> <thead> <tr> <th data-bbox="608 224 879 246">HTML (tanpa skrip)</th> <th data-bbox="879 224 1362 246">HTML (berserta skrip)</th> </tr> </thead> <tbody> <tr> <td data-bbox="608 246 879 510"> <pre><html> <body> <p>Nombor: </p> 5 </body> </html></pre>  </td> <td data-bbox="879 246 1362 510"> <pre><html> <body> <p>Nombor: </p> <script> document.write("5"); </script> </body> </html></pre> <div data-bbox="1182 304 1378 367" style="border: 1px solid red; padding: 2px;"> Kod JavaScript untuk mencetak "5" dalam dokumen HTML (laman web) </div>  </td> </tr> </tbody> </table> <p style="text-align: center;">Rajah 3.1.6 Alur cara HTML, JavaScript dan contoh output (<i>Explicit Embedding</i>)</p>	HTML (tanpa skrip)	HTML (berserta skrip)	<pre><html> <body> <p>Nombor: </p> 5 </body> </html></pre> 	<pre><html> <body> <p>Nombor: </p> <script> document.write("5"); </script> </body> </html></pre> <div data-bbox="1182 304 1378 367" style="border: 1px solid red; padding: 2px;"> Kod JavaScript untuk mencetak "5" dalam dokumen HTML (laman web) </div> 
HTML (tanpa skrip)	HTML (berserta skrip)					
<pre><html> <body> <p>Nombor: </p> 5 </body> </html></pre> 	<pre><html> <body> <p>Nombor: </p> <script> document.write("5"); </script> </body> </html></pre> <div data-bbox="1182 304 1378 367" style="border: 1px solid red; padding: 2px;"> Kod JavaScript untuk mencetak "5" dalam dokumen HTML (laman web) </div> 					
	<p><i>Implicit embedding</i></p>	<table border="1"> <thead> <tr> <th data-bbox="608 589 879 611">Fail "myScript.js" (skrip)</th> <th data-bbox="879 589 1362 611">HTML</th> </tr> </thead> <tbody> <tr> <td data-bbox="608 611 879 887"> <pre>function f1() { document.write("5"); }</pre> </td> <td data-bbox="879 611 1362 887"> <pre><html> <head> <title>Script Luaran</title> <script src="myScript.js"> </script> </head> <body> <p>Nombor: </p> <script type="text/javascript"> f1(); </script> </body> </html></pre> <div data-bbox="1214 656 1337 719" style="border: 1px solid red; padding: 2px;"> Rujuk kod JavaScript dalam fail myScript.js </div> <div data-bbox="1257 752 1362 853" style="border: 1px solid red; padding: 2px;"> Panggil dan laksana kod subatur cara f1() dalam fail myScript.js </div> </td> </tr> </tbody> </table> <div data-bbox="608 920 1378 1133" style="border: 1px solid green; padding: 5px;"> <p style="text-align: center;">Contoh output</p>  </div> <p style="text-align: center;">Rajah 3.1.7 Alur cara HTML, JavaScript dan contoh output (<i>Implicit Embedding</i>)</p>	Fail "myScript.js" (skrip)	HTML	<pre>function f1() { document.write("5"); }</pre>	<pre><html> <head> <title>Script Luaran</title> <script src="myScript.js"> </script> </head> <body> <p>Nombor: </p> <script type="text/javascript"> f1(); </script> </body> </html></pre> <div data-bbox="1214 656 1337 719" style="border: 1px solid red; padding: 2px;"> Rujuk kod JavaScript dalam fail myScript.js </div> <div data-bbox="1257 752 1362 853" style="border: 1px solid red; padding: 2px;"> Panggil dan laksana kod subatur cara f1() dalam fail myScript.js </div>
Fail "myScript.js" (skrip)	HTML					
<pre>function f1() { document.write("5"); }</pre>	<pre><html> <head> <title>Script Luaran</title> <script src="myScript.js"> </script> </head> <body> <p>Nombor: </p> <script type="text/javascript"> f1(); </script> </body> </html></pre> <div data-bbox="1214 656 1337 719" style="border: 1px solid red; padding: 2px;"> Rujuk kod JavaScript dalam fail myScript.js </div> <div data-bbox="1257 752 1362 853" style="border: 1px solid red; padding: 2px;"> Panggil dan laksana kod subatur cara f1() dalam fail myScript.js </div>					
	<p><i>Konsep aplikasi Pelanggan/ Pelayan (Client/Server)</i></p>	 <p style="text-align: center;">Rajah 3.1.8 Konsep aplikasi Pelanggan/Pelayan (<i>Client/Server</i>)</p>				
	<p><i>Pelanggan/client</i></p>	<p>Pelayar (<i>browser</i>) yang digunakan oleh pengguna bagi melayari web dengan menggunakan sama ada peranti mudah alih, komputer riba atau komputer meja.</p>				
	<p><i>Pelayan/server</i></p>	<p>Berada di mana-mana di seluruh dunia yang bertindak untuk memproses permintaan atau menghantar laman web yang diminta oleh klien.</p>				
	<p><i>Laman web statik</i></p>	<p>Kandungan paparan mengikut tag HTML yang digunakan</p> <p>Dibina menggunakan HTML sahaja</p> <p>Contoh: Laman web yang memaparkan latar belakang organisasi</p>				

	<p><i>Laman web dinamik</i></p>	<p>Kandungan paparan boleh dipanggil dari pangkalan data atau mengikut tindakan pengguna</p> <p>Dibina menggunakan penskripan pelayan</p> <p>Contoh: Aktiviti akan datang yang dipaparkan di laman utama web yang berubah mengikut tarikh kalendar</p>
	<p><i>Pelaksanaan penskripan klien dan penskripan pelayan</i></p>	 <p>Rajah 3.1.9 Pelaksanaan penskripan klien dan penskripan pelayan</p>
	<p><i>Penerangan gambar rajah</i></p>	<ol style="list-style-type: none"> 1. Pengguna komputer menaip URL di pelayar web 2. Pelayar web memajukan URL tersebut kepada pelayan web untuk diproses 3. Pelayan web akan memproses permintaan pengguna komputer dengan melaksanakan bahasa penskripan pelayan 4. Pelayan mengembalikan hasil pelaksanaan penskripan pelayan kepada pelayar web dengan menghantar laman web yang spesifik berdasarkan URL yang ditaip oleh pengguna komputer 5. Pelayar web akan menterjemahkan dan melaksanakan kandungan laman HTML termasuk mana-mana penskripan klien yang ada pada web tersebut untuk dibaca oleh pengguna komputer
	<p><i>Bahasa penskripan klien</i></p>	<p>Program komputer dalam web yang dilaksanakan pada komputer klien. Direka bagi memanipulasikan dan memaparkan kandungan laman web yang telah dimuat turun dalam komputer klien.</p>
	<p><i>Kelebihan bahasa penskripan</i></p>	<p>Laman web menjadi lebih interaktif</p> <p>Pembangun web berkeupayaan menentukan reka bentuk dan ciri-ciri web</p> <p>Meningkatkan kebolegunaan web bagi pelayar yang menyokong pelbagai jenis bahasa penskripan klien</p> <p>Pelaksanaan tugas yang pantas tanpa penglibatan komputer pelayan</p>
<p>3.1.2</p>	<p><i>Pengisihan</i></p>	<ul style="list-style-type: none"> - Isihan buih (<i>bubble sort</i>) - Isihan pilih (<i>selection sort</i>)

	<p><i>Cara pengisihan</i></p>	<p>Turutan menaik (1, 2, 3 ...)</p> <hr/> <p>Turutan menurun (3, 2, 1 ...)</p>
	<p><i>Isihan buih (bubble sort)</i></p>	<div data-bbox="836 344 1171 465" style="text-align: center;"> </div> <p style="text-align: center;">Rajah 3.1.13 Proses isihan buih</p> <p>Terdapat dua proses perbandingan yang boleh dilaksanakan untuk menukar kedudukan, iaitu:</p> <ol style="list-style-type: none"> 1 Banding unsur pertama sama ada lebih kecil daripada unsur kedua (untuk turutan menurun) 2 Banding unsur pertama sama ada lebih besar daripada unsur kedua (untuk turutan menaik)
	<p><i>Atur cara dan output isihan buih (turutan menaik)</i></p>	<div data-bbox="612 645 1362 1160" style="border: 1px solid black; padding: 10px;"> <pre> <html> <body> <script> var no = [5,1,4,3,2]; var bilNombor = no.length, i, j; var sementara = no[0]; for(i=0; i<bilNombor-1; i++) { for(j=0; j<bilNombor-i-1; j++) { if (no[j] > no[j+1]) { semantara = no[j]; no[j] = no[j+1]; no[j+1] = sementara; } } } document.write (no); </script> </body> </html> </pre> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div data-bbox="632 763 756 846" style="border: 1px solid red; padding: 2px; font-size: small;">no.length untuk mengira bilangan nombor dalam senarai</div> <div data-bbox="1193 696 1318 741" style="border: 1px solid red; padding: 2px; font-size: small;">Dapatkan senarai yang belum diisih</div> </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div data-bbox="1171 837 1295 860" style="border: 1px solid red; padding: 2px; font-size: small;">Bandingkan unsur</div> <div data-bbox="1203 891 1327 913" style="border: 1px solid red; padding: 2px; font-size: small;">Tukar kedudukan</div> </div> <div style="text-align: center; margin-top: 10px;"> </div> </div> <p style="text-align: center;">Rajah 3.1.14 Atur cara dan output bagi isihan buih (Turutan menaik)</p>

		<pre> graph TD Mula([Mula]) --> A[no = 5, 1, 4, 3, 2] A --> B["bilNombor = no.length sementara = no[0] i = 0"] B --> C{"i < bilNombor-1?"} C -- BENAR --> D[j = 0] D --> E{"j < bilNombor-i-1?"} E -- PALSU --> F[i = i + 1] F --> C E -- BENAR --> G{"no[j] > no[j+1]?"} G -- PALSU --> H[j = j + 1] H --> E G -- BENAR --> I["sementara = no[j] no[j] = no[j+1] no[j+1] = sementara"] I --> H I --> J[/Paparkan no/] J --> K([Tamat]) </pre> <p>Rajah 3.1.15 Carta alir bagi isihan buih (turutan menaik)</p>
<p><i>Kelebihan isihan buih</i></p>		<p>Mudah dibina, mudah difahami</p> <p>Sesuai digunakan apabila bilangan dalam unsur adalah kecil kerana perbandingan dibuat satu persatu hingga akhir</p> <p>Sesuai digunakan untuk pembelajaran dan bukan untuk aplikasi sebenar</p>

Atur cara
JavaScript dan
output bagi isihan
buih (turutan
menurun)

```
<html>
  <body>
    <script>
      var kodBuku = [110, 102, 105, 101, 100];
      var bilKod = kodBuku.length, i, j;
      var sementara = kodBuku[0];

      document.write("Senarai kod buku sebelum isih: " + kodBuku);

      for(i=0; i<bilKod-1; i++)
      {
        for (j=0; j<bilKod-i-1; j++)
        {
          if (kodBuku[j] < kodBuku[j+1])
          {
            sementara = kodBuku[j];
            kodBuku[j] = kodBuku[j+1];
            kodBuku[j+1] = sementara;
          }
        }
      }

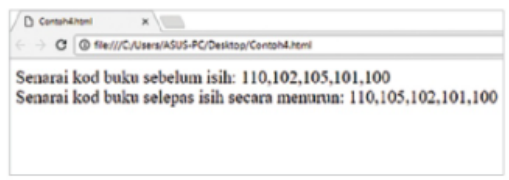
      document.write ("<br>Senarai kod buku selepas isih secara
      menurun: " + kodBuku);
    </script>
  </body>
</html>
```

kodBuku.length
untuk mengira
bilangan
kodBuku dalam
senarai

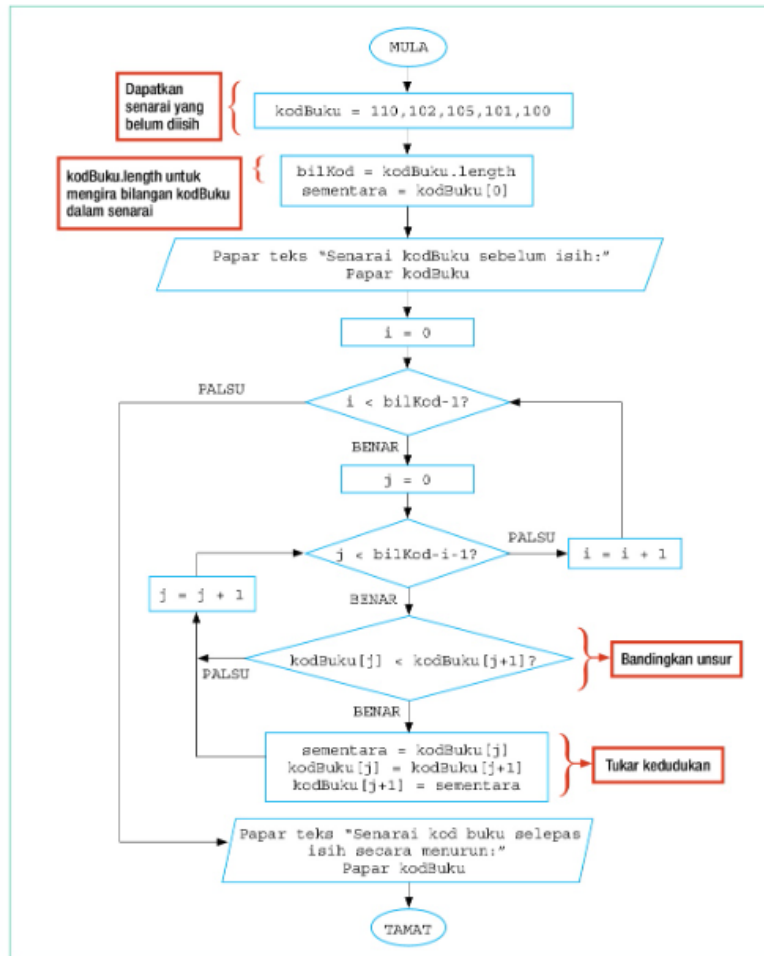
Dapatkan senarai
yang belum diisih

Bandingkan unsur

Tukar kedudukan



Rajah 3.1.16 Atur cara JavaScript dan output bagi isihan buih (turutan menurun)



Rajah 3.1.17 Carta alir bagi isihan buih (turutan menurun)

*Isihan pilih
(selection sort)*



Rajah 3.1.18 Proses isihan pilih

Terdapat dua proses perbandingan yang boleh dilaksanakan untuk menukar kedudukan, iaitu:

- 1 Banding unsur dirujuk sama ada **lebih besar** daripada unsur terkecil (untuk turutan menaik)
- 2 Banding unsur dirujuk sama ada **lebih kecil** daripada unsur terbesar (untuk turutan menurun)

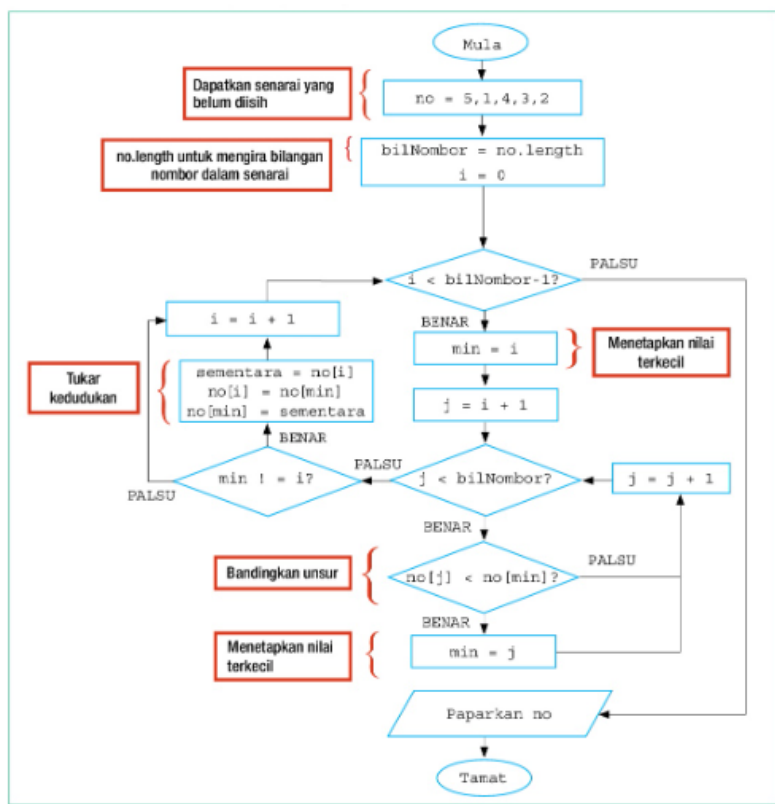
Atur cara JavaScript dan output bagi isihan pilih (turut menaik)

```

<html>
<body>
<script>
var no = [5,1,4,3,2], i, j, min, sementara;
var bilNombor = no.length;
for (i = 0; i < bilNombor-1; i++)
{
    min = i;
    for (j = i + 1; j < bilNombor; j++)
    {
        if (no[j] < no[min])
        {
            min = j;
        }
    }
    if (min != i)
    {
        sementara = no[i];
        no[i] = no[min];
        no[min] = sementara;
    }
}
document.write(no);
</script>
</body>
</html>

```

Carta alir isihan pilih (turut menaik)

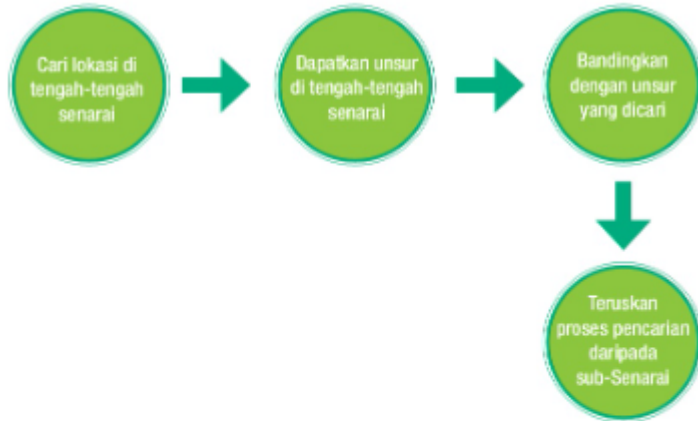


Rajah 3.1.20 Carta alir isihan pilih (turut menaik)

Kelebihan isihan pilih

Sesuai apabila bilangan unsur adalah kecil

Carian perduaan (Binary search)



Rajah 3.1.23 Proses carian perduaan

Terdapat tiga proses perbandingan yang boleh dilaksanakan, iaitu:

- 1 Bandingkan unsur di tengah sama ada **sama dengan** unsur yang dicari.
- 2 Bandingkan unsur di tengah sama ada **lebih kecil daripada** unsur yang dicari.
- 3 Bandingkan unsur di tengah sama ada **lebih besar daripada** unsur dicari.

Atur cara JavaScript carian perduaan

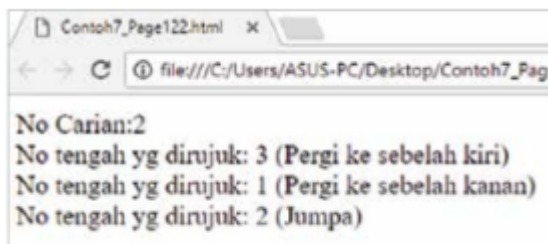
```

<html>
<body>
<script>
var no = [1,2,3,4,5];
var noCarian = 2;
var indeksAwal = 0, indeksAkhir = no.length - 1,
indeksTengah = Math.floor((indeksAkhir + indeksAwal) / 2);
document.write("No Carian:", noCarian);
while(no[indeksTengah] != noCarian && indeksAwal < indeksAkhir)
{
if (no[indeksTengah] > noCarian)
{
indeksAkhir = indeksTengah - 1;
document.write("<br>No tengah yg dirujuk: ",
no[indeksTengah]," (Pergi ke sebelah kiri) ");
}
else
{
if (no[indeksTengah] < noCarian)
{
indeksAwal = indeksTengah + 1;
document.write("<br>No tengah yg dirujuk: ",
no[indeksTengah]," (Pergi ke sebelah kanan) ");
}
}
indeksTengah = Math.floor((indeksAkhir + indeksAwal)/2);
}
if (no[indeksTengah] == noCarian)
{
document.write("<br>No tengah yg dirujuk:',no[indeksTengah],' (Jumpa) ');
}
}
</script>
</body>
</html>
  
```

Annotations in the code block:

- Red box: "Nombor yang telah diisih" points to the array `no = [1,2,3,4,5];`
- Red box: "Selagi nombor yang dirujuk TIDAK SAMA dengan nombor yang dicari" points to the `while` loop condition.
- Red box: "Kiraan semula indeksTengah" points to the `indeksTengah = Math.floor((indeksAkhir + indeksAwal)/2);` line.
- Red box: "Jika nombor yang dirujuk SAMA dengan nombor yang dicari" points to the `if (no[indeksTengah] == noCarian)` block.

Output



<p><i>Carta alir carian perduaan</i></p>	<p style="text-align: center;">Rajah 3.1.25 Carta alir (carian perduaan)</p>
<p><i>Nombor genap</i></p>	<p>Nombor dibahagi 2 dan baki mesti 0 Operator % digunakan untuk operasi mendapatkan baki</p>
<p></p>	<p>$no[i] \% 2 == 0$</p>
<p><i>Nombor ganjil</i></p>	<p>Nombor dibahagi 2 dan baki mesti 1 Operator % digunakan untuk operasi mendapatkan baki</p>
<p></p>	<p>$no[i] \% 2 == 1$</p>
<p><i>Mendapatkan nilai maksimum</i></p>	<p style="text-align: center;">Rajah 3.1.28 Proses mencari nilai maksimum</p>

Atur cara
dapatkan nilai
maksimum

```
<html>
<body>
<script>
var no = [5,1,4,12,8], noMaksimum = -1,
i,bilNombor = no.length;
for(i=0; i<bilNombor; i++)
{
if (no[i] > noMaksimum)
{
noMaksimum = no[i];
}
}
document.write(noMaksimum);
</script>
</body>
</html>
```

Beri nilai awalan kepada noMaksimum.
Dianggap -1 adalah noMaksimum (pilih nombor paling rendah)

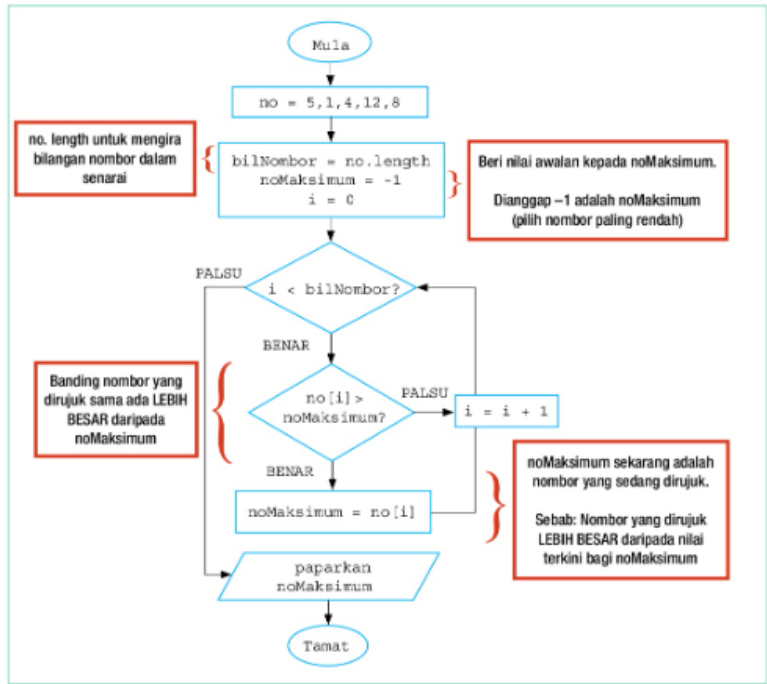
no.length untuk mengira bilangan nombor dalam senarai

Banding nombor yang dirujuk LEBIH BESAR daripada noMaksimum

noMaksimum sekarang adalah nombor yang sedang dirujuk. Sebab: Nombor yang dirujuk LEBIH BESAR daripada nilai terkini bagi noMaksimum



Carta alir
dapatkan nilai
maksimum

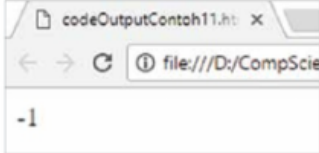
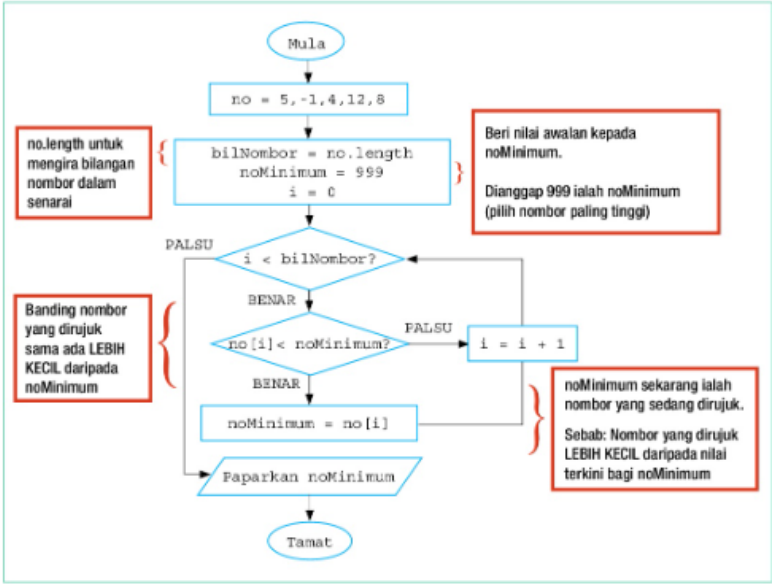


Rajah 3.1.30 Carta alir (mendapatkan nilai maksimum)

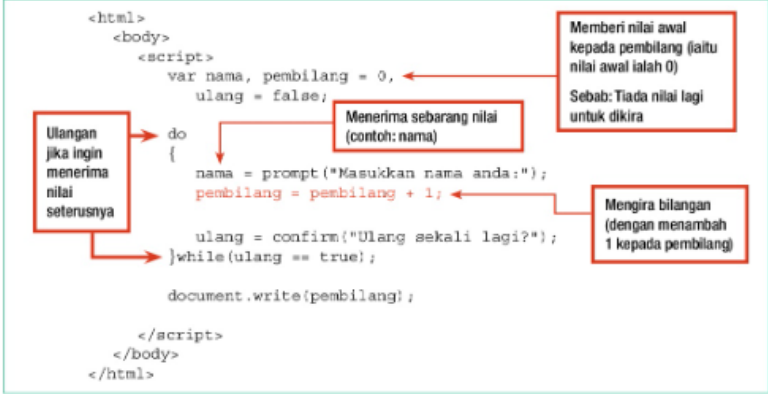
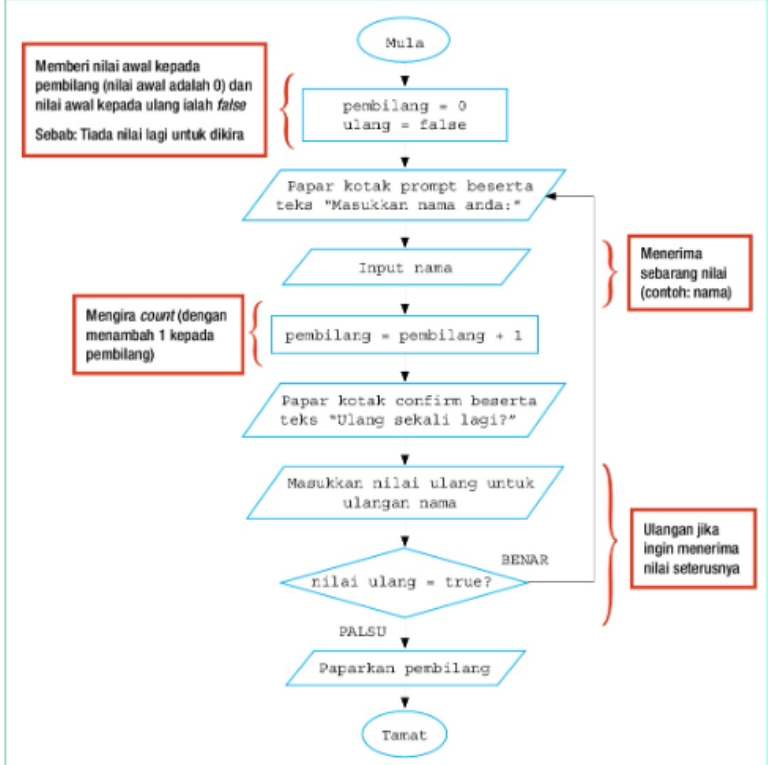
Mendapatkan
nilai minimum



Rajah 3.1.33 Proses mencari nilai minimum

<p><i>Atur cara dapatkan nilai minimum</i></p>	<p><i>Atur cara dapatkan nilai minimum</i></p>	<pre> <html> <body> <script> var no = [5, -1, 4, 12, 8], noMinimum = 999, i, bilNombor = no.length; for (i = 0; i < bilNombor; i++) { if (no[i] < noMinimum) { noMinimum = no[i]; } } document.write(noMinimum); </script> </body> </html> </pre> <p>Beri nilai awalan kepada noMinimum. Dianggap 999 adalah noMinimum (pilih nombor paling tinggi)</p> <p>no.length untuk mengira bilangan nombor dalam senarai</p> <p>Banding nombor yang dirujuk sama ada LEBIH KECIL daripada noMinimum</p> <p>noMinimum sekarang ialah nombor yang sedang dirujuk. Sebab: Nombor yang dirujuk LEBIH KECIL daripada nilai terkini bagi noMinimum</p> 
<p><i>Carta alir dapatkan nilai minimum</i></p>	<p><i>Carta alir dapatkan nilai minimum</i></p>	 <p>no.length untuk mengira bilangan nombor dalam senarai</p> <p>Beri nilai awalan kepada noMinimum. Dianggap 999 ialah noMinimum (pilih nombor paling tinggi)</p> <p>Banding nombor yang dirujuk sama ada LEBIH KECIL daripada noMinimum</p> <p>noMinimum sekarang ialah nombor yang sedang dirujuk. Sebab: Nombor yang dirujuk LEBIH KECIL daripada nilai terkini bagi noMinimum</p> <p>Rajah 3.1.35 Carta alir (mendapatkan nilai minimum)</p>
<p><i>Mendapatkan nilai Mean</i></p>	<p><i>Mendapatkan nilai Mean</i></p>	<p><i>Mean</i>: purata nilai dalam suatu senarai nilai</p> <ol style="list-style-type: none"> 1. Jumlahkan nilai-nilai dalam senarai nilai tersebut 2. Dapatkan bilangan nilai yang terdapat dalam senarai 3. Nilai mean dikira dengan membahagikan nilai jumlah dengan bilangan nilai yang terdapat dalam senarai <p><i>Formula</i>:</p> <p><i>Mean = Jumlah/Bilangan</i></p>

<p>Atur cara mendapatkan nilai mean</p>	<pre> <html> <body> <script> var no, pembilang=0, ulang=false, jumlah=0, purata=0; do { no = parseInt (prompt("Masukkan satu nombor:")); pembilang = pembilang + 1; jumlah = jumlah + no; ulang = confirm("Ulang sekali lagi?"); }while(ulang == true); purata = jumlah/pembilang; document.write (purata) ; </script> </body> </html> </pre> <p>Memberi nilai awal kepada pembilang, jumlah dan purata (iaitu nilai awal ialah 0) Sebab: Tiada nilai lagi untuk dikira</p> <p>Mengira bilangan (dengan menambah 1 kepada pembilang) dan mengira jumlah (dengan menambah nilai kepada jumlah)</p> <p>Mengira purata nilai. Formula: jumlah/pembilang</p> <p>nombor dimasukkan oleh pengguna (prompt) dan pastikan bernilai integer (parseInt)</p> <p>Rajah 3.1.38(a) Atur cara JavaScript (mendapatkan nilai mean)</p>
<p>Carta alir mendapatkan nilai mean</p>	<p>Memberi nilai awal kepada pembilang, jumlah dan purata.</p> <p>Mengira bilangan (dengan menambah 1 kepada pembilang) dan mengira jumlah (dengan menambah nilai kepada jumlah)</p> <p>Ulangan jika ingin menerima nilai seterusnya</p> <p>Mengira purata nilai. Formula: Jumlah/pembilang</p> <p>Rajah 3.1.39 Carta alir (mendapatkan nilai mean)</p>
<p>Mendapatkan nilai Count</p>	<p>Count: mengira bilangan nilai yang terdapat dalam suatu senarai</p> <ol style="list-style-type: none"> 1. Menggunakan suatu perwakilan sebagai pembilang dan memberikan nilai awalan pembilang sebagai 1 2. Nilai count dikira dengan menambah pembilang dengan 1 bagi setiap nilai yang dicapai atau dirujuk 3. Memaparkan nilai akhir pembilang <p>Formula: $\text{pembilang} = \text{pembilang} + 1$</p>

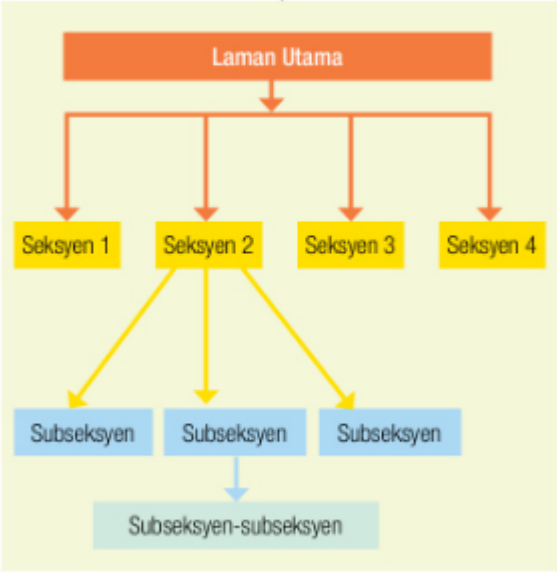
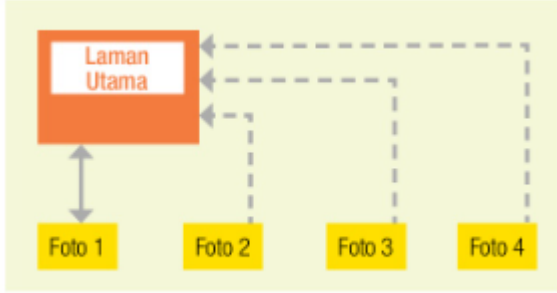
	<p>Atur cara mendapatkan nilai count</p>	 <pre> <html> <body> <script> var nama, pembilang = 0, ulang = false; do { nama = prompt("Masukkan nama anda:"); pembilang = pembilang + 1; ulang = confirm("Ulang sekali lagi?"); }while(ulang == true); document.write(pembilang); </script> </body> </html> </pre> <p>Memberi nilai awal kepada pembilang (iaitu nilai awal ialah 0) Sebab: Tiada nilai lagi untuk dikira</p> <p>Menerima sebarang nilai (contoh: nama)</p> <p>Mengira bitangan (dengan menambah 1 kepada pembilang)</p> <p>Ulangan jika ingin menerima nilai seterusnya</p> <p>Rajah 3.1.40(a) Atur cara JavaScript (mendapatkan nilai count)</p>						
	<p>Carta alir mendapatkan nilai count</p>	 <pre> graph TD Mula([Mula]) --> Init[pembilang = 0 ulang = false] Init --> Prompt[/Papar kotak prompt beserta teks "Masukkan nama anda:"/] Prompt --> Input[/Input nama/] Input --> Calc[pembilang = pembilang + 1] Calc --> Confirm[/Papar kotak confirm beserta teks "Ulang sekali lagi?"/] Confirm --> InputUlang[/Masukkan nilai ulang untuk ulangan nama/] InputUlang --> Decision{nilai ulang = true?} Decision -- BENAR --> Prompt Decision -- PALSU --> Output[/Paparkan pembilang/] Output --> Tamat([Tamat]) </pre> <p>Memberi nilai awal kepada pembilang (nilai awal adalah 0) dan nilai awal kepada ulang ialah false Sebab: Tiada nilai lagi untuk dikira</p> <p>Menerima sebarang nilai (contoh: nama)</p> <p>Mengira count (dengan menambah 1 kepada pembilang)</p> <p>Ulangan jika ingin menerima nilai seterusnya</p> <p>Rajah 3.1.41 Carta alir (mendapatkan nilai count)</p>						
	<p><i>Enqueue</i></p>	<p>Memasukkan data dalam queue</p>						
	<p><i>Dequeue</i></p>	<p>Membuang data lama dalam queue</p>						
<p>3.1.3</p>	<p><i>Pemboleh ubah</i></p>	<p>Nilai boleh berubah-ubah</p>						
	<p><i>Pemalar</i></p>	<p>Nilai tidak boleh berubah-ubah</p>						
	<p><i>Jenis data</i></p>	<table border="1"> <tr> <td>Nombor</td> <td>Contoh: 5, 100.50</td> </tr> <tr> <td>Rentetan (<i>String</i>)</td> <td>Contoh: "Ali bin Abu" "BMW 318i"</td> </tr> <tr> <td>Boolean</td> <td>Contoh: true, false</td> </tr> </table>	Nombor	Contoh: 5, 100.50	Rentetan (<i>String</i>)	Contoh: "Ali bin Abu" "BMW 318i"	Boolean	Contoh: true, false
Nombor	Contoh: 5, 100.50							
Rentetan (<i>String</i>)	Contoh: "Ali bin Abu" "BMW 318i"							
Boolean	Contoh: true, false							
<p>3.1.4</p>	<p><i>Struktur kawalan</i></p>	<p>Berjukkan (urutan), Pilihan, Ulangan</p>						

	Operator hubungan	<p>Jadual 3.1.13 Operator hubungan yang terdapat dalam ungkapan logik</p> <table border="1"> <thead> <tr> <th>Syarat</th> <th>Simbol Operator Hubungan</th> </tr> </thead> <tbody> <tr> <td>Sama dengan</td> <td>==</td> </tr> <tr> <td>Tidak sama dengan</td> <td>!=</td> </tr> <tr> <td>Lebih besar daripada</td> <td>></td> </tr> <tr> <td>Lebih kecil daripada</td> <td><</td> </tr> <tr> <td>Lebih besar atau sama dengan</td> <td>>=</td> </tr> <tr> <td>Lebih kecil atau sama dengan</td> <td><=</td> </tr> </tbody> </table>	Syarat	Simbol Operator Hubungan	Sama dengan	==	Tidak sama dengan	!=	Lebih besar daripada	>	Lebih kecil daripada	<	Lebih besar atau sama dengan	>=	Lebih kecil atau sama dengan	<=		
Syarat	Simbol Operator Hubungan																	
Sama dengan	==																	
Tidak sama dengan	!=																	
Lebih besar daripada	>																	
Lebih kecil daripada	<																	
Lebih besar atau sama dengan	>=																	
Lebih kecil atau sama dengan	<=																	
3.1.5	Standard library	Koleksi kaedah atau fungsi yang disediakan dan diguna pakai sewaktu implementasi dalam kod atur cara																
	<i>math.js</i>	<code>math.sqrt()</code> <code>math.pow()</code>																
	<i>date.js</i>	<code>date.now()</code> <code>date.format()</code>																
	Kelebihan standard library	<p>Digunakan tanpa mengetahui cara pelaksanaannya</p> <p>Boleh digunakan berulang-ulang kali apabila diperlukan dalam atur cara dengan hanya memanggil fungsinya</p> <p>Dapat mengurangkan masa pembangunan atur cara kerana tidak perlu menulis pernyataan berulang-ulang kali</p>																
3.1.6	<i>math.js</i>	<p>Jadual 3.1.30 Fungsi-fungsi dalam <i>math.js</i></p> <table border="1"> <thead> <tr> <th>Fungsi</th> <th>Penerangan</th> </tr> </thead> <tbody> <tr> <td><code>math.add(x,y)</code></td> <td>Menambah dua nombor</td> </tr> <tr> <td><code>math.divide(x,y)</code></td> <td>Membahagi dua nombor</td> </tr> <tr> <td><code>math.subtract(x,y)</code></td> <td>Mencikak dua nombor</td> </tr> <tr> <td><code>math.pow(x,y)</code></td> <td>Mengira kuasa kepada nombor</td> </tr> <tr> <td><code>math.sqrt(x)</code></td> <td>Mengira punca kuasa bagi nombor</td> </tr> <tr> <td><code>math.cube(x)</code></td> <td>Mendarab tiga nombor yang sama ($x * x * x$)</td> </tr> <tr> <td><code>math.sort(x)</code></td> <td>Mengisih unsur-unsur dalam satu matriks</td> </tr> </tbody> </table>	Fungsi	Penerangan	<code>math.add(x,y)</code>	Menambah dua nombor	<code>math.divide(x,y)</code>	Membahagi dua nombor	<code>math.subtract(x,y)</code>	Mencikak dua nombor	<code>math.pow(x,y)</code>	Mengira kuasa kepada nombor	<code>math.sqrt(x)</code>	Mengira punca kuasa bagi nombor	<code>math.cube(x)</code>	Mendarab tiga nombor yang sama ($x * x * x$)	<code>math.sort(x)</code>	Mengisih unsur-unsur dalam satu matriks
Fungsi	Penerangan																	
<code>math.add(x,y)</code>	Menambah dua nombor																	
<code>math.divide(x,y)</code>	Membahagi dua nombor																	
<code>math.subtract(x,y)</code>	Mencikak dua nombor																	
<code>math.pow(x,y)</code>	Mengira kuasa kepada nombor																	
<code>math.sqrt(x)</code>	Mengira punca kuasa bagi nombor																	
<code>math.cube(x)</code>	Mendarab tiga nombor yang sama ($x * x * x$)																	
<code>math.sort(x)</code>	Mengisih unsur-unsur dalam satu matriks																	
3.1.7	Jenis subatur cara	<p>Prosedur (<i>procedure</i>)</p> <p>Fungsi (<i>function</i>)</p>																
	Beza prosedur dan fungsi	<p>Jadual 3.1.35 Perbandingan prosedur dan fungsi</p> <table border="1"> <thead> <tr> <th>Aspek perbandingan</th> <th>Prosedur</th> <th>Fungsi</th> </tr> </thead> <tbody> <tr> <td>Memulangnkan nilai</td> <td>Tidak memulangnkan nilai</td> <td>Memulangnkan nilai</td> </tr> <tr> <td>Ccontoh definisi</td> <td> <pre>function toCelcius (fahrenheit) { var celcius; celcius = (5/9)* fahrenheit - 32; document.write ("Suhu adalah" + celcius + " darjah Celcius); }</pre> </td> <td> <pre>function toCelcius (fahrenheit) { var celcius; celcius = (5/9)* fahrenheit - 32); return celcius; }</pre> </td> </tr> <tr> <td>Ccontoh panggilan</td> <td><code>toCelcius (fahrenheit);</code></td> <td><code>celcius = toCelcius (fahrenheit);</code></td> </tr> </tbody> </table>	Aspek perbandingan	Prosedur	Fungsi	Memulangnkan nilai	Tidak memulangnkan nilai	Memulangnkan nilai	Ccontoh definisi	<pre>function toCelcius (fahrenheit) { var celcius; celcius = (5/9)* fahrenheit - 32; document.write ("Suhu adalah" + celcius + " darjah Celcius); }</pre>	<pre>function toCelcius (fahrenheit) { var celcius; celcius = (5/9)* fahrenheit - 32); return celcius; }</pre>	Ccontoh panggilan	<code>toCelcius (fahrenheit);</code>	<code>celcius = toCelcius (fahrenheit);</code>				
Aspek perbandingan	Prosedur	Fungsi																
Memulangnkan nilai	Tidak memulangnkan nilai	Memulangnkan nilai																
Ccontoh definisi	<pre>function toCelcius (fahrenheit) { var celcius; celcius = (5/9)* fahrenheit - 32; document.write ("Suhu adalah" + celcius + " darjah Celcius); }</pre>	<pre>function toCelcius (fahrenheit) { var celcius; celcius = (5/9)* fahrenheit - 32); return celcius; }</pre>																
Ccontoh panggilan	<code>toCelcius (fahrenheit);</code>	<code>celcius = toCelcius (fahrenheit);</code>																

	<i>Parameter</i>	function nama_function (parameter1, parameter2, ...)													
3.1.8 3.1.9 3.2.1	-	-													
3.2.2	<i>Operasi asas</i>	Menulis data terhadap fail teks													
		Mencipta/membuka fail teks													
		Membaca data daripada fail teks													
		Menyunting (mengubah dan menambah) fail teks													
	<i>Mod capaian dan penerangan</i>	<p style="text-align: center;">Jadual 3.2.1 Mod capaian dan penerangan</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #008000; color: white;"> <th>Mod capaian</th> <th>Penerangan</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">r</td> <td>Fail dibuka hanya untuk dibaca sahaja</td> </tr> <tr> <td style="text-align: center;">w</td> <td>Fail dibuka hanya untuk ditulis sahaja (bermula dari awal fail)</td> </tr> <tr> <td style="text-align: center;">a</td> <td>Fail dibuka untuk ditulis sahaja (bermula di akhir fail)</td> </tr> <tr> <td style="text-align: center;">r +</td> <td>Fail dibuka untuk dibaca/ditulis. Penunjuk fail bermula daripada permulaan fail tersebut.</td> </tr> <tr> <td style="text-align: center;">w +</td> <td>Fail dibuka untuk dibaca/ditulis. Memadamkan kandungan fail atau mencipta fail baharu sekiranya fail belum wujud. Penunjuk fail bermula daripada permulaan fail tersebut.</td> </tr> <tr> <td style="text-align: center;">a +</td> <td>Fail dibuka untuk dibaca/ditulis. Kandungan fail tersebut terpelihara. Mencipta fail baharu sekiranya fail belum wujud. Penunjuk fail bermula daripada penghujung fail tersebut.</td> </tr> </tbody> </table>	Mod capaian	Penerangan	r	Fail dibuka hanya untuk dibaca sahaja	w	Fail dibuka hanya untuk ditulis sahaja (bermula dari awal fail)	a	Fail dibuka untuk ditulis sahaja (bermula di akhir fail)	r +	Fail dibuka untuk dibaca/ditulis. Penunjuk fail bermula daripada permulaan fail tersebut.	w +	Fail dibuka untuk dibaca/ditulis. Memadamkan kandungan fail atau mencipta fail baharu sekiranya fail belum wujud. Penunjuk fail bermula daripada permulaan fail tersebut.	a +
Mod capaian	Penerangan														
r	Fail dibuka hanya untuk dibaca sahaja														
w	Fail dibuka hanya untuk ditulis sahaja (bermula dari awal fail)														
a	Fail dibuka untuk ditulis sahaja (bermula di akhir fail)														
r +	Fail dibuka untuk dibaca/ditulis. Penunjuk fail bermula daripada permulaan fail tersebut.														
w +	Fail dibuka untuk dibaca/ditulis. Memadamkan kandungan fail atau mencipta fail baharu sekiranya fail belum wujud. Penunjuk fail bermula daripada permulaan fail tersebut.														
a +	Fail dibuka untuk dibaca/ditulis. Kandungan fail tersebut terpelihara. Mencipta fail baharu sekiranya fail belum wujud. Penunjuk fail bermula daripada penghujung fail tersebut.														
<i>Date</i>	d	Day (01 - 31)													
	m	Month (10 - 12)													
	Y	Year (any 4 digits)													
	h	Hour (01 - 12)													
	i	Minute (1 - 59)													
	a	am / pm													
	d/m/Y h:i:s:a														
3.2.3 3.2.4 3.2.5 3.2.6 3.2.7	-	-													
3.2.8	<i>Jenis pernyataan</i>	Pernyataan tindakan													
		Pernyataan memilih													

	<i>Pernyataan tindakan</i>	Menjalankan tugas bersama-sama data yang sedia ada dari pangkalan data
	<i>Pernyataan memilih</i>	Mengambil data sedia ada dari pangkalan data untuk digunakan
3.2.9	-	-
3.2.10	<i>Pengemaskinian data</i>	<p>Rajah 3.2.55 Proses pengemaskinian data</p>
3.3.1	<i>Prinsip asas reka bentuk laman web</i>	Jenis persekitaran web yang dibina
		Perbezaan elemen dalam web
		Penekanan maklumat dalam web
		<i>Typography</i>
		Warna dan grafik
		Navigasi
		Keseimbangan visual
	<i>Jenis persekitaran web yang dibina</i>	Kesesuaian penggunaan pelbagai jenis pelayar
		Kelajuan capaian internet
		Keperluan perisian sokongan untuk paparan
		Kepelbagaian sistem komputer dan peranti mudah alih yang digunakan oleh pengguna
		Kepelbagaian saiz paparan monitor yang digunakan
	<i>Penekanan maklumat dalam web</i>	Penggunaan kesan khas (<i>drop</i> , <i>shadow</i> , tekstur atau cahaya dan bentuk)
		Penggunaan ruang putih untuk menjadikan elemen-elemen kelihatan lebih terserlah
		Penggunaan fungsi bold , <i>italic</i> , warna yang berbeza dan <i>border</i>
	<i>Perbezaan elemen dalam web</i>	Penggunaan label dan pautan berbeza
		Penggunaan <i>border</i> , warna berbeza dan kesan khas
		Penggunaan ruang putih, teks terbalik, <i>italic</i> , saiz berbeza

		Penggunaan prinsip perbezaan warna dalam memberi kelainan pada butang login													
	<i>Typography</i>	Corak paparan berasaskan kertas													
		Corak paparan secara <i>Z-shaped</i>													
		Corak paparan secara <i>F-shaped</i>													
	<i>Navigasi</i>	Bar navigasi													
		Menu <i>Drop-down</i>													
		<i>Breadcrumbs</i>													
		Pautan menggunakan grafik													
		<i>Site map</i>													
3.2.3	<i>Proses mereka bentuk kerangka aplikasi</i>	Pengstrukturkan kandungan atau maklumat													
		Penstrukturkan halaman													
		Reka bentuk navigasi													
		Reka bentuk antara muka													
	<i>Pengstrukturkan kandungan atau maklumat</i>	Membentuk inventori maklumat													
		Pengelasan maklumat Jadual 3.3.1 Pengelasan maklumat <table border="1" data-bbox="609 1205 1382 1590"> <thead> <tr> <th>Kaedah</th> <th>Contoh</th> </tr> </thead> <tbody> <tr> <td>Mengikut turutan abjad</td> <td>Pengelasan maklumat jualan berasaskan nama pengguna</td> </tr> <tr> <td>Kronologi</td> <td>Pengelasan hasil jualan mengikut tarikh</td> </tr> <tr> <td>Kategori atau jenis</td> <td>Pengelasan jualan mengikut kategori produk</td> </tr> <tr> <td>Hierarki</td> <td>Jenis baju muslimah yang dikelaskan kepada subkategori seperti blaus dan kurung moden</td> </tr> <tr> <td><i>Spatial</i></td> <td>Pengelasan maklumat mengikut geografi, contohnya pengelasan jualan mengikut negeri dan tempat</td> </tr> <tr> <td>Mengikut urutan magnitud</td> <td>Pengelasan maklumat jualan bermula daripada jumlah yang kecil kepada jumlah jualan yang besar.</td> </tr> </tbody> </table>	Kaedah	Contoh	Mengikut turutan abjad	Pengelasan maklumat jualan berasaskan nama pengguna	Kronologi	Pengelasan hasil jualan mengikut tarikh	Kategori atau jenis	Pengelasan jualan mengikut kategori produk	Hierarki	Jenis baju muslimah yang dikelaskan kepada subkategori seperti blaus dan kurung moden	<i>Spatial</i>	Pengelasan maklumat mengikut geografi, contohnya pengelasan jualan mengikut negeri dan tempat	Mengikut urutan magnitud
	Kaedah	Contoh													
	Mengikut turutan abjad	Pengelasan maklumat jualan berasaskan nama pengguna													
	Kronologi	Pengelasan hasil jualan mengikut tarikh													
	Kategori atau jenis	Pengelasan jualan mengikut kategori produk													
Hierarki	Jenis baju muslimah yang dikelaskan kepada subkategori seperti blaus dan kurung moden														
<i>Spatial</i>	Pengelasan maklumat mengikut geografi, contohnya pengelasan jualan mengikut negeri dan tempat														
Mengikut urutan magnitud	Pengelasan maklumat jualan bermula daripada jumlah yang kecil kepada jumlah jualan yang besar.														
<i>Penstrukturkan halaman</i>	Penstrukturkan secara hierarki														
	Penstrukturkan secara jujukan														
	Penstrukturkan kompleks														

	Penstrukturan secara hierarki	 <p style="text-align: center;">Rajah 3.3.17 Contoh penstrukturan secara hierarki</p>
	Penstrukturan secara jujukan	 <p style="text-align: center;">Rajah 3.3.18 Penstrukturan secara jujukan</p>
3.3.3	-	-
3.3.4	<i>Cara penulisan CSS dalam HTML</i>	Helaian gaya luar (<i>External Style Sheet</i>)
		Helaian gaya dalaman (<i>Internal Style Sheet</i>)
		Gaya dalam barisan (<i>Inline Style</i>)
	<i>Jenis kedudukan</i>	Static
		Fixed
		Relative
		Absolute
3.3.5	<i>Butang pilihan</i>	Butang radio
		Butang <i>checkbox</i>
	<i>Kotak popup</i>	Kotak <i>alert</i>
		Kotak <i>confirm</i>
		Kotak <i>prompt</i>

3.3.6	<i>Ciri laman web interaktif</i>	Menggunakan bahasa pengaturcaraan web seperti PHP, ASP dan lain-lain
		Mempunyai capaian kepada pangkalan data
		Kandungan laman web berubah mengikut permintaan pengguna
		Kandungan laman web sering dikemaskinikan
	<i>Elemen dalam laman web interaktif</i>	Kemudahan mengemas kini data yang diperolehi dan simpan semula ke dalam pangkalan data
		Paparan maklumat yang diperolehi atas permintaan daripada pengguna
		Carian maklumat berdasarkan permintaan daripada pengguna
		Sambungan kepada komputer pelayan (<i>web server</i>)
		Objek borang (<i>form</i>) dalam laman web
	3.3.7	-
<i>I love you</i> 🍷🌻🧡 Good luck!!! ✨❤		

