

RINGKASAN BUKU TEKS



SAINS KOMPUTER TINGKATAN 5

MOHD RAMADAN BIN SAIDIN

1.0

PENGKOMPUTERAN





1.0 PENGKOMPUTERAN

1.1 Komputer
dan Impak

1.2 Seni Bina
Komputer

1.3 Get Logik

KOMPUTER DAN IMPAK

1.1.1

- Menganalisis Kesan Inovasi dalam Pengkomputeran.

1.1.2

- Mengenal pasti aktiviti ttidak beretika dalam pengkomputeran.

1.1.3

- Menjelaskan keperluan keselamatan data dalam rangkaian komputer dan langkah keselamatan (*encrypted email, double verification, password*)

1.1.4

- Melaksanakan penyulitan (*encryption*) dan nyahsulit (*decryption*) data menggunakan *Ceaser Cipher*.

1.1.5

- Membincangkan implikasi sosial berkaitan penggodaman dan cetak rompak perisian.

1.1.6

- Mengkaji undang-undang siber diantara negara.

1.1.7

- Mengkaji kerjaya yang berkaitan dengan bidang pengkomputeran masa hadapan.

KESAN INOVASI DALAM PENGKOMPUTERAN

KESAN
POSITIF



KESAN
NEGATIF



INOVASI TEKNOLOGI
RANGKAIAN

INOVASI TEKNOLOGI
STORAN

INOVASI SISTEM
PENGOPERASIAN

PERKEMBANGAN
SISTEM DALAM TALIAN

JENAYAH SIBER



KESAN POSITIF



INOVASI DALAM TEKNOLOGI RANGKAIAN

- Terciptanya internet.



INOVASI DALAM TEKNOLOGI STORAN

- Jenis dan bentuk storan yang lebih baik.
- Wujudnya Storan awan.



INOVASI SISTEM PENGOPERASIAN

- Tercipta Sistem Pengoperasian Rangkaian.
- Tercipta Sistem Pengoperasian untuk peranti mudah alih.



PERKEMBANGAN SISTEM DALAM TALIAN

- Wujudnya pelbagai sistem atas talian (e-perbankan,e-dagang, system hiburan elektronik, e-pembelajaran, e-kerajaan)
- Jaringan komunikasi tanpa sempadan.

KESAN NEGATIF



JENAYAH SIBER

- Penyalahgunaan teknologi pengkomputeran dan internet bagi melaksanakan aktiviti jenayah iaitu merosakkan, mencuri atau mengubah data dalam sistem komputer bagi tujuan tertentu

PENGGUNA PERSENDIRIAN DAN INSTITUSI

- Penipuan, pembulian siber, penipuan e-mel, kecurian maklumat (phishing), perjudian atas talian, imej penderaan kanak-kanak, pornografi, skim cepat kaya, menyalin perisian secara tidak sah, pengubahsuaian atau merosakkan program dan data komputer, capaian yang tidak dibenarkan, malware atau penyebaran virus dan DDos, laman web palsu dan pengodaman.

KESELAMATAN NEGARA

- Serangan pengganas siber terhadap infrastruktur kritikal negara seperti sistem kerajaan, system kawalan trafik udara, sistem penjana elektrik, infrastuktur rangkaian telekomunikasi, system kawalan industry dan hacktivism.

ETIKA DALAM PENGKOMPUTERAN

- Standard panduan bagi mengawal perlakuan pengguna komputer supaya lebih bermoral.

CONTOH



ETIKA	TIDAK BERTIKA
Dilarang menggunakan komputer dengan niat untuk membahayakan orang lain.	• Mencipta virus. • Menyebarkan virus. • Merosak atau memanipulasi data atau fail pengguna lain.
Dilarang mengganggu urusan privasi atau tugas komputer orang lain.	• Mengganggu hak urusan peribadi orang lain. • Memuat turun perisian tidak sah ke dalam komputer orang lain sehingga mengganggu fungsi komputer.
Dilarang mencero bph ke dalam fail komputer pengguna lain.	• Mendapatkan data daripada fail peribadi orang lain tanpa kebenaran. • Mengintip fail orang lain. • Membaca mesej peribadi orang lain.
Dilarang menggunakan komputer untuk mencuri.	• Mencuri data peribadi bagi tujuan mendapatkan sumber kewangan.
Dilarang menggunakan komputer untuk bertindak sebagai saksi palsu.	• Menyebarkan berita palsu dengan mengubah maklumat.
Dilarang menggunakan/ menyalin perisian orang lain tanpa kebenaran.	• Membuat salinan haram (cetak rompak) terhadap perisian hakcipta tanpa kebenaran penciptanya.
Tidak menggunakan sumber komputer orang lain	• Menggunakan komputer tanpa meminta izin dari pemiliknya.
Dilarang menggunakan harta intelek orang lain sebagai hak milik.	• Membuat salinan/mengkomersialkan perisian tersebut tanpa kebenaran pemiliknya.
Mempertimbangkan kesan sosial daripada program atau sistem yang dibangunkan.	• Pembangun perisian tidak mempertimbangkan jenis kandungan yang mempunyai unsur-unsur yang tidak baik/ beri kesan negatif.
Disaran menggunakan komputer dengan menunjukkan pertimbangan dan menghormati pengguna lain.	• tidak melayan orang lain dengan hormat. • mencero boh ruang peribadi orang lain. • menggunakan bahasa kesat. • membuat kenyataan palsu. • tidak berbudi Bahasa. • tidak menghormati masa dan sumber orang lain. • tidak bertimbangrasa dengan pengguna baharu.

ISU KESELAMATAN

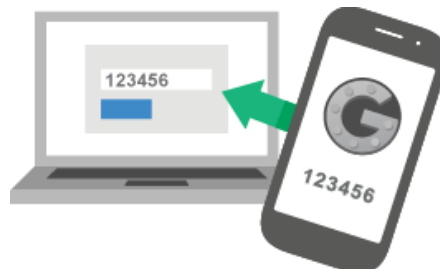
- Isu keselamatan data dalam rangkaian perlu diberi perhatian kerana kelemahan ini memudahkan penjenayah siber untuk membuat pintasan terhadap rangkaian komputer dan mencuri maklumat peribadi yang berharga.

ANTARA LANGKAH-LANGKAH
KESELAMATAN DATA

**Penyulitan
E-mel**
(*Encrypted Email*)



Dua Pengesahan
(*Double
Verification*)



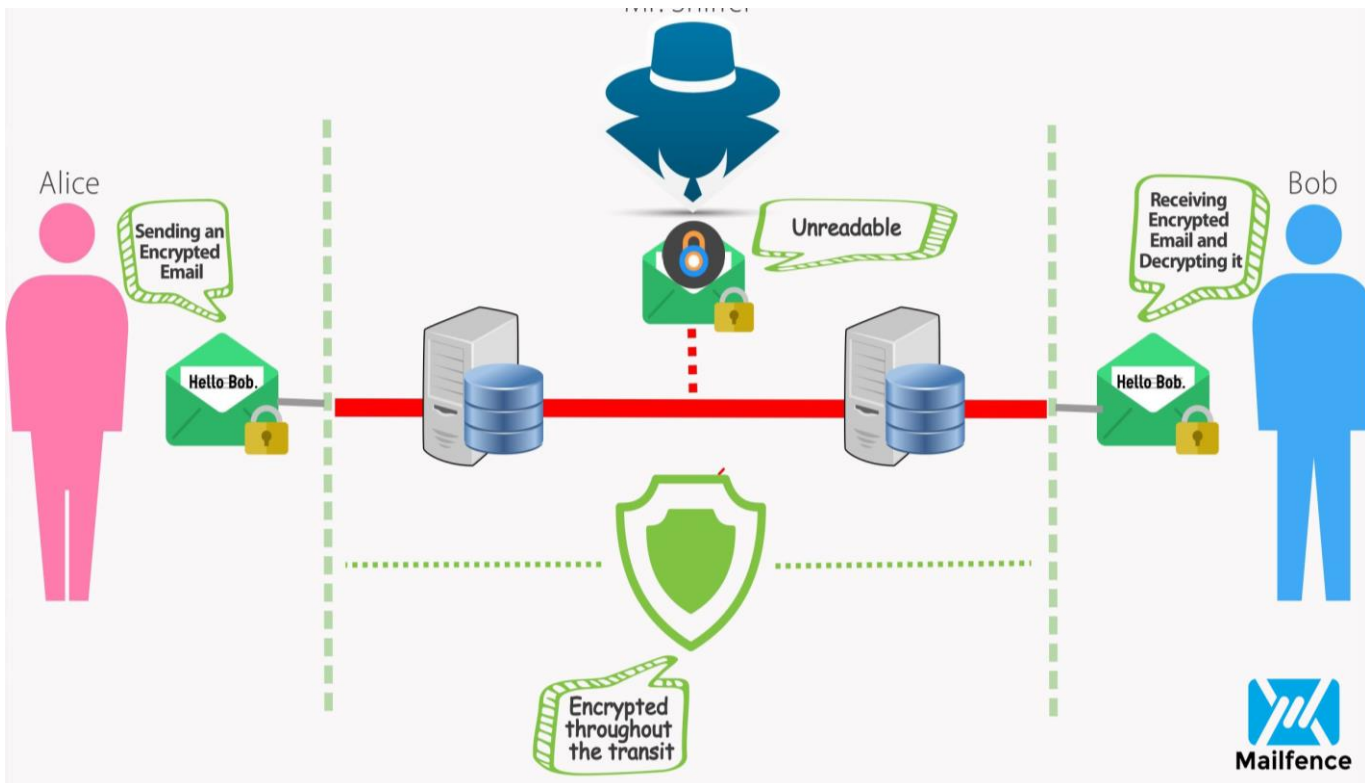
Kata Laluan
(*Password*)



PENYULITAN E-MEL

PENYULITAN E-MEL

- Proses penyulitan maklumat yang akan dihantar melalui e-mel.
- Biasanya melibatkan maklumat yang sensitif atau peribadi.
- Contoh : *SafeGmail, Rmail, infoencrypt, Hushmail, iSafeguard, safe-mail.net.*



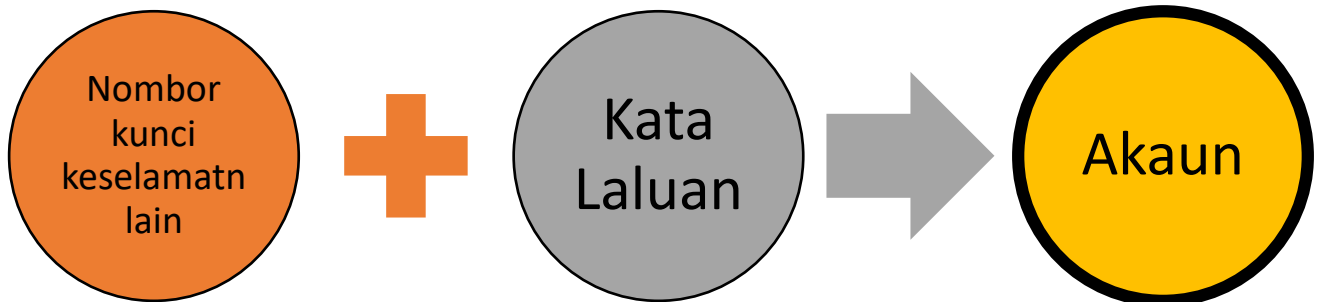
safegmail.com



DUA PENGESAHAN

DUA PENGESAHAN

- Merupakan lapisan pengesahan tambahan yang perlu ditentukan oleh pengguna apabila ingin memasuki akaun atau mengakses data dalam talian.
- Dengan adanya dua lapisan tersebut, langkah pengesahan untuk mengakses data menjadi lebih sukar.
- Semakin banyak lapisan pengesahan, semakin sukar data untuk diceroboh.



KATA LALUAN

KATA LALUAN

- Merupakan lapisan pengesanan keselamatan yang utama untuk memasuki akaun pengguna.
- Memudahkan penceroboh memasuki akaun pengguna jika hanya satu lapisan keselamatan sahaja.
- Pemilihan kata laluan amatlah penting.
- Elak pembinaan kata laluan berdasarkan nama, nombor kad pengenalan, tarikh lahir atau sebarang data yang berkaitan dengan maklumat peribadi pengguna.
- Kata laluan yang baik dibina daripada gabungan huruf, simbol dan nombor iaitu antara 8 hingga 12 aksara.
- Kebarangkalian gabungan aksara ini akan menyukarkan penggodam menekanya.



Strong password

Password	secure %
↓	↓
Rohitkumar	10%
Rohit123	15%
Rohit@123	20%
RO"h"1"t"@123	99.99%

Include:

More Characters

More Numbers/Symbols

Memorable Combinations

Uppercase & Lowercase

Do Not Include:

Dictionary Words

Repeated Words

Personal Information

The Word "Password"

1.1

1.1.4

MELAKSANAKAN PENYULITAN (*ENCRYPTION*) DAN NYAHSULIT (*DECRYPTION*) DATA MENGGUNAKAN *CEASER CIPHER*

KRIPTOGRAFI

- Proses penyulitan data bagi melindungi privasi data pengguna

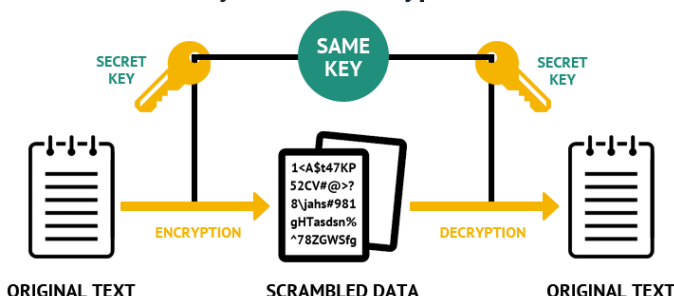
PENYULITAN

- Proses penterjemahan data kepada bentuk kod rahsia yang tidak dapat difahami.
- Merupakan satu cara yang berkesan untuk melindungi data.
- Teks biasa (Plaintext)** : Data yang belum melalui proses penyulitan.
- Teks Sifer (Ciphertext)** : Data yang telah melalui proses penyulitan.

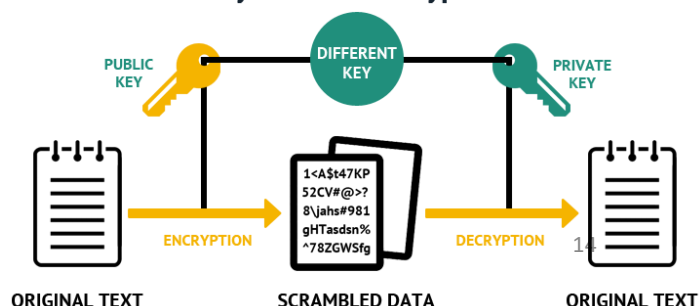


JENIS PENYULITAN

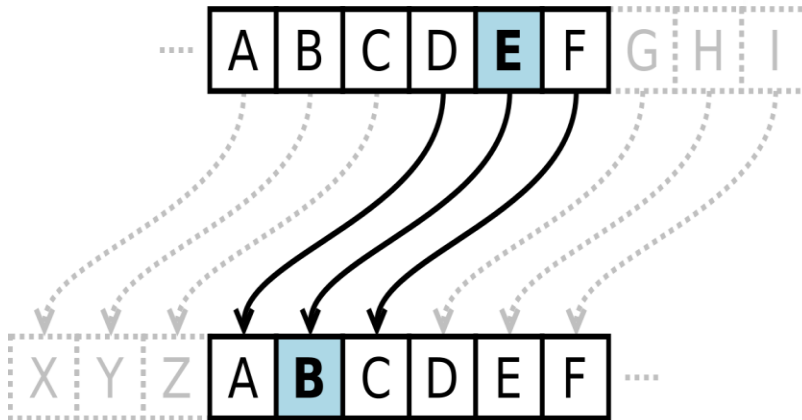
Symmetric Encryption



Asymmetric Encryption



CEASER CIPHER



CEASAR CIPHER



- Salah satu bentuk penyulitan yang paling mudah.
- Merupakan *cipher* penggantian (*substitution*) iaitu setiap huruf teks biasa digantikan dengan huruf yang sepadan dengan huruf dalam abjad (A-Z).
- Formula : $E_n(X) = (X + n) \bmod 26$.
- n : Merujuk bilangan anjakan.
- Mod 26 : merujuk kepada jumlah huruf A hingga Z iaitu 26.
- Proses nyahsulit (decryption) terhadap teks sifer dilakukan dengan cara yang sama tetapi dengan menolak jumlah anjakan.



Teks
Biasa

Teks
Sifer

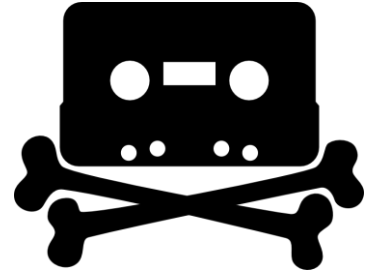
PENGGOUDAM

- Penggodam (*hackers*) ialah penjenayah siber yang bertindak memintas komunikasi rangkaian bagi memperoleh data penting pengguna untuk dieksploitasikan.



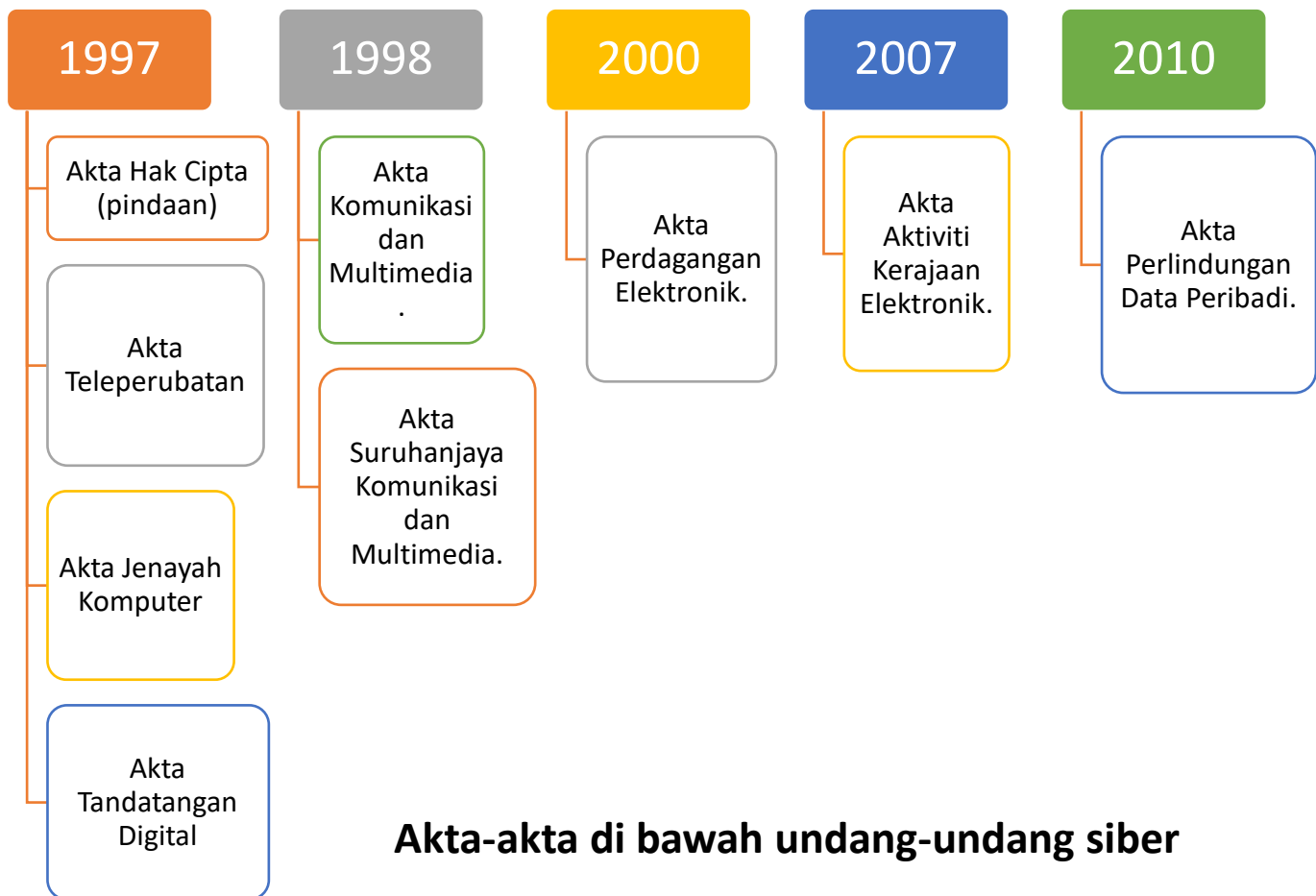
CETAK ROMPAK

- Merupakan pencabulan hak cipta – perbuatan menyalin semula, menerbitkan atau menggunakan bahan bercetak atau bahan rakaman digital seperti video dan perisian komputer yang dilakukan tanpa kebenaran pemegang hak cipta bagi harta intelek tersebut dengan matlamat mengaut keuntungan.



UNDANG-UNDANG SIBER

- Merupakan peraturan yang ditetapkan oleh pihak berkuasa di sesebuah negara untuk memberikan hukuman kepada pesalah-pesalah yang melakukan kesalahan dalam talian.



PERBEZAAN UNDANG-UNDANG SIBER ANTARA NEGARA

NEGARA	PENUBUHAN UNDANG- UNDANG SIBER	KAWALAN TERHADAP MEDIA SOSIAL	ISU KEBEBASAN BERSUARA	ISU KELUCAHAN	PERJUDIAN
AMERIKA SYARIKAT 	Akta Jenayah Komputer 1986, <i>Department of Justice</i>	Tiada sekatan	Ada kebebasan.	Ada kebebasan – kecuali pornografi kanak-kanak.	Tidak menghalang – kecuali perjudian haram dibawah akta penguatkuasaan perjudian internet tidak sah 2006.
CHINA 	1997 Cybersecurity Law oleh Public Security Bureau (PBS) .	Menyekat Capaian internet secara bebas. <i>(The Great Firewall of China)</i>	Tiada kebebasan secara mutlak.	Tidak dibenarkan	Tidak dibenarkan
ARAB SAUDI 	Anti Cyber Crime Law by Royal	Menyekat Capaian internet secara bebas. Tidak membenarkan capaian media social.	Tiada kebebasan secara mutlak.	Tidak dibenarkan	Tidak dibenarkan
MALAYSIA 	Akta Jenayah Komputer 1997	Tiada sekatan	Tiada kebebasan secara mutlak.	Tidak dibenarkan	Tidak dibenarkan



BIDANG	PERANAN DAN TANGGUNGJAWAB	KERJAYA	KEMAHIRAN
SISTEM RANGKAIAN DAN KESELAMATAN RANGKAIAN	- Mengendalikan semua aspek keselamatan rangkaian. - Membangunkan dasar keselamatan. - Mengesyorkan peningkatan keselamatan data.	Juruanalisis Keselamatan Maklumat	Kepakaran dalam bidang rangkaian dan keselamatan komputer.
SOKONGAN MAKLUMAT DAN PERKHIDMATAN	- Mengintegrasikan pengurusan perniagaan dan analisis data. - menilai pelaksanaan sistem komputer. - Menyediakan sokongan pengkomputeran & teknikal. - Menyelenggara sistem maklumat.	- Pentadbir pangkalan data. - Juruanalisis system komputer. - Juruanalisis sokongan komputer. - Pengurus pengkomputeran - Pentadbir system komputer.	Perisian aplikasi perniagaan yang melibatkan pengurusan pangkalan data.
PENGATURCARAAN DAN PEMBANGUNAN PERISIAN	Merancang, mereka bentuk, mengemaskini dan menguruskan perisian komputer dan system melalui pengaturcaraan dan pembangunan perisian.	- Juruanalisis system. - Jurutera system komputer. - Pembangun perisian. - Pengatur cara komputer.	- Java - C# - C++ .Net - Sharepoint - Python
WEB DAN KOMUNIKASI DIGITAL	- Membangunkan aplikasi berasaskan web. - Menulis kod. - Mengintegrasikan sistem. - Menyediakan bantuan teknikal. - Membangunkan media interaktif, produk digital dan multimedia.	- Pembangun laman web. - Pembangun perisian multimedia. - Pakar media digital. - Pereka grafik dan multimedia.	- Javascript - HTML. - AJAX - PHP .Net - Python - Perisian & perkakasan multimedia.

Tiga elemen telah dikenal pasti menjadi komponen utama infrastruktur bagi produk dan perkhidmatan dalam sektor perindustrian masa kini.



MUDAH ALIH

- Penghasilan aplikasi dan kandungan mudah alih bagi peranti selain komputer.



KESELAMATAN DATA

- Keperluan kepakaran dalam bidang penyulitan, pengesahan, pengurusan sijil, infrastruktur *firewall* dan *anti-malware*.
- Keperluan terhadap kerjaya penganalisis keselamatan data, pentadbir keselamatan, pentadbir keselamatan rangkaian dan pentadbir keselamatan sistem.



DATA RAYA (BIG DATA) DAN KECERDASAN PERNIAGAAN & ANALISTIK (BUSINESS INTELLIGENCE & ANALYTICS).

- Kemampuan menterjemah, menganalisis dan keupayaan untuk mengubah kuantiti data mentah yang besar ke bentuk maklumat.
- Keperluan set kemahiran pentafsir yang memahami cara untuk mendapatkan data, menganalisis dan melaporkan data.

SENI BINA KOMPUTER

1.2.1

- Menjelaskan kitaran capai-nyahkod-laksana (*FDE – Fetch, Decode, Execute cycle*)

1.2.2

- Menerangkan konsep asas seni bina (Von Neumann) – Unit Kawalan, ALU (*Arithmetic Logic Unit*), *register, clock, address bus* dan *data bus*.

1.2.3

- Mengkaji perbezaan pemproses (*processor*) linear dan semasa.

1.2.4

- Membincangkan penterjemah pelbagai aras Bahasa pengaturcaraan.

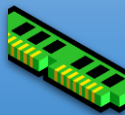
1.2.5

- Merumuskan perlaksanaan atur cara dalam sebuah pemproses.



UNIT PEMROSESAN PUSAT

- ✓ Juga dikenali sebagai pemproses @ CPU.
- ✓ Terdiri dari 3 komponen.
- ✓ **Unit Aritmetik dan Logik (ALU) :** Melakukan operasi aritmetik dan logik.
- ✓ **Unit Kawalan :** mengawal operasi ingatan, pemproses dan peranti input dan output.
- ✓ **Daftar (Register) :** berfungsi untuk menyimpan data secara sementara, menghantar arahan yang dicapai daripada ingatan utama sebelum diproses di ALU dan menerima hasil daripada ALU sebelum dihantar ke ingatan utama.



UNIT INGATAN

- ✓ **FUNGSI :** menyimpan data dan arahan semasa komputer digunakan.
- ✓ Menyimpan/ memegang data secara sementara selagi arahan yang merujuknya masih beroperasi.
- ✓ Tidak digunakan untuk menyimpan data secara kekal.
- ✓ **BERSIFAT MERUAP :** hilang kandungannya jika bekalan elektrik terputus.
- ✓ Ingatan utama terbahagi kepada 2 : RAM dan ROM.
- ✓ Set arahan dalam ROM akan kekal.

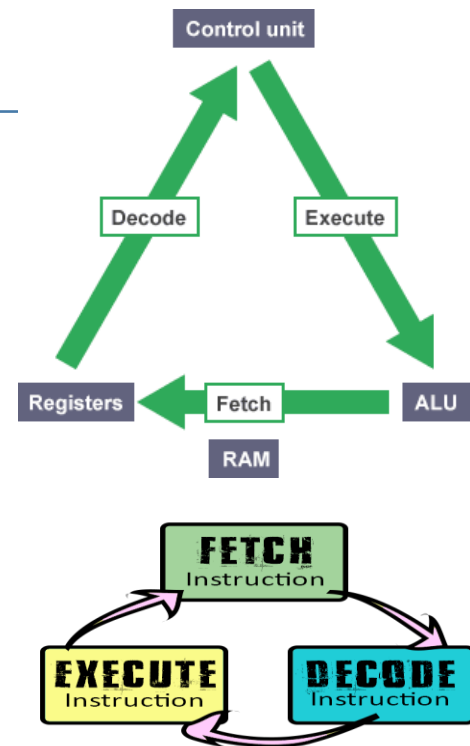


PERANTI INPUT DAN OUTPUT

- ✓ Komponen yang digunakan untuk memasukkan data dan memaparkan hasil daripada sistem komputer.
- ✓ **PERANTI INPUT :** menukar data luaran kepada satu bentuk yang boleh difahami dan kemudian diproses oleh komputer.
- ✓ Contoh : papan kekunci, skrin sentuh, pengimbas, perakam suara, tetikus.
- ✓ **PERANTI OUTPUT :** memaparkan hasil yang diproses oleh komputer kepada bentuk yang boleh difahami oleh manusia.
- ✓ Contoh : pencetak, monitor, pembesar suara.

KITARAN MESIN

- Tugas utama CPU adalah untuk melaksanakan arahan pemprosesan data.
- Pemprosesan data dan arahan mengikut kitaran mesin → kitaran capai-nyahkod-laksana (FDE).
- Juga dikenali sebagai kitaran arahan.
- Pemproses menghantar isyarat supaya data dan arahan yang dimasukkan melalui peranti input disalin ke dalam ingatan utama.
- Data dan arahan tersebut diproses melalui 3 proses utama dalam kitaran mesin untuk menghasilkan output.

**CAPAI**
(Fetch)

- Unit kawalan memberikan isyarat supaya pemproses mencapai data dan arahan dari ingatan utama dan disimpan dalam daftar sementara menunggu untuk diproses.

NYAHKOD
(Decode)

- Unit kawalan menyahkod atau menterjemah arahan bagi menentukan jenis operasi yang akan dilaksanakan oleh ALU.

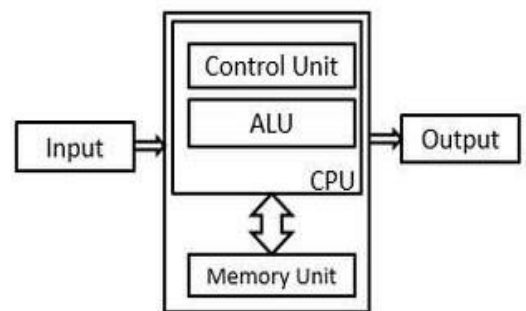
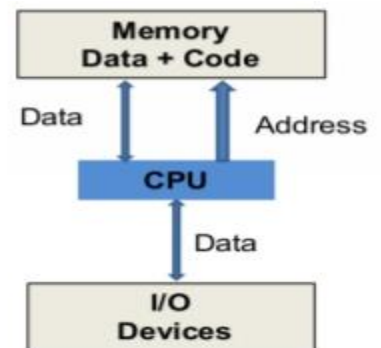
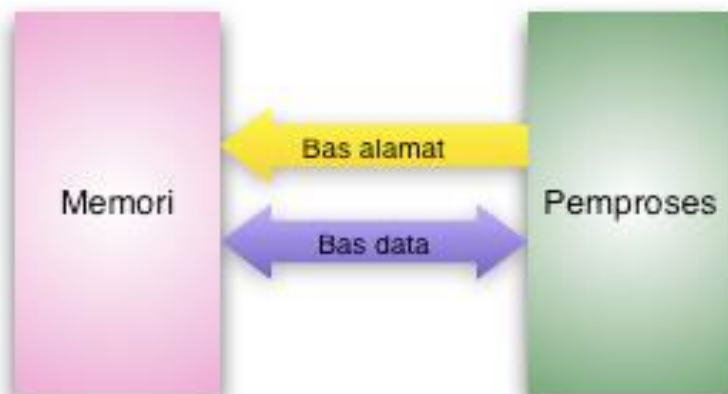
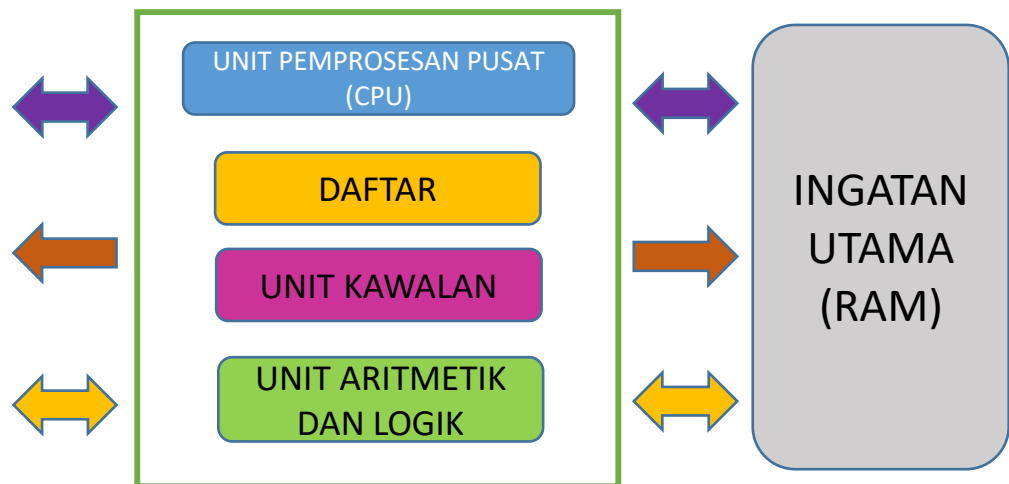
LAKSANA
(Execute)

- Pemproses melaksanakan operasi aritmetik dan logik terhadap data seperti yang dinyatakan oleh arahan.

- ✓ Pemproses akan mengulangi proses capai-nyahkod-laksana sehingga semua data dan arahan selesai diproses.
- ✓ Semasa proses ini diulangi, hasil pemprosesan disimpan sementara di dalam daftar sehingga selesai dan hasil akhir akan dihantar ke ingatan utama untuk dipamerkan melalui peranti output.

SENI BINA VON NEUMANN

- 1945 : Jon Von Neumann mendapat idea mencipta kaedah untuk menyimpan atur cara komputer dalam system komputer.
- Atur cara dan data disimpan dalam ingatan dan digerakkan dari unit ingatan ke pemproses.
- **3 komponen utama dalam seni bina Von Neuman** : CPU, Ingatan dan peranti input dan output.
- Komponen-komponen ini dihubungkan dalam sistem bas.

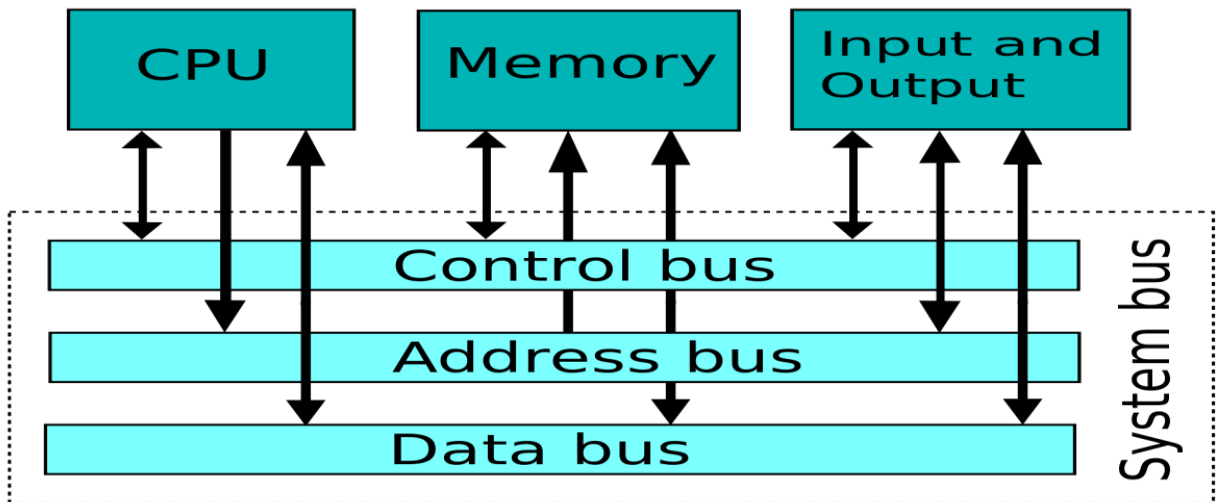
UNIT INPUT
DAN OUTPUT

Von Neumann Machine

1.2

1.2.2

MENERANGKAN KONSEP ASAS SENI BINA VON NEUMANN



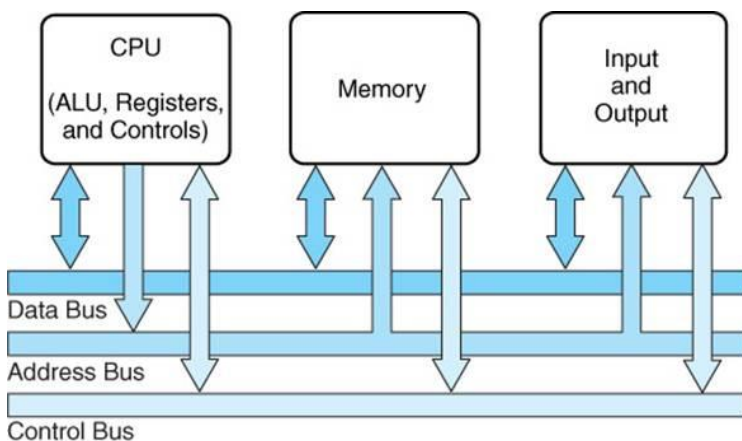
SISTEM BAS

BAS DALAMAN

Menghubungkan komponen-komponen dalam pemproses dan membenarkan komponen tersebut bertukar-tukar maklumat.

BAS LUARAN

Menghubungkan pemproses dengan komponen luaran seperti ingatan dan peranti input dan output.



BAS DATA

Digunakan untuk memindahkan
 * antara pemproses, ingatan dan peranti input/output.
 * antara daftar, unit kawalan dan ALU

BAS ALAMAT

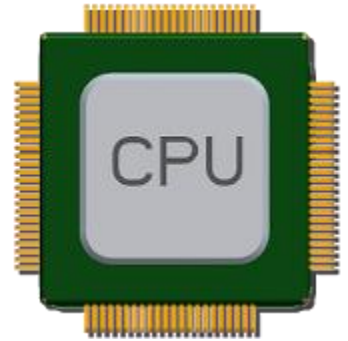
Digunakan bagi
 * membawa alamat lokasi yang menempatkan data dan arahan dalam ingatan utama.
 * mengandungi alamat peranti input/output

BAS KAWALAN

Bas yang mengandungi isyarat kawalan yang digunakan bagi mengawal operasi keseluruhan sistem.

UNIT PEMROSESAN PUSAT (CPU)

- Komponen yang paling penting dalam sistem komputer.
- Juga dikenali sebagai pemproses.
- **3 Komponen utama** : Unit Aritmetik dan Logik (ALU), daftar dan Unit Kawalan.

UNIT ARITMETIK
DAN LOGIK (ALU)

- Melakukan operasi-operasi aritmetik (tambah, tolak, bahagi, darab, nilai negatif dan sebagainya).
- Melakukan operasi logik (DAN, ATAU, TAK, XATAU dan sebagainya).
- ALU menerima input dari daftar-daftar dalam CPU dan menghasilkan maklumat untuk dihantar ke daftar sebelum disimpan dalam ingatan.

DAFTAR

- Menyimpan data dan arahan yang dicapai dari ingatan utama sebelum dihantar ke ALU untuk diproses dan menyimpan hasil pemprosesan sementara sebelum dihantar ke ingatan utama dan dipamerkan melalui peranti output.

UNIT KAWALAN

- Menerima dan menjana isyarat-isyarat bagi mengawal operasi keseluruhan sistem komputer.
- Inout utamanya ialah denyutan isyarat dari sistem jam.
- Unit kawalan mengandungi penyahkod arahan dan logik pemasaan serta kawalan.

JENIS DAFTAR

PC

(Program Counter/
Pembilang atur cara)

- ✓ Memegang alamat ingatan pada peringkat awal proses kitaran mesin.
- ✓ Salin alamat tadi ke daftar MAR. Nilai PC bertambah.
- ✓ Akan sentiasa menunjuk/ memegang alamat bagi arahan seterusnya.

ACC

(Accumulator / Daftar
Pengumpul)

- ✓ Menyimpan data yang menjadi input dan output semasa operasi ALU.

MAR

(Memory Address
Register / Daftar
Alamat Ingatan)

- ✓ Memegang alamat lokasi maklumat dalam ingatan utama.

MDR

(Memory Data
Register/ Daftar Data
Ingatan)

- ✓ Memegang data dan arahan yang dibaca dari ingatan @ yang akan ditulis.
- ✓ Kandungan arahan dalam MDR akan dihantar ke IR dan kandungan data akan dihantar ke ACC.

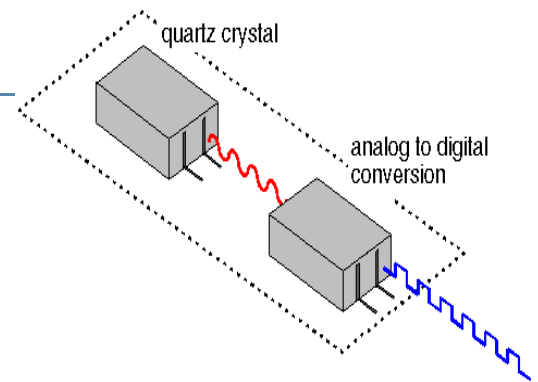
IR

(Instruction Register/
Daftar Arahan)

- ✓ Mengandungi arahan terkini yang akan dilaksanakan.
- ✓ Unit kawalan akan menyahkod arahan ini bagi menentukan jenis operasi yang perlu dilaksanakan oleh ALU.

SISTEM JAM (CLOCK SYSTEM)

- Merupakan satu peranti cip dalam pemproses yang menjana denyutan elektronik pada kadar yang tetap bagi menyeragamkan semua operasi komputer.
- **Hertz** : Unit yang digunakan untuk menghitung kadar kelajuan jam dalam pemproses.
- **1 Hertz = 1 denyutan/ kitaran per saat.**
- Contoh : 160 MHz = 160,000,000 denyutan/kiraan sesaat.
- Pemproses akan menggunakan denyutan ini semasa melaksanakan arahan dalam kitaran mesin.
- Satu arahan memerlukan satu kitaran mesin untuk dilaksanakan.
- Dengan itu, lebih tinggi kelajuan hertz, lebih laju komputer tersebut.



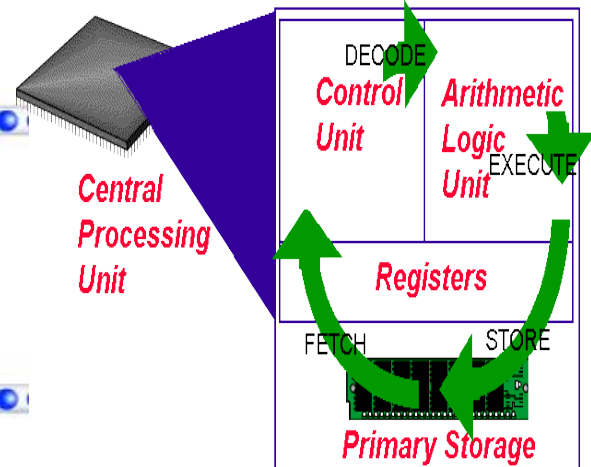
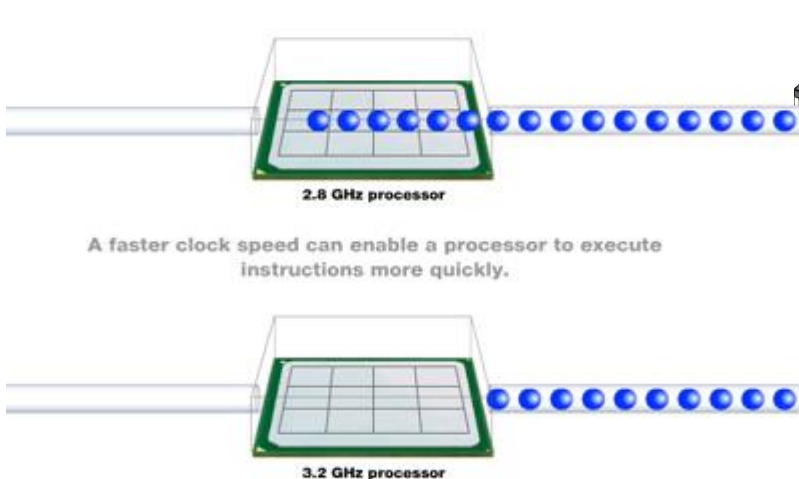
Units of Clock Speed Measurement

$$1 \text{ hertz} = \frac{1 \text{ cycle}}{1 \text{ second}}$$

$$1 \text{ MHz} = \frac{1,000,000 \text{ cycles}}{1 \text{ second}}$$

$$1 \text{ GHz} = \frac{1,000,000,000 \text{ cycles}}{1 \text{ second}}$$

$$1 \text{ GHz} = 1,000 \text{ MHz}$$

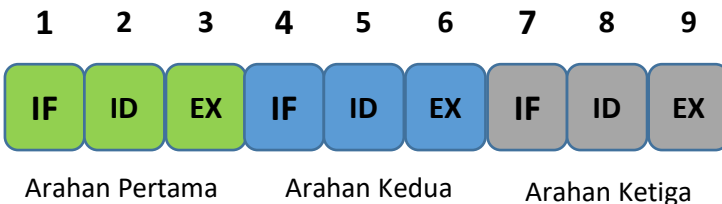


PEMROSESAN

PEMROSESAN LINEAR

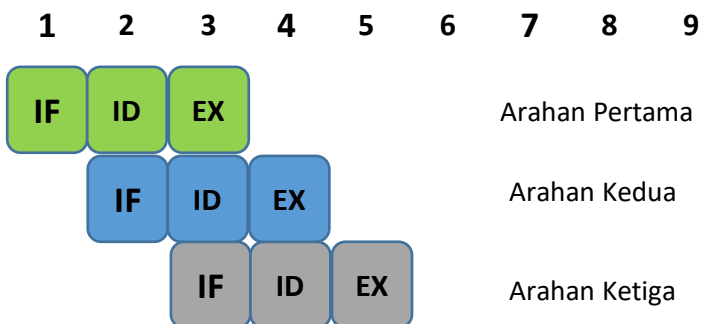
- ✓ Juga dikenali sebagai **pemprosesan bersiri**.
- ✓ Pemproses (CPU) memproses data dan arahan secara berjujukan.
- ✓ Mengikut Kitaran Mesin : **Capai (IF) – Nyahkod (ID) – Laksana (EX)**.
- ✓ Hanya sesuai untuk melaksanakan arahan yang ringkas.

Masa (Milisaat) →



PELAKSANAAN ARAHAN SECARA LINEAR

Masa (Milisaat) →



PELAKSANAAN ARAHAN MENGGUNAKAN TEKNIK TALIAN PAIP (PIPELINING)

PEMROSESAN SEMASA

- ✓ Juga dikenali sebagai **pemprosesan tidak linear atau pemprosesan selari**.
- ✓ Boleh dikonfigurasi untuk melaksanakan pelbagai tugas dalam satu masa (*multitasking*).
- ✓ Mampu melaksanakan beberapa arahan secara serentak dengan menggunakan Teknik talian paip (*pipelining*).
- ✓ Teknik ini digunakan untuk mempertingkatkan prestasi pemprosesan semasa.
- ✓ Teknik ini membenarkan pemprosesan arahan-arahan secara bertindih (*overlapping*).

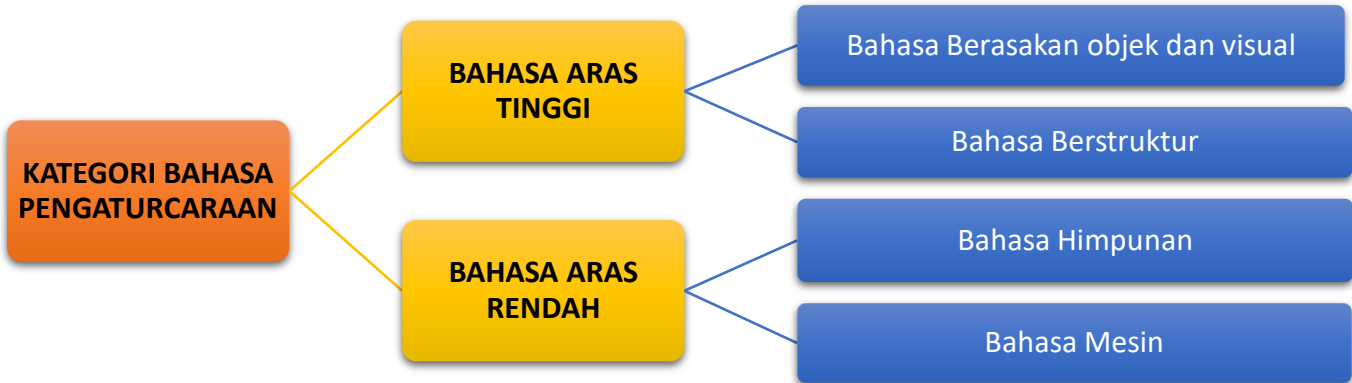
PERBEZAAN

PEMROSESAN LINEAR

- Kurang efektif dalam pemprosesan data dan arahan yang panjang dan kompleks.
- Hanya boleh melakukan satu tugas pada satu masa.
- Proses lebih perlahan

PEMROSESAN SEMASA

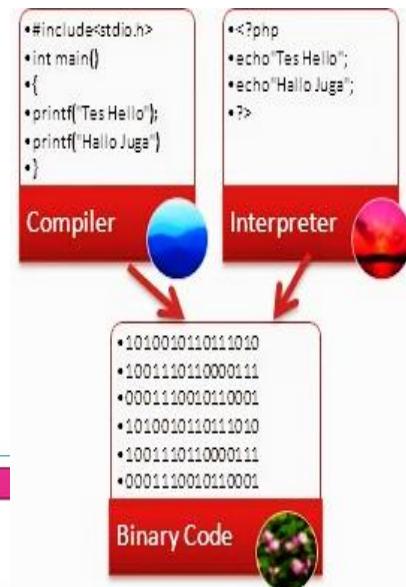
- Efektif dalam pemprosesan data dan arahan yang panjang dan kompleks.
- Arahan –arahan dibahagikan kepada beberapa bahagian dan dilaksanakan secara berasingan. Melaksanakan pelbagai tugas dalam satu masa.
- Proses lebih cepat.



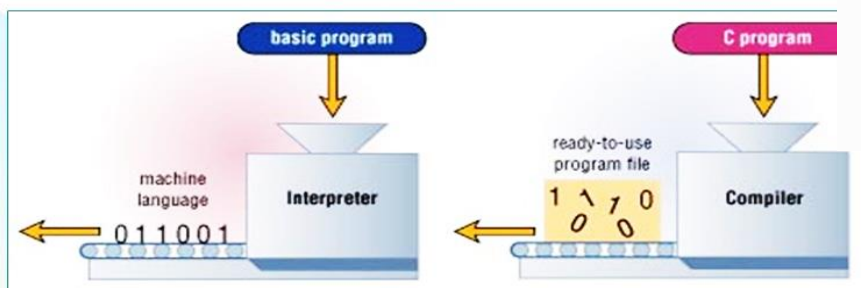
KATEGORI BAHASA PENGATURCARAAN		
BAHASA PENGATURCARAAN ARAS TINGGI	✓ Menggunakan pernyataan yang mudah difahami oleh pengatur cara. ✓ Bahasaanya hampir sama dengan Bahasa tabii manusia. ✓ 2 Jenis : Bahasa Pengaturcaraan Berasaskan Objek dan Bahasa Berstruktur. ✓ Lebih mudah difahami dan digunakan untuk membina atur cara.	<pre> def add5(x): return x+5 def dotwrite(ast): nodename = getNodeName() label=symbol.sym_name.get(int(ast[0]),ast[0]) print ' %s [label=%s]' % (nodename, label) if isinstance(ast[1], str): if ast[1].strip(): print ' %s' % ast[1] else: print ' ' else: print ' ' children = [] for i in n, childrenumerate(ast[1:]): children.append(dotwrite(child)) print ' %s -> {' % nodename for i in namechildren: print ' %s' % name, </pre>
BAHASA PENGATURCARAAN ARAS RENDAH	BAHASA HIMPUNAN ✓ Pernyataan yang digunakan untuk membina arahan bagi komputer dalam bentuk simbol yang ringkas. ✓ Pengatur cara perlu memahami seni bina pemproses dan fungsi setiap perkakasan dalam sistem komputer.	<pre> lw Sto, 0(S2) lw St1, 4(S2) sw St1, 0(S2) sw St0, 4(S2) </pre>
	BAHASA MESIN ✓ Bahasa yang sukar difahami dan dimanipulasi. ✓ Bentuk pernyataan Bahasa mesin bergantung pada jenis komputer. ✓ Atur cara dalam Bahasa mesin dibina daripada jujukan digit perduaan (0, 1)	<pre> 0110 1010 1111 0 1000 0000 1001 1 1111 0101 1000 0 1001 1100 0110 1 </pre>

PENTERJEMAH BAHASA PENGATURCARAAN

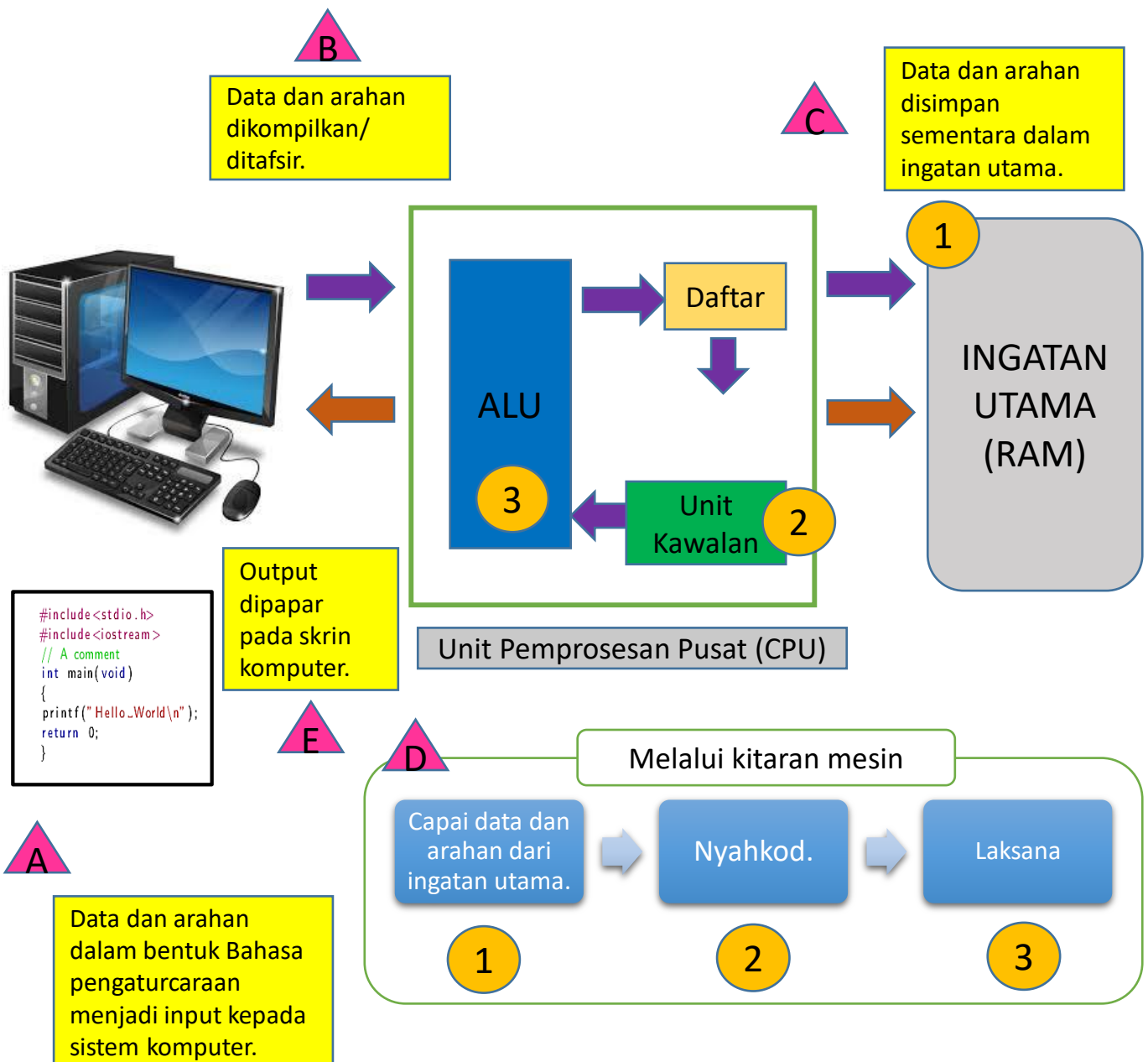
- Komputer melaksanakan tugas dengan mengikut arahan atau set arahan yang ditulis oleh pengatur cara dengan menggunakan pelbagai jenis Bahasa pengaturcaraan.
- **Komputer hanya memahami Bahasa dalam bentuk kod mesin (digit perduaan 0, 1).**
- Bahasa pengaturcaraan perlu diterjemahkan kepada bentuk Bahasa mesin sebelum difahami oleh komputer.
- Tugas penterjemah : menukarkan arahan yang difahami manusia kepada bentuk yang difahami oleh mesin.
- **Bahasa aras tinggi, Bahasa aras rendah → Bahasa Mesin.**
- Proses penterjemahan Bahasa aras tinggi dilakukan samada oleh **pengkompil (compiler)** atau **pentafsir (interpreter)**.
- **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.



Compilers and Interpreters



BAHASA PENGATURCARAAN ARAS TINGGI	PENKOMPIL	PENTAFSIR
C , C++ , COBOL	✓	
Python, Perl, PHP, Ruby		✓
Java, Visual Basic, C#	✓	✓



GET LOGIK

1.3.1

- Menerangkan get logik sebagai binaan asas litar bersepadu

1.3.2

- Menyenaraikan dan melukis simbol get logik berikut:
 - (i) get logik asas (TAK, DAN,ATAU)
 - (ii) kombinasi get logik (TAK DAN,TAK ATAU, XATAU, XTAKATAU)

1.3.3

- Membina Jadual Kebenaran dan Ungkapan Boolean bagi menerangkan tindakan get logik berikut:
 - (i) get logik asas (TAK, DAN, ATAU)
 - (ii) kombinasi get logik (TAK DAN,TAK ATAU, XATAU, XTAKATAU)

1.3.4

- Membina Jadual Kebenaran bagi satu litar get logik gabungan yang mempunyai dua input.

1.3.5

- Membina Ungkapan Boolean bagi satu litar get logik gabungan yang mempunyai dua input

1.3.6

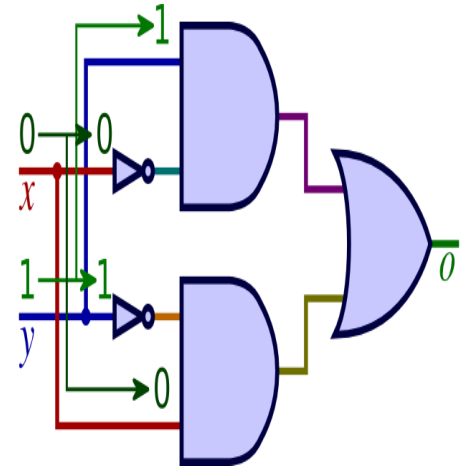
- Menghasilkan litar get logik dan jadual kebenaran berdasarkan pernyataan logik

1.3.7

- Menghasilkan litar get logik dengan jadual kebenaran dan Ungkapan Boolean yang lengkap dalam menyelesaikan masalah

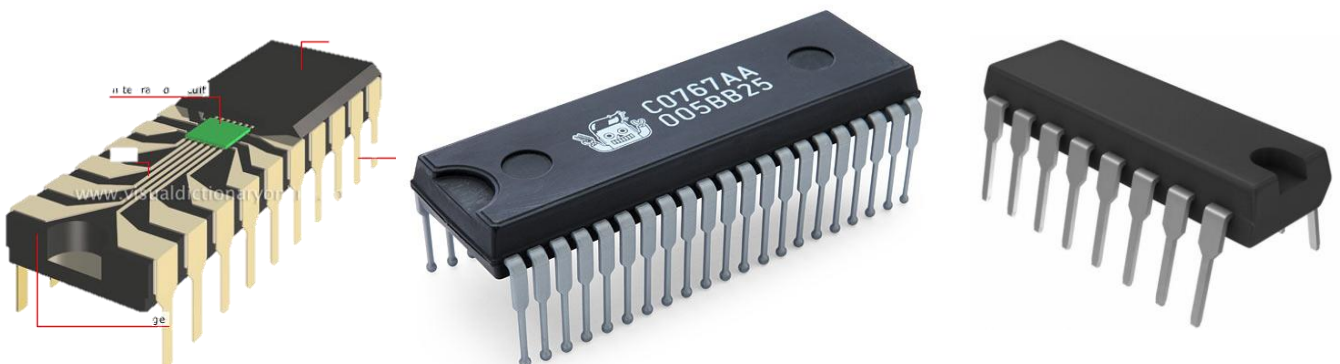
GET LOGIK

- Merupakan blok binaan asas litar bersepadu.
- Mempunyai 2 input dan 1 output.
- Menerima input dan menghasilkan output dalam bentuk perduaan.
- Logik perduaan menggunakan 2 keadaan.
- 0 = tiada arus @ arus rendah.
- 1 = ada arus @ arus tinggi.



LITAR BERSEPADU

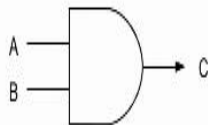
- Merupakan litar elektronik yang terdiri daripada gabungan komponen elektronik (transistor, perintang, diod dan kapasitor) yang perlu untuk melaksanakan pelbagai fungsi get logik.
- Get-get ini disambungkan menjadi satu litar yang lengkap dan dikecilkan beribu-ribu kali ganda daripada saiz asal.
- Kemudian ditempatkan pada satu kepingan silikon dan dinamakan cip atau cip mikro (serpihan).
- Lembaran emas digunakan untuk menyambung serpihan pada pin-pin litar bersepadu.
- Perumah plastik atau logam biasanya digunakan untuk melindungi serpihan agar tidak rosak.
- Semakin banyak pin pada litar bersepadu, semakin banyak fungsinya.
- Cip litar bersepadu terbahagi kepada beberapa jenis berdasarkan bilangan get di dalamnya.
- Penggunaan litar bersepadu sangat meluas dalam pelbagai peranti elektronik.



GET LOGIK ASAS

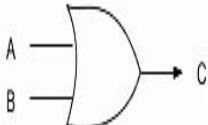
GET DAN(AND GATE)

Simbol



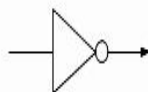
GET ATAU(OR GATE)

Simbol



GET TAK(NOT GATE)

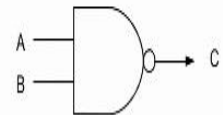
Simbol



KOMBINASI GET LOGIK

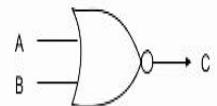
GET TAK DAN(NAND GATE)

Simbol



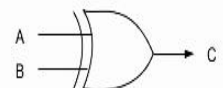
GET TAK ATAU(NOR GATE)

Simbol



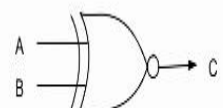
GET EKSklusif ATAU(EX-OR)

Simbol



Get Eksklusif TAK ATAU

Simbol



JADUAL KEBENARAN

- Digunakan untuk menyemak output yang dihasilkan daripada get-get logik atau daripada litar logik.
- Juga digunakan untuk membuktikan teorem algebra *boolean*.
- Bagi membina jadual kebenaran, bilangan input perlu dipertimbangkan.
- Input boleh diwakilkan dalam bentuk pemboleh ubah (A,B, C, ..X,Y,Z).
- Bilangan input akan menentukan kebarangkalian bagi gabungan input yang melalui get.

A	B	C	$Y = A\bar{B} + \bar{A}C$
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

BILANGAN PEMBOLEH UBAH INPUT	JUMLAH KEBARANGKALIAN INPUT
1	$2^1 = 2$
2	$2^2 = 4$
3	$2^3 = 8$
4	$2^4 = 16$

CONTOH

INPUT
A
0
1

INPUT	
A	B
0	0
0	1
1	0
1	1

INPUT		
A	B	C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

INPUT			
A	B	C	D
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

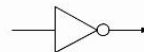
GET DAN(AND GATE)

Simbol			
Jadual kebenaran	A	B	C
	0	0	0
	0	1	0
	1	0	0
	1	1	1
Formula	$C = A \cdot B$ @ $C = A \times B$		

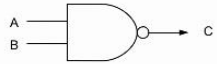
GET ATAU(OR GATE)

Simbol			
Jadual kebenaran	A	B	C
	0	0	0
	0	1	1
	1	0	1
	1	1	1
Formula	$C = A + B$		

GET TAK(NOT GATE)

Simbol			
Jadual kebenaran	A	C	
	0	1	
	1	0	
Formula	$C = \overline{A}$		

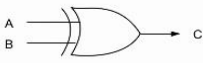
GET TAK DAN(NAND GATE)

Simbol			
Jadual kebenaran	A	B	C
	0	0	1
	0	1	1
	1	0	1
	1	1	0
Formula	$C = \overline{A \cdot B}$		

GET TAK ATAU(NOR GATE)

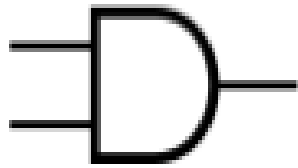
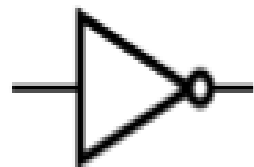
Simbol			
Jadual kebenaran	A	B	C
	0	0	1
	0	1	0
	1	0	0
	1	1	0
Formula	$C = \overline{A + B}$		

GET EKSKLUSIF ATAU(EX-OR)

Simbol			
Jadual kebenaran	A	B	C
	0	0	0
	0	1	1
	1	0	1
	1	1	0
Formula	$C = \overline{A} \cdot B + A \cdot \overline{B} = A \oplus B$		

Get Eksklusif TAK ATAU

Simbol			
Jadual kebenaran	A	B	C
	0	0	1
	0	1	0
	1	0	0
	1	1	1
Formula	$C = \overline{A} \cdot \overline{B} + A \cdot B = A \oplus B$		

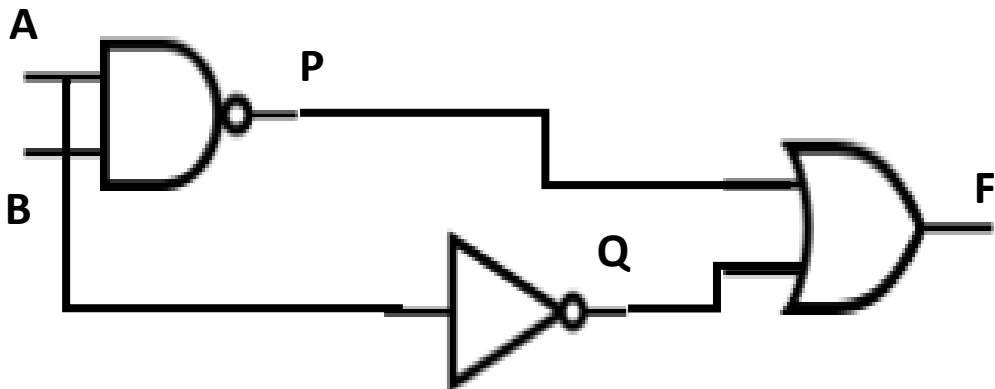
NAMA GET	SIMBOL GET	UNGKAPAN BOOLEAN	JADUAL KEBENARAN																		
DAN		$F = A.B$ Dibaca : A DAN B A X B	<table><tr><th colspan="2">INPUT</th><th>OUTPUT</th></tr><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></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><tr><th colspan="2">INPUT</th><th>OUTPUT</th></tr><tr><th>A</th><th>B</th><th>F</th></tr><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></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																			
TAK		$F = \overline{A}$ Dibaca : F mempunyai nilai yang bertentangan dengan A	<table><tr><th>INPUT</th><th>OUTPUT</th></tr><tr><th>A</th><th>$F = \overline{A}$</th></tr><tr><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td></tr></table>	INPUT	OUTPUT	A	$F = \overline{A}$	0	0	0	1										
INPUT	OUTPUT																				
A	$F = \overline{A}$																				
0	0																				
0	1																				

NAMA GET	SIMBOL GET	UNGKAPAN BOOLEAN	JADUAL KEBENARAN																								
TAK DAN		$F = \overline{(A \cdot B)}$ Dibaca : A DAN B bar	<table> <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> <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> </table>	INPUT		OUTPUT		A	B	A.B	F	0	0	0	1	0	1	0	1	1	0	0	1	1	1	1	0
INPUT		OUTPUT																									
A	B	A.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> <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> <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> </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		$F = A \oplus B$ Dibaca : Jika kedua-duanya sama, Output = 0	<table> <tr> <th colspan="2">INPUT</th><th colspan="2">OUTPUT</th></tr> <tr> <th>A</th><th>B</th><th>A ⊕ B</th><th></th></tr> <tr> <td>0</td><td>0</td><td>0</td><td></td></tr> <tr> <td>0</td><td>1</td><td>1</td><td></td></tr> <tr> <td>1</td><td>0</td><td>1</td><td></td></tr> <tr> <td>1</td><td>1</td><td>0</td><td></td></tr> </table>	INPUT		OUTPUT		A	B	A ⊕ B		0	0	0		0	1	1		1	0	1		1	1	0	
INPUT		OUTPUT																									
A	B	A ⊕ B																									
0	0	0																									
0	1	1																									
1	0	1																									
1	1	0																									
XTAKATAU		$F = \overline{(A \oplus B)}$ Dibaca : Output XATAU diterbalikkan	<table> <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> <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> </table>	INPUT		OUTPUT		A	B	A ⊕ B	F	0	0	0	1	0	1	1	0	1	0	1	0	1	1	0	1
INPUT		OUTPUT																									
A	B	A ⊕ B	F																								
0	0	0	1																								
0	1	1	0																								
1	0	1	0																								
1	1	0	1																								

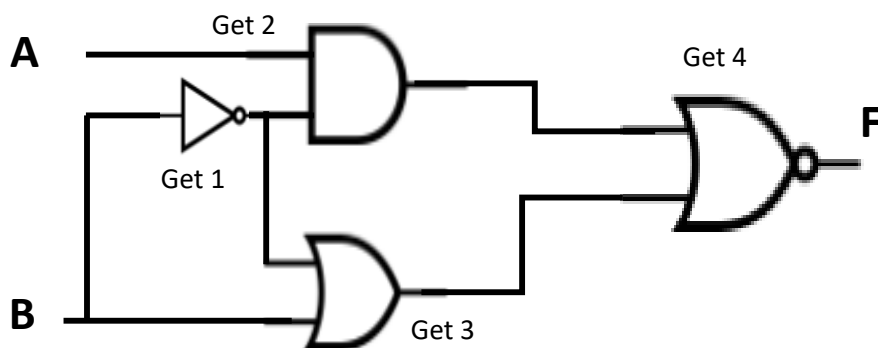
1.3

1.3.4

MEMBINA JADUAL KEBENARAN BAGI SATU LITAR LOGIK GABUNGAN YANG MEMPUNYAI DUA INPUT



INPUT		P	Q	OUTPUT
A	B	$(A \cdot B)$	A	$F = AB + A$
0	0	1	1	1
0	1	1	1	1
1	0	1	0	1
1	1	0	0	0

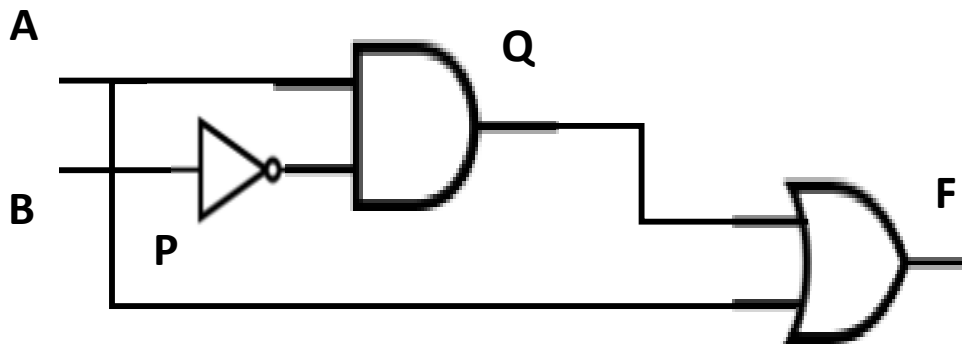


INPUT		GET 1	GET 2	GET 3	OUTPUT
A	B				
0	0				
0	1				
1	0				
1	1				

1.3

1.3.5

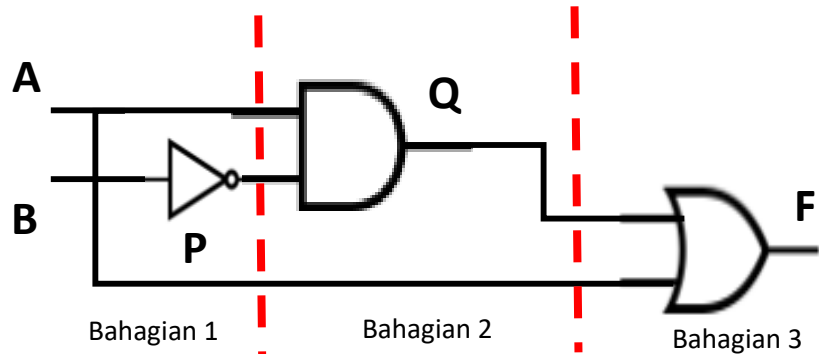
MEMBINA UNGKAPAN BOOLEAN BAGI SATU LITAR LOGIK GABUNGAN YANG MEMPUNYAI DUA INPUT



LANGKAH PENYELESAIAN

LANGKAH 1

Bahagikan litar kepada 3 bahagian berdasarkan get logik yang terlibat.



LANGKAH 2

Selesaikan Bahagian 1

- Melibatkan **get logik TAK**.

INPUT	OUTPUT
B	$\overline{B}$

LANGKAH 3

Selesaikan Bahagian 2

- Melibatkan **get logik DAN**.

INPUT		OUTPUT
A	P	$Q = A \cdot \overline{B}$

LANGKAH 4

Selesaikan Bahagian 3

- Melibatkan **get logik ATAU**.

INPUT		OUTPUT
Q	A	$F = A \cdot \overline{B} + A$

JAWAPAN

$$F = A \cdot \overline{B} + A$$

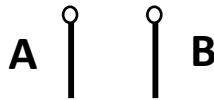
CONTOH

UNGKAPAN BOOLEAN	$F = \bar{A}\bar{B} + AB + A\bar{B}$
UNGKAPAN LOGIK	Nilai F = 1 jika (A = 0 DAN B = 0) ATAU (A = A DAN B = 1) ATAU (A = 1 DAN B = 0)

LANGKAH PENYELESAIAN

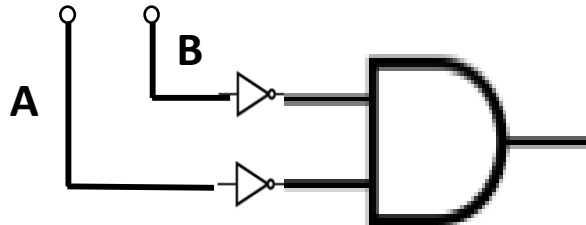
LANGKAH 1

Kira bilangan pemboleh ubah atau input yang terlibat.



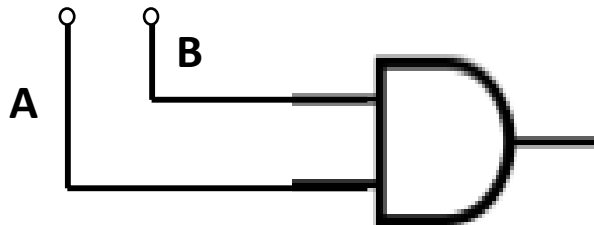
LANGKAH 2

Tentukan get yang terlibat bagi $\bar{A}\bar{B}$



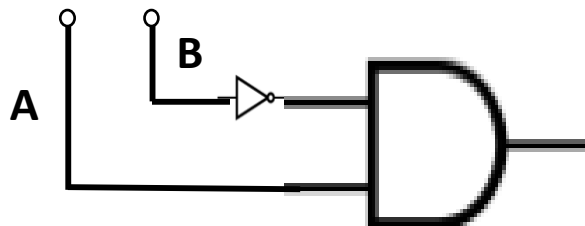
LANGKAH 3

Tentukan get yang terlibat bagi AB



LANGKAH 4

Tentukan get yang terlibat bagi $A\bar{B}$



LANGKAH PENYELESAIAN

LANGKAH 5

Kenalpasti get yang menghubungkan kesemua ungkapan yang terlibat.

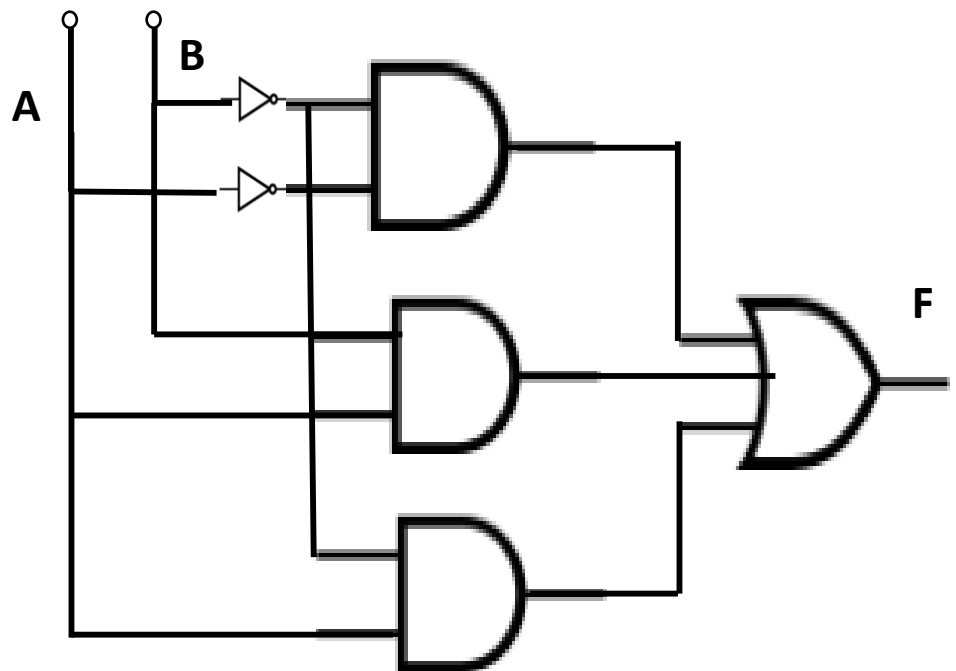
$$F = \bar{A}\bar{B} + AB + A\bar{B}$$

Get ATAU

Get ATAU

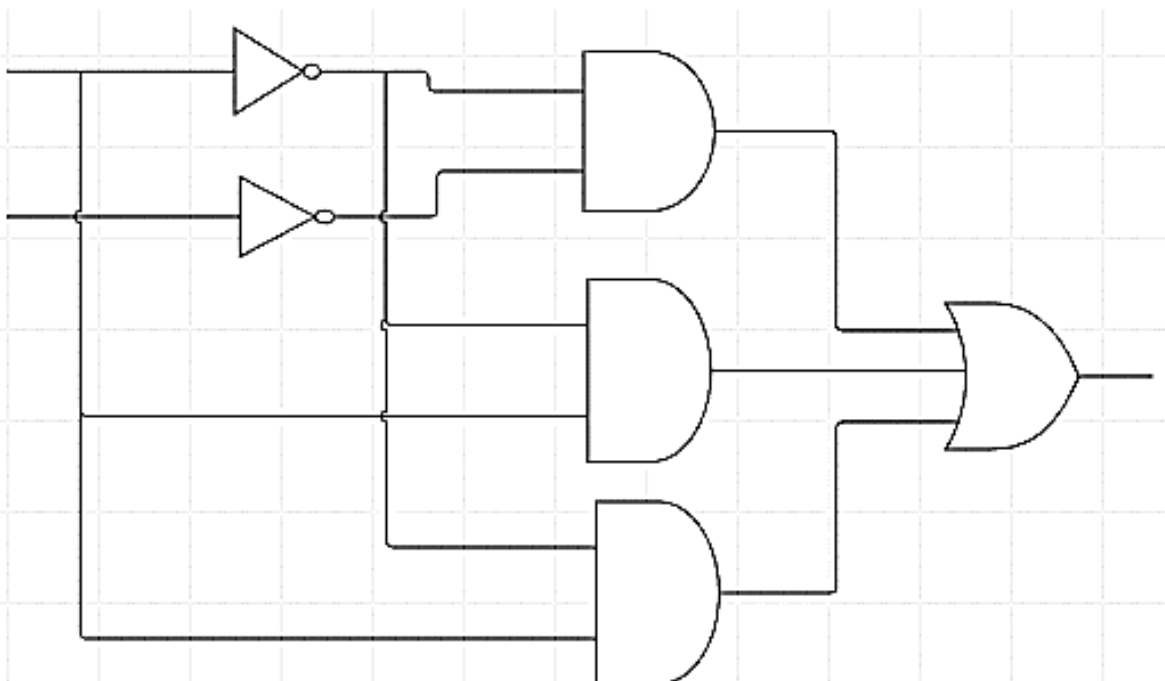
LANGKAH 6

Gabungkan ketiga-tiga rajah.



JADUAL KEBENARAN

A	B	$\overline{A}$	$\overline{B}$	$\overline{A} \overline{B}$	$A \overline{B}$	$A B$	F
0	0	1	1	1	0	0	1
0	1	1	0	0	0	0	0
1	0	0	1	0	0	1	1
1	1	0	0	0	1	0	1



CONTOH MASALAH

Sistem penggera dibina menggunakan litar logik 2 input iaitu A dan B. Sistem akan mengeluarkan isyarat bunyi F, jika input bagi A mewakili ON dan input B mewakili ON atau jika input A mewakili OFF dan input B mewakili OFF.

LANGKAH 1

Tulis ungkapan logik bagi mewakili senario yang diberikan.

$$F = 1, \text{ jika } (A = 1 \text{ DAN } B = 1) \text{ ATAU } (A = 0 \text{ DAN } B = 0)$$

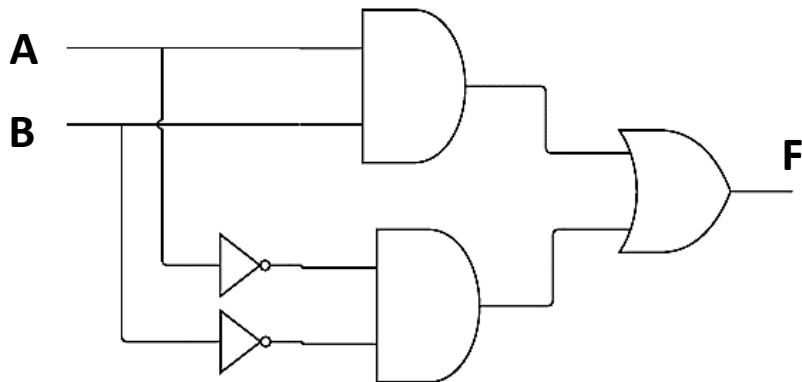
LANGKAH 2

Tulis semula dalam bentuk ungkapan *Boolean*.

$$F = A.B + \overline{A}.\overline{B}$$

LANGKAH 3

Lukis rajah litar logik bagi mewakili ungkapan Boolean



LANGKAH 4

Bina jadual kebenaran

INPUT		PERALIHAN		OUTPUT
A	B	A . B	$\overline{A}.\overline{B}$	$F = A.B + \overline{A}.\overline{B}$
0	0	0	1	1 (Siren berbunyi)
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1 (Siren berbunyi)

2.0

PANGKALAN DATA LANJUTAN





2.0

**PANGKALAN
DATA LANJUTAN**

**2.1 Bahasa
Pertanyaan
Berstruktur : SQL**

BAHASA PERTANYAAN BERSTRUKTUR : SQL (STRUCTURED QUERY LANGUAGE)

2.1.1

- Melakar ERD(*Entity Relationship Diagram*) bagi permasalahan yang diberi .

2.1.2

- Menghasilkan skema hubungan yang ternormal (*normalize*) daripada permasalahan yang diberi

2.1.3

- Membina pangkalan data hubungan menggunakan SQL: Table

2.1.4

- Menggunakan SQL untuk memanipulasi data:
(i) Sisip/Tambah (ii) Padam (iii) Kemaskini

2.1.5

- Menggunakan SQL SELECT untuk mendapatkan semula data menggunakan:
(i) GROUP BY (ii) ORDER BY

1.3.6

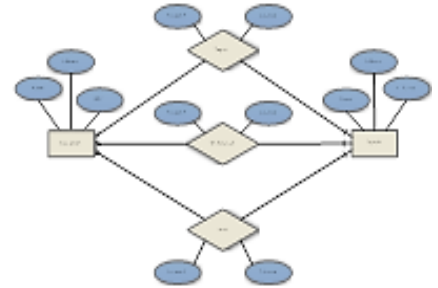
- Menghasilkan litar get logik dan jadual kebenaran berdasarkan pernyataan logik

1.3.7

- Menghasilkan litar get logik dengan jadual kebenaran dan Ungkapan Boolean yang lengkap dalam menyelesaikan masalah

ERD

ULANGKAJI



KOMPONEN ERD

ENTITI

- ✓ Objek atau konsep yang unik dan dapat dikenal pasti dalam persekitaran tertentu seperti orang, tempat, benda atau peristiwa yang dikutip dan disimpan sebagai data.
- ✓ Contoh : MURID, GURU, BUKU, PEPERIKSAAN.
- ✓ Diwakili dalam bentuk 4 segi empat tepat.

MURID

ATRIBUT

- ✓ Ciri-ciri yang menghuraikan sesuatu entiti.
- ✓ Atribut ialah sifat yang memerihalkan sesuatu set entiti.
- ✓ Diwakili dalam bentuk bujur.
- ✓ **KUNCI PRIMER** : atribut unik yang membezakan rekod dalam pangkalan data.
- ✓ Setiap entiti hanya mempunyai satu kunci primer sahaja.

Nama

HUBUNGAN

- ✓ Hubungan ialah digunakan untuk menyatukan satu atau lebih entiti yang terdapat dalam ERD.
- ✓ Menunjukkan bagaimana dua entiti berkongsi maklumat dalam pangkalan data.
- ✓ Hubungan dilabelkan dengan kata kerja.
- ✓ Contoh : mengajar, mendaftar, menginap.

Hubungan

JENIS ENTITI	
ENTITI KUAT	Boleh wujud tanpa bersandar pada entiti lain.
ENTITI LEMAH	Bersandar pada entiti kuat.



JENIS ATRIBUT

JENIS-JENIS ATRIBUT

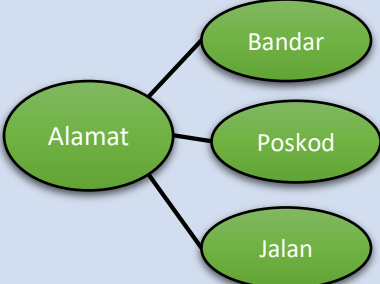
ATRIBUT
MUDAH

ATRIBUT NILAI
GANDAAN

ATRIBUT
TERBITAN

ATRIBUT
KUNCI

ATRIBUT
KOMPOSIT

NOTASI	KETERANGAN
	✓ Digunakan untuk mewakili atribut mudah dan atribut bernilai tunggal. ✓ ATRIBUT MUDAH : Tidak boleh dibahagikan kepada beberapa bahagian. Cth : Nama, NoKadPengenalan. ✓ ATRIBUT BERNILAI TUNGGAL : Mempunyai satu nilai sahaja. Cth : NOKadPengenalan.
	✓ ATRIBUT NILAI GANDAAN : Mempunyai lebih daripada satu nilai . ✓ Contoh : Kemahiran – Seorang guru mempunyai lebih daripada satu kemahiran.
	✓ ATRIBUT TERBITAN: Terhasil berasaskan atribut lain. ✓ Contoh : Gaji Tahunan – Gaji Tahunan seorang guru diperoleh dengan mengira gaji bulanan x 12.
	✓ ATRIBUT KUNCI: Adalah unik. Membezakan setiap rekod dalam pangkalan data. ✓ Contoh : NoKadPengenalan, NoKakitangan
	✓ ATRIBUT KOMPOSIT : Atribut yang boleh dipecahkan kepada beberapa bahagian. ✓ Contoh : Alamat boleh dibahagikan kepada Jalan, Bandar dan Poskod.

KEKARDINALAN

- Menunjukkan hubungan dari segi kuantiti data antara suatu entity dengan entiti yang lain.

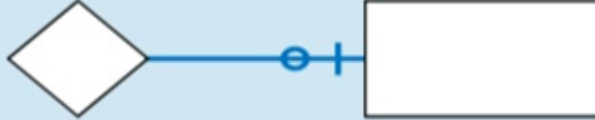
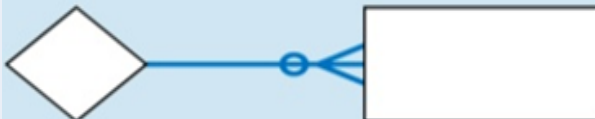
JENIS KEKARDINALAN

KEKARDINALAN MINIMUM

- ✓ Minimum data suatu entiti berkait dengan entiti lain.
- ✓ Contoh : Seorang GURU mengajar sekurang-kurangnya satu MATAPELAJARAN.
- ✓ Maka kekardinalan minimum ialah satu kerana seorang guru mesti mengajar sekurang-kurangnya satu mata pelajaran.
- ✓ Tidak ada guru yang tidak akan mengajar.

KEKARDINALAN MAKSIMUM

- ✓ Maksimum data suatu entiti berkait dengan entiti lain.
- ✓ Contoh : Satu MATAPELAJARAN boleh diajar oleh seorang GURU atau lebih. Seorang guru boleh mengajar maksimum lebih dari satu mata pelajaran.
- ✓ Maka kekardinalan maksimum adalah banyak.

SIMBOL	KETERANGAN
	MANDATORI SATU Minimum dan maksimum 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.

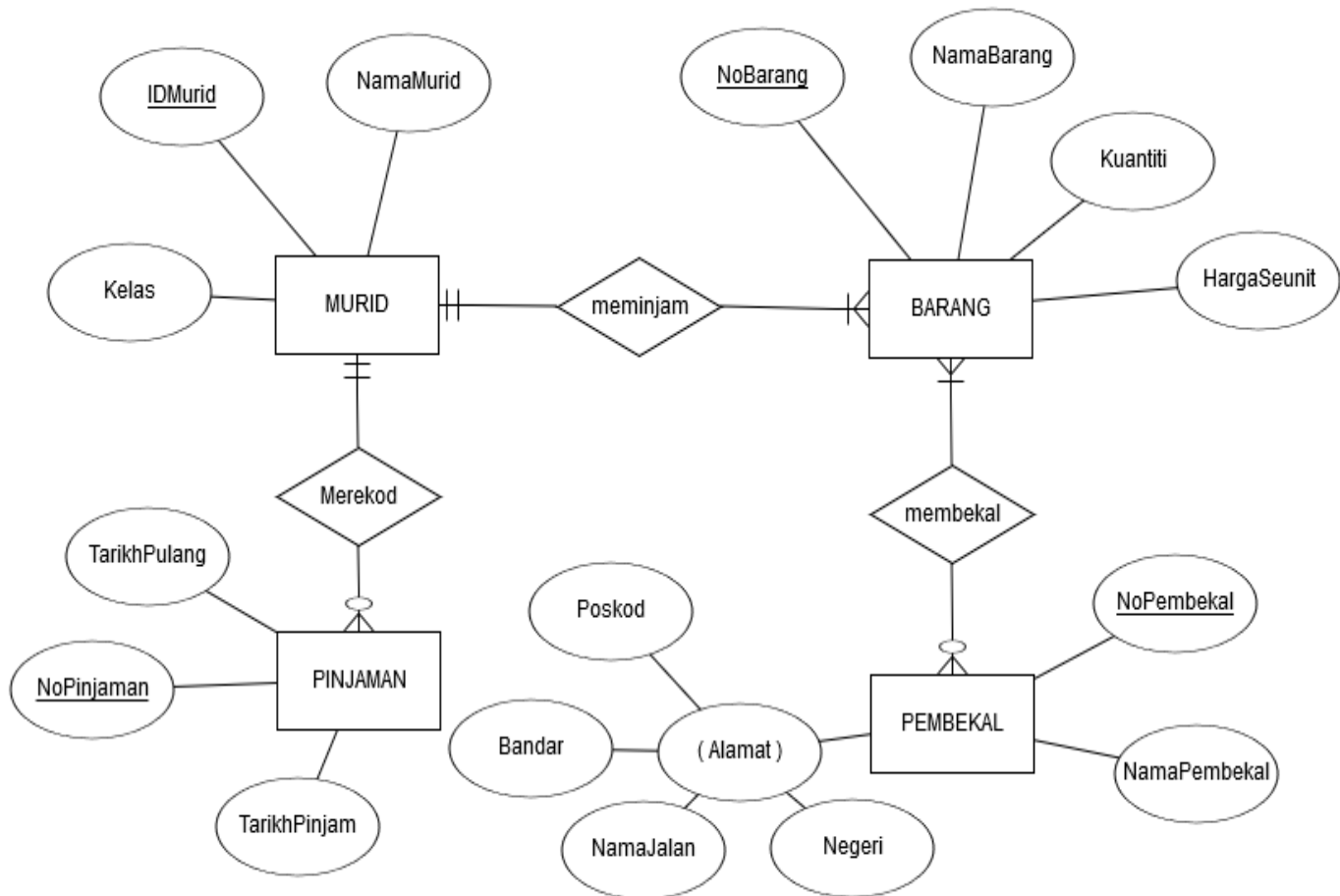
CONTOH

SOALAN

Pengurusan stok dan inventori alatan sukan amat penting dalam kejayaan sesuatu aktiviti sukan dan kokurikulum. Kekurangan peralatan sukan menyebabkan sesuatu sukan sukar dijalankan. Pangkalan data stor sukan mengandungi rekod pelbagai alatan sukan sama ada untuk sukan sekolah ataupun untuk pembelajaran dan pengajaran mata pelajaran Pendidikan jasmani. Dalam Pangkalan data ini juga, maklumat barangan sukan, pembekal, peminjam dan rekod pinjaman yang dibuat oleh semua murid disimpan.

ENTITI DAN ATRIBUT YANG TERLIBAT	ENTITI	ATRIBUT
	BARANG	NoBarang , NamaBarang , Kuantiti , HargaSeunit
	MURID	IDMurid , NamaMurid , Kelas
	PINJAMAN	NoPinjaman , TarikhPinjaman , TarikhPulang
	PEMBEKAL	NoPembekal , NamaPembekal dan Alamat (NamaJalan, Bandar , Poskod dan Negeri)
KENALPASTI KUNCI PRIMER	ENTITI	ATRIBUT
	BARANG	NoBarang
	MURID	IDMurid
	PINJAMAN	NoPinjaman
	PEMBEKAL	NoPembekal
KENALPASTI HUBUNGAN ANTARA ENTITI	▪ Meminjam ▪ Merekod ▪ Membekal	

CONTOH



2.1

2.1.2

MENGHASILKAN SKEMA HUBUNGAN YANG TERNORMAL (*NORMALIZE*) DARIPADA PERMASALAHAN YANG DIBERI

MODEL HUBUNGAN DATA

- Menyimpan data dalam bentuk jadual hubungan.
- Dibina selepas melakar ERD.
- Memberi gambaran tentang struktur jadual dalam sistem pangkalan data yang ingin dibangunkan.

Activity Code	Activity Name
23	Patching
24	Overlay
25	Crack Sealing

Key = 24

Activity Code	Date	Route No.
24	01/12/01	I-95
24	02/08/01	I-66

Date	Activity Code	Route No.
01/12/01	24	I-95
01/15/01	23	I-495
02/08/01	24	I-66

KUNCI HUBUNGAN

- Setiap hubungan perlu ada kunci primer yang membolehkan menyimpan dan mencapai data dalam hubungan.
- **KUNCI PRIMER** : Atribut unik yang dipilih untuk mengenal pasti rekod dalam hubungan.
- Kunci Primer digariskan seperti contoh dibawah.

MURID (NoKadPengenalan, Nama, Alamat)

- **KUNCI ASING** : Kunci Primer sesuatu hubungan yang muncul dalam hubungan lain dalam pangkalan data yang sama.
- Kunci ditandakan dengan garis putus-putus.

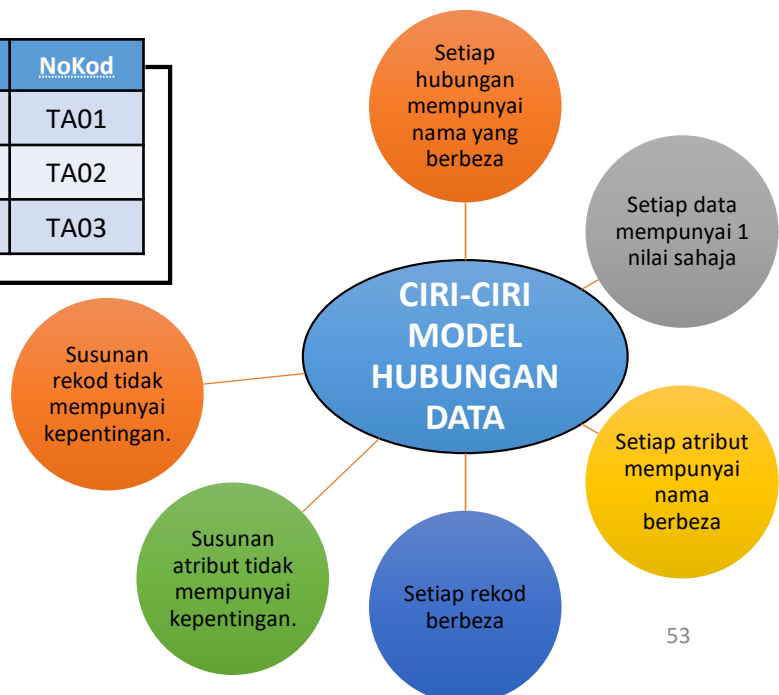
MURID (NoKadPengenalan, Nama, Alamat, NoKod)
 MATAPELAJARAN (NoKod, NamaMataPelajaran

MURID

NoKadPengenalan	Nama	Alamat	NoKod
991002021111	Airis	Shah Alam	TA01
990203042222	Amir	PJ	TA02
990110105555	Naufal	Kelana Jaya	TA03

MATAPELAJARAN

<u>NoKod</u>	Nama
TA01	Bahasa Melayu
TA02	Matematik
TA03	Sejarah



SKEMA HUBUNGAN

- Menghuraikan struktur logik secara keseluruhan sesuatu pangkalan data.

MURID

NoKadPengenalan	Nama	Alamat
-----------------	------	--------

PEMETAAN ERD KEPADA SKEMA HUBUNGAN

Entiti ditukarkan
menjadi nama
hubungan.



Atribut akan menjadi
nama medan.



Jenis hubungan dalam
ERD akan menentukan
pemilihan kunci asing.

JENIS-JENIS HUBUNGAN ERD

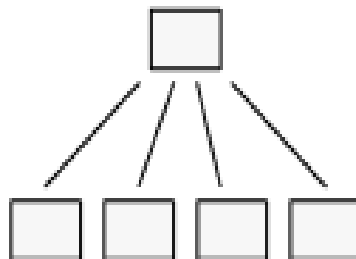
Hubungan
satu-ke-satu

Hubungan
satu-ke-banyak

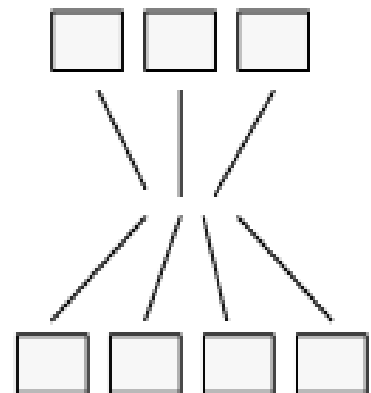
Hubungan
banyak-ke-banyak



one-to-one



one-to-many



many-to-many

CONTOH

ERD	SKEMA HUBUNGAN								
<pre>graph LR P((PENGETUA)) --- NoID PK1(()) P --- Nama PK2(()) P --- Alamat PK3(()) S((SEKOLAH)) --- NoSekolah PK4(()) S --- Nama Sekolah PK5(()) P --- mengurus S style PK1 fill:none,stroke:none style PK2 fill:none,stroke:none style PK3 fill:none,stroke:none style PK4 fill:none,stroke:none style PK5 fill:none,stroke:none</pre>	PENGETUA <table><tr><td><u>NoID</u></td><td>Nama</td><td>Alamat</td></tr></table> SEKOLAH <table><tr><td><u>NoSekolah</u></td><td>NamaSekolah</td><td>NoID</td></tr></table> ✓ Kunci asing perlu dipilih pada entiti yang berada pada bahagian mandatori - (PENGETUA)	<u>NoID</u>	Nama	Alamat	<u>NoSekolah</u>	NamaSekolah	NoID		
<u>NoID</u>	Nama	Alamat							
<u>NoSekolah</u>	NamaSekolah	NoID							
<pre>graph LR P((PELANGGAN)) --- NoPelanggan PK1(()) P --- Nama PK2(()) P --- Alamat PK3(()) S((PESANAN)) --- NoPesanan PK4(()) S --- TarikhPesanan PK5(()) P --- mengurus S style PK1 fill:none,stroke:none style PK2 fill:none,stroke:none style PK3 fill:none,stroke:none style PK4 fill:none,stroke:none style PK5 fill:none,stroke:none</pre>	PELANGGAN <table><tr><td><u>NoPelanggan</u></td><td>Nama</td><td>Alamat</td></tr></table> PESANAN <table><tr><td><u>NoPesanan</u></td><td>TarikhPesanan</td><td>NoPelanggan</td></tr></table> ✓ Kunci primer pada bahagian satu akan menjadi kunci asing pada bahagian banyak.	<u>NoPelanggan</u>	Nama	Alamat	<u>NoPesanan</u>	TarikhPesanan	NoPelanggan		
<u>NoPelanggan</u>	Nama	Alamat							
<u>NoPesanan</u>	TarikhPesanan	NoPelanggan							
<pre>graph LR P((PEMBEKAL)) --- NoPembekal PK1(()) P --- Nama PK2(()) P --- Alamat PK3(()) S((BAHAN MENTAH)) --- NoBahan PK4(()) S --- Kos PK5(()) S --- JumlahUnit PK6(()) P --- membekal S style PK1 fill:none,stroke:none style PK2 fill:none,stroke:none style PK3 fill:none,stroke:none style PK4 fill:none,stroke:none style PK5 fill:none,stroke:none style PK6 fill:none,stroke:none</pre>	PEMBEKAL <table><tr><td><u>NoPembekal</u></td><td>Nama</td><td>Alamat</td></tr></table> PEMBEKAL_BAHAN_MENTAH <table><tr><td><u>NoPembekal</u></td><td><u>NoBahan</u></td></tr></table> BAHAN_MENTAH <table><tr><td><u>NoPelanggan</u></td><td>JumlahUnit</td><td>Kos</td></tr></table> ✓ Jadual baharu perlu dibentuk dan kunci primer yang menghubungkan kedua-dua entiti akan menjadi asing dalam jadual ini.	<u>NoPembekal</u>	Nama	Alamat	<u>NoPembekal</u>	<u>NoBahan</u>	<u>NoPelanggan</u>	JumlahUnit	Kos
<u>NoPembekal</u>	Nama	Alamat							
<u>NoPembekal</u>	<u>NoBahan</u>								
<u>NoPelanggan</u>	JumlahUnit	Kos							

SKEMA HUBUNGAN TERNORMAL

- Boleh dihasilkan berdasarkan lakaran ERD.

CONTOH

MURID

<u>IDMurid</u>	NamaMurid	Kelas
----------------	-----------	-------

PINJAMAN

<u>NoPinjaman</u>	TarikhPinjam	TarikhPulang	<u>IDMurid</u>
-------------------	--------------	--------------	----------------

BARANG

<u>NoBarang</u>	NamaBarang	Kuantiti	HargaSeunit	<u>IDMurid</u>
-----------------	------------	----------	-------------	----------------

PEMBEKAL_BARANG

<u>NoBarang</u>	<u>NoPembekal</u>
-----------------	-------------------

PEMBEKAL

<u>NoPembekal</u>	NamaPembekal	NamaJalan	Bandar	Poskod	Negeri
-------------------	--------------	-----------	--------	--------	--------

PENORMALAN

- Proses untuk memastikan model data logikal yang terhasil mempunyai anomali yang paling minimum.
- Penormalan bertujuan untuk menghasilkan skema hubungan dengan menempatkan attribut ke dalam jadual dengan betul bagi mengelakkan masalah anomaly.
- **ANOMALI** – Ralat yang mungkin berlaku apabila pengguna ingin mengemas kini jadual yang mengandungi data berulang.

BENTUK PENORMALAN PERTAMA (1NF)

- **1NF** – Jika tiada atribut yang mempunyai kumpulan data yang berulang.
- Setiap atribut mempunyai satu nilai sahaja.
- Jadual dikatakan dalam bentuk normal pertama apabila kumpulan data berulang dihapuskan.
- Ini dilakukan dengan membentuk **kunci primer**.

CONTOH : Jadual Inventori dalam bentuk ONF

No Barang	Nama Barang	Kuantiti	Harga Seunit	Nama Pembekal	Alamat	IDMurid	Nam Murid	Kelas	Tarikh Pinjam	Tarikh Pulang
A01	Bola Sepak	10	30.50	Angsa Maju Setia Sukan	Alor Setar	M1000	Lim	1 Merah	02/7/2017	02/7/2017
	Bola Jaring	5	15.30		Shah Alam					
A03	Bola Hoki	10	20.00	Indah Sukan	Kuching	M2000	Elisa	4 Biru	07/7/2017	07/7/2017
A05	Baton	20	10.30	Setia Sukan	Shah Alam	M3000	Devi	3 Ungu	09/8/2017	12/8/2017

CONTOH : Jadual BARANG dalam bentuk 1NF (Wujudkan Kunci Primer)

No <u>Barang</u>	Nama Barang	Kuantiti	Harga Seunit	Nama Pembekal	Alamat	IDMurid	Nam Murid	Kelas	Tarikh Pinjam	Tarikh Pulang
A01	Bola Sepak	10	30.50	Angsa Maju	Alor Setar	M1000	Lim	1 Merah	02/7/2017	02/7/2017
A02	Bola Jaring	5	15.30	Setia Sukan	Shah Alam	M1000	Lim	1 Merah	02/7/2017	02/7/2017
A03	Bola Hoki	10	20.00	Indah Sukan	Kuching	M2000	Elisa	4 Biru	07/7/2017	07/7/2017
A05	Baton	20	10.30	Setia Sukan	Shah Alam	M3000	Devi	3 Ungu	09/8/2017	12/8/2017

BENTUK PENORMALAN KEDUA (2NF)

- **2NF** – Hubungan kriteria berikut dipenuhi → Jadual adalah dalam bentuk normal pertama dan semua atribut bukan kunci bergantung sepenuhnya kepada atribut kunci.
- **Kebergantungan separa** - Jika ada atribut bukan kunci hanya bergantung pada salah satu daripada atribut kunci.
- **CONTOH** : Jadual BARANG mempunyai 2 dua kunci primer – NoBarang dan IDMurid.
- Akan tetapi atribut NamaBarang, Kuantiti, HargaSeunit, NamaPembekal dan Alamat hanya bergantung pada NoBarang dan atribut NamaMurid dan Kelas hanya bergantung pada pada IDMurid.
- Hanya atribut TarikhPinjam dan TarikhPulang bergantung bergantung pada kedua-dua kunci.
- **PENYELESAIAN** : Cipta jadual baharu untuk mewakili TarikhPinjam dan tarikhPulang.

BARANG

<u>NoBarang</u>	Namabarang	Kuantiti	HargaSeunit	NamaPembekal	Alamat
-----------------	------------	----------	-------------	--------------	--------

PINJAMAN

<u>NoBarang</u>	<u>IDMurid</u>	TarikhPinjam	TarikhPulang
-----------------	----------------	--------------	--------------

MURID

<u>IDMurid</u>	TarikhPinjam	TarikhPulang
----------------	--------------	--------------

BENTUK PENORMALAN KETIGA (3NF)

- **3NF** – Hubungan kriteria berikut dipenuhi → Jadual adalah dalam bentuk normal kedua dan tidak wujud kebergantungan transitif.
- **Kebergantungan Transitif** – Wujudnya kebergantungan antara atribut bukan kunci.
- **CONTOH** : Jadual BARANG yang berada dalam bentuk normal kedua dan terdapat kebergantungan transitif iaitu atribut Alamat bergantung NamaPembekal dan kedua-duanya adalah atribut bukan kunci.
- **PENYELESAIAN** : Cipta jadual baharu bagi atribut bukan kunci yang saling bergantung.

BARANG

<u>NoBarang</u>	Namabarang	Kuantiti	HargaSeunit	<u>NamaPembekal</u>
-----------------	------------	----------	-------------	---------------------

PINJAMAN

<u>NoBarang</u>	<u>IDMurid</u>	TarikhPinjam	TarikhPulang
-----------------	----------------	--------------	--------------

MURID

<u>IDMurid</u>	TarikhPinjam	TarikhPulang
----------------	--------------	--------------

PEMBEKAL

<u>NamaPembekal</u>	Alamat
---------------------	--------

SQL

- **Structured Query Language / Bahasa Pertanyaan Berstruktur** : Digunakan untuk mencapai dan mengemaskini data dalam Pangkalan data.
- Bahasa Pengaturcaraan piawai yang digunakan untuk berhubung dengan pangkalan data melalui sistem pengurusan pangkalan data.
- Hampir semua sistem pengurusan pangkalan data mengenali pernyataan SQL.

BAHASA
ARAHAN SQLBahasa Takrifan Data
(Data Definition Language – DDL)

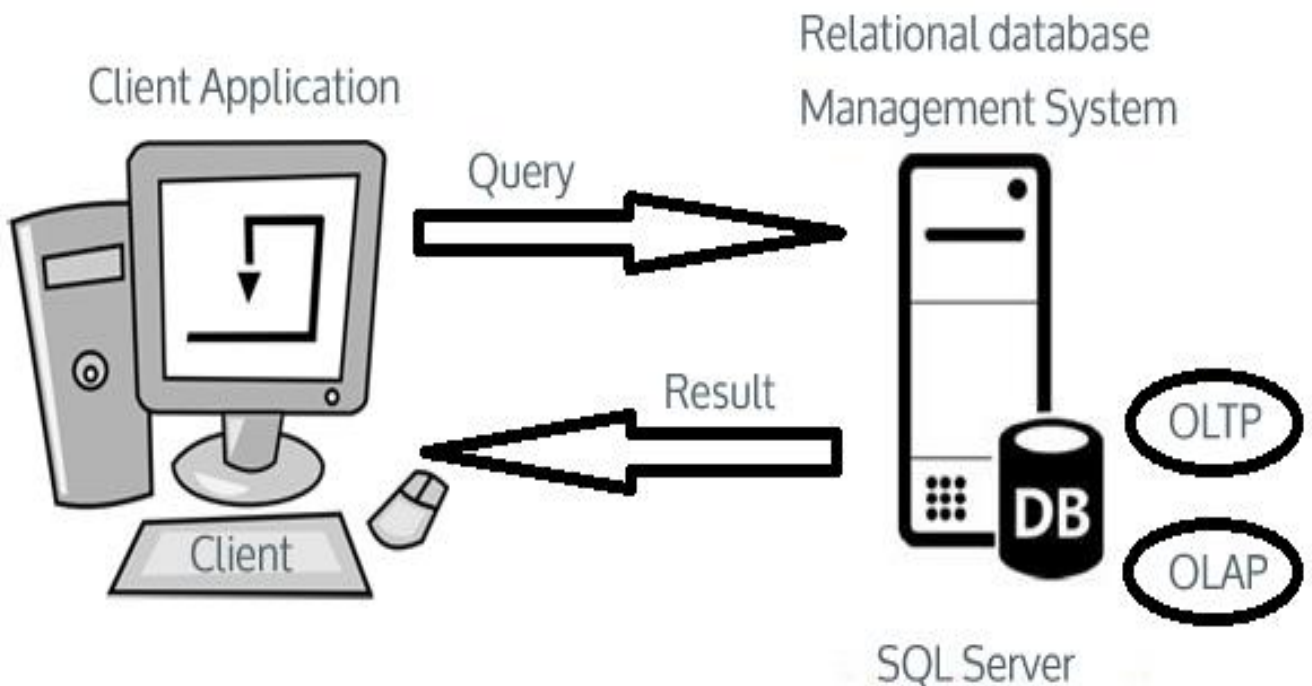
- ✓ Untuk mencipta dan mengubah struktur pangkalan data.
- ✓ Contoh : **CREATE** , **ALTER** , **DROP**

Bahasa Pengolahan Data
(Data Manipulation Language - DML)

- ✓ Untuk memanipulasi data dalam jadual.
- ✓ Contoh : **SELECT** , **INSERT** , **UPDATE** , **DELETE**

Bahasa Kawalan Data
(Data Control Language – DCL)

- ✓ Untuk mengawal capaian data pangkalan data.
- ✓ Contoh : **GRANT** , **REVOKE**



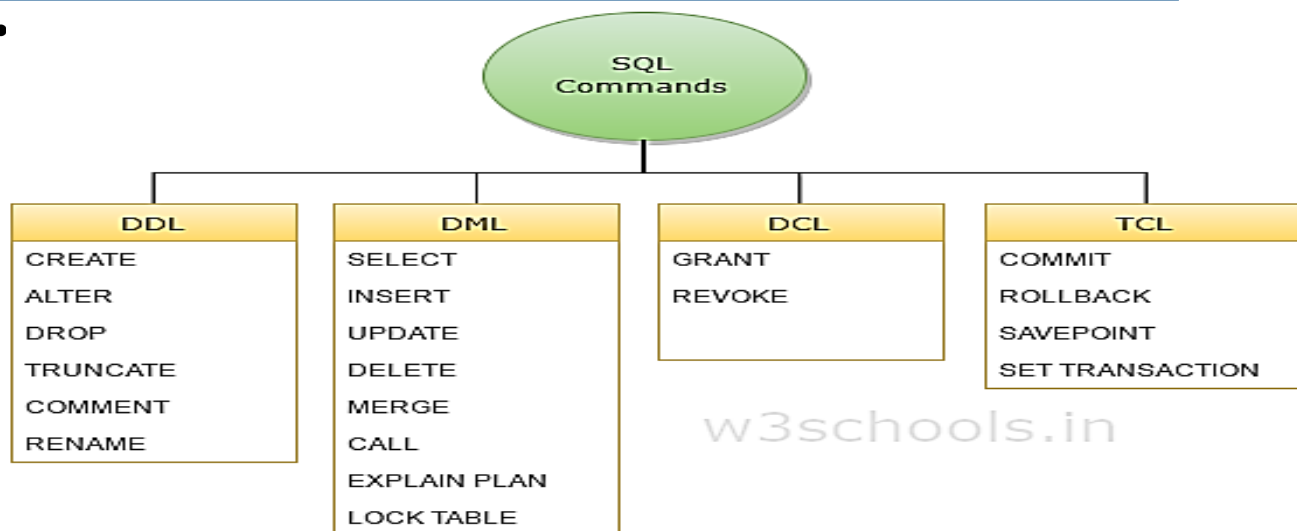
JENIS-JENIS DATA

- Pemilihan jenis data yang betul adalah penting kerana pemilihannya memberikan kesan kepada prestasi keseluruhan pangkalan data.

JENIS DATA	KETERANGAN
CHAR	• <i>String</i> yang antara 0 hingga 255 patah perkataan. • CHAR digunakan untuk lajur yang mengandungi data yang tetap dan seragam. • Contoh : Jantina
VARCHAR	• <i>String</i> yang boleh diubah antara 0 hingga 255 patah perkataan. • VARCHAR digunakan untuk lajur yang mengandungi data yang tidak seragam. • Contoh : Nama , Alamat
TINYTEXT	• <i>String</i> dengan panjang maksimum 255 patah perkataan.
TEXT	• <i>String</i> 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 dengan format YYYY-MM-DD/HH:MM:SS
TIME	• Waktu dengan format HH:MM:SS

MEMBINA PANGKALAN DATA

ARAHAN SQL



ARAHAN	FORMAT
Mencipta jadual (<i>Create table</i>)	CREATE TABLE nama jadual (Nama_medan1 jenis data(saiz) PRIMARY KEY , Nama_medan2 jenis data(saiz) NOT NULL , Nama_medan3 jenis data(saiz) , ...)
Mengubah Struktur Jadual	ALTER TABLE nama_jadual pilihan perubahan; - ALTER TABLE : arahan asas untuk mengubah struktur jadual. - nama jadual – nama jadual yang akan diubah. - Pilihan perubahan – contoh : menambah medan, meminda jenis data dan menghapus medan.
Sisip/Tambah Medan	ALTER TABLE nama_jadual ADD nama_medan jenis data (saiz);
Pinda/Ubah Medan	ALTER TABLE nama_jadual MODIFY nama medan jenis data baharu (saiz baharu); @ ALTER TABLE nama_jadual CHANGE nama medan jenis data baharu (saiz baharu);
Hapus Medan	ALTER TABLE nama_jadual DROP nama medan;

2.1

2.1.4

MENGGUNAKAN SQL UNTUK MEMANIPULASI DATA SEPERTI SISIP/TAMBAH, PADAM DAN KEMASKINI

ARAHAN SQL

ARAHAN	FORMAT
Sisip/Tambah	INSERT INTO nama_jadual (medan 1, medan2.... medan...N) 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

2.1.5

MENGGUNAKAN SQL *SELECT* UNTUK MENDAPATKAN DATA SEMULA DENGAN MENGGUNAKAN *GROUP BY* DAN *ORDER BY*.

PERNYATAAN	FORMAT
SELECT	SELECT nama_medan FROM nama_jadual WHERE syarat1 AND syarat 2 AND syarat..... ** Digunakan untuk mencapai rekod dari jadual.
Padam	DELETE FROM nama_jadual WHERE nama_medan = nilai;
Kemaskini	UPDATE nama_jadual SET nama_medan = nilai_baharu WHERE nama_medan = nilai_tertentu;
GROUP BY	SELECT nama_medan FROM nama_jadual GRPOUP BY Nama_medan;
ORDER BY	SELECT nama_medan FROM nama_jadual ORFER BY Nama_medan;

FUNGSI AGREGAT

- Fungsi agregat dilaksanakan untuk menghasilkan satu nilai penyatuan bagi medan berkenaan.
- Fungsi ini menggabungkan sekumpulan data dan mengembalikan nilai tunggal sebagai hasilnya.

FUNGSI AGREGAT YANG SERING DIGUNAKAN

FUNGSI	KETERANGAN	FORMAT
SUM	Mengembalikan jumlah nilai medan tertentu	SELECT SUM (nama medan) FROM nama jadual;
MIN	Mengembalikan nilai terendah medan tertentu	SELECT MIN (nama medan) FROM nama jadual;
MAX	Mengembalikan nilai tertinggi medan tertentu	SELECT MAX (nama medan) FROM nama jadual;
AVG	Mengembalikan nilai purata medan tertentu	SELECT AVG (nama medan) FROM nama jadual;
COUNT	Mengembalikan jumlah bilangan rekod	SELECT COUNT (nama medan) FROM nama jadual;

```
SELECT c.Country
, COUNT(*) as CustomerInEachCountry
, MAX(CustomerID) as TheHighestID
FROM dbo.Customer as c
GROUP BY c.Country
```

Country	CustomerInEachCountry	TheHighestID
UK	3	3
Germany	2	6
Spain	1	7

```
SELECT address_state,
SUM(fees_paid)
FROM student
GROUP BY address_state;
```

```
SELECT address_state,
SUM(fees_paid)
FROM student
GROUP BY address_state
HAVING SUM(fees_paid) > 500
```

CONTOH

JADUAL DALAM BENTUK ONF

IDPeserta	NamaPeserta	Kelas	Jantina	Kategori	RumahSukan	NoAcara	NamaAcara	Pingat
1000	Komala	5 Cergas	P	Bawah 18	Merah	A01	100 M	Gangsa
1000	Komala	5 Cergas	P	Bawah 18	Merah	A02	400 M	Emas
2000	Nisrina	2 Bistari	P	Bawah 15	Hijau	A02	400 M	Emas
2000	Nisrina	2 Bistari	P	Bawah 15	Hijau	A03	Bola Jaring	Perak
3000	Andy	1 Anjung	L	Bawah 15	Kuning	A04	Bola Sepak	Tiada
4000	Andang	1 Cergas	L	Bawah 18	Kuning	A04	Bola Sepak	Tiada

JADUAL DALAM BENTUK 1NF

IDPeserta	NamaPeserta	Kelas	Jantina	Kategori	RumahSukan	NoAcara	NamaAcara	Pingat
1000	Komala	5 Cergas	P	Bawah 18	Merah	A01	100 M	Gangsa
1000	Komala	5 Cergas	P	Bawah 18	Merah	A02	400 M	Emas
2000	Nisrina	2 Bistari	P	Bawah 15	Hijau	A02	400 M	Emas
2000	Nisrina	2 Bistari	P	Bawah 15	Hijau	A03	Bola Jaring	Perak
3000	Andy	1 Anjung	L	Bawah 15	Kuning	A04	Bola Sepak	Tiada
4000	Andang	1 Cergas	L	Bawah 18	Kuning	A04	Bola Sepak	Tiada

JADUAL DALAM BENTUK 2NF

PESERTA

<u>IDPeserta</u>	NamaPeserta	Kelas	Jantina	Kategori	RumahSukan
------------------	-------------	-------	---------	----------	------------

KEDUDUKAN

<u>IDPeserta</u>	<u>NoAcara</u>	Pingat
------------------	----------------	--------

ACARA

<u>NoAcara</u>	NamaAcara
----------------	-----------

***Kategori bergantung pada kelas yang menentukan umur murid sama ada di bawah 18 atau di bawah 15*

JADUAL DALAM BENTUK 3NF

PESERTA

<u>IDPeserta</u>	NamaPeserta	Jantina	RumahSukan	<u>Kategori</u>
------------------	-------------	---------	------------	-----------------

KEDUDUKAN

<u>IDPeserta</u>	<u>NoAcara</u>	Pingat
------------------	----------------	--------

ACARA

<u>NoAcara</u>	NamaAcara
----------------	-----------

KATEGORI PESERTA

<u>Kategori</u>	Kelas
-----------------	-------

MEMBINA PANGKALAN DATA

•
~Rujuk Buku Teks~ ms 114 - 116

3.0

PENGATURCARAAN BERASASKAN WEB





3.0 PENGATURCARAAN BERASASKAN WEB

3.1 Bahasa
Penskripan Klien

3.2 Bahasa
Penskripan
Pelayan

3.3 Laman Web
Interaktif

BAHASA PENSKRIPAN KLIEN

3.1.1

- Menerangkan keperluan Bahasa Penskripan Klien dalam laman web.

3.1.2

- Menterjemah atur cara mudah yang diberi dari Bahasa Penskripan Klien kepada carta alir (*bubble sort* , *selection sort* , *binary search* , *max / min / mean* , *count* , *queue*)

3.1.3

- Menggunakan pemalar, pemboleh ubah dan jenis data yang berlainan dalam atur cara untuk melaksanakan *bubble sort* , *selection sort* , *binary search* , *max / min / mean* , *count* , *queue*

3.1.4

- Menggunakan struktur kawalan (jujukan, pilihan, ulangan) dalam atur cara .

3.1.5

- Menerangkan kelebihan standard library kepada pengaturcara.

3.1.6

- Menggunakan standard library dalam atur cara .

3.1.7

- Meringkaskan atur cara dengan menggunakan:
(i) procedure (ii) function

3.1.8

- Membina atur cara yang melibatkan penggunaan tatasusunan (array)

3.1.9

- Menghasilkan atur cara interaktif yang mengandungi pengistiharan data, struktur kawalan, procedure dan function serta tatasusunan

3.1

3.1.1

MENERANGKAN KEPERLUAN BAHASA PENGATURCARAAN PENSKRIPAN KLIEN DALAM LAMAN WEB

Pengenalan kepada HTML dan CSS

- **HTML** (*Hypertext markup Language*) : Bahasa yang *markup standard* yang digunakan oleh pereka web bagi mereka bentuk laman web.
- Dokumen HTML adalah mengenai konsep bagaimana dokumen dipaparkan dan bagaimana elemen teks, audio, video serta grafik disusun agar mudah dibaca dan difahami oleh pelawat laman web.
- Untuk menerangkan struktur bagi setiap laman web, **kod HTML** atau **elemen HTML** digunakan.
- Elemen terdiri daripada 2 tag berpasangan : **start tag** dan **close tag**.
- Semua elemen HTML mengandungi atribut.
- Atribut mengandungi maklumat tambahan bagi setiap elemen HTML dan biasanya ditempatkan di *start tag*.
- Terdapat 2 bahagian atribut : **Atribut nama** dan **Atribut nilai**.



Pengenalan kepada CSS

- **CSS** (*Cascading Style Sheet*) : Salah satu Bahasa pengaturcaraan yang merupakan tambahan (*extension*) kepada HTML.
- CSS membenarkan perubahan kepada gaya sesuatu halaman web dan menerangkan persembahan bagi sesuatu dokumen HTML.

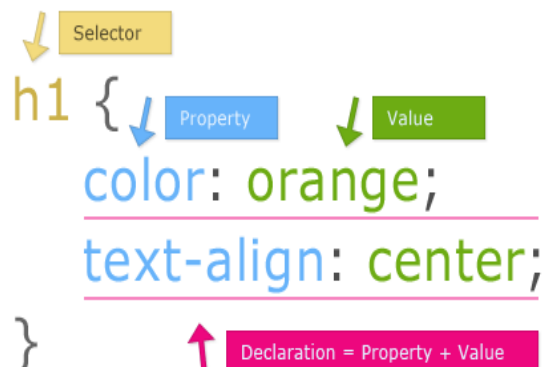
What is CSS

- **C**ascading **S**tyle **S**heets
- Contains the rules for the **presentation** of HTML.



- CSS was introduced to keep the **presentation** information **separate** from **HTML** markup (content).

Anatomy of a CSS Rule



3.1

3.1.1

MENERANGKAN KEPERLUAN BAHASA PENGATURCARAAN PENSKRIPAN KLIEN DALAM LAMAN WEB

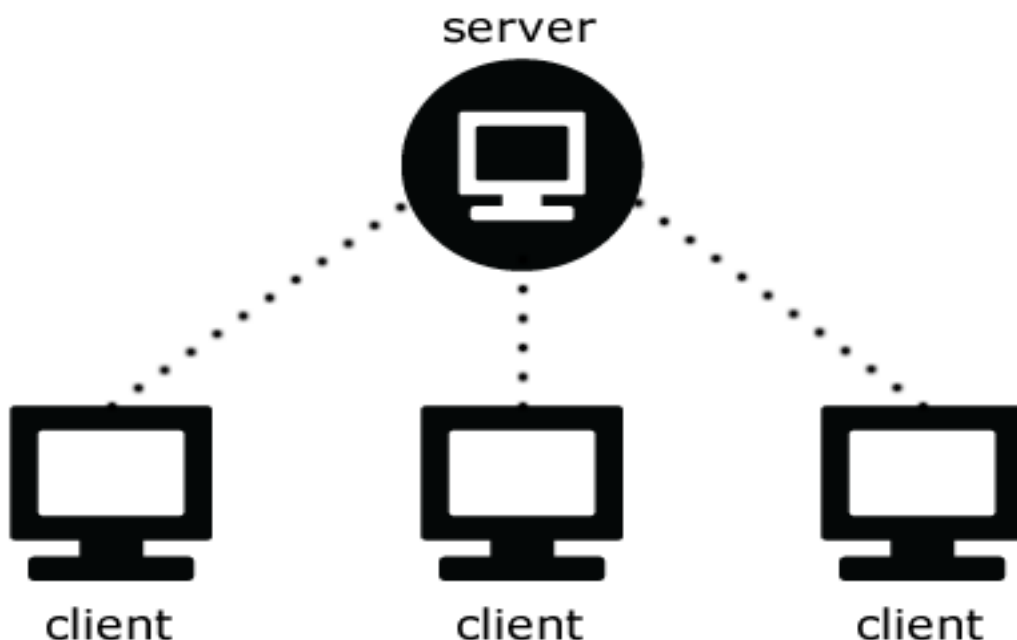
PENGENALAN KEPADA JAVASCRIPT

- **JavaScript** : menyediakan keupayaan pengaturcaraan untuk memproses sesuatu tugas berdasarkan permohonan yang diterima daripada klien.
- Kod *JavaScript* akan ditulis bersama-sama kod HTML.
- Kod *JavaScript* ditulis di antara tag `<script>` dengan `</script>`.
- 2 cara Kod *JavaScript* ditulis : *Explicit embedding* dan *Implicit embedding*.



KONSEP APLIKASI PELANGGAN/PELAYAN (CLIENT/SERVER)

- **Pelanggan (Client)** : biasanya ialah pelayar (*browser*) yang digunakan oleh pengguna bagi melayari web.
- Aplikasi **pelayan (server)** : boleh berada di mana-mana jua di seluruh dunia yang bertindak untuk memproses permintaan atau menghantar laman web yang diminta oleh klien.
- Satu lagi komponen laman web ialah pangkalan data yang digunakan untuk menyimpan data.



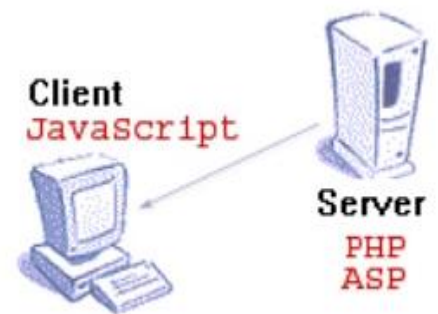
3.1

3.1.1

MENERANGKAN KEPERLUAN BAHASA PENGATURCARAAN PENSKRIPAN KLIEN DALAM LAMAN WEB

BAHASA PENSKRIPAN

- Konsep penskripan web digunakan secara meluas dalam pembangunan laman web.
- **PENSKRIPAN** : sebahagian kecil program yang dilaksanakan berdasarkan interaksi pengguna dengan laman web.
- Penskripan ini ditempatkan di dua bahagian :
penskripan klien di sebelah pelanggan dan
penskripan pelayan di sebelah pelayan..



BAHASA PENSKRIPAN

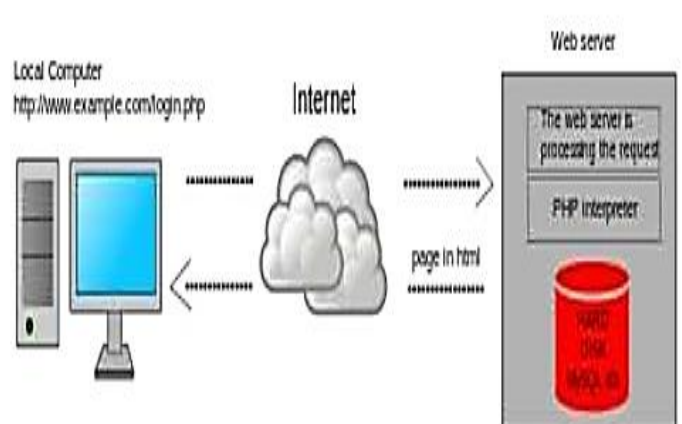
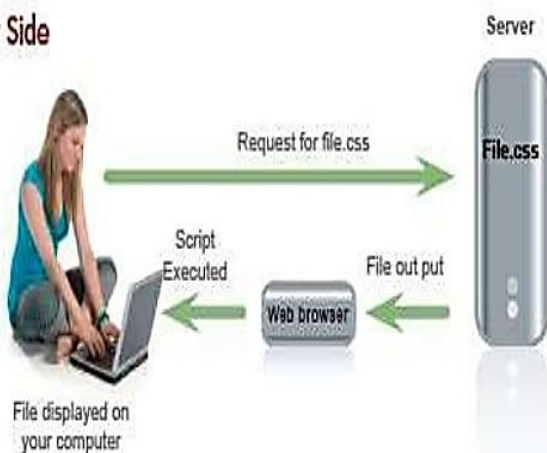
Bahasa Penskripan Klien

- ✓ Penskripan klien pula bekerja di bahagian hadapan pembangunan web.
- ✓ dilaksanakan oleh pelayar web.

Bahasa Penskripan Pelayan

- ✓ bekerja di bahagian belakang pembangunan web.
- ✓ dilaksanakan oleh pelayan web.

Client Side



3.1

3.1.1

MENERANGKAN KEPERLUAN BAHASA PENGATURCARAAN PENSKRIPAN KLIEN DALAM LAMAN WEB

BAHASA PENSKRIPAN KLIEN

The process of client side scripting



- Merujuk kepada program komputer dalam web yang dilaksanakan pada komputer klien.
- Program ini direka bagi memanipulasikan dan memaparkan kandungan laman web yang telah dimuat turun ke komputer klien.
- Penskripan klien biasanya digunakan untuk proses pengesahan input yang dimasukkan oleh pengguna apabila mengisi borang dalam talian.
- Juga digunakan untuk melaksanakan tugas-tugas yang mudah seperti menukar elemen. Contoh : jenis font atau warna latar tanpa melibatkan pelayan.
- Contoh lain : *Shopping cart* – Setiap kali kita menambah atau mengurangkan item, jumlah kos akan berubah secara automatic.
- Bukan semua tugas dapat dilaksanakan oleh Bahasa penskripan klien.
- Tugas-tugas yang melibatkan pangkalan data pada sebelah pelayan web atau capaian ke atas fail sistem yang terdapat pada pelayan web tidak mampu dilaksanakan oleh Bahasa Penskripan Klien.

Laman menjadi lebih interaktif.

Pembangun web berkeupayaan menentukan reka bentuk dan ciri-ciri web.

KELEBIHAN BAHASA PENSKRIPAN

Meningkatkan kebolegunaan web bagi pelayar yang menyokong pelbagai jenis Bahasa penskripan klien

Pelaksanaan tugas yang pantas tanpa penglibatan komputer pelayan.

3.1

3.1.2

MENTERJEMAH ATUR CARA MUDAH YANG DIBERI DARIPADA BAHASA PENSKRIPAN KLIEN KEPADA CARTA ALIR

ISIHAN BUIH (*BUBBLE SORT*)

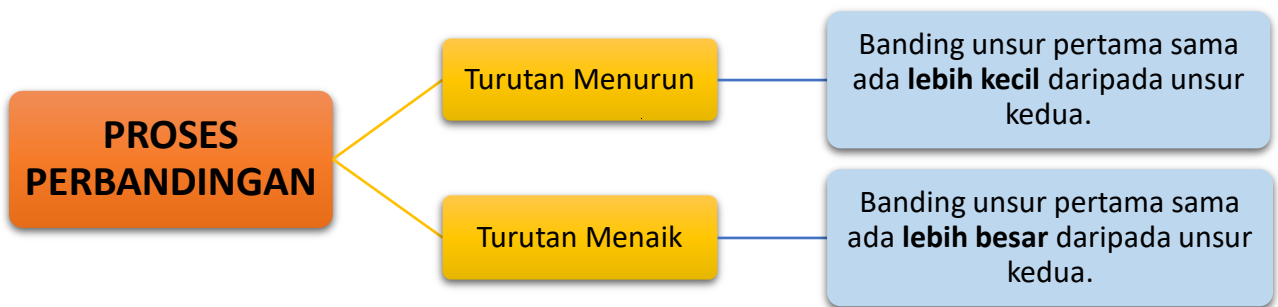
- **Bubble Sort** : Satu algoritma mudah dalam pengisihan suatu senarai yang prosesnya dilaksanakan satu persatu secara berturutan dengan membandingkan unsur-unsur bersebelahan dan akan menukar kedudukan (*swap*) sekiranya berlaku turutan yang salah.

Banding satu persatu unsur
bersebelahan



Tukar kedudukan jika
kedudukan salah.

PROSES ISIHAN BUIH



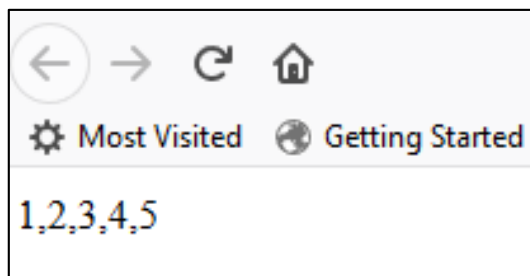
5	4	2	1	3
4	5	2	1	3
4	2	5	1	3
4	2	1	5	3
4	2	1	3	5
2	4	1	3	5
2	4	1	3	5
2	1	4	3	5
2	1	3	4	5
2	1	3	4	5
1	2	3	4	5
1	2	3	4	5

ATURCARA BAGI ISIHAN BUIH (TURUTAN MENAIK)

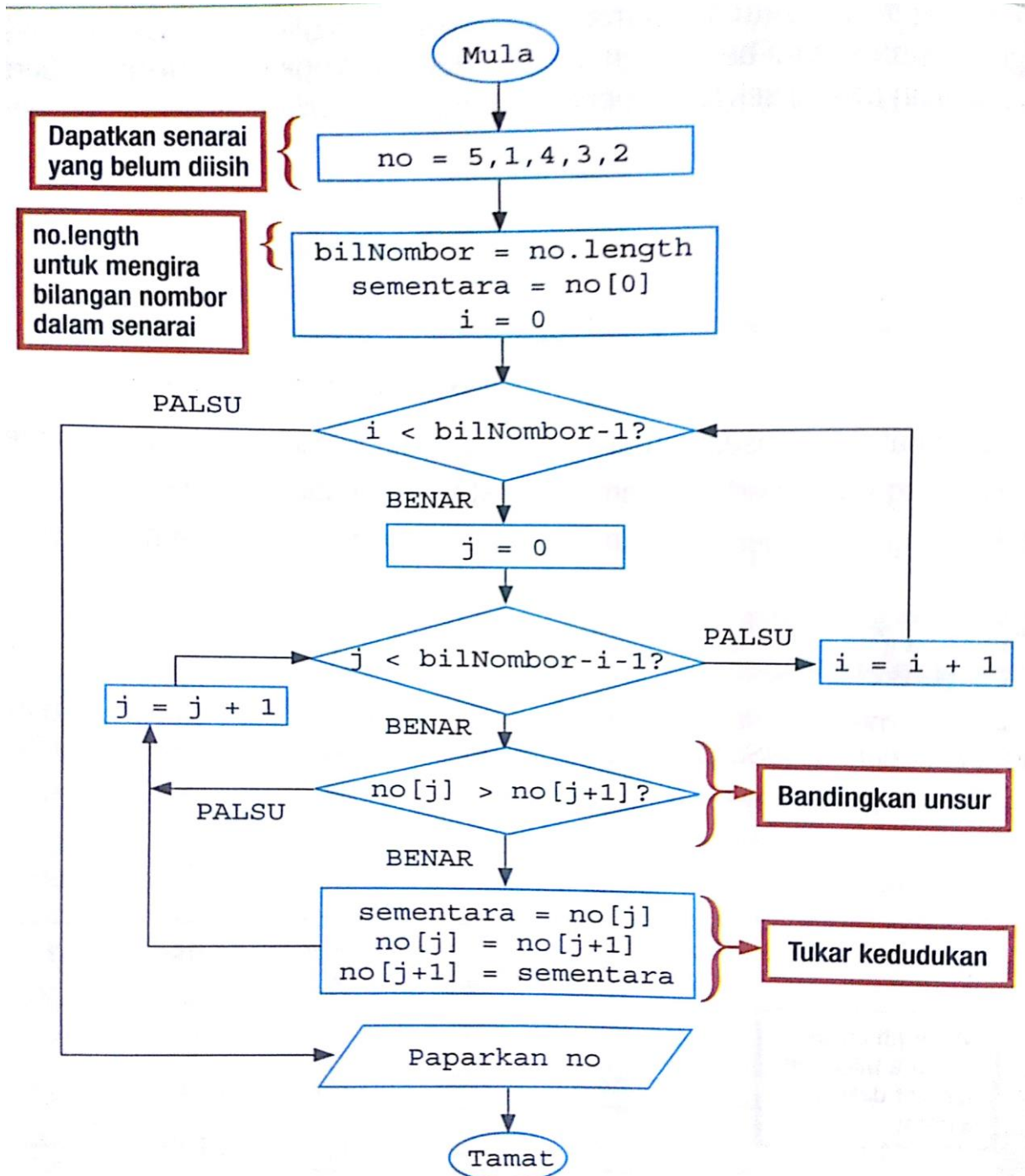
```
<html>
  <body>
    <script>
      var no = [ 5,1,4,3,2];
      var bilNombor = no.length,i,j;
      var sementara = no [0];

      for (i = 0; i < bilNombor-1; i++)
      {
        for (j=0;j<bilNombor-i-1;j++)
        {
          if (no[j] > no[j+1])
          {
            sementara = no[j];
            no[j]= no[j+1];
            no[j+1] = sementara;
          }
        }
      }
      document.write(no);
    </script>
  </body>
</html>
```

OUTPUT



CARTA ALIR BAGI ISIHAN BUIH (TURUTAN MENAIK)



ATURCARA 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];

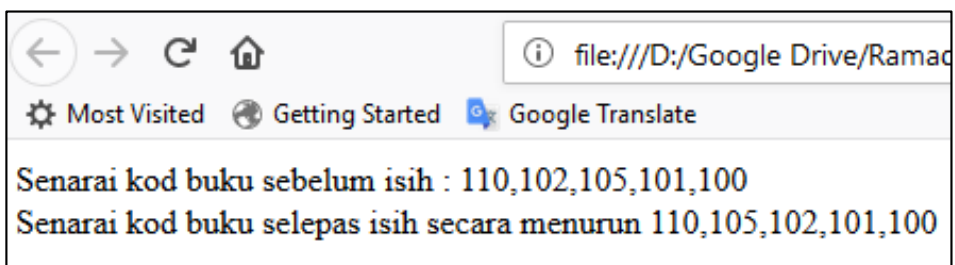
      document.write ("Senarai kod buku sebelum isih : " + kodBuku);

      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;
          }
        }
      }

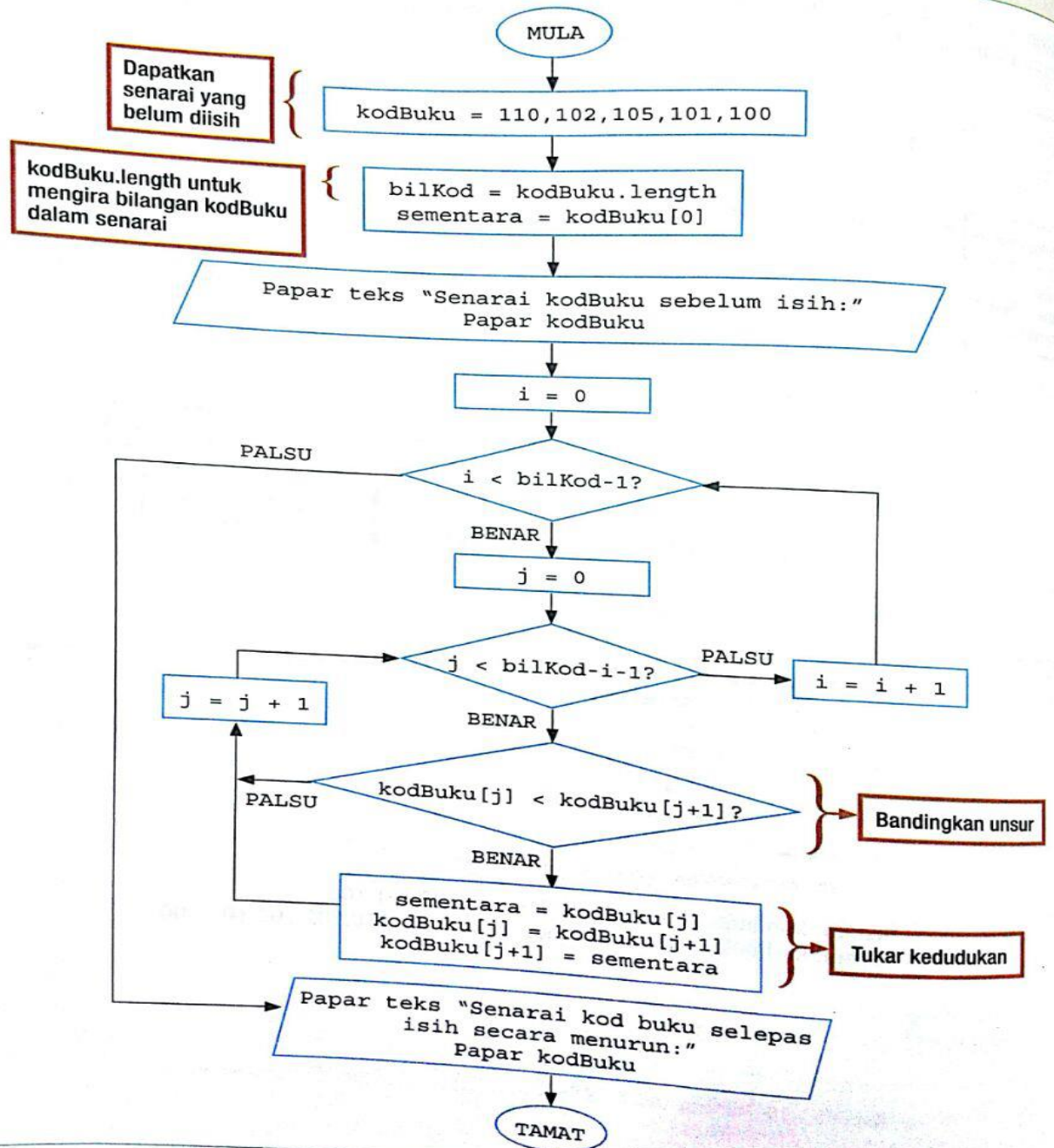
      document.write(" <br> Senarai kod buku selepas isih secara menurun " + kodBuku );

    </script>
  </body>
</html>
```

OUTPUT



CARTA ALIR BAGI ISIHAN BUIH (TURUTAN MENURUN)



Rajah 3.1.17 Carta alir bagi isihan buih (turutan menurun)

3.1

3.1.2

MENTERJEMAH ATUR CARA MUDAH YANG DIBERI DARIPADA BAHASA PENSKRIPAN KLIEN KEPADA CARTA ALIR

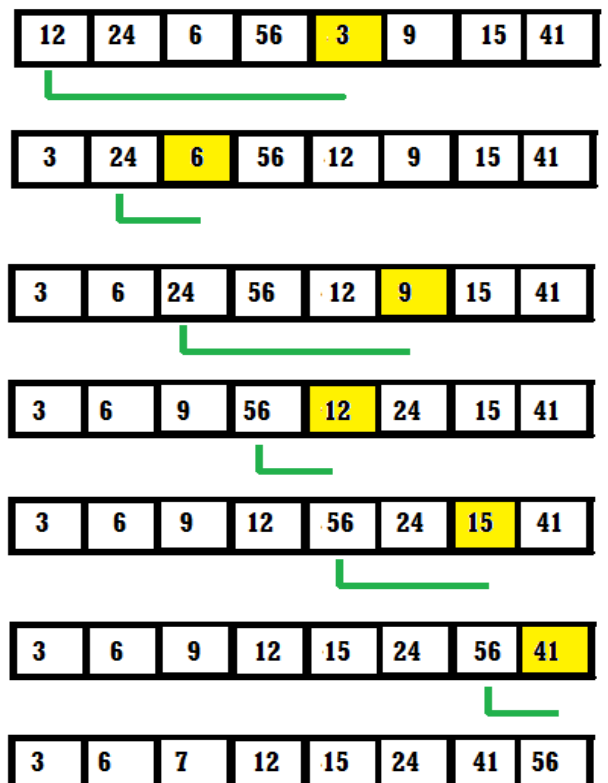
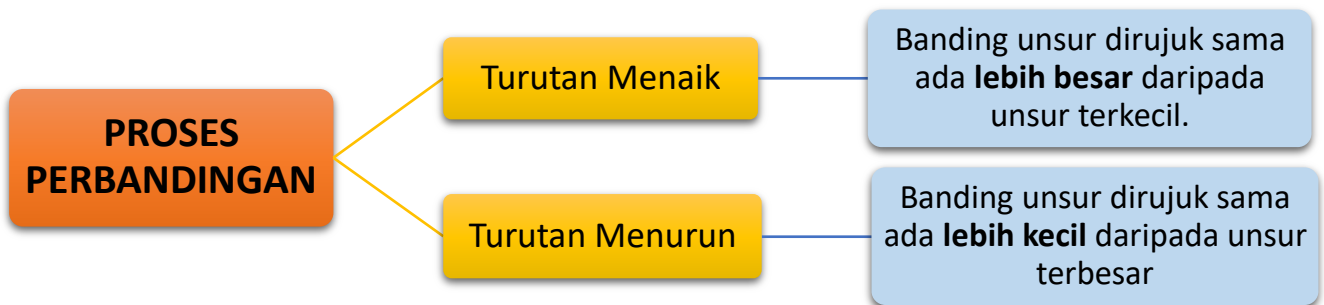
ISIHAN PILIH (*SELECTION SORT*)

Banding satu unsur dengan
unsur terkecil / terbesar
daripada unsur lain dalam
senarai.



Tukar kedudukan jika turutan
salah.

PROSES ISIHAN PILIH



ATURCARA BAGI ISIHAN PILIH (TURUTAN MENAIK)

```
<html>
  <body>
    <script>
      var no = [5,1,4,3,2],i,j,min,sementara;
      /*dapatkan senarai nama yang belum diisih*/

      var bilNombor = no.length;
      /*no.length untuk mengira bilangan nombor dalam senarai*/

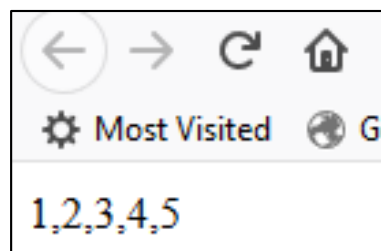
      for (i = 0; i < bilNombor-1; i++)
      {
        min = i;
        /*menetapkan nilai terkecil*/

        for (j = i + 1; j < bilNombor; j++)
        {
          if (no[j] < no[min])
            /*bandingkan unsur*/

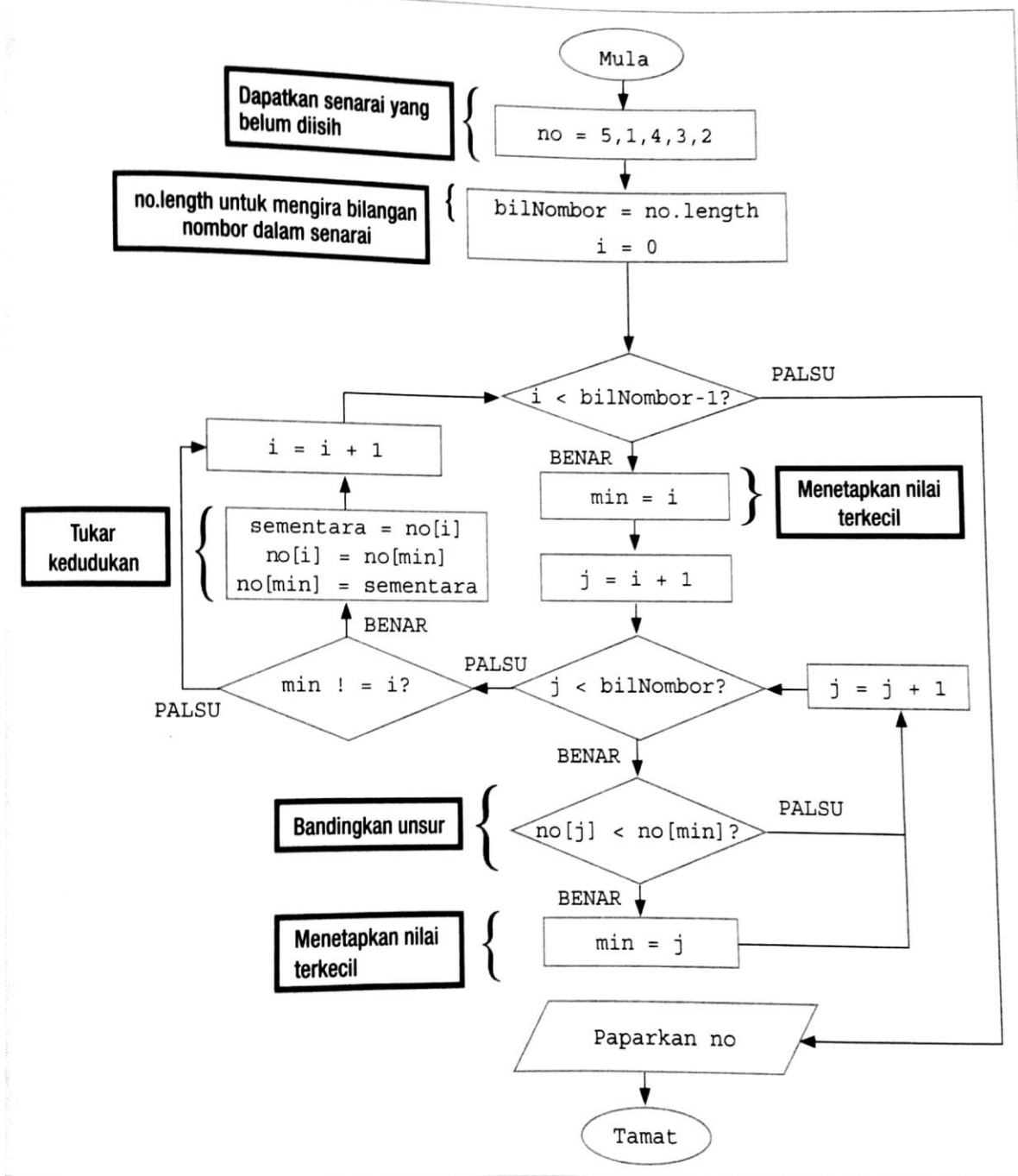
            {
              min = j;
              /*menetapkan nilai terkecil*/
            }

          if (min != i)
          {
            sementara = no[i];
            no[i]= no[min];
            no[min] = sementara;
            /*tukar kedudukan*/
          }
        }
        document.write(no) ;
      }
    </script>
  </body>
</html>
```

OUTPUT



CARTA ALIR BAGI ISIHAN PILIH (TURUTAN MENAIK)



Rajah 3.1.20 Carta alir isihan pilih (turutan menaik)

ATURCARA BAGI ISIHAN PILIH (TURUTAN MENAIK)

```
<html>
  <body>
    <script>
      var tinggi = [ 6,7,7,6,5,6], i,j,min, sementara;
      /*dapatkan senarai yang belum diisih*/

      var bilKetinggian = tinggi.length;
      /* tinggi.length untuk mengira bilangan ketinggian dalam senarai */

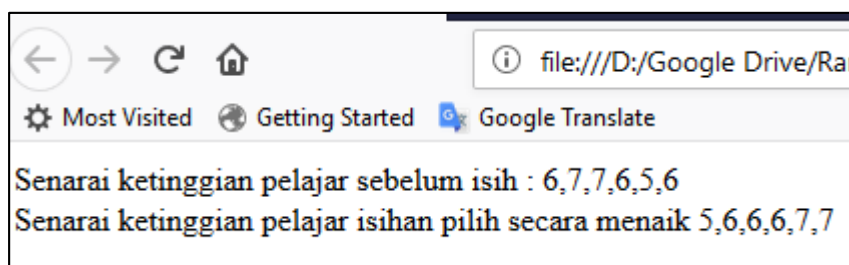
      document.write ("Senarai ketinggian pelajar sebelum isih : " + tinggi);

      for (i = 0; i < bilKetinggian-1; i++)
      {
        min = i;
        /* menetapkan nilai terkecil */

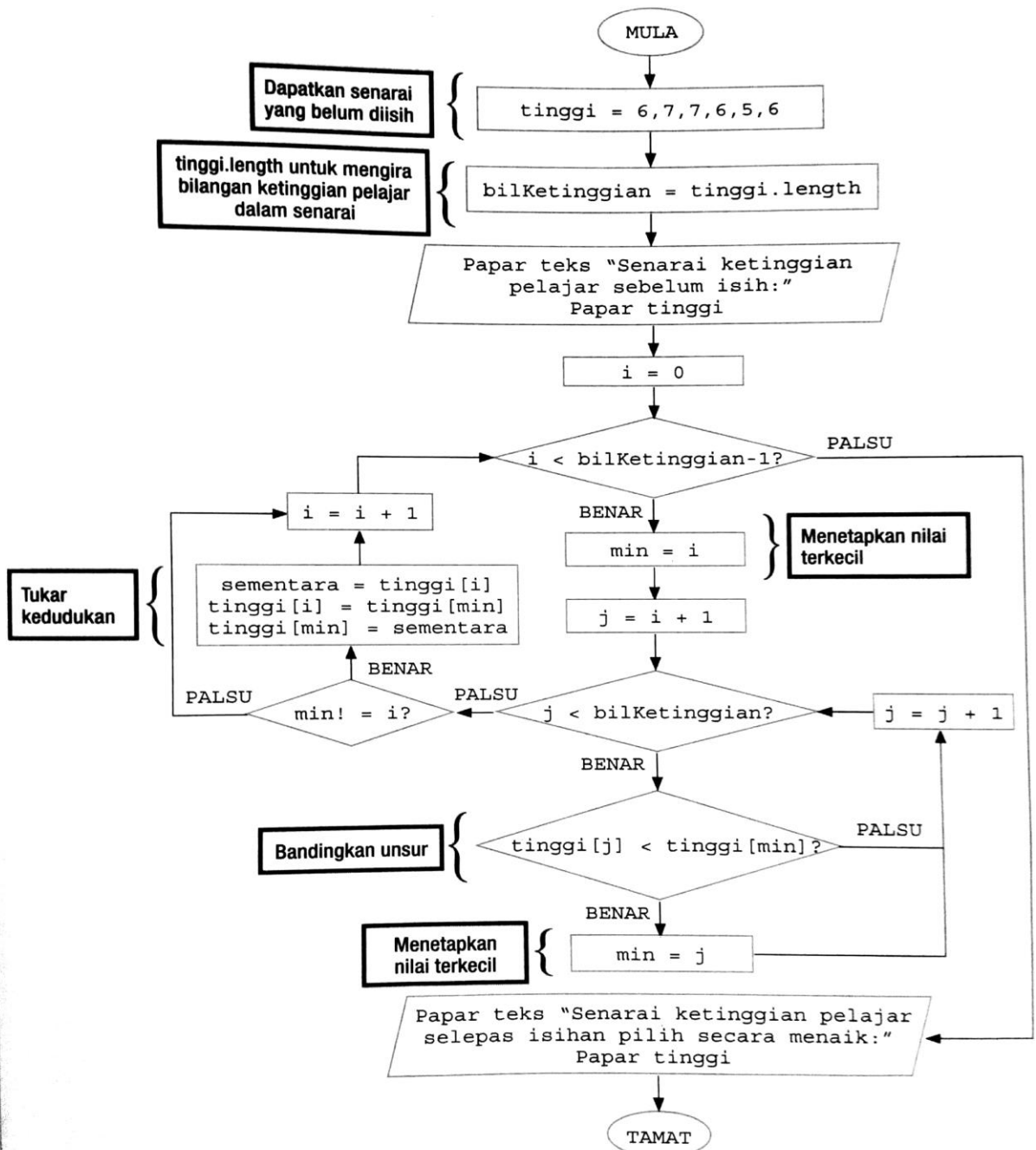
        for (j= i + 1; j < bilKetinggian; j++)
        {
          if (tinggi[j] < tinggi[min])
            /* bandingkan unsur*/
            {
              min = j;
              /* menetapkan nilai terkecil */
            }
        }
        if (min != i)
        {
          sementara = tinggi[i];
          tinggi[i]= tinggi[min];
          tinggi[min] = sementara;
          /* tukar kedudukan */
        }
      }

      document.write(" <br> Senarai ketinggian pelajar isihan pilih secara menaik " + tinggi );
    </script>
  </body>
</html>
```

OUTPUT



CARTA ALIR BAGI ISIHAN BUIH (TURUTAN MENAIK)



Rajah 3.1.22 Carta alir isihan pilih (turutan menaik)

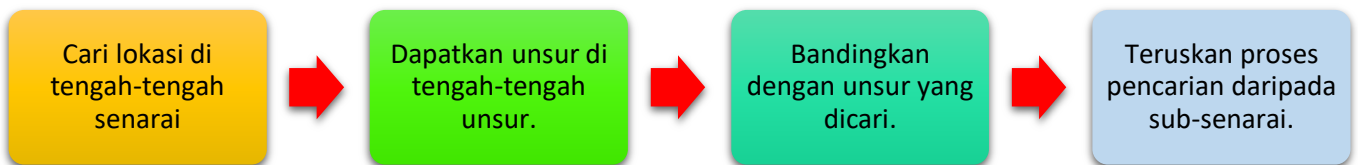
3.1

3.1.2

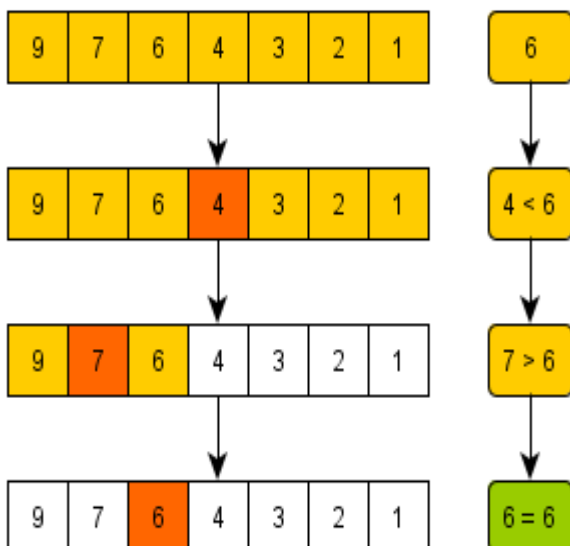
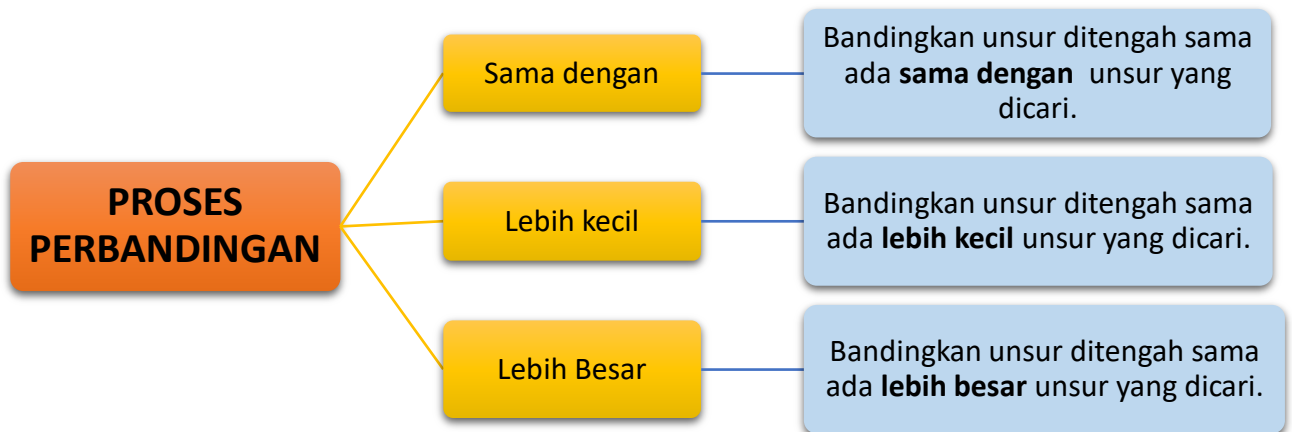
MENTERJEMAH ATUR CARA MUDAH YANG DIBERI DARIPADA BAHASA PENSKRIPAN KLIEN KEPADA CARTA ALIR

CARIAN PERDUAAN (*BINARY SEARCH*)

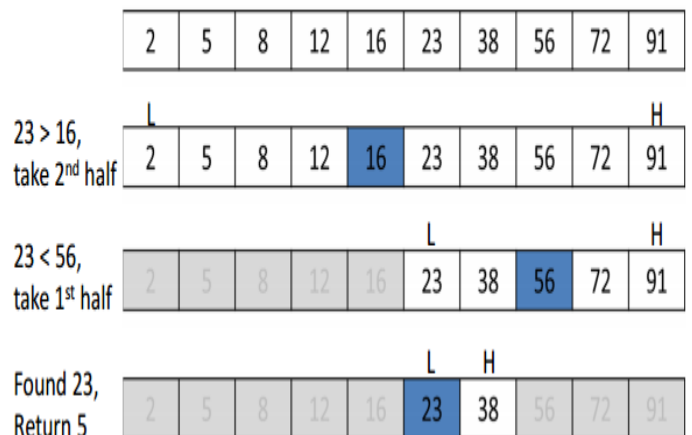
- **Binary Search:** Sesuai digunakan apabila melibatkan **bilangan data yang banyak dan telah diisih**.



Proses Carian Perdua (Binary Search)



If searching for 23 in the 10-element array:



ATURCARA BAGI CARIAN PERDUAAN

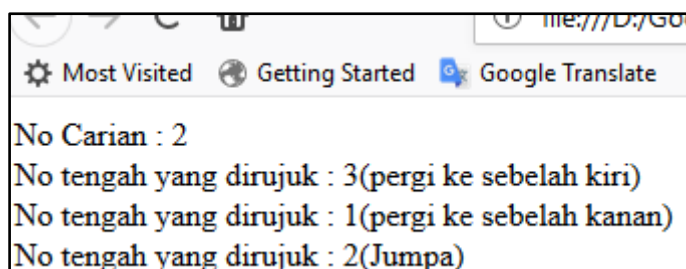
```
<html>
  <body>
    <script>
      var no = [1,2,3,4,5];
      var noCarian = 2;
      var indeksAwal = 0, indeksAkhir = no.length-1, indeksTengah = Math.floor ((indeksAkhir + indeksAwal)/2);

      document.write ("No Carian : " , noCarian);

      while (no[indeksTengah] != noCarian && indeksAwal < indeksAkhir)
      {
        if(no[indeksTengah]> noCarian)
        {
          indeksAkhir = indeksTengah-1;
          document.write ("<br> No tengah yang dirujuk : ", no[indeksTengah], "(pergi ke sebelah kiri)");
        }

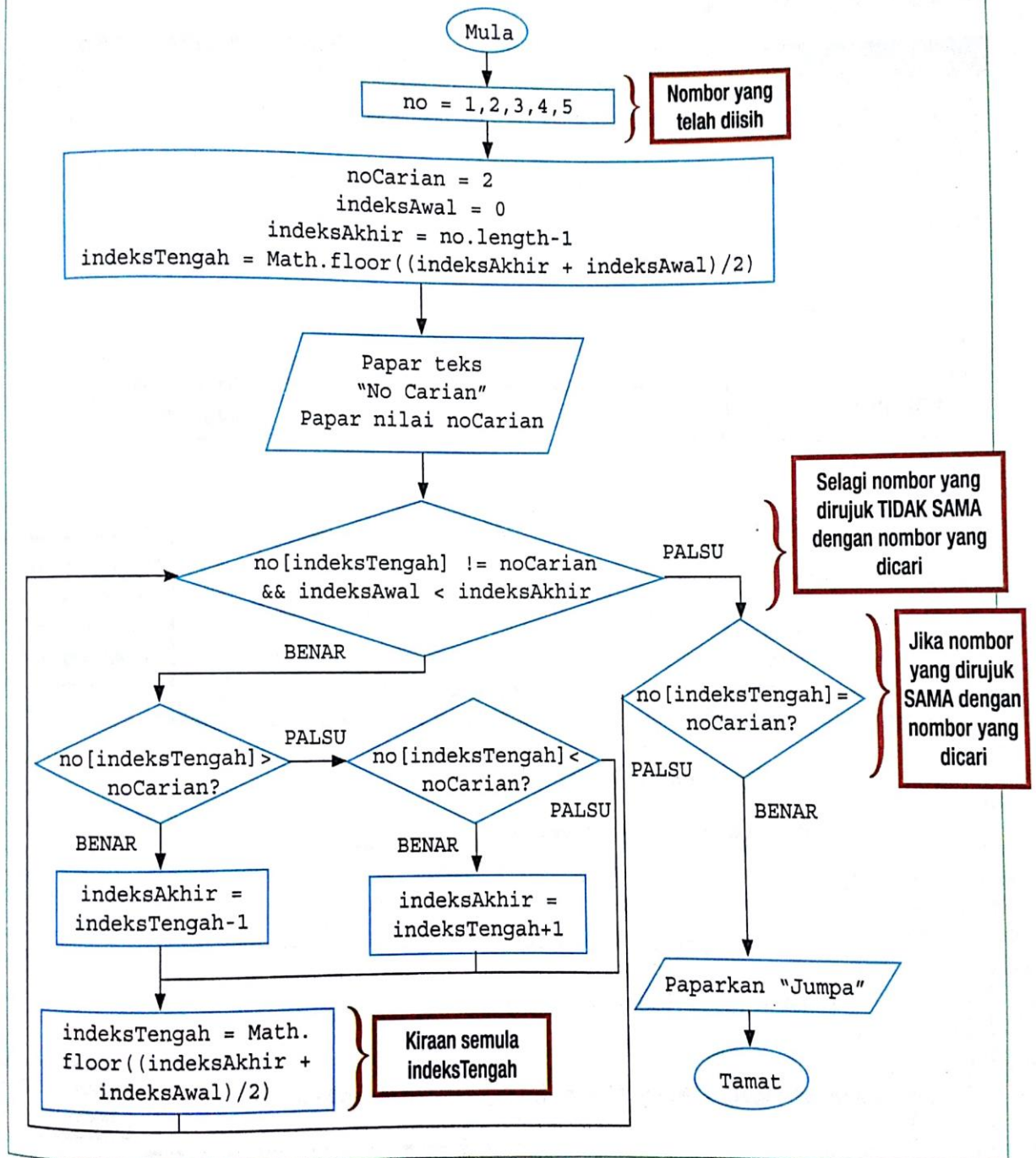
        else
        {
          if (no[indeksTengah] <noCarian)
          {
            indeksAwal = indeksTengah + 1;
            document.write ("<br> No tengah yang dirujuk : ", no[indeksTengah], "(pergi ke sebelah kanan)");
          }
          indeksTengah = Math.floor ((indeksAkhir + indeksAwal)/2);
        }
        if (no[indeksTengah] == noCarian)
        {
          document.write ("<br> No tengah yang dirujuk : ", no[indeksTengah], "(Jumpa)");
        }
      }
    </script>
  </body>
</html>
```

OUTPUT



```
No Carian : 2
No tengah yang dirujuk : 3(pergi ke sebelah kiri)
No tengah yang dirujuk : 1(pergi ke sebelah kanan)
No tengah yang dirujuk : 2(Jumpa)
```

CARTA ALIR CARIAN PERDUAAN



Rajah 3.1.25 Carta alir (carian perduaan)

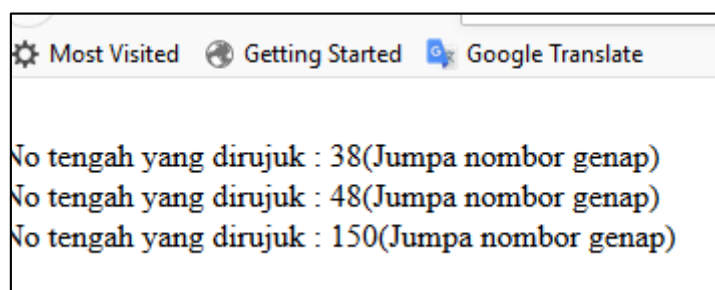
ATURCARA BAGI CARIAN PERDUAAN NOMBOR GENAP

```
<html>
  <body>
    <script>
      var no = [11,23,38,48,150],i;
      var noGenap;

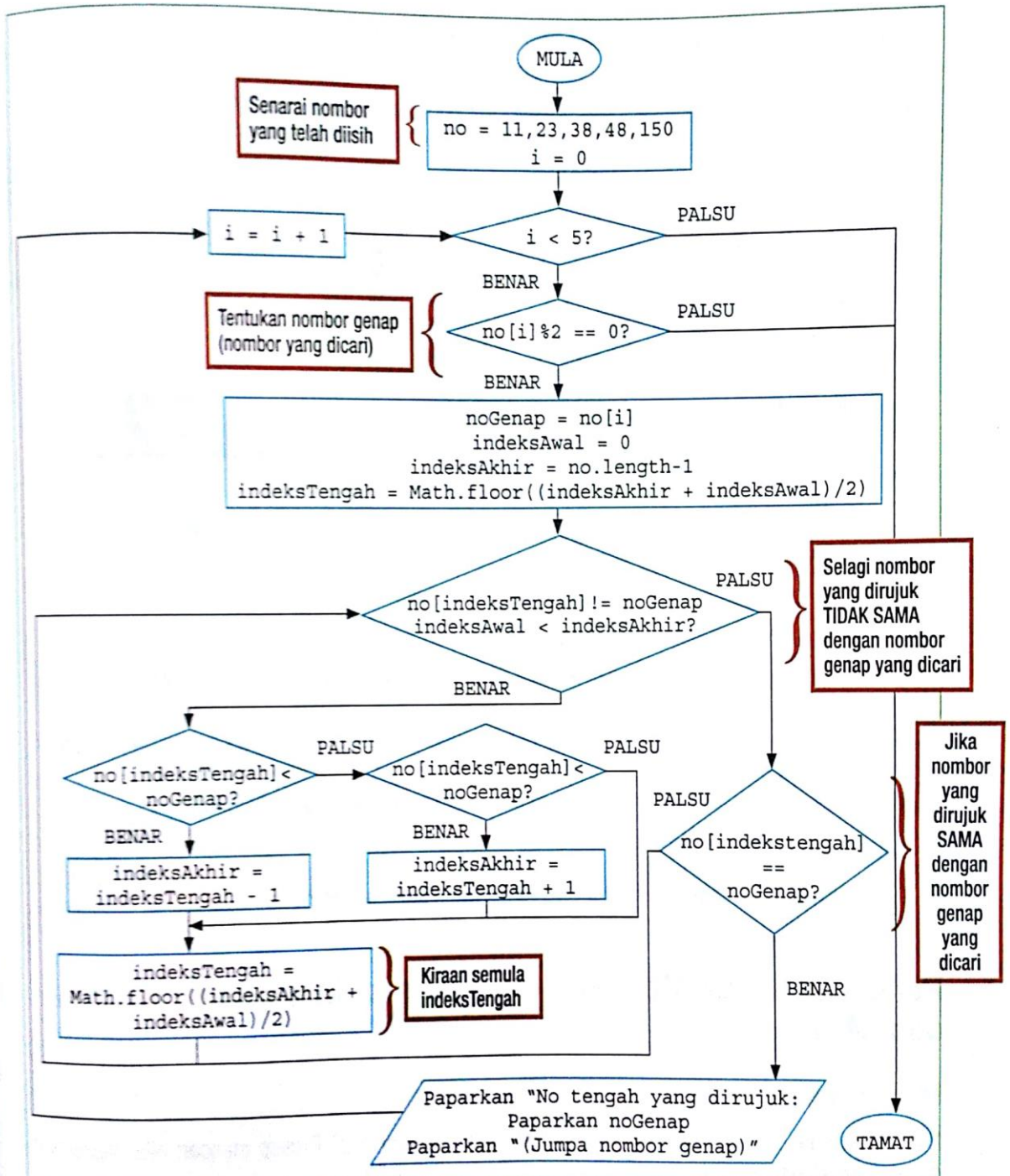
      for (i = 0; i<5; i++)
      {
        if ((no[i]%2 == 0))
        {
          noGenap = no[i];
          var indeksAwal =0, indeksAkhir = no.length-1, indeksTengah = Math.floor ((indeksAkhir + indeksAwal)/2);

          while (no[indeksTengah] != noGenap && indeksAwal < indeksAkhir)
          {
            if(no[indeksTengah]> noGenap)
            {
              indeksAkhir = indeksTengah-1;
            }
            else
            {
              if (no[indeksTengah] < noGenap)
              {
                indeksAwal = indeksTengah + 1;
              }
            }
            indeksTengah = Math.floor ((indeksAkhir + indeksAwal)/2);
          }
          if (no[indeksTengah] == noGenap)
          {
            document.write ("<br> No tengah yang dirujuk : ", no[indeksTengah], "(Jumpa nombor genap)");
          }
        }
      }
    </script>
  </body>
</html>
```

OUTPUT



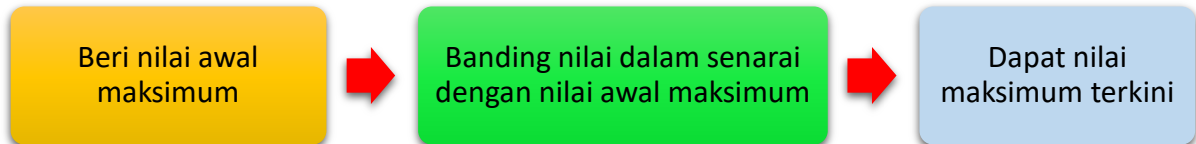
CARTA ALIR CARIAN PERDUAAN NOMBOR GENAP



Rajah 3.1.27 Carta alir carian perdua (carian nombor genap)

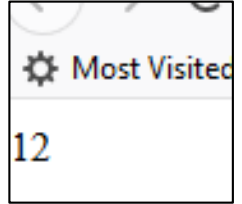
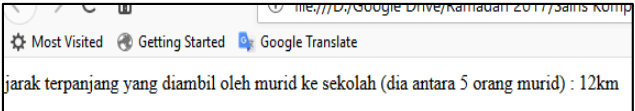
MENDAPATKAN NILAI MAKSIMUM

- Nilai maksimum merujuk nilai terbesar dalam suatu senarai nilai.
- Terdapat hanya satu proses perbandingan : bandingkan nilai yang dirujuk dalam senarai sama ada lebih besar daripada nilai maksimum.

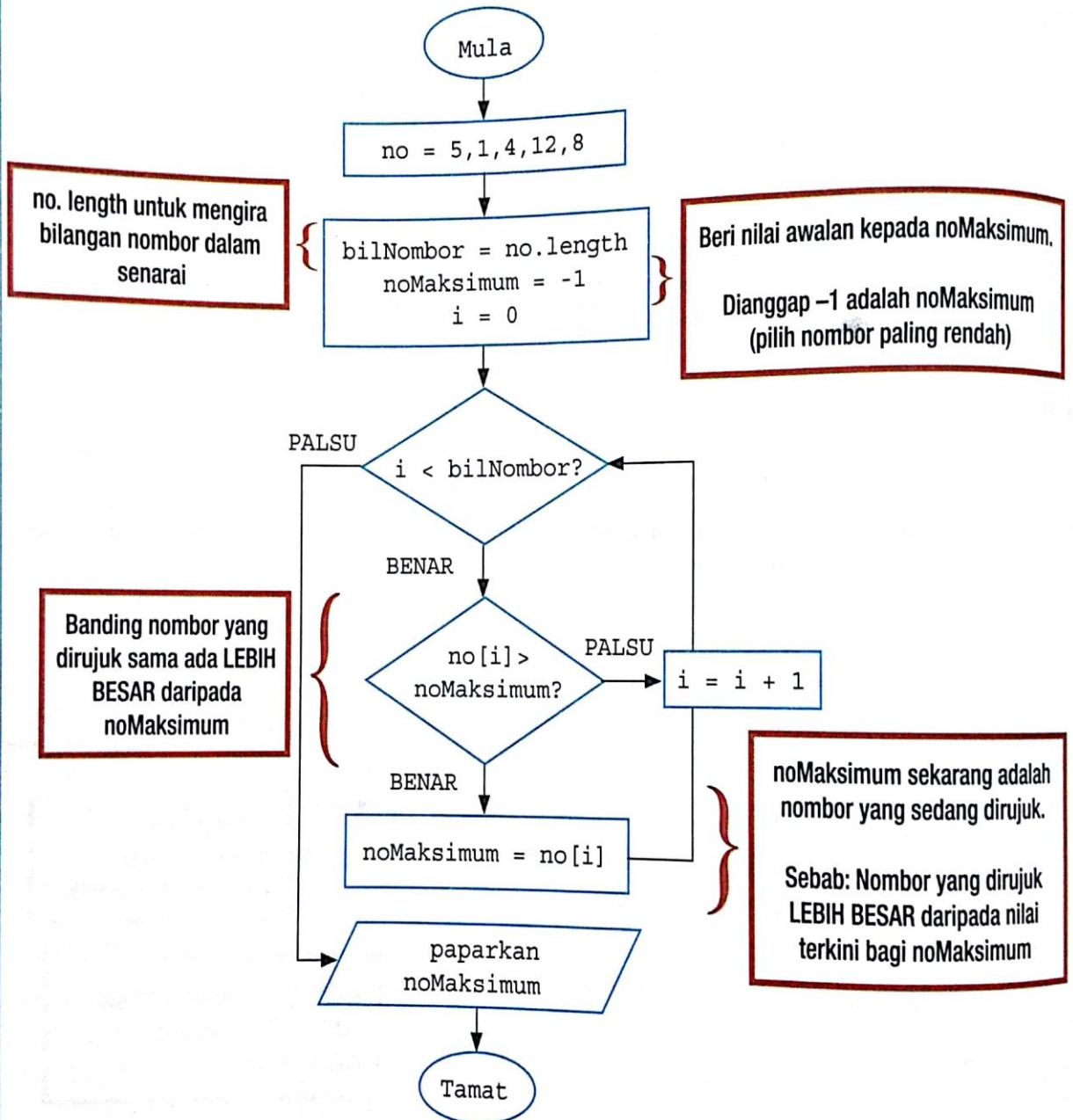


PROSES Mencari Nilai Maksimum

ATURCARA Mencari Nilai Maksimum

ATURCARA	OUTPUT
<pre> <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> </pre>	
<pre> <html> <body> <script> var jarak=[10,8,8,12,5],jarakTerpanjang = -1,i,bilJarak = jarak.length; for (i=0 ; i < bilJarak ; i++) { if (jarak[i] > jarakTerpanjang) { jarakTerpanjang = jarak[i]; } } document.write(" jarak terpanjang yang diambil oleh murid ke sekolah (dia antara 5 orang murid) : " + jarakTerpanjang + "km"); </script> </body> </html> </pre>	

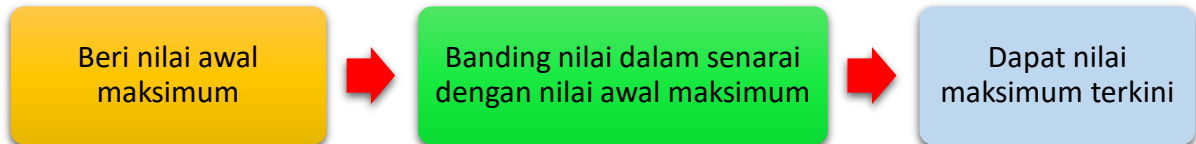
CARTA ALIR MENDAPATKAN NILAI MAKSIMUM



Rajah 3.1.30 Carta alir (mendapatkan nilai maksimum)

MENDAPATKAN NILAI MAKSIMUM

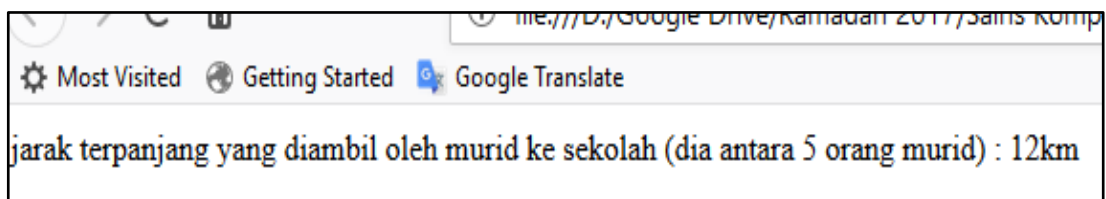
- Nilai maksimum merujuk nilai terbesar dalam suatu senarai nilai.
- Terdapat hanya satu proses perbandingan : bandingkan nilai yang dirujuk dalam senarai sama ada lebih besar daripada nilai maksimum.



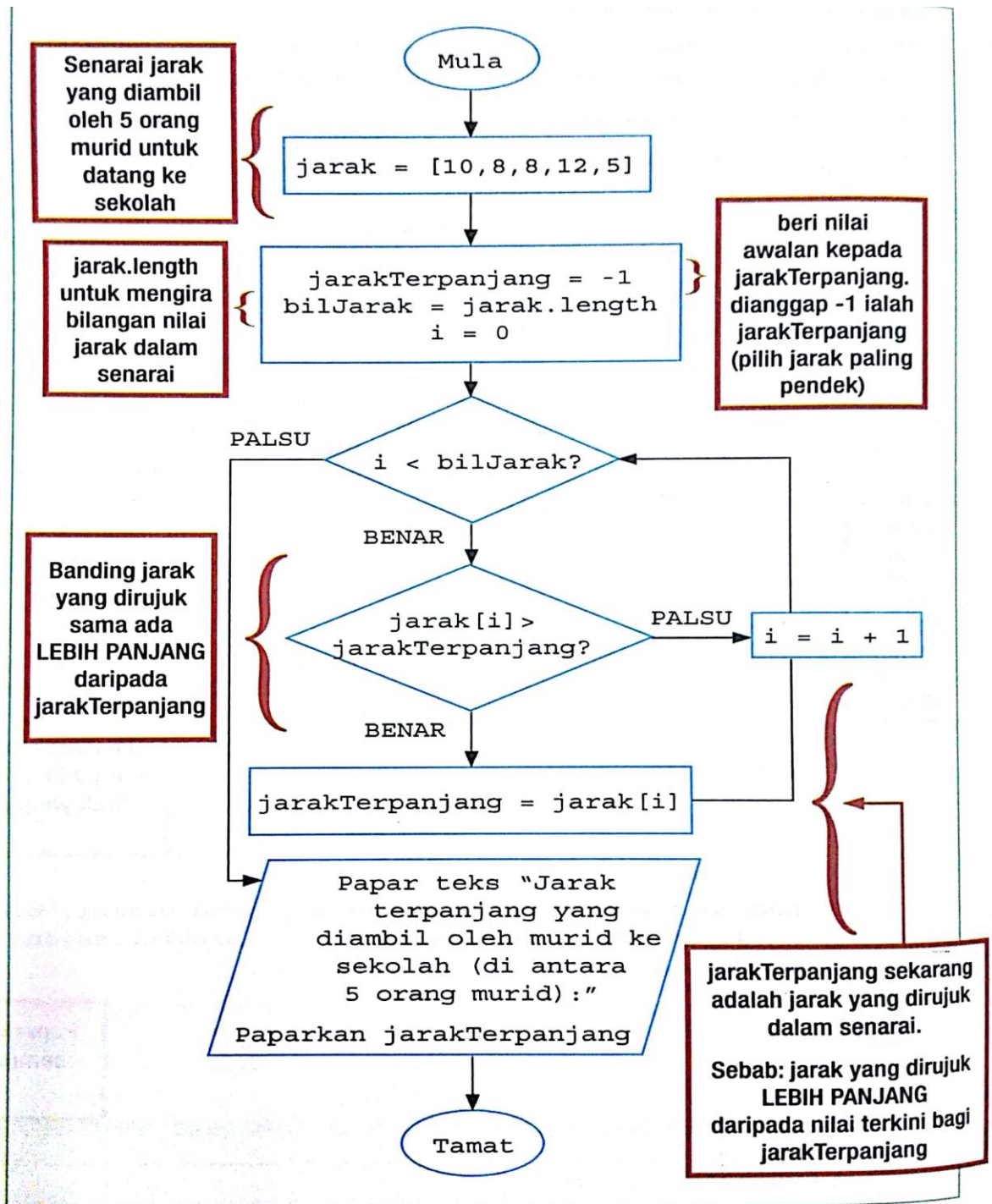
PROSES Mencari Nilai Maksimum

ATURCARA Mencari Nilai Maksimum

```
<html>
  <body>
    <script>
      var jarak =[10,8,8,12,5],jarakTerpanjang = -1,i,bilJarak = jarak.length;
      for (i=0 ; i < bilJarak ;i++)
      {
        if (jarak[i] > jarakTerpanjang)
        {
          jarakTerpanjang = jarak[i];
        }
      }
      document.write(" jarak terpanjang yang diambil oleh murid ke sekolah (dia antara 5 orang
murid) : " + jarakTerpanjang + "km" );
    </script>
  </body>
</html>
```



CARTA ALIR (JARAK TERPANJANG KE SEKOLAH)



Rajah 3.1.32 Carta alir (mendapatkan jarak terpanjang ke sekolah)

MENDAPATKAN NILAI MINIMUM

- Nilai minimum merujuk nilai terkecil dalam suatu senarai nilai.
- Terdapat hanya satu proses perbandingan : bandingkan nilai yang dirujuk dalam senarai sama ada lebih kecil daripada nilai minimum.

Beri nilai awal minimum

Banding nilai dalam senarai dengan nilai awal minimum

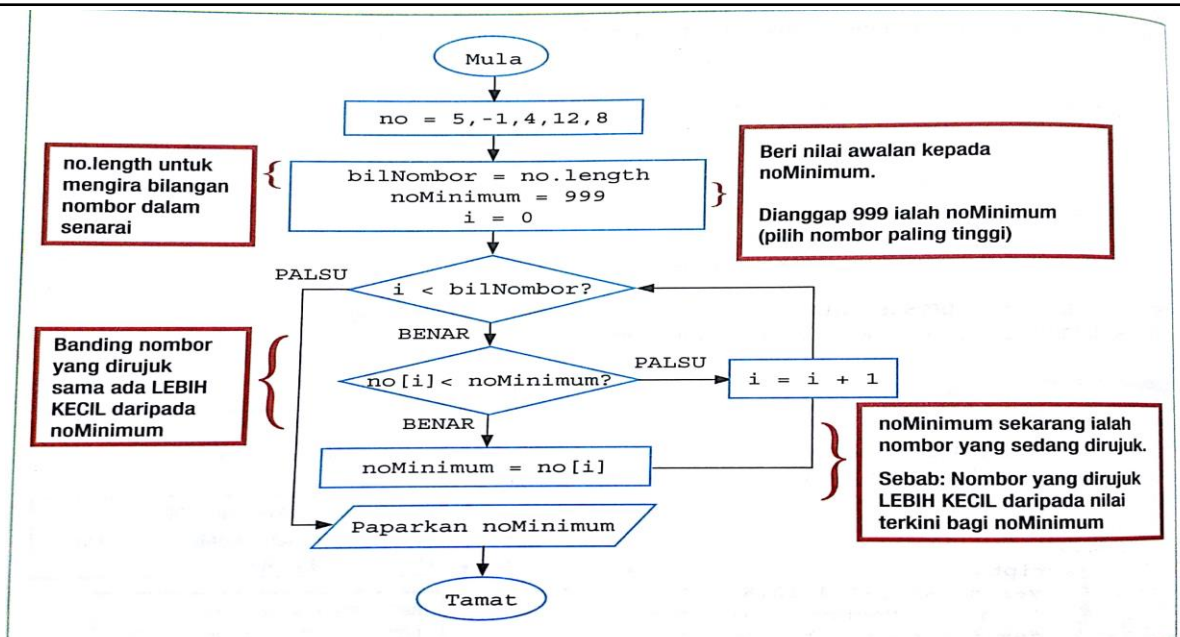
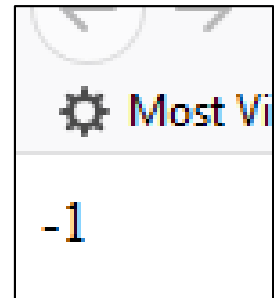
Dapat nilai minimum terkini

PROSES Mencari Nilai Minimum

ATURCARA Mencari Nilai Minimum

```
<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>
```



Rajah 3.1.35 Carta alir (mendapatkan nilai minimum)

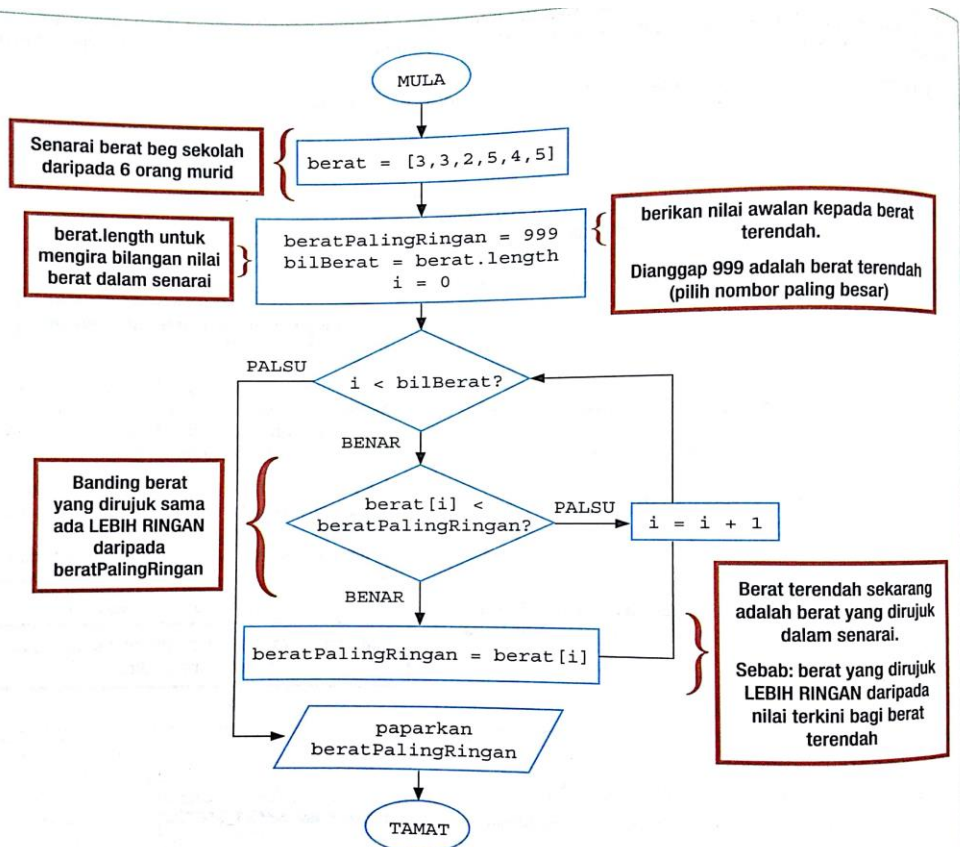
ATURCARA MENDAPATKAN BEG PALING RINGAN

```
<html>
  <body>
    <script>
      var berat =[3,3,2,5,4,5],beratPalingRingan= 999,i, bilBerat = berat.length;

      for (i=0; i<bilBerat;i++)
      {
        if (berat[i] < beratPalingRingan)
        {
          beratPalingRingan = berat[i];
        }
      }
      document.write("Berat beg paling ringan (dikalangan 6 orang murid) : " + beratPalingRingan + "kg");
    </script>
  </body>
</html>
```

Most Visited Getting Started Google Translate

Berat beg paling ringan (dikalangan 6 orang murid) : 2kg



Rajah 3.1.37 Carta alir (mendapatkan berat beg paling ringan)

MENDAPATKAN NILAI *MEAN*

- **Mean** : merujuk purata nilai dalam suatu senarai nilai .

Jumlahkan nilai-nilai
dalam senarai nilai.



Dapatkan bilangan nilai yang
terdapat dalam senarai.



$Mean = \text{Jumlah} / \text{Bilangan}$

Proses mencari nilai *mean*ATURCARA MENCARI NILAI *MEAN*

```
<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>
```

OUTPUT

Masukkan satu nombor :

OK

Cancel

Ulang sekali lagi?

☐ Prevent this page from creating additional dialogs

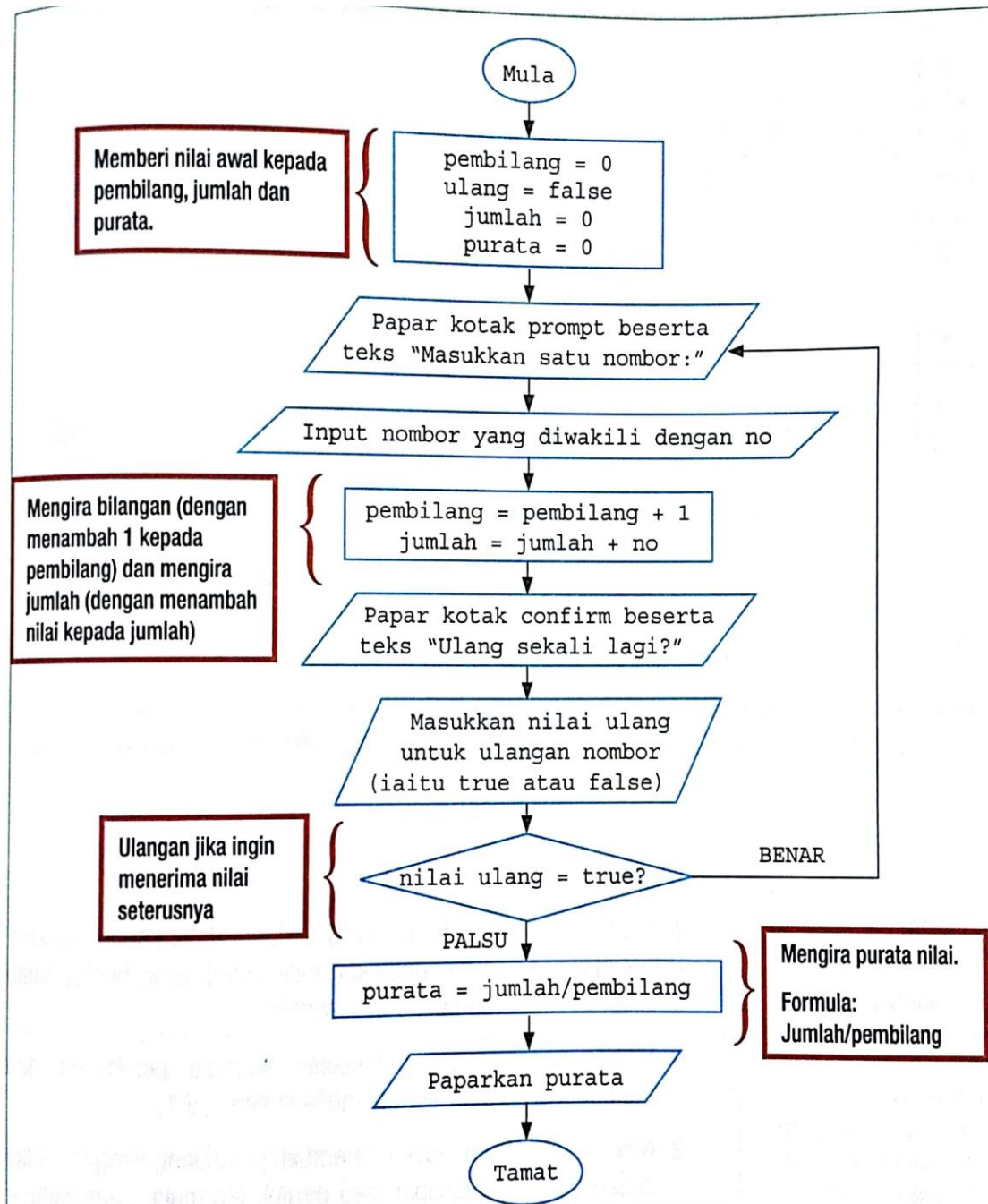
OK

Cancel

Most Visited

3.5

CARTA ALIR (MENDAPATKAN NILAI *MEAN*)



Rajah 3.1.39 Carta alir (mendapatkan nilai *mean*)

3.1

3.1.2

MENTERJEMAH ATUR CARA MUDAH YANG DIBERI DARIPADA BAHASA PENSKRIPAN KLIEN KEPADA CARTA ALIR

MENDAPATKAN NILAI *COUNT*

- **Count** : mengira bilangan nilai yang terdapat dalam suatu nilai.

Guna satu perwakilan sebagai pembilang dan berikan nilai awalan 1.



Tambah pembilang dengan 1 bagi setiap nilai yang dicapai atau dirujuk.
($Pembilang = pembilang + 1$)



Papar nilai akhir pembilang

Proses mencari nilai *count*

ATURCARA MENCARI NILAI *COUNT*

```
<html>
  <body>
    <script>
      var nama, pembilang = 0, ulang =false;

      do
      {
        nama = parseInt (prompt("Masukkan nama anda : "));
        pembilang = pembilang +1;
        ulang = confirm ("Ulang sekali lagi?");
      } while (ulang == true);

      document.write(pembilang);
    </script>
  </body>
</html>
```

OUTPUT

Masukkan nama anda :

OK Cancel

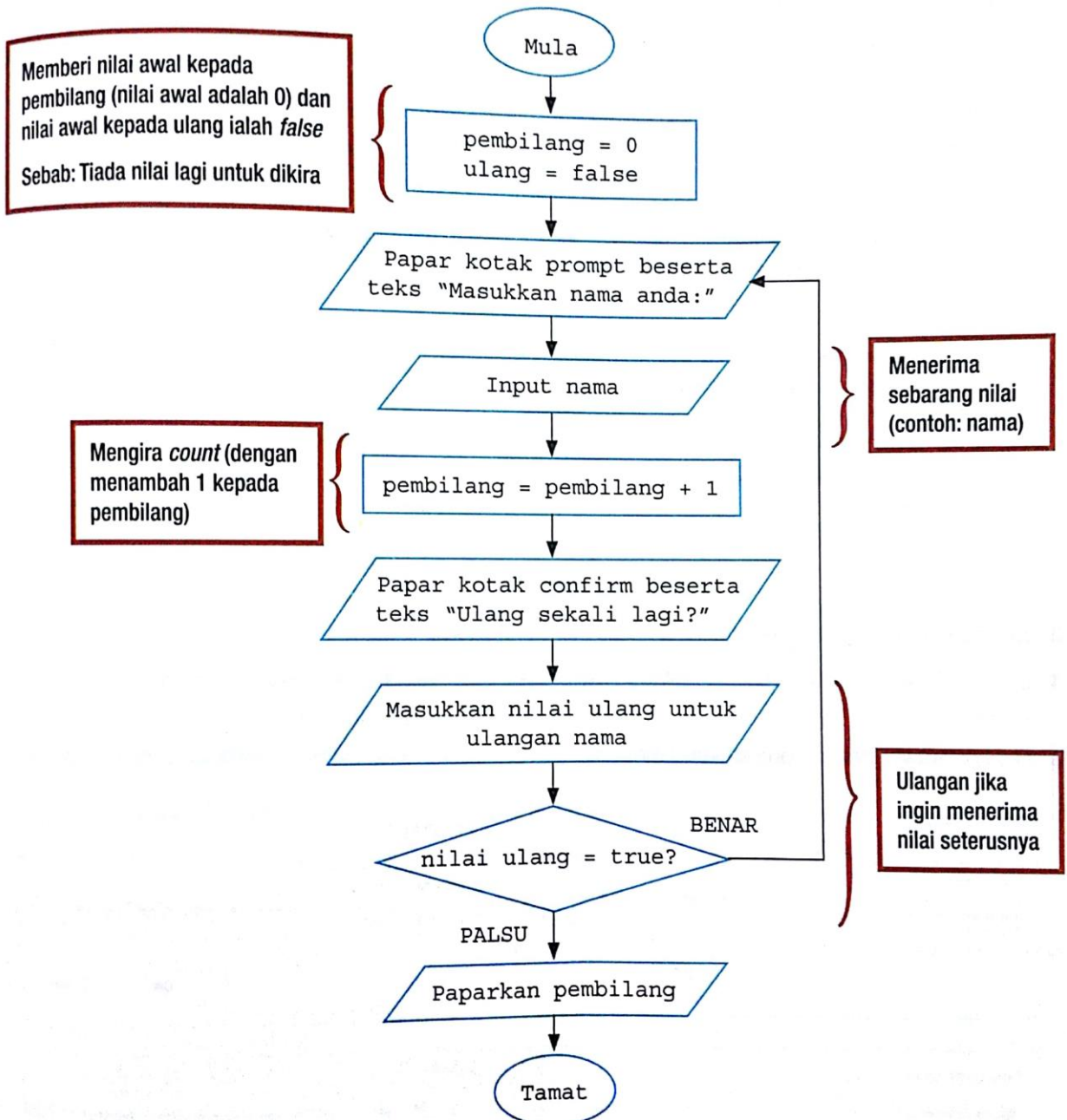
Ulang sekali lagi?

☐ Prevent this page from creating additional dialogs

OK Cancel

Most Visited

3

CARTA ALIR (MENDAPATKAN NILAI *COUNT*)Rajah 3.1.41 Carta alir (mendapatkan nilai *count*)

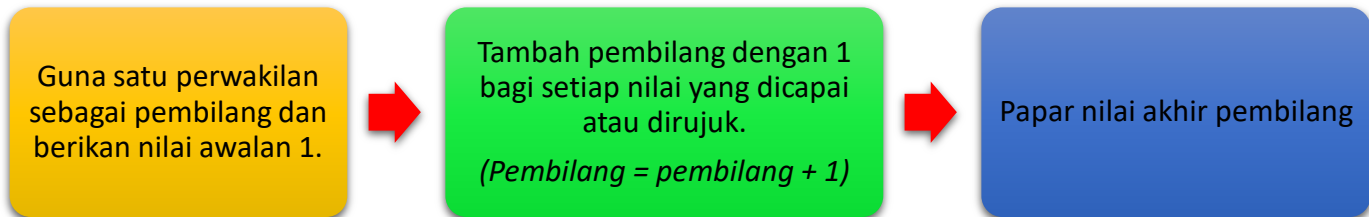
3.1

3.1.2

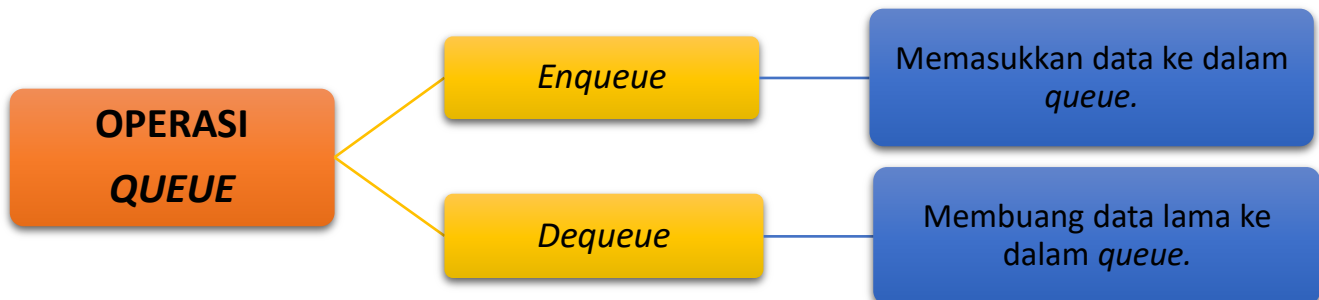
MENTERJEMAH ATUR CARA MUDAH YANG DIBERI DARIPADA BAHASA PENSKRIPAN KLIEN KEPADA CARTA ALIR

PENGUMPUKAN NILAI DALAM STRUKTUR GILIRAN (*QUEUE*)

- **Queue** : merupakan struktur data linear – pengumpulan nilai dalam ruang ingatan secara teratur.
- *Queue* boleh membuang nilai lama → nilai yang terawal dimasukkan.
- Kaedah ini dipanggil FIFO – First-In , First-Out



Proses *Queue*



PENGUNAAN push () dan shift () dalam JavaScript

- Nilai awal dimasukkan melalui bahagian belakang ruang ingatan *array* .
- Nilai awal dikeluarkan melalui bahagian depan ruang ingatan *array*.

	depan		belakang
push (5)			5
	depan		belakang
push (-1)		5	-1
	depan		belakang
push (12)	5	-1	12
	depan		belakang
shift ()	-1	12	

PENGUNAAN unshift () dan pop () dalam JavaScript

- Nilai awal dimasukkan melalui bahagian depan ruang ingatan *array* .
- Nilai awal dikeluarkan melalui bahagian belakang ruang ingatan *array*.

	depan		belakang
unshift (5)	5		
	depan		belakang
unshift (-1)	-1	5	
	depan		belakang
unshift(12)	12	-1	5
	depan		belakang
pop ()		12	-1

ATURCARA PENGUMPULAN NILAI-NILAI DALAM STRUKTUR *QUEUE*

```
<html>
  <body>
    <script>
      var queueNo = [ ] , noBaru, noKeluar;

      noBaru = 5;
      queueNo.push(noBaru);
      document.write("<br> Nombor Baru Masuk : " , noBaru);

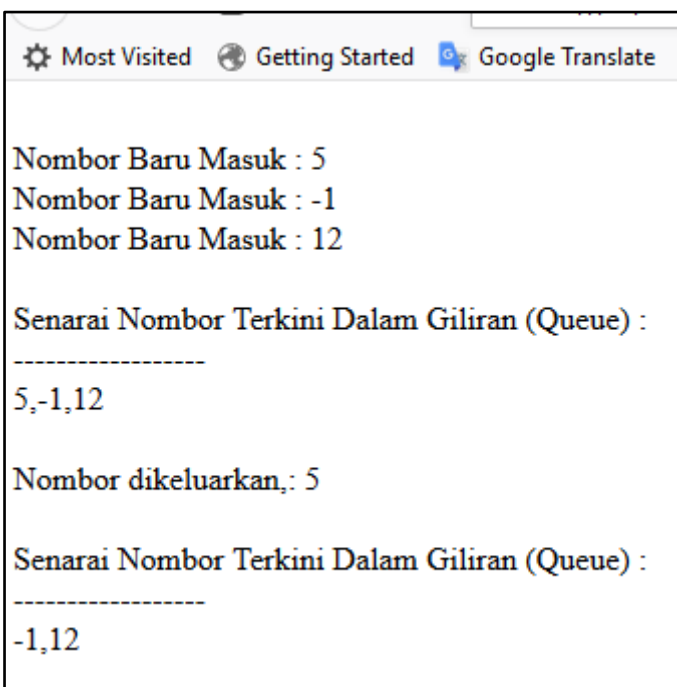
      noBaru = -1;
      queueNo.push(noBaru);
      document.write("<br> Nombor Baru Masuk : " , noBaru);

      noBaru = 12;
      queueNo.push(noBaru);
      document.write("<br> Nombor Baru Masuk : " , noBaru);

      document.write ("<br><br> Senarai Nombor Terkini Dalam Giliran (Queue) : <br>-----");
      document.write ("<br>", queueNo);

      noKeluar = queueNo.shift();
      document.write ("<br><br> Nombor dikeluarkan,: " , noKeluar);
      document.write ("<br><br> Senarai Nombor Terkini Dalam Giliran (Queue) : <br>-----");
      document.write ("<br>", queueNo) ;
    </script>
  </body>
</html>
```

OUTPUT



Most Visited Getting Started Google Translate

Nombor Baru Masuk : 5
 Nombor Baru Masuk : -1
 Nombor Baru Masuk : 12

Senarai Nombor Terkini Dalam Giliran (Queue) :

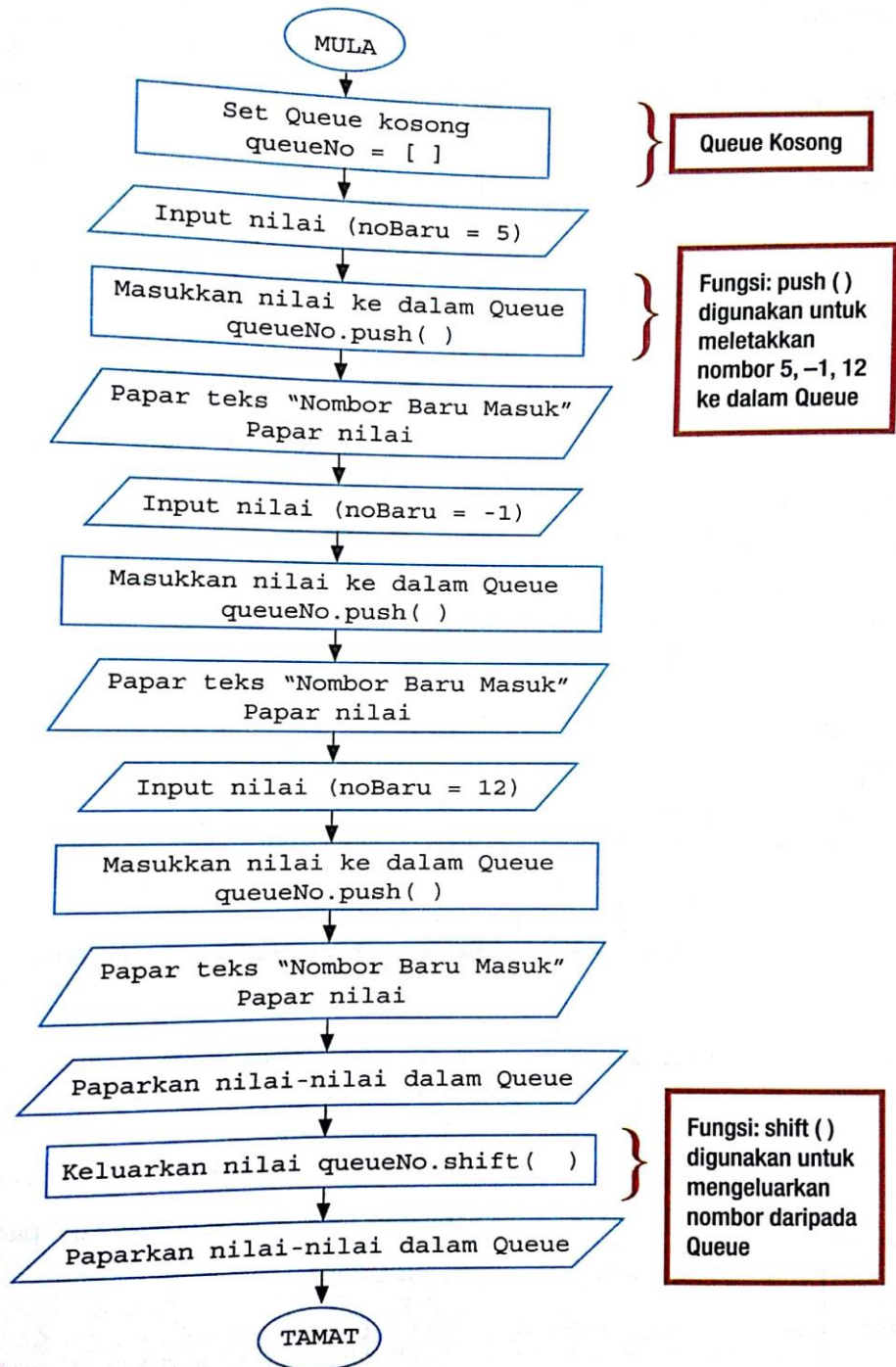
 5,-1,12

Nombor dikeluarkan,: 5

Senarai Nombor Terkini Dalam Giliran (Queue) :

 -1,12

CARTA ALIR PENGUMPULAN NILAI-NILAI DALAM STRUKTUR *QUEUE*



Rajah 3.1.43 Carta alir (pengumpulan nilai-nilai dalam struktur *queue*)

ATURCARA PENGUMPULAN NILAI-NILAI DALAM STRUKTUR *QUEUE*

```
<html>
  <body>
    <script>
      var queueNo = [ ] , noBaru, noKeluar;

      noBaru = 5;
      queueNo.unshift(noBaru);
      document.write("<br> Nombor Baru Masuk : " , noBaru);

      noBaru = -1;
      queueNo.unshift(noBaru);
      document.write("<br> Nombor Baru Masuk : " , noBaru);

      noBaru = 12;
      queueNo.unshift(noBaru);
      document.write("<br> Nombor Baru Masuk : " , noBaru);

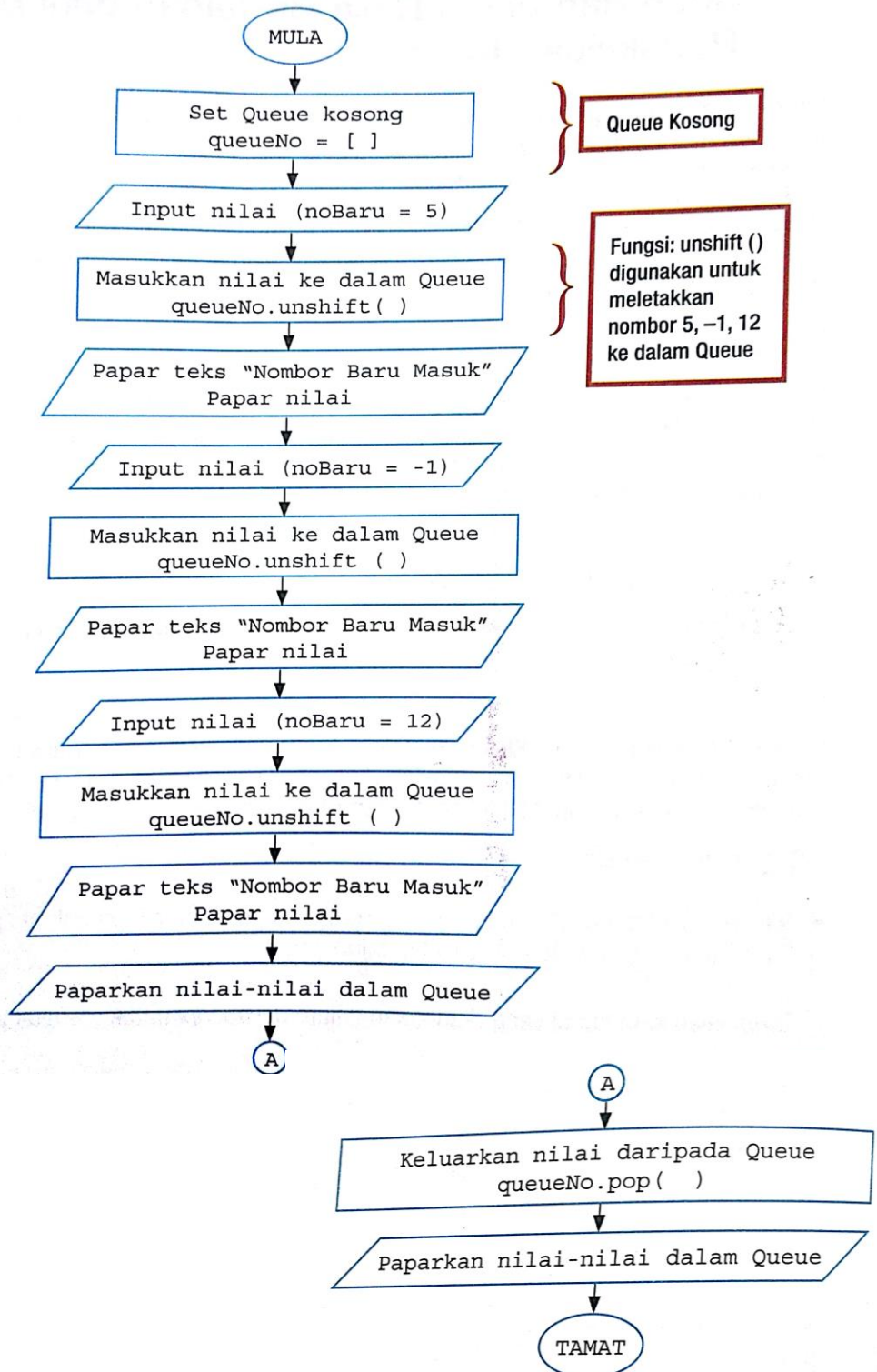
      document.write ("<br><br> Senarai Nombor Terkini Dalam Giliran (Queue) : <br>-----");
      document.write ("<br>", queueNo);

      noKeluar = queueNo.pop();
      document.write ("<br><br> Nombor dikeluarkan,: " , noKeluar);
      document.write ("<br><br> Senarai Nombor Terkini Dalam Giliran (Queue) : <br>-----");
      document.write ("<br>", queueNo);
    </script>
  </body>
</html>
```

OUTPUT



CARTA ALIR PENGUMPULAN NILAI-NILAI DALAM STRUKTUR QUEUE



Rajah 3.1.45 Carta alir (pengumpulan nilai-nilai dalam struktur queue)

PEMBOLEH UBAH



- Ruang ingatan yang digunakan untuk menyimpan nilai sementara sewaktu atur cara diproses.
- Nilainya boleh berubah-ubah mengikut penggunaan pemboleh ubah atau arahan kepada pemboleh ubah.
- Setiap pemboleh ubah akan diberi nama mengikut nilai yang disimpan.
- **Pengecam** : Nama yang diberi kepada pemboleh ubah.

CONTOH PEMBOLEH UBAH

NILAI	NAMA PEMBOLEH UBAH
Nilai Gaji Pekerja	gajiPekerja
Nama Pelajar	namaPelajar

PEMALAR

- Ruang ingatan yang digunakan untuk menyimpan nilai sementara sewaktu atur cara diproses.
- Sama seperti pemboleh ubah tetapi nilainya tidak boleh berubah-ubah kerana nilai yang ditentukan itu merupakan satu nilai yang telah ditetapkan dalam atur cara.

CONTOH PEMALAR

- Nilai $\pi \rightarrow 3.142$
- Bilangan hari dalam seminggu $\rightarrow 7$ hari

JENIS DATA

- Jenis nilai atau pelbagai set data yang digunakan dalam atur cara.

JENIS DATA	CONTOH DATA	
Nombor	5	100.50
Rentetan (<i>String</i>)	" Ali bin Abu "	" BMW 318i "
<i>Boolean</i>	true	false

ISIHAN BUIH (*BUBBLE SORT*) – JENIS DATA NOMBOR

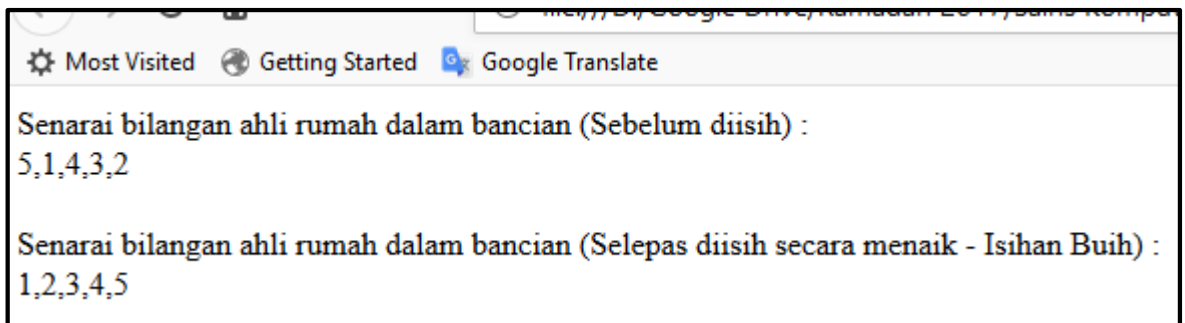
```
<html>
<body>
  <script>
    var banciPenduduk = [5,1,4,3,2];
    var bilNombor = banciPenduduk.length,i, j;
    var sementara = banciPenduduk [0];

    document.write ("Senarai bilangan ahli rumah dalam bancian (Sebelum diisih) : <br> ");
    document.write (banciPenduduk);

    document.write (" <br><br> Senarai bilangan ahli rumah dalam bancian (Selepas diisih secara menaik - Isihan Buih) : <br> ");

    for ( i=0; i < bilNombor - 1; i++)
    {
      for (j = 0; j < bilNombor -i-1 ; j++)
      {
        if (banciPenduduk[j] > banciPenduduk[j+1])
        {
          sementara = banciPenduduk [j];
          banciPenduduk [j] = banciPenduduk [j+1];
          banciPenduduk [ j+1] = sementara;
        }
      }
    }
    document.write (banciPenduduk);
  </script>
</body>
</html>
```

OUTPUT



Senarai bilangan ahli rumah dalam bancian (Sebelum diisih) :
5,1,4,3,2

Senarai bilangan ahli rumah dalam bancian (Selepas diisih secara menaik - Isihan Buih) :
1,2,3,4,5

ISIHAN BUIH (*BUBBLE SORT*) – JENIS DATA *STRING*

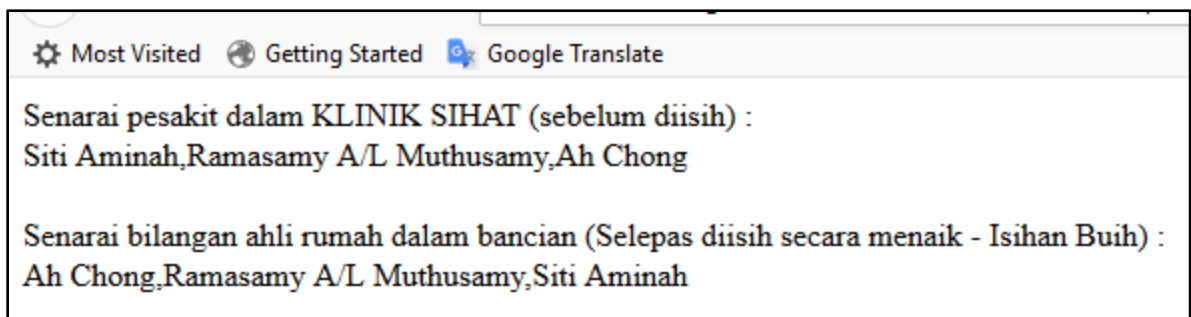
```
<html>
<body>
  <script>
    var namaPesakit = ["Siti Aminah", "Ramasamy A/L Muthusamy", "Ah Chong"];
    var bilPesakit = namaPesakit.length,i, j;
    var sementara = bilPesakit [0];

    document.write ("Senarai pesakit dalam KLINIK SIHAT (sebelum diisih) : <br> ");
    document.write (namaPesakit);

    document.write ("<br><br> Senarai bilangan ahli rumah dalam banciaan (Selepas diisih secara menaik - Isihan Buih) : <br> ");

    for ( i=0; i < bilPesakit - 1; i++)
    {
      for (j = 0; j < bilPesakit -i-1 ; j++)
      {
        if (namaPesakit[j] > namaPesakit[j+1])
        {
          sementara = namaPesakit [j];
          namaPesakit [j] = namaPesakit [j+1];
          namaPesakit [j+1] = sementara;
        }
      }
    }
    document.write (namaPesakit);
  </script>
</body>
</html>
```

OUTPUT



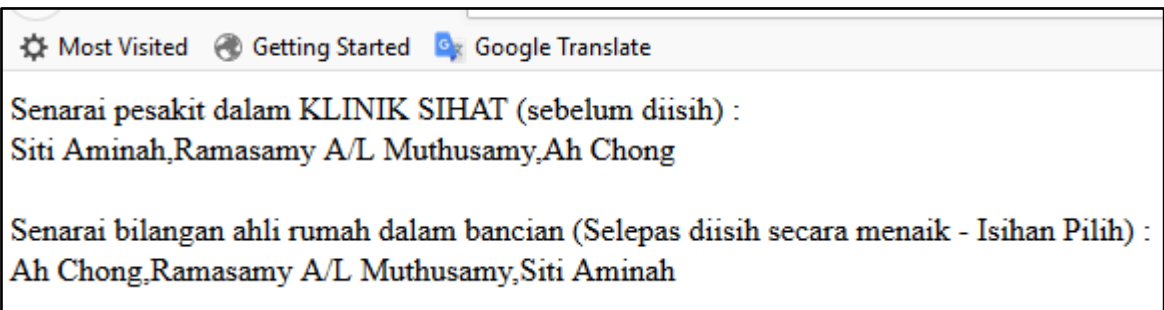
ISIHAN PILIH (*SELECTION SORT*) – JENIS DATA *STRING*

```
.html>
<body>
  <script>
    var namaPesakit = ["Siti Aminah", "Ramasamy A/L Muthusamy", "Ah Chong"];
    var i,j,min,sementara;
    var bilPesakit = namaPesakit.length;

    document.write ("Senarai pesakit dalam KLINIK SIHAT (sebelum diisih) : <br> ");
    document.write (namaPesakit);
    document.write ("<br><br> Senarai bilangan ahli rumah dalam banciaan (Selepas diisih secara menaik - Isihan Pilih) : <br> ");

    for ( i=0; i < bilPesakit - 1; i++)
    {
      min = i;
      for (j = i + 1; j < bilPesakit; j++)
      {
        if (namaPesakit[j] < namaPesakit[min])
        {
          min = j;
        }
      }
      if (min != i)
      {
        sementara = namaPesakit [i];
        namaPesakit [i] = namaPesakit [min];
        namaPesakit [min] = sementara;
      }
    }
    document.write (namaPesakit);
  </script>
</body>
./html>
```

OUTPUT



Senarai pesakit dalam KLINIK SIHAT (sebelum diisih) :
Siti Aminah,Ramasamy A/L Muthusamy,Ah Chong

Senarai bilangan ahli rumah dalam banciaan (Selepas diisih secara menaik - Isihan Pilih) :
Ah Chong,Ramasamy A/L Muthusamy,Siti Aminah

CARIAN PERDUAAN (*BINARY SEARCH*) – JENIS DATA *STRING*

```
<html>
<body>
  <script>
    var namaPesakit = ["Ah Chong","Ramasamy A/L Muthusamy","Siti Aminah"];
    var namaPesakitCarian = "Ramasamy A/L Muthusamy";
    var indeksAwal = 0, indeksAkhir = namaPesakit.length-1, indeksTengah = Math.floor((indeksAkhir + indeksAwal) / 2);

    document.write ("Senarai pesakit dalam KLINIK SIHAT : <br> ");
    document.write (namaPesakit);
    document.write ("<br><br> Nama Pesakit yang dicari : ", namaPesakitCarian);

    while (namaPesakit [indeksTengah] != namaPesakitCarian && indeksAwal < indeksAkhir)
    {
      if (namaPesakit[indeksTengah] > namaPesakitCarian)
      {
        indeksAkhir = indeksTengah;
      }
      else
      {
        if (namaPesakit[indeksTengah] < namaPesakitCarian)
        {
          indeksAwal = indeksTengah + 1;
        }
      }
      indeksTengah = Math.floor ((indeksAkhir + indeksAwal) / 2);
    }
    if (namaPesakit [indeksTengah] == namaPesakitCarian)
    {
      document.write (" (masih dalam giliran)");
    }
  </script>
</body>
</html>
```

OUTPUT

⚙ Most Visited 🌐 Getting Started 🗨 Google Translate

Senarai pesakit dalam KLINIK SIHAT :

Ah Chong,Ramasamy A/L Muthusamy,Siti Aminah

Nama Pesakit yang dicari : Ramasamy A/L Muthusamy (masih dalam giliran)

PENGUMPUMAN NILAI DALAM QUEUE– JENIS DATA NOMBOR

```

<html>
<body>
<script>
    var queueNo = [],noBaru,noKeluar;

    noBaru = 5.5;
    queueNo.unshift(noBaru);
    document.write ("<br>Nombor baru masuk : " , noBaru);

    noBaru = -1.3;
    queueNo.unshift(noBaru);
    document.write ("<br>Nombor pelajar masuk : " , noBaru);

    noBaru = 12.95;
    queueNo.unshift(noBaru);
    document.write ("<br>Nombor pelajar masuk : " , noBaru);

    document.write ("<br><br> Senarai Nombor Pelajar Terkini dalam Giliran (Queue) : <br>----- ");
    document.write ("<br>",queueNo);

    noKeluar = queueNo.pop();
    document.write ("<br><br> Nombor Pelajar dikeluarkan:", noKeluar);

    document.write ("<br><br> Senarai Nombor Pelajar Terkini dalam Giliran (Queue) : <br>----- ");
    document.write ("<br>",queueNo);

</script>
</body>
</html>

```

OUTPUT

```

Nombor baru masuk : 5.5
Nombor pelajar masuk : -1.3
Nombor pelajar masuk : 12.95

Senarai Nombor Pelajar Terkini dalam Giliran (Queue) :
-----
12.95,-1.3,5.5

Nombor Pelajar dikeluarkan:5.5

Senarai Nombor Pelajar Terkini dalam Giliran (Queue) :
-----
12.95,-1.3

```

PENGUMPUMAN NILAI DALAM QUEUE– JENIS DATA *STRING*

```

<html>
<body>
<script>
    var queueNama = [],namaBaru,namaKeluar;

    namaBaru = "Siti Maimunah";
    queueNama.unshift(namaBaru);
    document.write ("<br>Nombor pesakit baru masuk : " , namaBaru);

    namaBaru = "Ramasamy A/L Muthusamy";
    queueNama.unshift(namaBaru);
    document.write ("<br>Nombor pesakit masuk : " , namaBaru);

    namaBaru = "Ah Chong";
    queueNama.unshift(namaBaru);
    document.write ("<br>Nombor pesakit masuk : " , namaBaru);

    document.write ("<br><br> Senarai Nama Pesakit Terkini dalam Giliran (Queue) : <br>----- ");
    document.write ("<br>",queueNama);

    namaKeluar = queueNama.pop();
    document.write ("<br><br> Nombor Pesakit dikeluarkan:", namaKeluar);

    document.write ("<br><br> Senarai Nama Pesakit Terkini dalam Giliran (Queue) : <br>----- ");
    document.write ("<br>",queueNama);

</script>
</body>
</html>

```

OUTPUT

```

Nombor pesakit baru masuk : Siti Maimunah
Nombor pesakit masuk : Ramasamy A/L Muthusamy
Nombor pesakit masuk : Ah Chong

Senarai Nama Pesakit Terkini dalam Giliran (Queue) :
-----
Ah Chong,Ramasamy A/L Muthusamy,Siti Maimunah

Nombor Pesakit dikeluarkan:Siti Maimunah

Senarai Nama Pesakit Terkini dalam Giliran (Queue) :
-----
Ah Chong,Ramasamy A/L Muthusamy

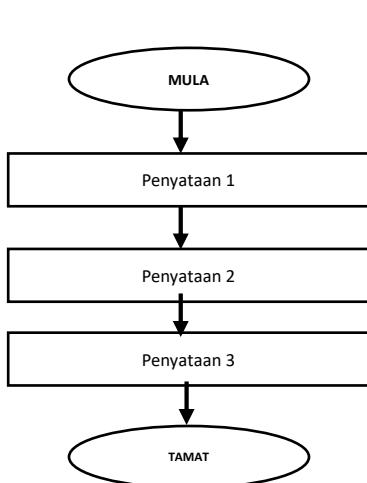
```

STRUKTUR KAWALAN

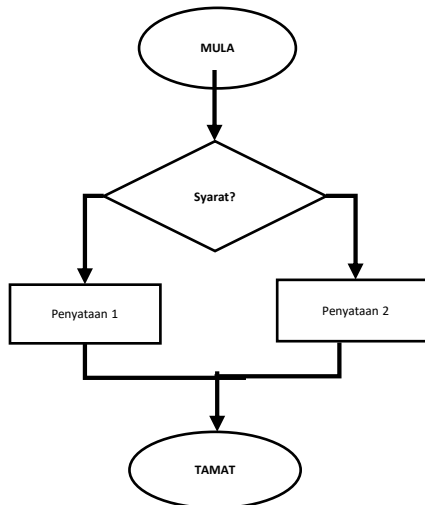
- Akan mengawal aliran atur cara satu blok kod atur cara.
- Pernyataan ini akan menentukan sama ada pernyataan akan dilaksanakan atau tidak, termasuk mengulang pernyataan dalam satu blok kod.

STRUKTUR KAWALAN
(dalam Javascript)

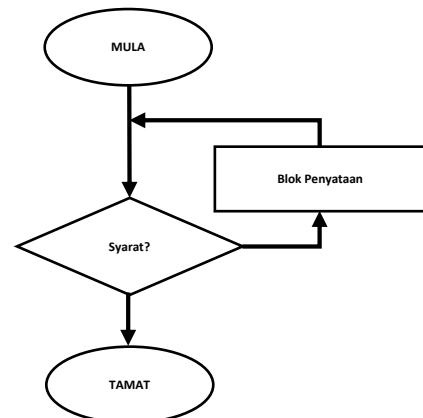
Berjujukan



Pilihan



Ulangan

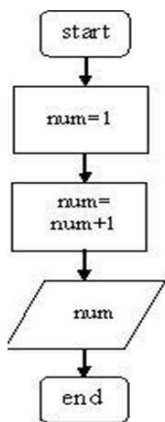


STRUKTUR KAWALAN JUJUKAN

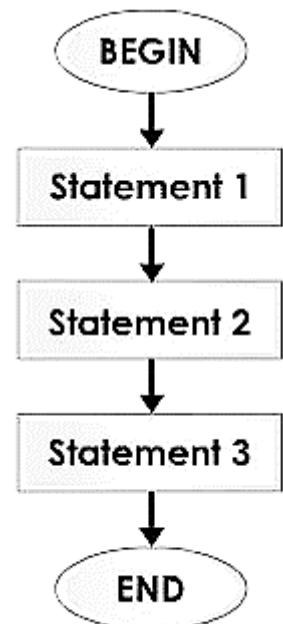
ULANGKAJI

- Arahan atau pernyataan dilaksanakan mengikut tertib sepertimana yang diarahkan dalam suatu kod atur cara tanpa perlu memeriksa sebarang syarat.
- Antara pernyataan yang terdapat dalam struktur kawalan jujukan ialah input data, proses seperti pengiraan (operasi matematik) dan paparan cetakan output.
- Input data boleh diterima daripada pengguna (secara interaktif) atau ditetapkan dalam atur cara.

Sequence Control Structure



- All actions must be executed
- Only one way to enter this sequence structure (at the start)
- Only one way to exit the structure (at the end)
- What is the output?

Pseudocodes: The **Sequence** control structure

- A series of steps or statements that are executed in the order they are written in an algorithm.
- The beginning and end of a block of statements can be optionally marked with the keywords *begin* and *end*.
- Example-1:

Begin

Read the birth date from the user.
 Calculate the difference between the birth date and today's date.
 Print the user age.

End

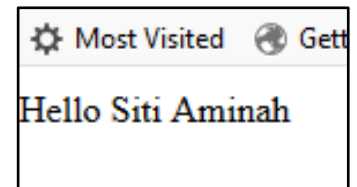
CONTOH ATUR CARA STRUKTUR KAWALAN JUJUKAN

ATURCARA

OUTPUT

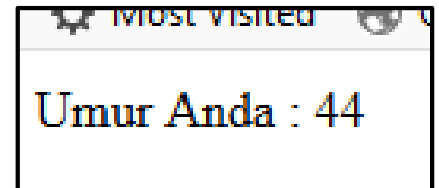
```
<html>
  <head>
    <title>Struktur Kawalan Jujukan </title>
  </head>

  <body>
    <script>
      var namaAnda;
      namaAnda = "Siti Aminah";
      document.write ("Hello " + namaAnda);
    </script>
  </body>
</html>
```



```
<html>
  <head>
    <title>Struktur Kawalan Jujukan </title>
  </head>

  <body>
    <script>
      var tahunSemasa,tahunLahir,umur;
      tahunSemasa = 2017;
      tahunLahir = 1973;
      umur = tahunSemasa - tahunLahir;
      document.write ("Umur Anda : " + umur);
    </script>
  </body>
</html>
```

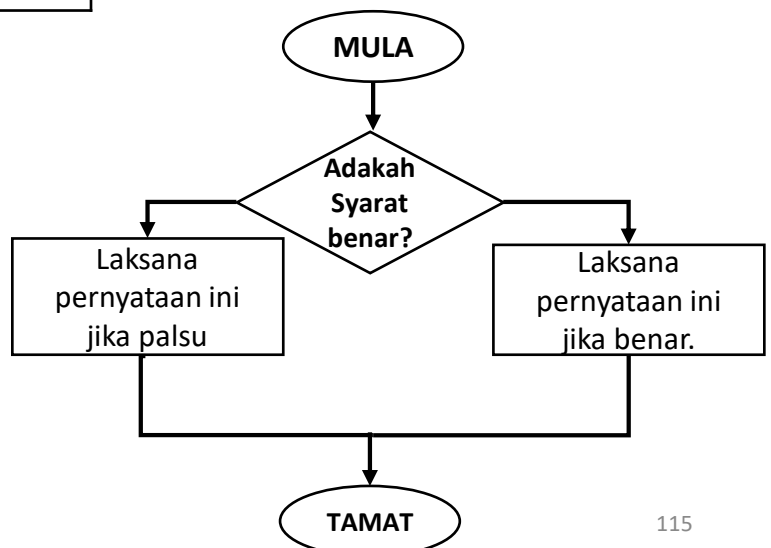
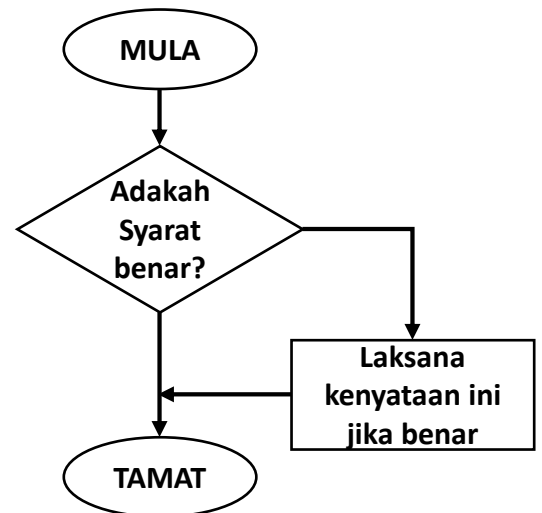


STRUKTUR KAWALAN PILIHAN

- Memerlukan pernyataan bersyarat yang akan melaksanakan satu pernyataan atau satu blok yang ditutupi “{” dan “}”.
- Pernyataan bersyarat dilaksanakan dengan menggunakan pernyataan yang dipanggil ungkapan logik yang terdiri daripada **operator hubungan**.
- Ungkapan logik akan menghasilkan nilai *Boolean* iaitu *true* atau *false* untuk menentukan sama ada pernyataan atau satu blok kod akan dilaksanakan atau tidak.
- **Pernyataan** : *if ...* dan *if ...else...*
- Dalam pernyataan *if..else..*, dua atau lebih syarat boolean boleh diuji serentak dengan menggabungkan syarat boolean melalui **operator logikal**.

OPERATOR HUBUNGAN	
Digunakan untuk membandingkan dua nilai bagi menghasilkan keputusan Boolean.	
==	sama dengan
!=	tidak sama dengan
>	lebih besar daripada
>=	lebih besar daripada atau sama dengan
<	Kurang daripada
<=	Kurang atau sama dengan

OPERATOR LOGIKAL	
&&	AND
	OR
!	NOT



CONTOH ATUR CARA STRUKTUR KAWALAN PILIHAN

ATURAN

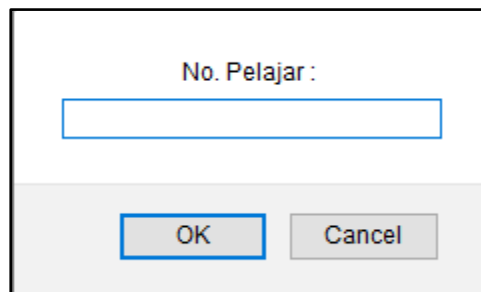
```
<html>
  <head>
    <title>Struktur Kawalan Pilihan </title>
  </head>

  <body>
    <script>
      var noPelajar;

      noPelajar = prompt ("No. Pelajar : ");

      if (noPelajar == 123)
        document.write(" Anda adalah pelajar rendah Sekolah Taman ABC ");
    </script>
  </body>
</html>
```

OUTPUT



Most Visited Getting Started Google Translate

Anda adalah BUKAN pelajar rendah Sekolah Taman ABC

CONTOH ATUR CARA STRUKTUR KAWALAN PILIHAN

ATURAN

```
<html>
  <head>
    <title>Struktur Kawalan Pilihan </title>
  </head>

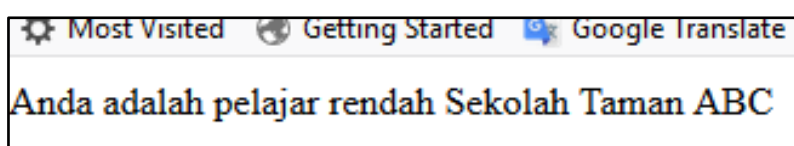
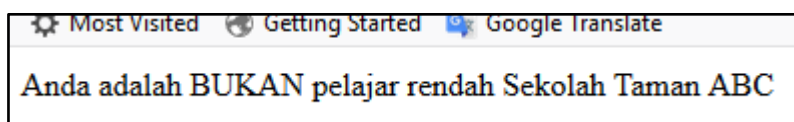
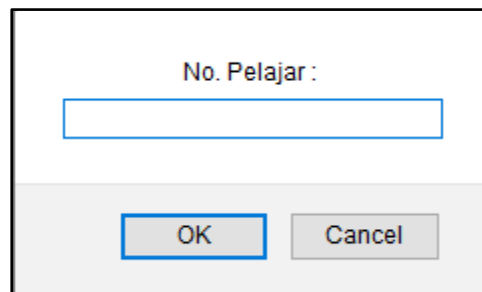
  <body>
    <script>
      var noPelajar;

      noPelajar = prompt ("No. Pelajar : ");

      if (noPelajar == 123)
        document.write(" Anda adalah pelajar rendah Sekolah Taman ABC ");
      else
        document.write(" Anda adalah BUKAN pelajar rendah Sekolah Taman ABC");

    </script>
  </body>
</html>
```

OUTPUT



CONTOH ATUR CARA STRUKTUR KAWALAN PILIHAN

ATURCARA

Markah Peperiksaan : 67 (Gred B)

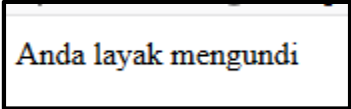
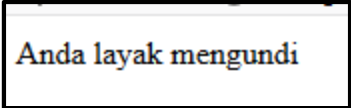
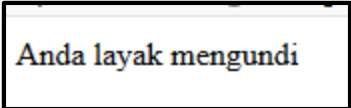
OUTPUT

```
<html>
  <body>
    <script>
      var markahPeperiksaan = 67, gred;

      if (markahPeperiksaan >= 0 && markahPeperiksaan <= 39)
        {gred = " Gred G"; }
      else
      {
        if (markahPeperiksaan >= 40 && markahPeperiksaan <= 44)
          {gred = "Gred E"; }
        else
        {
          if (markahPeperiksaan >= 45 && markahPeperiksaan <= 49)
            {gred = " Gred D";}
          else
          {
            if (markahPeperiksaan >= 50 && markahPeperiksaan <= 59)
              {gred = " Gred C";}
            else
            {
              if (markahPeperiksaan >= 60 && markahPeperiksaan <= 64)
                {gred = " Gred C+"; }
              else
              {
                if (markahPeperiksaan >= 65 && markahPeperiksaan <= 69)
                  {gred = "Gred B";}
                else
                {
                  if (markahPeperiksaan >= 70 && markahPeperiksaan <= 75)
                    {gred = " Gred B+";}
                  else
                  {
                    if (markahPeperiksaan >= 75 && markahPeperiksaan <= 79)
                      {gred = " Gred A-";}
                    else
                    {
                      if (markahPeperiksaan >= 80 && markahPeperiksaan <= 89)
                        {gred = " Gred A";}
                      else
                      {
                        if (markahPeperiksaan >= 90 && markahPeperiksaan <= 100)
                          {gred = " Gred A+ "; }
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }

      document.write ("Markah Peperiksaan : " + markahPeperiksaan + " (" + gred + " ) ");
    </script>
  </body>
</html>
```

CONTOH ATUR CARA STRUKTUR KAWALAN PILIHAN

ATURCARA	OUTPUT
<pre> <html> <body> <script> var umur = 25, statusDaftar = true; if (umur >= 21 statusDaftar == true) { document.write (" Anda layak mengundi"); } else { document.write (" Anda tidak layak mengundi"); } </script> </body> </html> </pre>	
<pre> <html> <body> <script> var umur = 25, statusDaftar = false; if (umur >= 21 statusDaftar == true) { document.write (" Anda layak mengundi"); } else { document.write (" Anda tidak layak mengundi"); } </script> </body> </html> </pre>	
<pre> <html> <body> <script> var umur = 19, statusDaftar = true; if (umur >= 21 statusDaftar == true) { document.write (" Anda layak mengundi"); } else { document.write (" Anda tidak layak mengundi"); } </script> </body> </html> </pre>	

CONTOH ATUR CARA STRUKTUR KAWALAN PILIHAN

ATURCARA	OUTPUT
<pre><html> <body> <script> var umur = 19, statusDaftar = false; if (umur >= 21 statusDaftar == true) { document.write (" Anda layak mengundi"); } else { document.write (" Anda tidak layak mengundi"); } </script> </body> </html></pre>	Anda tidak layak mengundi

STRUKTUR KAWALAN ULANGAN

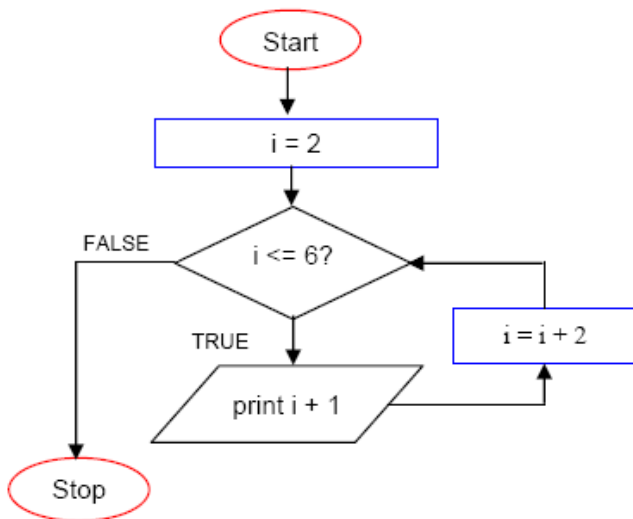
ULANGKAJI

- Memerlukan pernyataan bersyarat yang akan melaksanakan satu pernyataan atau satu blok berulang kali.
- Melaksanakan satu pernyataan yang diberi selagi syaratnya adalah *true*.
- **2 Jenis Struktur Kawalan Ulangan** : Ulangan berdasarkan pembilang dan Ulangan berdasarkan syarat

STRUKTUR KAWALAN ULANGAN

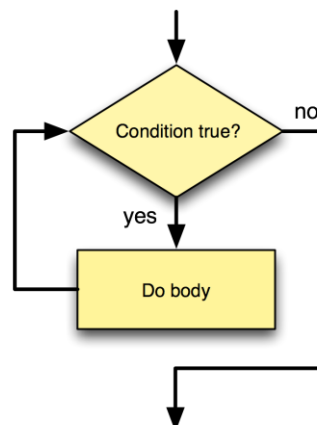
ULANGAN BERASASKAN PEMBILANG

- menggunakan pernyataan *for..*
- Mempunyai 3 komponen : Nilai awal, nilai tamat dan nilai kemas kini.

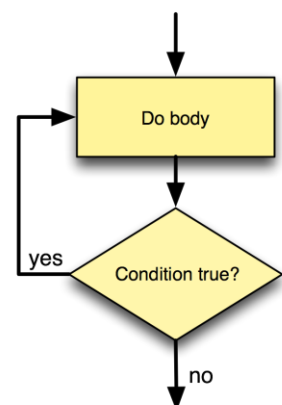


ULANGAN BERASASKAN SYARAT

- menggunakan pernyataan *while..* dan *do...while*.
- **Pernyataan While** : Melaksanakan suatu pernyataan sekali selagi syarat adalah *true*.
- **Pernyataan do..While** : Melaksanakan suatu pernyataan sekurang-kurangnya sekali selagi syarat adalah *true*



while flowchart



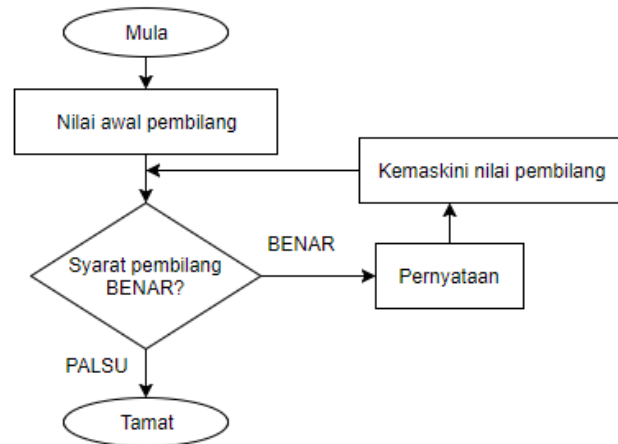
do/while flowchart

ULANGAN BERASASKAN PEMBILANG

ULANGKAJI

- Menggunakan satu pemboleh ubah untuk membuat ulangan kepada pernyataan dan syaratnya bergantung kepada pembilang.

PEMBILANG	
NILAI AWAL	Nilai permulaan untuk membilang.
NILAI TAMAT	Nilai untuk menamatkan proses membilang yang menentukan ulangan tamat atau diteruskan.
NILAI AKHIR	Proses untuk mengemas kini nilai pembilang.



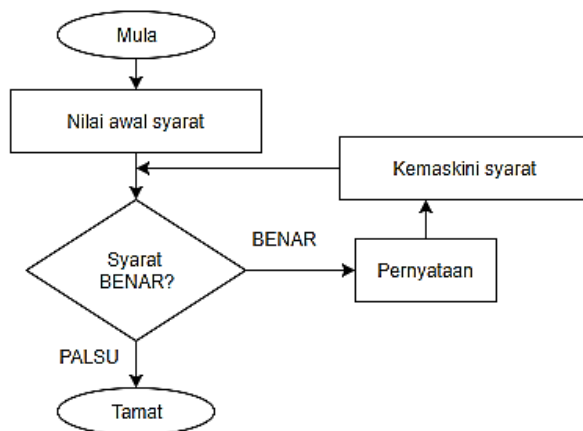
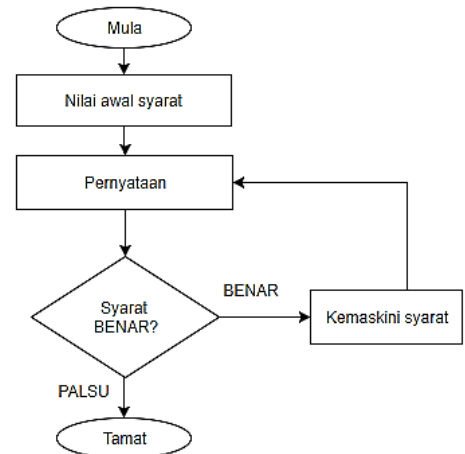
CONTOH ATUR CARA STRUKTUR KAWALAN ULANGAN

ATURCARA	OUTPUT
<pre> <html> <body> <script> var pembilang; for (pembilang = 1; pembilang <= 3; pembilang++) { document.write("Salam Sejahtera!
"); } </script> </body> </html> </pre>	
<pre> <html> <body> <script> var pembilang; for (pembilang = 1; pembilang <= 3; pembilang++) { document.write("Salam Sejahtera!
"); if (pembilang == 2) break; } </script> </body> </html> </pre>	Kata kunci <i>break</i> akan menyebabkan penamatan @ perberhentian suatu ulangan. Berlaku di mana-mana sahaja dalam blok kod tersebut
<pre> <html> <body> <script> var pembilang; for (pembilang = 1; pembilang <= 3; pembilang++) { document.write("Salam Sejahtera!
"); if (pembilang == 2) continue; } </script> </body> </html> </pre>	Kata kunci <i>continue</i> akan terus melaksanakan arahan dan menghabiskan ulangan blok kod

ULANGAN BERASASKAN SYARAT



- menggunakan pernyataan *while*...dan *do..while*

*while...**do...while*

CONTOH ATUR CARA STRUKTUR KAWALAN ULANGAN

ATURCARA	OUTPUT
<pre> <html> <body> <script> var ulang = 1; do { document.write("Salam Sejahtera!
"); ulang++; }while (ulang<= 3); </script> </body> </html> </pre>	
<pre> <html> <body> <script> var ulang = 4; do { document.write("Salam Sejahtera!
"); ulang++; }while (ulang<= 3); </script> </body> </html> </pre>	
<pre> <html> <body> <script> var ulang = 1; while (ulang <=3) { document.write("Salam Sejahtera!
"); ulang++; } </script> </body> </html> </pre>	

STANDARD LIBRARY

- Merupakan koleksi atau kaedah atau fungsi yang disediakan dan digunakan sewaktu implementasi dalam kod atur cara.
- *Standard Library* perlu ditakrifkan dalam spesifikasi bahasa pengaturcaraan.
- *Standard Library* juga merupakan sebahagian daripada arahan dalam atur cara termasuk definisi bagi algoritma yang biasa digunakan, struktur data dan mekanisme bagi input dan output.

Tak perlu tahu cara perlaksanaannya.

Mengurangkan masa pembangunan atur cara

Boleh digunakan berulang kali

KELEBIHAN
STANDARD LIBRARY

CONTOH ATURCARA

```
<html>
  <head>
    <script src=https://cdnjs.cloudflare.com/ajax/libs/mathjs/3.3.0/math.min.js>
    </script>
  </head>
  <body>
    <script>
      document.write("Punca kuasa dua bagi 4 ialah " , math.sqrt(4));
      document.write("<br> 3 Kuasa 2 ialah " , math.pow(3,2));
    </script>
  </body>
</html>
```

OUTPUT

Punca kuasa dua bagi 4 ialah 2
3 Kuasa 2 ialah 9

math.js**mathjs**

- **math.js** ialah *standard library* yang kerap digunakan dalam *Javascript*.
- Terdiri daripada fungsi *build-in* dan pemalar dengan pelbagai jenis data seperti nombor.
- *Standard Library* dalam *Javascript* berada pakej “.js” dan perlu disimpan dahulu.
- Seterusnya **math.js** akan dimuatkan dan seterusnya boleh digunakan dalam atur cara.

FUNGSI	PENERANGAN
math.add (x,y)	Menambah dua nombor
math.divide (x,y)	Membahagi dua nombor
math.subtract(x,y)	Menolak dua nombor
math.pow (x,y)	Mengira kuasa kepada nombor
math.sqrt (x)	Mengira punca kuasa bagi nombor
math.cube(x)	Mendarab tiga nombor yang sama (x * x *)
math.sort (x)	Mengisih unsur-unsur dalam satu matriks

Fungsi-fungsi dalam math.js**date.js**

FUNGSI	PENERANGAN
Date.today ()	Menghasilkan Tarikh hari ini
Date.parse (' today ')	Menukar bentuk objek kepada bentuk objek <i>Date</i>
Date.today ().add ().days ()	Menambah hari kepada hari ini

CONTOH ATURCARA

```

<html>
  <head>
    <script src=https://cdnjs.cloudflare.com/ajax/libs/datejs/1.0/date.min.js>
    </script>
  </head>
  <body>
    <script>
      var hariIni = Date.today();
      document.write(hariIni);
    </script>
  </body>
</html>

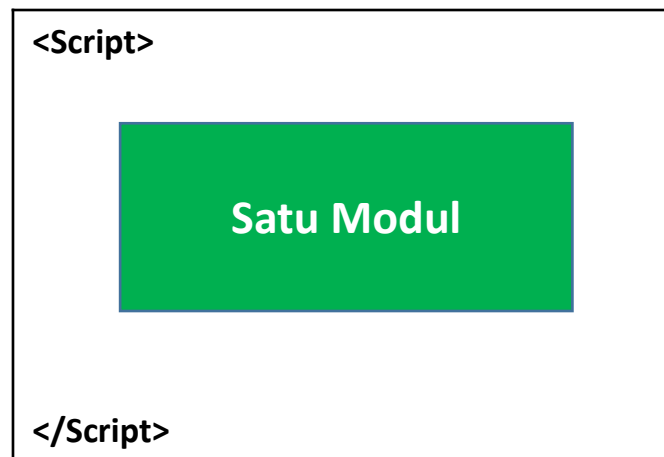
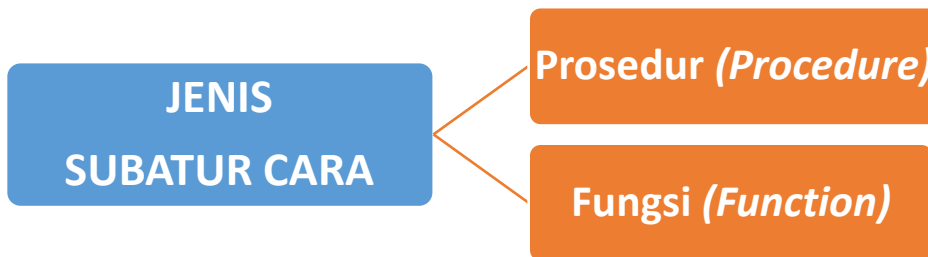
```

OUTPUT

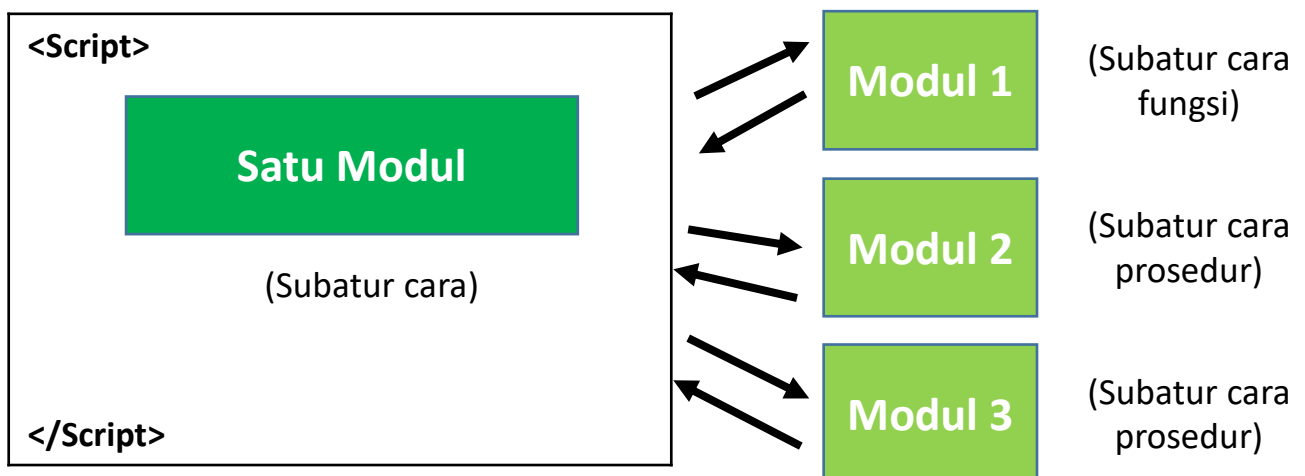
Sun Jan 07 2018 00:00:00 GMT+0800 (Malay Peninsula Standard Time)

SUBATUR CARA

- Satu atur cara *Javascript* mempunyai banyak barisan arahan yang dihipunkan dalam satu blok kod yang dikenali sebagai **modul**.
- Barisan arahan tersebut boleh dipecahkan mengikut tugas masing-masing dan dihipunkan dalam modul yang berasingan yang dipanggil **subatur cara**.



Modul



Modul utama yang dipecahkan

PERBEZAAN ANTARA PROSEDUR DAN FUNGSI

ASPEK PERBANDINGAN	PROSEDUR	FUNGSI
Memulangkan Nilai	Tidak memulangkan nilai	Memulangkan Nilai
Contoh 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>
Contoh Panggilan	toCelcius (fahrenheit);	Celcius = toCelcius (Fahrenheit)

CARA MEMANGGIL/MENGUNAKAN PROSEDUR DAN FUNGSI

- Subatur cara akan menggunakan subatur cara lain bertujuan supaya kod pernyataan-pernyataan dalam subatur cara lain akan dilaksanakan.

Subatur cara
(Pemanggil)



Subatur cara
(Dipanggil)

CONTOH ATURCARA

```
<html>
  <body>
    <script>
      var SenaraiPertama = [2,4,12,77];
      var SenaraiKedua = [333,222,111];
      var SenaraiKetiga = [5.1,5.2,5.6,5.0];
      var jum1,jum2,jum3;

      jum1 = JumlahNo (SenaraiPertama);
      jum2 = JumlahNo (SenaraiKedua);
      jum3 = JumlahNo (SenaraiKetiga);

      document.write(jum1 + "<br>");
      document.write(jum2 + "<br>");
      document.write(jum3 + "<br>");

      function JumlahNo(nombor)
      {
        var i, bilNo,jumlah = 0;
        bilNo = nombor.length;

        for (i=0;i < bilNo; i++)
        {
          jumlah = jumlah + nombor [i];
        }
        return jumlah;
      }
    </script>
  </body>
</html>
```

OUTPUT

95
666
20.9

PARAMETER

- **Parameter** atau **argument** membolehkan prosedur dan fungsi menghantar nilai (pemanggil) dan diterima oleh prosedur dan fungsi yang dipanggil.
- **Parameter rasmi** (*formal parameter*) digunakan untuk merujuk parameter bagi subatur cara yang dipanggil.
- **Parameter sebenar** (*actual parameter*) digunakan untuk merujuk pemboleh ubah didalam subatur cara pemanggil.

CONTOH ATURCARA

```
<html>
  <body>
    <script>
      var fahrenheit = 77;

      document.write ("Suhu adalah ", + toCelcius (fahrenheit) + " darjah celcius");

      function toCelcius (fahrenheit)
      {
        return (5/9) * (fahrenheit -32);
      }
    </script>
  </body>
</html>
```

OUTPUT

Suhu adalah 25 darjah celcius

MEMULANGKAN NILAI

- Fungsi (*function*) akan mengembalikan kawalan kepada pemanggil.
- Sesetengah Fungsi (*function*) juga mengembalikan nilai.

CONTOH ATURCARA

```
<html>
  <body>
    <script>
      var fahrenheit = 77;

      document.write ("Suhu adalah ", + toCelcius (fahrenheit) + " darjah celcius");

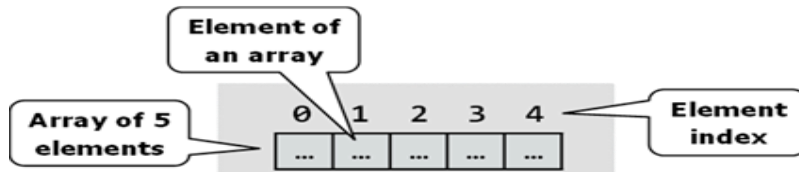
      function toCelcius (fahrenheit)
      {
        return (5/9) * (fahrenheit -32);
      }
    </script>
  </body>
</html>
```

OUTPUT

Suhu adalah 25 darjah celcius

TATASUSUNAN

- Penggunaan tatasusunan adalah untuk menyimpan satu senarai nilai dalam satu pemboleh ubah.



SINTAKS	CONTOH PENGISYTIHARAN
<pre>var nama_tatasusunan = [unsur1, unsur2,...]</pre>	<pre>var no = [5, -1, 4, 12,8] @ var no []; no [0] = 5; no [1] = -1; no [2] = 4; no [3] = 12; no [4] = 8;</pre>

CONTOH ATURCARA

```
<html>
  <body>
    <script>
      var no = [5,-1,4,12,8];
      var jumlah = 0;

      jumlah = no[0]+no[1]+no[2]+no[3]+no[4];

      document.write (jumlah);

    </script>
  </body>
</html>
```

OUTPUT

28

```
<html>
  <body>
    <script>
      var no = [5,-1,4,12,8];
      var jumlah = 0;
      var i;

      for (i =0;i<5;i++)
      {
        jumlah = jumlah + no[i];
      }

      document.write (jumlah);

    </script>
  </body>
</html>
```

OUTPUT

28

CONTOH ATURCARA

```

<html>
  <body>
    <script>
      var no = [5,-1,4,12,8];
      var i;

      for (i =0;i<5;i++)
      {
        if (no[i] == 12)
          document.write (no [i] + " dijumpai");
      }
    </script>
  </body>
</html>

```

OUTPUT

12 dijumpai

```

<html>
  <body>
    <script>
      var no = [5,-1,4,12,8];
      var i,j,sementara;

      for (i =0;i<5;i++)
      {
        document.write (" " + no[i]);
      }

      for (i = 0; i < 5-1; i++)
      {
        for (j = 0; j < 5 - i -1; j++)
        {
          if (no[j] > no[j+1])
          {
            sementara = no [j];
            no [j] = no [j +1];
            no [j+1] = sementara;
          }
        }
      }

      document.write ("<br><br> Senarai nombor (telah diisih secara menaik) : <br><br>");

      for (i=0;i<5;i++)
      {
        document.write (" " + no[i]);
      }
    </script>
  </body>
</html>

```

OUTPUT

5 -1 4 12 8

Senarai nombor (telah diisih secara menaik) :

-1 4 5 8 12

CONTOH ATURCARA

```

<html>
  <body>
    <p> Senarai negara dan suhu (belum diisi) : </p>
    <script>
      var negara = ["Algeria","Ghana","Nigeria","Afrika Selatan","Tunisia"];
      var suhu = [51.3,43.3,46.4,50.0,55.0];
      var i,j,sementara,sementara2;

      for (i =0;i<5;i++)
      {
        document.write (" " + negara[i] + " " + suhu[i]);
        document.write ("<br>");
      }

      for (i = 0; i < 5-1; i++)
      {
        for (j = 0; j < 5-i-1; j++)
        {
          if (suhu[j] > suhu[j+1])
          {
            sementara = suhu[j];
            suhu[j] = suhu [j +1];
            suhu [j+1] = sementara;

            sementara2 = negara[j];
            negara[j] = negara [j +1];
            negara [j+1] = sementara2;

          }
        }
      }

      document.write ("<br><br> Senarai negara beserta suhu (telah diisi) : <br><br>");

      for (i=0;i<5;i++)
      {
        document.write (" " + negara[i] + " " + suhu[i]);
        document.write ("<br>");
      }
    </script>
  </body>
</html>

```

OUTPUT

Senarai negara dan suhu (belum diisi) :

Algeria 51.3
Ghana 43.3
Nigeria 46.4
Afrika Selatan 50
Tunisia 55

Senarai negara beserta suhu (telah diisi) :

Ghana 43.3
Nigeria 46.4
Afrika Selatan 50
Algeria 51.3
Tunisia 55

TAG HTML <button>

- Tag HTML <button> boleh digunakan untuk laman web interaktif, iaitu butang akan terhasil dalam dokumen HTML (laman web)

CONTOH ATURCARA

```
<html>
  <body>
    <button onclick = "window.alert ('Hai')">
      Klik sini
    </button>
  </body>
</html>
```

Klik sini

Hai

OK

```
<html>
  <body>
    <form action = "\action_page.php">
      nama :
        <input type = "text" name = "nama pengguna" value = " " >
        <input type = "submit" value = "Hantar">
    </form>
  </body>
</html>
```

OUTPUT

nama : Ain

Hantar

Example

Link

Button

Input

Submit

1. `<a class="btn" href="">Link</a>`
2. `<button class="btn" type="submit">Button</button>`
3. `<input class="btn" type="button" value="Input">`
4. `<input class="btn" type="submit" value="Submit">`

```
<!DOCTYPE html>
<html>
<!-- created 2010-01-01 -->
<head>
  <title>sample</title>
</head>
<body>
  <p>Voluptatem accusantium
  totam rem aperiam.</p>
</body>
</html>
```

TAG HTML <button> (*memanggil prosedur/fungsi*)

- Setelah tombol diklik, suatu *procedure/function* akan dipanggil untuk melaksanakan sesuatu tugas/arahan.

CONTOH ATURCARA

```
<html>
  <body>
    <p> Senarai nama sebelum diisih : </p>
    <button onclick = "SebelumIsih()" > Sebelum Isih </button>

    <script>
      function sebelumIsih()
      {
        var no = [5,1,4,3,2];
        document.write(no);
      }
    </script>

    <p> Senarai nama selepas diisih (Isihan Buih): </p>
    <button onclick = "SelepasIsih()" > Selepas Isih </button>

    <script>
      function selepasIsih()
      {
        var no = [ 5,1,4,3,2];
        var panjang = no.length, i, j;
        var sementara;

        for (j =0;j < panjang ; j++)
        {
          sementara = no[j];
          no[j] = no [j + 1];
          no [j+1] = sementara;
        }
      }
      document.write (no);
    </script>
  </body>
</html>
```

OUTPUT

Senarai nama sebelum diisih :

Sebelum Isih

Senarai nama selepas diisih (Isihan Buih):

Selepas Isih

5 1 4 3 2

1 2 3 4 5

3.1

3.1.9

MENGHASILKAN ATUR CARA INTERAKTIF YANG MENGANDUNGI PENGISYTIHARAN DATA, STRUKTUR KAWALAN, *PROCEDURE* DAN *FUNCTION* SERTA TATASUSUNAN

CONTOH ATURCARA

```
<html>
  <body>
    <p> Senarai nama sebelum diisih : </p>
    <button onclick = "SebelumIsih()" > Sebelum Isih </button>

    <script>
      function sebelumIsih()
      {
        var no = [5,1,4,3,2];
        document.write(no);
      }
    </script>

    <p> Senarai nama selepas diisih (Isihan Buih): </p>
    <button onclick = "SelepasIsih()" > Selepas Isih </button>

    <script>
      function selepasIsih()
      {
        var no = [ 5,1,4,3,2];
        var panjang = no.length;
        var sementara;
        isihanBuih (no,panjang,sementara);
      }
    </script>

    <script>
      function isihanBuih (no,panjang,sementara)
      {
        var i,j;

        for (i =0;i < panjang ; i++)
        {
          for (j =0;j < panjang ; j++)
          {
            if (no[j] > no [j + 1]);
            {
              tukarKedudukan (no,sementara, j, j+1);
            }
          }
        }
        document.write (no);
      }
    </script>

    function tukarKedudukan (no,sementara,i,j)
    {
      sementara = no[i];
      no[i] = no [j];
      no [j] = sementara;
    }
  </script>
</body>
</html>
```

OUTPUT

Senarai nama sebelum diisih :

Sebelum Isih

Senarai nama selepas diisih (Isihan Buih):

Selepas Isih

5 1 4 3 2

1 2 3 4 5

3.1

3.1.9

MENGHASILKAN ATUR CARA INTERAKTIF YANG MENGANDUNGI PENGISYTIHARAN DATA, STRUKTUR KAWALAN, *PROCEDURE* DAN *FUNCTION* SERTA TATASUSUNAN

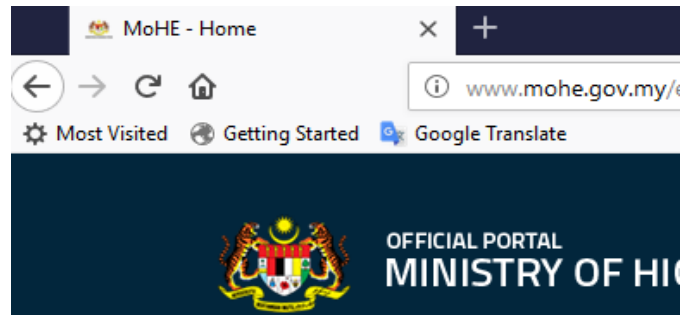
TAG HTML <a>... (*hyperlink*)

CONTOH ATURCARA

```
<html>
  <body>
    <a href = "http://www.mohe.gov.my/en/"> Laman Web KPT Malaysia </a>
  </body>
</html>
```

OUTPUT

[Laman Web KPT Malaysia](http://www.mohe.gov.my/en/)



CONTOH ATURCARA

```
<html>
  <body>
    <a href = "javascript: lamanWeb ()"> Laman Web KPT Malaysia </a>
    <script>
      function lamanWeb()
      {
        window.open ("http://www.mohe.gov.my/en/", "jav", "width = 400, height = 300, resizable = yes");
      }
    </script>
  </body>
</html>
```

OUTPUT

[Laman Web KPT Malaysia](http://www.mohe.gov.my/en/)



3.1

3.1.9

MENGHASILKAN ATUR CARA INTERAKTIF YANG MENGANDUNGI PENGISYTIHARAN DATA, STRUKTUR KAWALAN, *PROCEDURE* DAN *FUNCTION* SERTA TATASUSUNAN

TAG HTML <a>... (*hyperlink*)

CONTOH ATURCARA

```
<html>
  <body>
    <a href = "javascript: lamanWeb ()"> <img src = "logo_kpt_2012.png" border = "0" </a>
    <script>
      function lamanWeb()
      {
        window.open ("http://www.mohe.gov.my/en/", "jav", "width = 400, height = 300, resizable = yes");
      }
    </script>
  </body>
</html>
```



BAHASA PENSKRIPAN PELAYAN

3.2.1

- Menjelaskan fungsi laman web sebagai penghubung antara pengguna dan sistem dalam rangkaian

3.2.2

- Menghasil dan menyimpan **data dalam fail teks (.txt)**

3.2.3

- Menggunakan Bahasa Penskripan Pelayan untuk membina laman web yang unik bagi setiap pengguna

3.2.4

- Menggunakan data yang diimport **dari fail teks** untuk menyelesaikan masalah.

3.2.5

- Menggunakan data yang diimport **dari fail pangkalan data** untuk menyelesaikan masalah

3.2.6

- Melakukan pengesahsahihan (**validation**) pada data input dari pengguna

3.2.7

- Membina (**create**), mengemaskini (**update**), membuat sandaran (**backup**) dan memulihkan (**restore**) pangkalan data

3.2.8

- Melaksanakan beberapa bentuk pertanyaan (**query**) pangkalan data.

3.2.9

- Membina daftar (**sign up**) dan log masuk (**login**) pada laman web

3.2.10

- Menghasilkan atur cara yang dapat **mengesahkan pengguna** dan **mengemaskini data** di dalam pangkalan data

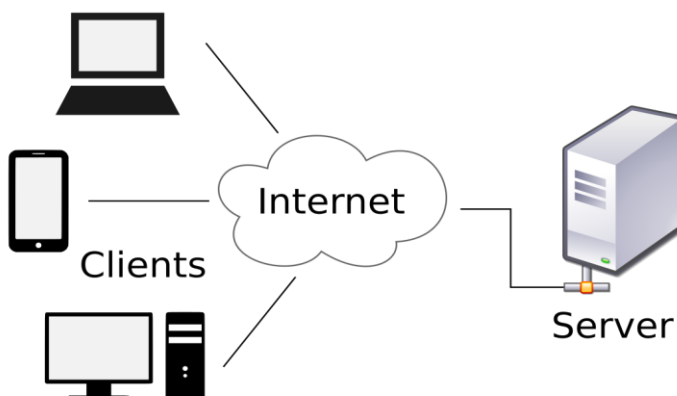
KEPERLUAN LAMAN WEB

- Kewujudan laman web adalah untuk menjadi perantara dalam menentukan bagaimana pengguna boleh berinteraksi dengan sesuatu sistem dalam internet.
- Paparan dalam laman web akan menentukan cara seseorang pengguna itu dapat memanipulasi maklumat dalam sesebuah sistem dan menentukan sejauh manakah capaian pengguna terhadap system tersebut.



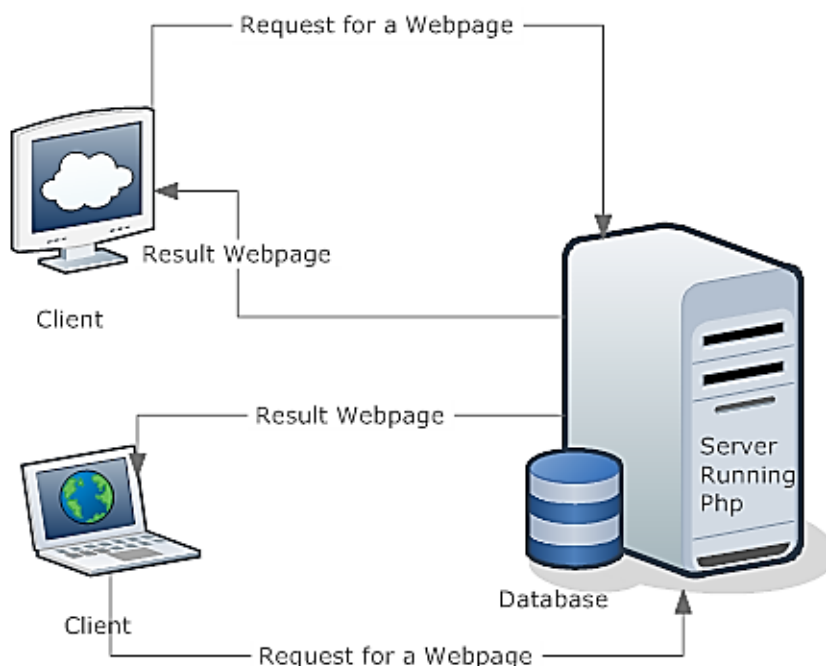
KONSEP PELAYAN

- Berasaskan pengagihan fungsi antara klien dan pelayan.
- **Klien** : apa-apa mesin atau proses yang memohon servis yang spesifik daripada pelayan.
- **Pelayan** : Mesin atau proses yang memberikan servis kepada klien.
- Proses klien dan pelayan ini boleh berlaku dalam satu system komputer yang sama atau dalam satu system komputer yang berbeza yang dihubungkan melalui rangkaian.
- Dalam persekitaran internet dan rangkaian, pelayar web (*web browser*) akan bertindak sebagai klien manakala pelayan web (*web server*) pula adalah pelayan.



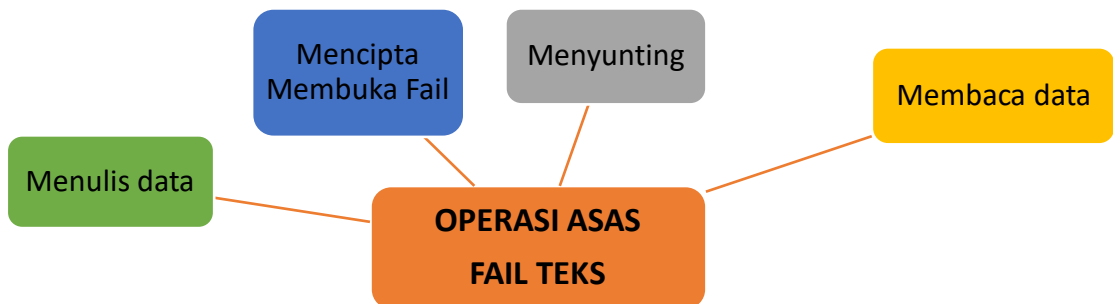
BAHASA PENSKRIPAN PELAYAN

- Merupakan program yang dilaksanakan disebelah pelayan atau pelayan web (*web server*) yang melatari pembangunan web.
- Salah satu fungsinya ialah mengakses data daripada pangkalan data.
- Sekiranya pengguna memohon suatu laman web dipaparkan dengan menggunakan pelayar (*web browser*, permintaan tersebut akan dihantar kepada pelayan web (*web server*) .
- Kemudian, pelayan web (*web server*) akan mencari laman web yang dikehendaki dan memindahkan laman web tersebut kepada pelayar (*web browser*) untuk diterjemahkan dan dipaparkan kepada pengguna.
- Bahasa Penskripan Pelayan tidak bergantung pada pelayar (*web browser*). Kesemua pemprosesan dilakukan disebelah pelayan, maka isu pemprosesan disekat oleh pelayar tidak akan timbul.
- Namun begitu, kadangkala Bahasa penskripan pelayan tidak begitu selamat memandangkan penskripan pelayan boleh digunakan untuk menggodam laman web.
- Oleh itu, laman web yang bagus perlu menggunakan kedua-dua jenis Bahasa penskripan klien dan pelayan bagi menyokong proses pengesahan disebelah klien dan juga disebelah pelayan untuk tujuan keselamatan laman web tersebut.
- Contoh Aplikasi yang boleh digunakan : Papan pesanan dan buku pelawat blog.
- Antara Bahasa penskripan pelayan : ASP , *Javascript*, PHP, Python, Perl.



PENGHASILAN DAN PENYIMPANAN DATA DALAM FAIL TEKS

- Sesuatu data yang dimasukkan oleh pengguna daripada laman web boleh disimpan ke dalam pelayan dalam bentuk fail teks.
- **Contoh** : Apabila pengguna memasukkan nama serta kata laluan, maklumat tersebut boleh disimpan dalam fail teks dalam komputer pelayan.
- Fungsi "*fopen*" akan digunakan digunakan sebagai arahan yang akan dilaksanakan dalam pelayan.



MENCIPTA / MEMBUKA FAIL TEKS

SINTAKS	CONTOH
\$f = fopen ("nama fail teks",mod capaian);	\$f = fopen ("LogMasuk.txt ", "w");

MOD CAPAIAN	PENERANGAN
r	Fail dibuka hanya untuk dibaca sahaja.
w	Fail dibuka hanya untuk ditulis sahaja (bermula dari awal fail)
a	Fail dibuka hanya 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.

MENCIPTA / MEMBUKA FAIL TEKS

CONTOH ATURCARA

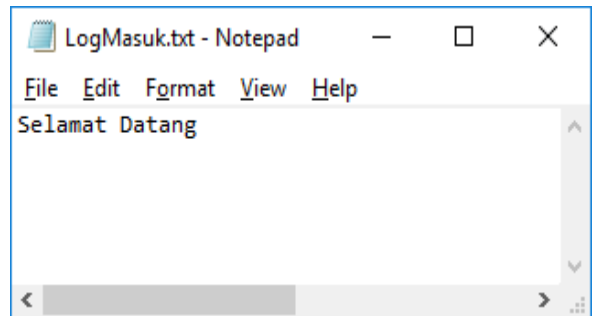
<?php

\$f = fopen ("LogMasuk.txt", "w");

fwrite (\$f, "Selamat Datang");

fclose (\$f);

?>



CONTOH ATURCARA

```
<html>
  <head><title> Write to text file</title></head>

  <body>
    <h1> Selamat Datang </h1>
    <h1> Sekolah Menengah Bistari Jaya  </h2>

    <form method = 'POST'>
      <p> Masukkan Nama Anda
      <input type = 'text' name = 'namapengguna' ></p>
      <input type = 'submit' name = 'Submit' value = 'Masuk'>
    </form>

    <?php
      if (isset($_POST["Submit"]))
      {
        $f = fopen ("LogMasuk.txt", "a");
        $nama = $_POST ['namapengguna'];
        $tarikh = date ('d/m/Y h:i:s', time());
        $log = $nama.":".$tarikh.PHP_EOL;
        fwrite($f,$log);
        fclose($f);
      }

    ?>
  </body>
</html>
```

OUTPUT

Selamat Datang**Sekolah Menengah Bistari Jaya**Masukkan Nama Anda

3.2

3.2.3

MENGGUNAKAN BAHASA PENSKRIPAN PELAYAN UNTUK MEMBINA LAMAN WEB YANG UNIK BAGI SETIAP PENGGUNA

UNIK BAGI SETIAP PENGGUNA

- Sesuatu laman web kelihatan unik bagi setiap pengguna apabila laman web memaparkan maklumat tentang pengguna tersebut sahaja.

CONTOH ATURCARA

~Rujuk Buku Teks~ ms 233 - 238

3.2

3.2.4

MENGGUNAKAN DATA YANG DIIMPORT DARIPADA FAIL TEKS UNTUK MENYELESAIKAN MASALAH

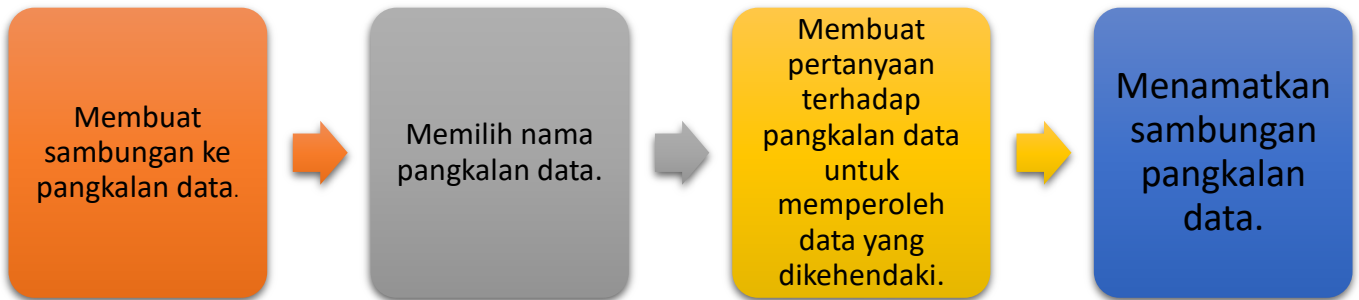
DATA YANG DIIMPORT

- Apabila kita mempunyai banyak data yang perlu diproses/dimanipulasikan, adalah lebih baik sekiranya data-data tersebut disimpan dalam fail (contohnya fail teks).
- Data-data dalam fail kemudiannya dapat diproses tanpa memerlukan satu persatu data yang diinput oleh pengguna.

CONTOH ATURCARA

~Rujuk Buku Teks~ ms 240 - 243

MENCAPAI DATA DARIPADA PANGKALAN DATA



Membuat Sambungan ke Pangkalan Data MySQL

SINTAKS	<code>\$con = mysql_connect ("hoskomputer", " namapengguna", "katalaluan");</code>
CONTOH	<code>\$con = mysql_connect ("localhost", " halim", "1234");</code>

Memilih Nama Pangkalan Data MySQL

SINTAKS	<code>mysqli_select_db ("namasambungan", " namapangkalan data");</code>
CONTOH	<code>mysqli_select_db (\$con, " dbPelajar");</code>

Membuat Pertanyaan terhadap Pangkalan Data MySQL

SINTAKS	<code>\$hasil = mysqli_query ("namasambungan", " pernyataan SQL ");</code>
CONTOH	<code>\$hasil = mysqli_query (\$con, " SELECT * from MURID ");</code>

CONTOH ATURCARA

~Rujuk Buku Teks~ ms 244 - 247

MELAKUKAN PENGESAHSAHIHAN (*VALIDATION*)

- Pengesahsahihan (*validation*) pada input daripada pengguna adalah untuk mengelakkan terjadinya ralat semasa proses penyimpanan data ke dalam pangkalan data.



Antara Cara Pengesahsahihan (*Validation*)

CONTOH ATURCARA

~Rujuk Buku Teks~ ms 249 - 251

3.2

3.2.7

MEMBINA (*CREATE*), MENGEMAS KINI (*UPDATE*), MEMBUAT SANDARAN (*BACKUP*), DAN MEMULIHKAN (*RESTORE*) PANGKALAN DATA

CONTOH

~Rujuk Buku Teks~ ms 251 - 257

3.2

3.2.8

MELAKSANAKAN BEBERAPA BENTUK PERTANYAAN (*QUERY*) PANGKALAN DATA

MELAKSANAKAN PERTANYAAN (*QUERY*)

- Sesuatu maklumat yang telah disimpan dalam pangkalan data boleh diperoleh semula dengan menggunakan mekanisme pertanyaan (*query*).
- Menggunakan format yang telah ditetapkan oleh kebanyakan system pengurusan pangkalan data menggunakan *Structured Query Language (SQL)*.
- Penggunaan pertanyaan (*query*) ini akan mencari serta menyusun data daripada satu atau lebih jadual (*table*).



CONTOH

~Rujuk Buku Teks~ ms 258 - 261

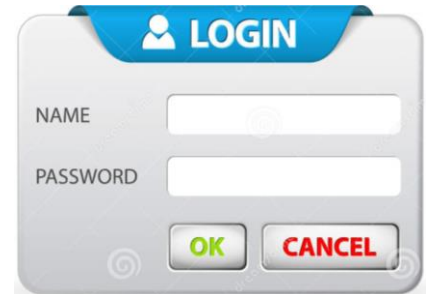
3.2

3.2.9

MEMBINA DAFTAR (*SIGN UP*) DAN LOG MASUK (*LOGIN*) PADA LAMAN WEB

DAFTAR (*SIGN UP*) DAN LOG MASUK (*LOGIN*)

- Kebanyakan laman web menghendaki pengguna mendaftar sebagai pengguna/ahli yang sah sebelum dibenarkan untuk mencapai data daripadanya.
- Setelah pengesahan dilakukan, barulah pengguna dapat melayari laman web tersebut.



CONTOH

~Rujuk Buku Teks~ ms 261 - 266

3.2

3.2.10

MENGHASILKAN ATUR CARA YANG DAPAT MENGESAHKAN PENGGUNA DAN MENGEMAS KINI DATA DALAM PANGKALAN DATA

MENGESAHKAN PENGGUNA (*USER VERIFICATION*)

- Merupakan salah satu ciri keselamatan dalam pangkalan data yang boleh dicapai oleh pengguna.
- Kemudahan capaian ini akan membataskan pengguna untuk melakukan perubahan ke atas pangkalan data.
- Contoh : Pengguna yang berbeza memberikan hak capaian yang berbeza dan aktiviti yang boleh dilakukan ke atas pangkalan data.



CONTOH

~Rujuk Buku Teks~ ms 267 - 268

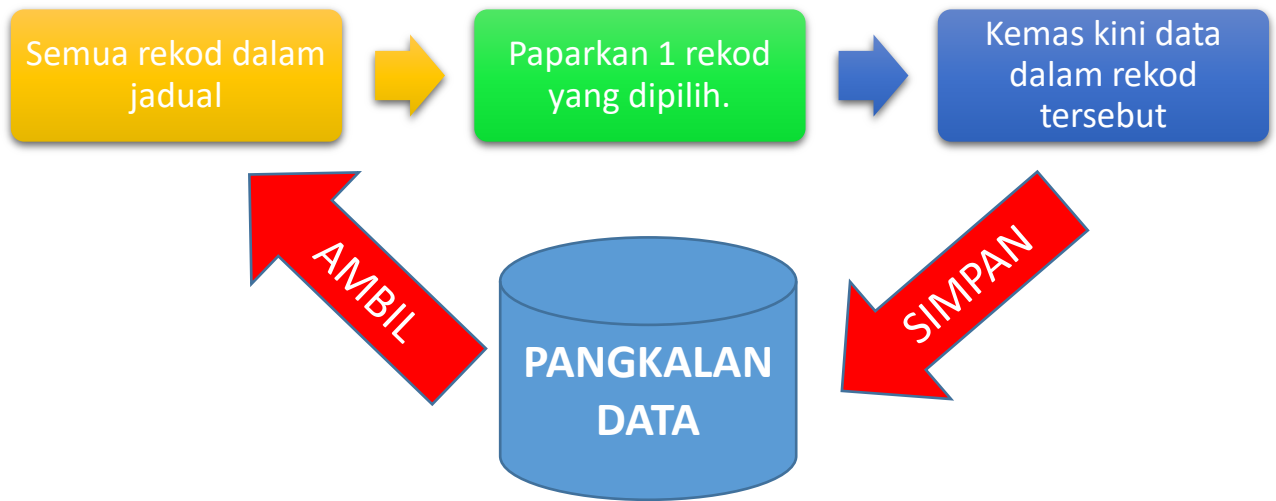
3.2

3.2.10

MENGHASILKAN ATUR CARA YANG DAPAT MENGESAHKAN PENGGUNA DAN MENGEMAS KINI DATA DALAM PANGKALAN DATA

PENGEMASKINIAN DATA

- Untuk mengemas kini data dalam pangkalan data, kita memerlukan pencarian rekod yang hendak dikemaskini terlebih dahulu dalam jadual (*table*) pangkalan data.



Proses pengemaskinian data

CONTOH

~Rujuk Buku Teks~ ms 269 - 272

LAMAN WEB INTERAKTIF

3.3.1

- Mengkaji dan merumus prinsip reka bentuk laman web dari sudut kesesuaian pengguna dan tujuan laman web

3.3.2

- Mereka bentuk kerangka aplikasi yang akan dibangunkan.

3.3.3

- Membina laman web mudah menggunakan Hypertext Markup Language (**HTML**) yang mengandungi: **frame**, **header**, **paragraph** dan **image**

3.3.4

- Menggunakan Cascading Style Sheets (**CSS**) untuk menggayakan text, **font**, **background**, **tables**, **borders** dan **position** .

3.3.5

- Menggunakan Bahasa Penskripan Klien untuk membina laman web pengguna yang interaktif mengandungi fitur berikut:
 - (i) **Pengesahan data**
 - (ii) **Butang pilihan**
 - (iii) **Kotak Popup**

3.3.6

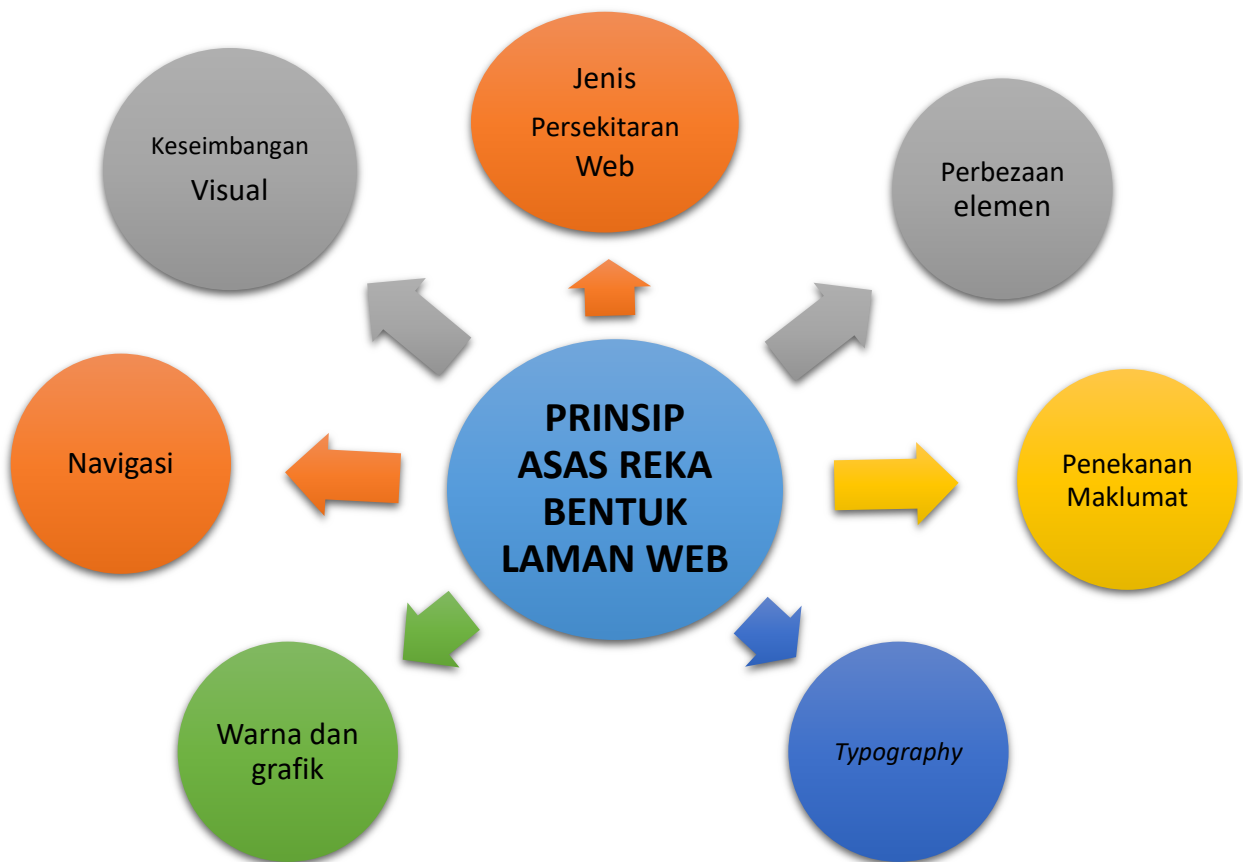
- Menggunakan Bahasa Penskripan Pelayan untuk membina lamanweb yang boleh **mencapai** dan **mengemaskini data** di dalam pangkalan data

3.3.7

- Menghasilkan sebuah laman web interaktif untuk kegunaan **pengguna** dan **pentadbir sistem** dalam menyelesaikan masalah

PRINSIP REKA BENTUK LAMAN WEB

- Mereka bentuk laman web boleh diibaratkan seperti proses merancang, membina dan mengubahsuai.
- Beberapa prinsip atau elemen reka bentuk laman web perlu diberi perhatian agar laman web yang dibina tidak kelihatan pelik atau memberikan pengalaman yang tidak menyenangkan kepada pengguna.
- Ini kerana elemen-elemen tersebut mampu memberikan pengalaman yang berbeza dan mempengaruhi kepuasan pelanggan untuk melayari laman web yang dibina.



JENIS PERSEKITARAN WEB YANG DIBINA

- Faktor-faktor teknikal mengenai persekitaran laman web yang perlu diambil kira.

**KESESUAIAN
PENGGUNAAN
PELBAGAI JENIS
PELAYAR**

- Pereka web mempertimbangkan jenis dan versi pelayar yang boleh digunakan untuk paparan.
- Tidak tertumpu pada 1 versi/ jenis sahaja.

**KESESUAIAN
CAPAIAN INTERNET**

- Kelajuan capaian perlu diambil kira.
- Contoh : Laman web permainan atas talian memerlukan capaian yang lebih laju berbanding laman web berita sahaja.

**KEPERLUAN PERISIAN
SOKONGAN UNTUK
PAPARAN**

- Kadangkala sesetengah laman web memerlukan perisian sokongan untuk menyokong paparan.
- Contoh : *Flash* - untuk memaparkan animasi/video

**KEPELBAGAIAN SISTEM
KOMPUTER DAN
PERANTI MUDAH ALIH
YANG DIGUNAKAN
OLEH PENGGUNA**

- Pereka web harus memastikan laman web yang dipaparkan perlu fleksibel, selari dengan kepelbagaian komputer dan peranti pengguna.

**KEPELBAGAIAN SAIZ
PAPARAN MONITOR
YANG DIGUNAKAN**

- Pereka web harus memastikan laman web yang dipaparkan perlu fleksibel, selari dengan kepelbagaian saiz skrin pengguna.

PENEKANAN MAKLUMAT DALAM WEB

- Merujuk kepada maklumat atau kandungan yang ingin diberikan perhatian dalam laman web tersebut.

TEKNIK PENEKANAN

Penggunaan kesan khas seperti *drop shadow*, tekstur atau cahaya dan bentuk

Penggunaan ruang putih untuk menjadikan elemen-elemen kelihatan lebih terserlah.

Penggunaan fungsi *bold*, *italic*, warna yang berbeza dan *border*.

PERBEZAAN ELEMEN DALAM WEB

- Konsep perbezaan digunakan untuk menghasilkan tarikan visual terhadap suatu elemen yang ingin ditekankan dalam web.
- Semakin besar perbezaan suatu elemen yang ingin ditonjolkan berbanding elemen lain di persekitarannya, maka elemen tersebut akan kelihatan lebih berbeza.
- Penggunaan konsep perbezaan ini mampu menghasilkan sistem navigasi lebih mudah dan dapat memandu pelawat untuk mencapai maklumat yang dikehendaki.

KAEDAH PERBEZAAN

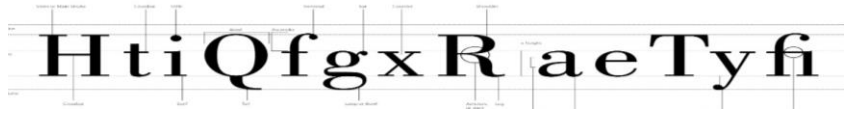
Penggunaan label dan pautan yang berbeza.

Penggunaan *border*, warna yang berbeza & kesan khas.

Penggunaan ruang putih, teks terbalik, teks huruf condong (*italic*) dan saiz teks yang berbeza.

Penggunaan prinsip perbezaan warna

TYPOGRAPHY



- Merujuk kepada penyusunan teks yang akan dipaparkan dalam laman web.
- Penyusunan teks yang baik memudahkan pemahaman dalam proses pembacaan dan juga meningkatkan elemen mesra pengguna dalam laman web.
- Terdapat beberapa corak penyusunan teks bagi laman web. Corak-corak ini boleh digunakan mengikut gaya pembacaan pembaca.
- Contoh corak penyusunan teks : Corak paparan berasaskan kertas, corak paparan secara Z-Shaped , Corak paparan secara Z-Shaped.
- Pemilihan jenis *font* juga penting.
- Pemilihan *font* yang kurang biasa digunakan boleh menyebabkan sesetengah sistem pengoperasian tidak menyokong jenis *font* tersebut dan mempunyai masalah untuk memaparkan kandungan web.

PANDUAN TYPOGRAPHY

Seragamkan jenis teks atau fon yang digunakan bagi setiap halaman yang dibina.

Gunakan saiz teks yang berbeza bagi membezakan tajuk dengan kandungan atau huraian.

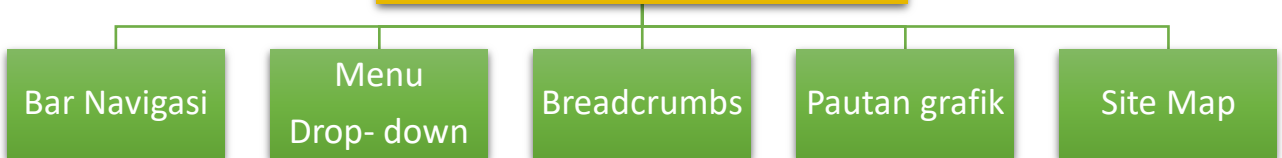
WARNA & GRAFIK

- Penggunaan warna dan grafik mampu menonjolkan imej dan identiti sesebuah laman web.
- Penggunaan warna yang bijak mampu menarik perhatian serta meningkatkan kebolehbacaan mereka.

NAVIGASI

- Merujuk kepada struktur laman web yang akan memudahkan seseorang pengguna melayarinya.
- Navigasi web yang bagus seharusnya memberikan petunjuk kepada pelawat laman web tentang kedudukan atau lokasi semasa pengguna.

TEKNIK NAVIGASI WEB



KESEIMBANGAN VISUAL

- Merujuk kepada gabungan elemen visual seperti garisan, imej, teks, bentuk dan warna dalam laman web.
- Pereka laman web perlu peka tentang pengguna sasaran laman web tersebut.

REKA BENTUK BERASASKAN PENGGUNA

- Pereka laman web harus mengenali pengguna sasaran dan perkara yang diingini oleh pengguna apabila mereka melayari laman web.

PENSTRUKTURAN KANDUNGAN

- Kunci kejayaan penghasilan suatu laman web bergantung kepada cara laman web itu berfungsi.
- Perhatian yang lebih perlu diberikan untuk menentukan cara maklumat dalam laman web dapat dicapai dan pendekatan yang sesuai bagi pengguna untuk mendapatkan maklumat.



Proses mereka bentuk kerangka aplikasi

PENSTRUKTURAN KANDUNGAN ATAU MAKLUMAT

- Setiap reka bentuk yang bakal dihasilkan perlu berasaskan pengguna atau pengalaman pengguna semasa melayari laman web.
- Penglibatan pengguna semasa proses pengumpulan maklumat sangat penting.
- Contoh aktiviti : mengadakan sesi temu bual, mengedarkan borang soal selidik atau membuat tinjauan cara laman web digunakan.

MEMBENTUK INVENTORI MAKLUMAT

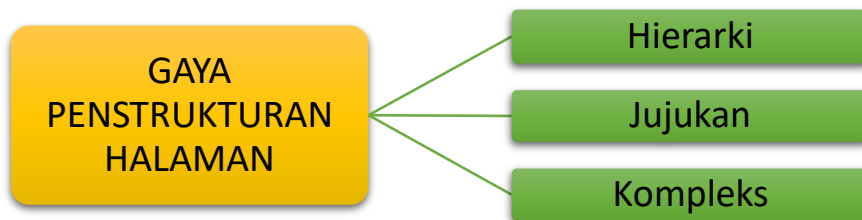
- Senaraikan semua maklumat yang bakal dimasukkan ke dalam laman web.
- Senaraikan semua perkara yang boleh dilakukan oleh pengguna.

PENGELASAN MAKLUMAT

- Mengelaskan senarai maklumat kepada beberapa kategori.
- Pereka bentuk maklumat boleh melihat persamaan maklumat yang ada bagi mewujudkan kemungkinan-kemungkinan kategori yang perlu ada.
- Maklumat atau informasi ialah suatu yang subjektif dan corak pengelasan maklumat boleh berbeza-beza bergantung pada perspektif pereka web.
- Pereka perlu menetapkan bilangan kategori maklumat yang perlu ada bersesuaian dengan senarai maklumat yang tersedia dan bagaimana setiap kategori tersebut akan divisualkan dalam laman web yang bakal dibina.
- Gunakan gambarajah atau carta alir bagi menunjukkan hubungkait bagi setiap kategori maklumat yang ada.
- Ini mampu memberikan idea atau gambaran kasar tentang corak pautan laman web yang bakal dibina.

PENSTRUKTURAN HALAMAN

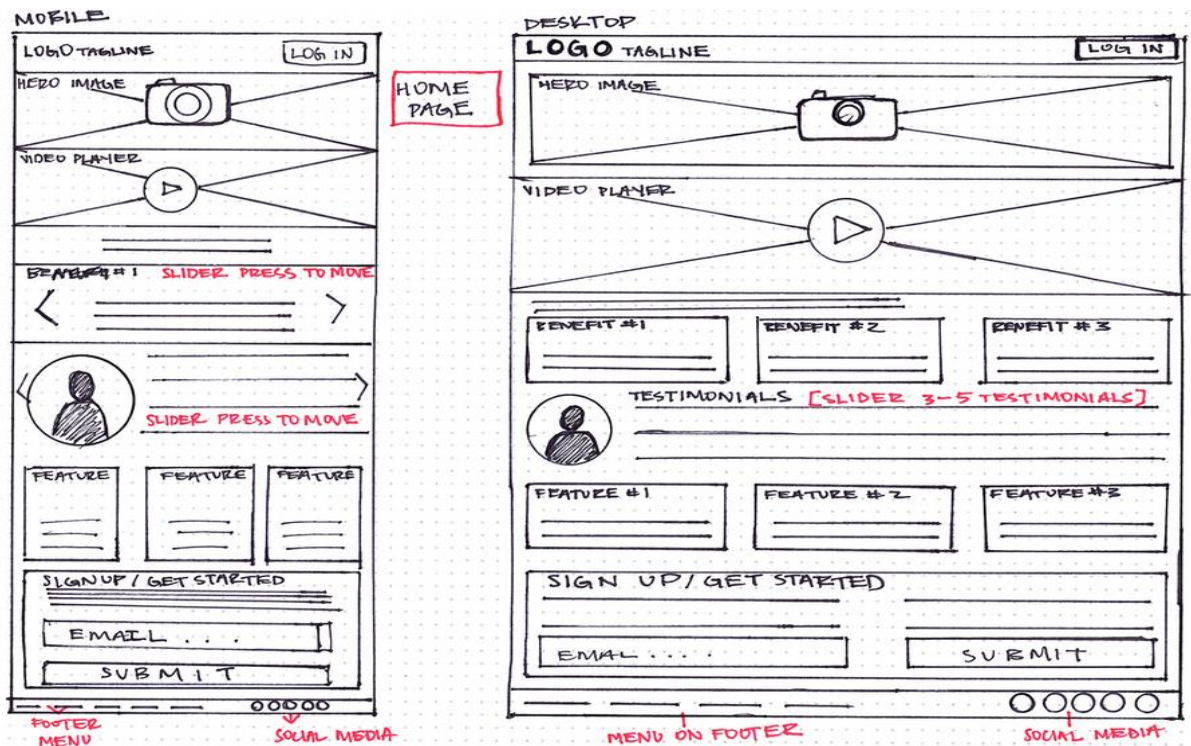
- Menekankan cara setiap halaman dalam laman web dihubungkan.
- Kebiasannya pereka bentuk akan menggunakan gambar rajah tapak (site diagram) sebagai alat untuk menyampaikan struktur laman web kepada klien dan sebagai panduan sepanjang proses pembangunan laman web.
- Gambar rajah tapak : diagram yang menggunakan simbol kotak bagi mewakili setiap laman web dengan garis atau anak panah bagi mewakili hubungan antara laman web.



GAYA	KETERANGAN
PENSTRUKTURAN SECARA HIERARKI	
PENSTRUKTURAN SECARA JUJUKAN • Pengguna dipandu dari halaman ke halaman. • Sesuai untuk laman web penceritaan / kronologi	
PENSTRUKTURAN KOMPLEKS	• Terdapat laman web komersial yang menawarkan maklumat yang padat dan fungsi agak kompleks.

REKA BENTUK ANTARA MUKA

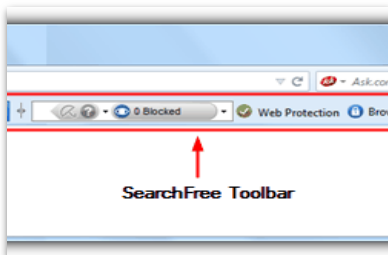
- Setelah halaman dalam laman web distrukturkan, fasa seterusnya adalah untuk menentukan bagaimana pengguna laman web dapat mencapai maklumat tersebut.
- Reka bentuk antara muka : menentukan bagaimana struktur laman web secara logikalnya digambarkan termasuk teknik navigasi yang memandu kepada pergerakan pengguna dalam laman web.
- Pereka bentuk antara muka biasanya akan menggunakan *diagram* atau carta alir bagi menggambarkan reka bentuknya.
- Salah satu rajah yang digunakan dalam proses reka bentuk antara muka ialah *wireframe* atau rangka wayar.
- Rangka wayar atau juga dikenali skema halaman atau pelan halaman ialah panduan visual yang mewakili rangka kerja keseluruhan laman web.
- Tujuan rangka wayar adalah untuk menyusun elemen-elemen yang akan dimasukkan dalam laman web bagi menepati konsep laman web tersebut.



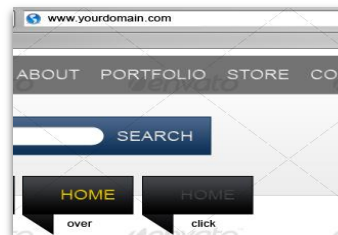
REKA BENTUK NAVIGASI

- Navigasi web merujuk kepada proses objek-objek dalam laman web yang terdiri daripada teks, rajah, audio, video dihubungkan antara satu sama lain dengan menggunakan pautan hiperteks atau hypermedia.
- Konsep navigasi web memberikan pengalaman kepada pelawat laman web untuk mengemudi laman web tersebut mengikut kecekapan masing-masing.
- Navigasi web yang bagus seharusnya memberikan petunjuk kepada pelawat laman web tentang kedudukan atau lokasi semasa pengguna.
- Fungsi utama konsep navigasi dalam laman web adalah untuk membantu pengguna atau pelawat mengetahui kedudukan semasa mereka dan memandu pelawat tentang pilihan yang disediakan untuk mereka menerokai laman web tersebut.

TEKNIK NAVIGASI



TOOLBAR



**BAR
NAVIGASI**



BREADCRUMBS



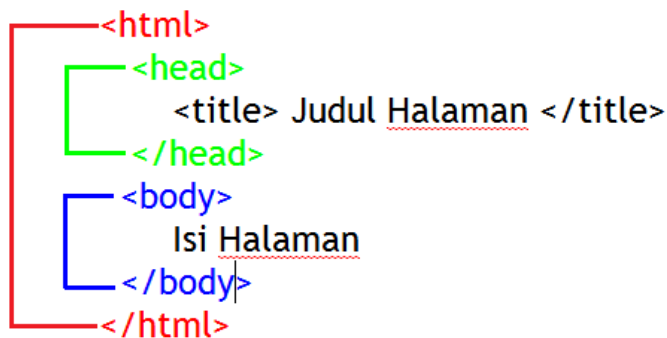
**PETA TAPAK
(SITEMAP)**



**MENU
DROP-DOWN**

Hypertext Markup Language (HTML)

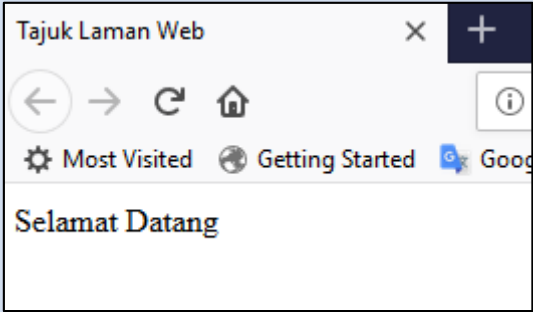
- Setiap laman web ialah fail teks yang ditulis menggunakan *Hypertext Markup Language* (HTML).
- *Markup Language* : Bahasa yang menghuraikan kandungan dan struktur dokumen.



Struktur HTML

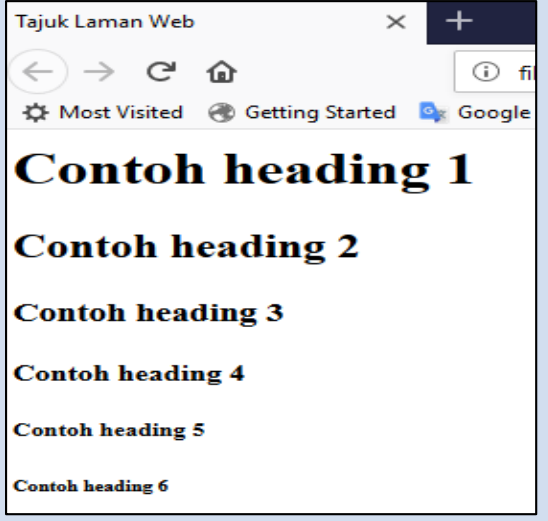
ELEMEN DAN ATRIBUT DALAM HTML

- Bagi menulis dokumen HTML, HTML editor diperlukan.
- Salah satu HTML editor ialah *text editor*. Contohnya *Notepad*.
- Dokumen HTML adalah mengenai konsep bagaimana dokumen dipaparkan.
- Untuk menerangkan struktur bagi setiap laman web, kod HTML atau elemen HTML digunakan.
- Elemen biasanya terdiri daripada dua tag yang berpasangan, *Start tag* dan *close tag*.

ATURCARA	OUTPUT
<pre><html> <head> <title> Tajuk Laman Web</title> </head> <body> Selamat Datang </body> </html></pre>	

ELEMEN *HEADING* DALAM HTML

- Elemen *heading* ialah elemen yang digunakan bagi menakrifkan “tajuk” dalam laman web.
- Terdapat 6 peringkat *heading* yang ditakrifkan dalam HTML.

ATURCARA	OUTPUT
<pre><html> <head> <title> Tajuk Laman Web</title> </head> <body> <h1> Contoh heading 1 </h1> <h2> Contoh heading 2 </h2> <h3> Contoh heading 3 </h3> <h4> Contoh heading 4 </h4> <h5> Contoh heading 5 </h5> <h6> Contoh heading 6 </h6> </body> </html></pre>	

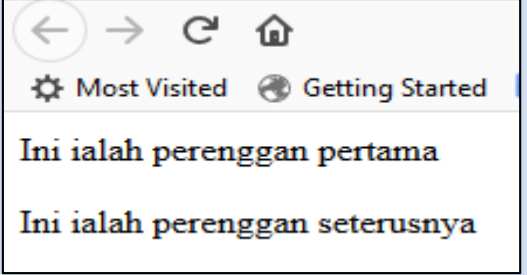
ELEMEN *HEADER* DALAM HTML

- Digunakan untuk memaparkan pengenalan kandungan bagi suatu laman web atau pautan navigasi.

ATURCARA	OUTPUT
<pre><html> <head></head> <body> <header> <h1> Nilai-nilai Murni </h1> <hr> <h4> Definisi Nilai-nilai Murni </h4> </header> <p> Nilai-nilai murni sewajarnya diterapkan dalam diri setiap murid sejak awal. </p> </body> </html></pre>	

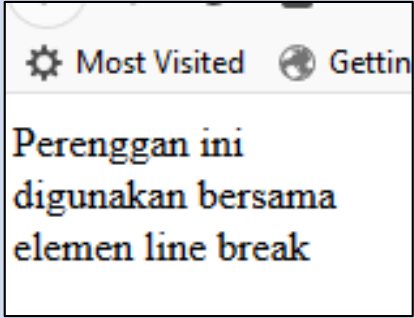
ELEMEN *PARAGRAPH* DALAM HTML

- Digunakan bagi menyusun teks dalam bentuk perenggan.

ATURCARA	OUTPUT
<pre><html> <head></head> <body> <p> Ini ialah perenggan pertama </p> <p> Ini ialah perenggan seterusnya </p> </body> </html></pre>	 A screenshot of a web browser interface. At the top, there are navigation icons: back, forward, refresh, and home. Below these are two tabs: 'Most Visited' and 'Getting Started'. The main content area displays two paragraphs of text: 'Ini ialah perenggan pertama' followed by 'Ini ialah perenggan seterusnya'.

ELEMEN *LINE BREAK* DALAM HTML

- Untuk memaparkan teks di baris baharu, elemen `<br>` digunakan.

ATURCARA	OUTPUT
<pre><html> <head></head> <body> <p> Perenggan ini
 digunakan bersama
 elemen line break </p> </body> </html></pre>	 A screenshot of a web browser interface. At the top, there are navigation icons: back, forward, refresh, and home. Below these are two tabs: 'Most Visited' and 'Getting Started'. The main content area displays a single paragraph of text that is wrapped across three lines: 'Perenggan ini', 'digunakan bersama', and 'elemen line break'.

ELEMEN *IMAGE* DALAM HTML

- Bagi memaparkan grafik dalam laman web, elemen `<img>` digunakan.
- Elemen `<img>` ialah *empty element*, tidak memerlukan *close tag*.
- Atribut `src` digunakan Bersama `<img>` tag untuk menentukan URL atau alamat imej tersebut diambil.
- Atribut `width` dan `height` juga boleh digunakan untuk menetapkan saiz paparan grafik.

ATURCARA	OUTPUT
<pre><html> <body> <img src = " http://hddfhm.com/images/clipart-html-5.jpg" style = width:128px;height :128px"> </body> </html>></pre>	

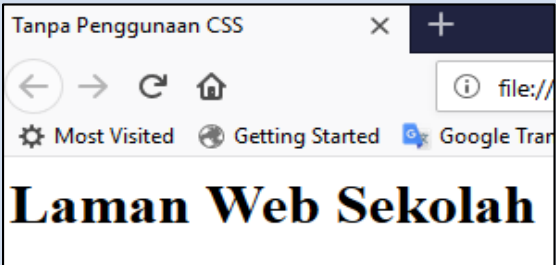
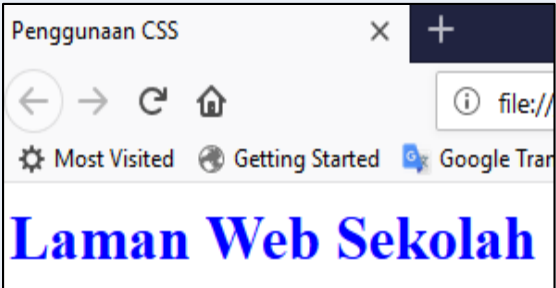
ELEMEN *FRAME* DALAM HTML

- Elemen *iframe*, Untuk memaparkan dokumen HTML lain dalam dokumen HTML semasa yang dipaparkan.
- Atribut `src`, `height` dan `width` digunakan bagi menentukan saiz paparan pada *iframe*.

ATURCARA
<pre><html> <head><title>Contoh Atur cara </title></head> <body> <h2>Tentang Saya</h2> <p> Nama saya Suriana binti Shuib. Saya berumur 17 tahun. Saya bersekolah di Sekolah Menengah Seri Intan. Saya mempunyai keluarga yang bahagia </P> <p> Saya gemar melayari Internetdi masa lapang. Namun, tidak semua yang berita yang dipaparkan itu betul. Untuk mengetahui kesahihannya, kita boleh merujuk kepada laman web <h4> sebenarnya.my</h4> </p>
 <center> <iframe height = "40%" src = "http://sebenarnya.my"></iframe> </center> </body> </html></pre>

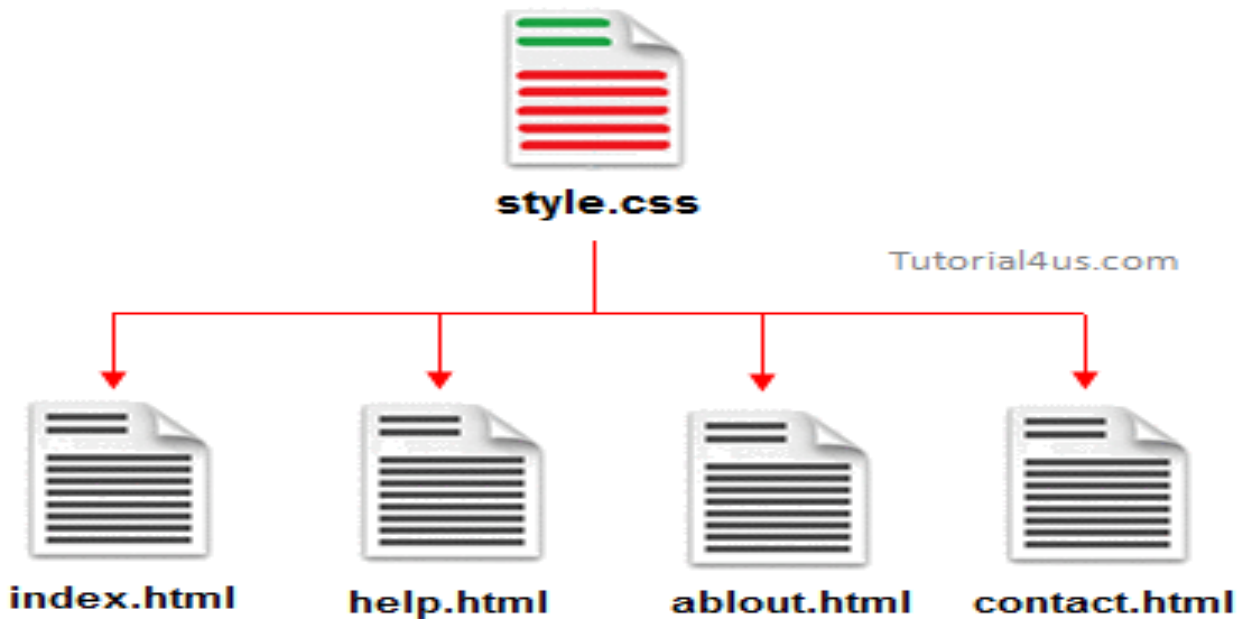
CASCADING STYLE SHEET (CSS)

- CSS : salah satu Bahasa pengaturcaraan tambahan (*extension*) bagi HTML.
- CSS membenarkan perubahan kepada gaya sesuatu laman web dan menerangkan persembahan bagi sesuatu dokumen HTML.

ATURCARA	OUTPUT
<pre><html> <head> <title>Tanpa Penggunaan CSS </title> </head> <body> <h1>Laman Web Sekolah</h1> </body> </html></pre>	
<pre><html> <head> <title>Penggunaan CSS </title> <style> h1 { color:blue;} </style> </head> <body> <h1>Laman Web Sekolah</h1> </body> </html></pre>	

CARA PENULISAN CSS

HELAIAN GAYA LUAR (<i>External Style Sheet</i>)	HELAIAN GAYA DALAMAN (<i>Internal Style Sheet</i>)	GAYA DALAM BARISAN (<i>Inline Style</i>)
• Tag gaya akan ditulis dan disimpan di dalam satu fail dan fail itu mesti mempunyai <i>extension</i> "CSS". • Fail tersebut akan dipanggil daripada tag HTML dengan menggunakan sintaks.	• Tag gaya boleh ditulis sama ada di dalam kawasan <head> atau <body>. • Letakkan <style>	• Tag gaya akan ditulis pada bahagian/elemen tag.

HELAIAN GAYA LUAR (*EXTERNAL STYLE SHEET*)

ATURCARA

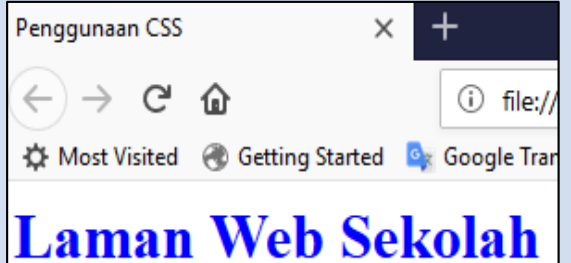
Sediakan satu fail bernama " *style1.css* "

```
h1 { color: blue;}
```

Sediakan satu fail bernama " *utama.html* "

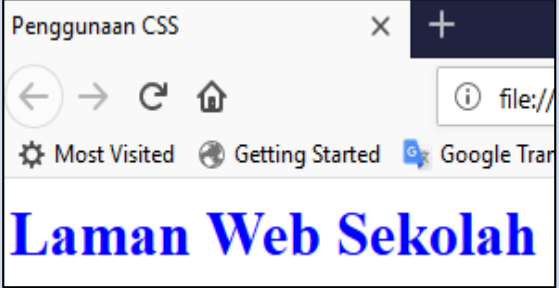
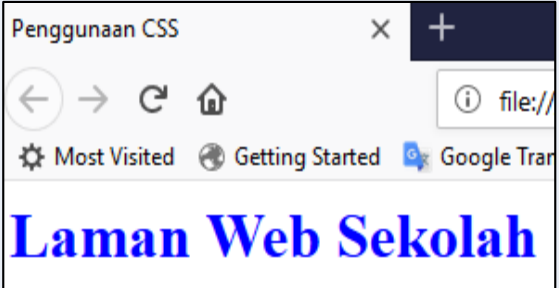
```
<html>
  <head>
    <title>Penggunaan CSS </title>
    <link rel = "stylesheet" type = text/css" href= "style1.css">
  </head>
  <body>
    <h1>Laman Web Sekolah</h1>
  </body>
</html>
```

OUTPUT



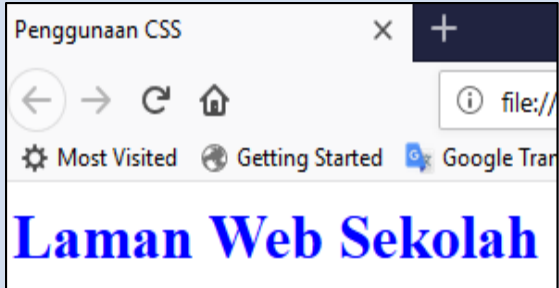
HELAIAN GAYA DALAMAN (INTERNAL STYLE SHEET)

- Boleh ditulis sama ada di dalam kawasan `<head>` atau `<body>`.

ATURCARA	OUTPUT
<pre><html> <head> <title>Penggunaan CSS </title> <style> h1 { color:blue;} </style> </head> <body> <h1>Laman Web Sekolah</h1> </body> </html></pre>	
<pre><html> <head> <title>Penggunaan CSS </title> </head> <body> <style> h1 { color:blue;} </style> <h1>Laman Web Sekolah</h1> </body> </html></pre>	

GAYA DALAM BARISAN (INLINE STYLE)

- Tag gaya akan ditulis pada bahagian/elemen tag.

ATURCARA	OUTPUT
<pre><html> <head> <title>Penggunaan CSS </title> </head> <body> <h1 style = "color:blue";> Laman Web Sekolah</h1> </body> </html></pre>	

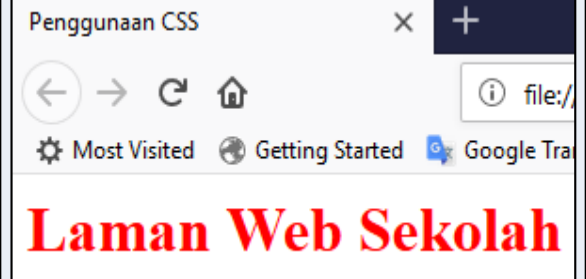
GAYA PADA TEXT

WARNA TEXT (COLOUR)

ATURAN

```
<html>
<head>
  <title>Penggunaan CSS </title>
</head>
<body>
  <h1>Laman Web Sekolah</h1>
</body>
</html>
```

OUTPUT



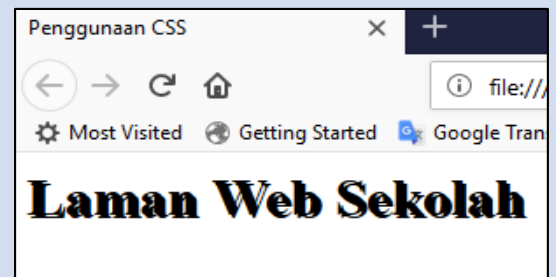
JAJARAN TEKS (TEXT ALIGNMENT)

```
<html>
<head>
  <title>Penggunaan CSS </title>
</head>
<body>
  <h1>Laman Web Sekolah</h1>
</body>
</html>
```



BAYANG TEKS (TEXT SHADOW)

```
<html>
<head>
  <title>Penggunaan CSS </title>
</head>
<body>
  <h1>Laman Web Sekolah</h1>
</body>
</html>
```



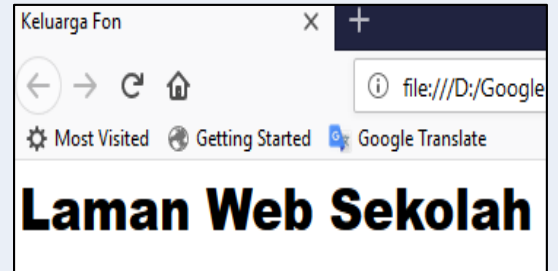
GAYA PADA FON (FONT)

KELUARGA FONT (FONT FAMILY)

ATURCARA

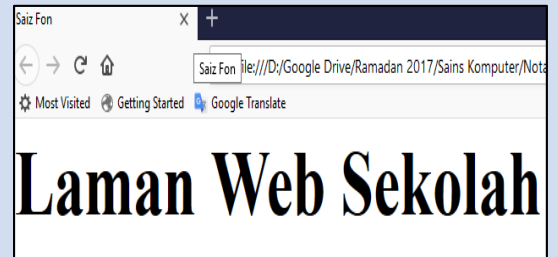
```
<html>
  <head>
    <title>Keluarga Fon </title>
  </head>
  <body>
    <h1 style="font-family:Arial Black;">Laman Web
    Sekolah</h1>
    <h>
  </body>
</html>
```

OUTPUT



SAIZ FON (FONT SIZE)

```
<html>
  <head>
    <title>Saiz Fon </title>
  </head>
  <body>
    <h1 style="font-size:70;" >Laman Web Sekolah</h1>
  </body>
</html>
```



STAIL FON (FONT STYLE)

```
<html>
  <head>
    <title>Stail Fon </title>
  </head>
  <body>
    <h1 style="font-style:italic;" >Laman Web Sekolah</h1>
  </body>
</html>
```



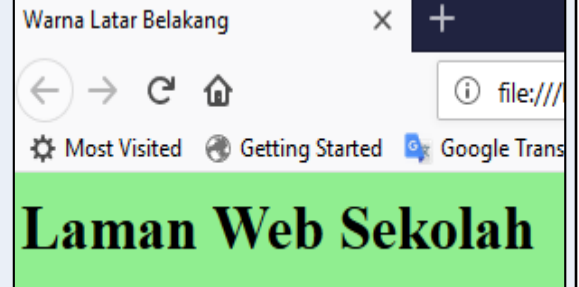
GAYA PADA *BACKGROUND*

WARNA LATAR BELAKANG

ATURAN

```
<html>
<head>
  <title>Warna Latar Belakang </title>
</head>
<Style>
  body
  { background-color : lightgreen;
  }
</style>
<body>
  <h1>Laman Web Sekolah</h1>
</body>
</html>
```

OUTPUT



IMEJ ATAU GAMBAR SEBAGAI LATAR BELAKANG

```
<html>
<head>
  <title>Latar Belakang Imej </title>
</head>
<Style>
  body
  {
    background-image : url("http://moziru.com/images/collage-
    clipart-school-13.jpg");
  }
</style>
<body>
  <h1>Laman Web Sekolah</h1>
</body>
</html>
```



* *sintak aturcara berbeza dari buku teks*

GAYA PADA TABLES

MELETAKKAN GARISAN LUAR ATAU *BORDER*

ATURCARA

```
<html>
<head>
  <title>Gaya Jadual </title>
</head>
<style>
  table
  { border : 1px solid; }
</style>
<body>
  <h1>Laman Web Sekolah</h1>
  <table>
    <th> KANDUNGAN LAMAN WEB </th>
    </tr>
    <tr>
      <td> Pengenalan Sekolah </td>
    </tr>
    <tr>
      <td> Senarai Guru </td>
    </tr>
  </table>
</body>
</html>
```

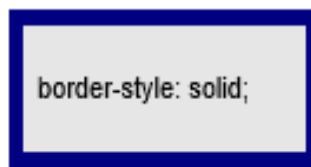
OUTPUT



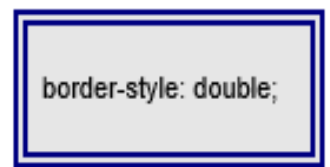
border-style: dotted;



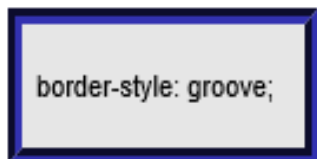
border-style: dashed;



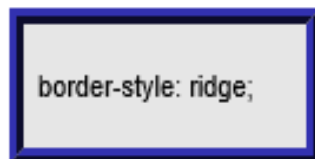
border-style: solid;



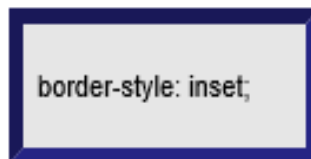
border-style: double;



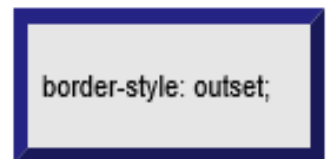
border-style: groove;



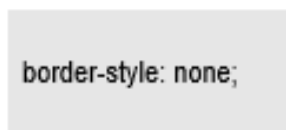
border-style: ridge;



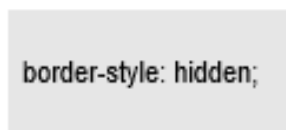
border-style: inset;



border-style: outset;



border-style: none;



border-style: hidden;

GAYA PADA TABLES

MELETAKKAN GARISAN LUAR ATAU *BORDER*

ATURCARA

```
html>
<head>
  <title>Gaya Jadual </title>
</head>
<Style>
  table, th,td
  { border : 1px solid;}
</style>

<body>
<h1>Laman Web Sekolah</h1>
<table>
  <th> KANDUNGAN LAMAN WEB </th>
  </tr>
  <tr>
    <td> Pengenalan Sekolah </td>
  </tr>
  <tr>
    <td> Senarai Guru </td>
  </tr>
</table>
</body>
</html>
```

OUTPUT



GAYA PADA TABLES

MELETAKKAN GARISAN LUAR ATAU *BORDER*

ATURCARA

```
html>
<head>
  <title>Gaya Jadual </title>
</head>
<Style>
  table {border-collapse: collapse;}
  table, th,td
    { border : 1px solid;}
</style>

<body>
<h1>Laman Web Sekolah</h1>
<table>
  <th> KANDUNGAN LAMAN WEB </th>
</tr>
<tr>
  <td> Pengenalan Sekolah </td>
</tr>
<tr>
  <td> Senarai Guru </td>
</tr>
</table>
</body>
</html>
```

OUTPUT



GAYA PADA TABLES

MELETAKKAN GARISAN LUAR ATAU BORDER

ATURCARA

```
html>
<head>
  <title>Gaya Jadual </title>
</head>
<style>
  table {border-collapse: collapse;}
  table, th,td
    { border : 3px solid;}
</style>

<body>
<h1>Laman Web Sekolah</h1>
<table>
  <th> KANDUNGAN LAMAN WEB </th>
  </tr>
  <tr>
    <td> Pengenalan Sekolah </td>
  </tr>
  <tr>
    <td> Senarai Guru </td>
  </tr>
</table>
</body>
</html>
```

OUTPUT



GAYA PADA TABLES

MELETAKKAN GARISAN LUAR ATAU *BORDER*

ATURCARA

```
html>
<head>
  <title>Gaya Jadual </title>
</head>
<Style>
  table {border-collapse: collapse;}
  table, th, { border : 3px solid;}
  td { border : 2px dashed;}
</style>

<body>
<h1>Laman Web Sekolah</h1>
<table>
  <th> KANDUNGAN LAMAN WEB </th>
  </tr>
  <tr>
    <td> Pengenalan Sekolah </td>
  </tr>
  <tr>
    <td> Senarai Guru </td>
  </tr>
</table>
</body>
</html>
```

OUTPUT



GAYA PADA TABLES

MELETAKKAN GARISAN LUAR ATAU *BORDER*
(Menetapkan kelebaran dan ketinggian sesuatu table)

ATURCARA

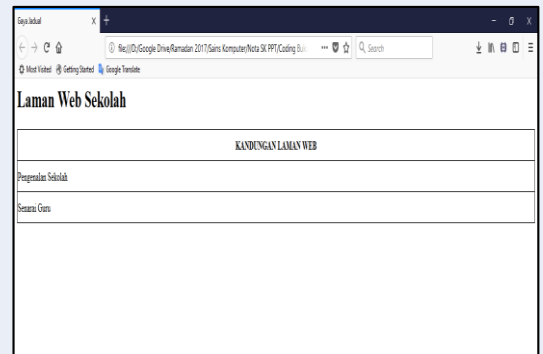
```

html>
<head>
  <title>Gaya Jadual </title>
</head>
<style>
  table {border-collapse: collapse;
  width : 100%; height : 20%}
  table, th, td { border : 1px solid;}
</style>

<body>
<h1>Laman Web Sekolah</h1>
<table>
  <th> KANDUNGAN LAMAN WEB </th>
  </tr>
  <tr>
    <td> Pengenalan Sekolah </td>
  </tr>
  <tr>
    <td> Senarai Guru </td>
  </tr>
</table>
</body>
</html>

```

OUTPUT



GAYA PADA TABLES

MELETAKKAN GARISAN LUAR ATAU *BORDER*
(Menetapkan jajaran teks di dalam table)

ATURCARA

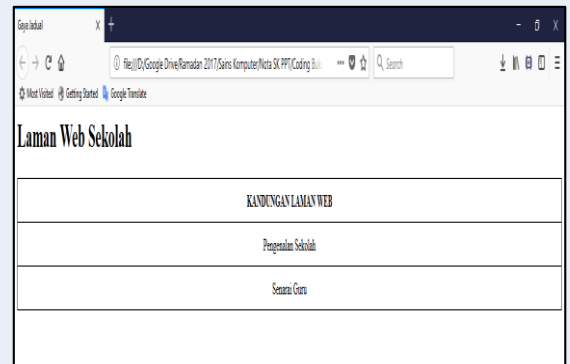
```

html>
<head>
  <title>Gaya Jadual </title>
</head>
<Style>
  table {border-collapse: collapse;
  width : 100%; height : 20%;
  text-align : center;}
  table, th, td{ border : 1px solid;}
</style>

<body>
<h1>Laman Web Sekolah</h1>
<table>
  <th> KANDUNGAN LAMAN WEB </th>
</tr>
<tr>
  <td> Pengenalan Sekolah </td>
</tr>
<tr>
  <td> Senarai Guru </td>
</tr>
</table>
</body>
</html>

```

OUTPUT

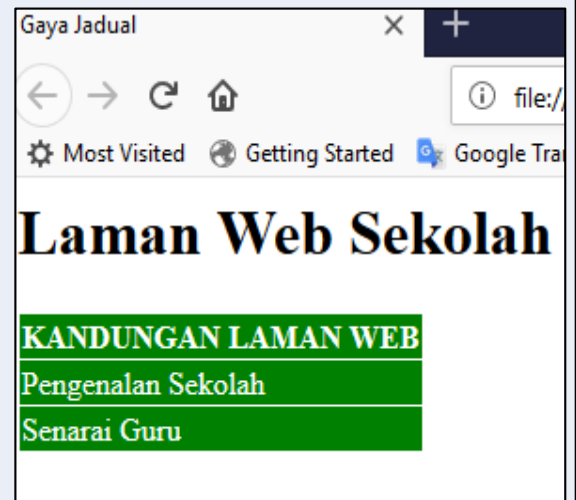


GAYA PADA TABLES

MELETAKKAN GARISAN LUAR ATAU BORDER
*(Menukar warna latar belakang dan warna teks dalam table)***ATURCARA**

```
html>
<head>
  <title>Gaya Jadual </title>
</head>
<Style>
  table
  {
    border-collapse: collapse;
    background-color : green;
    color : white;
  }
  table, th, td{ border : 1px solid;}
</style>

<body>
<h1>Laman Web Sekolah</h1>
<table>
  <th> KANDUNGAN LAMAN WEB </th>
  </tr>
  <tr>
    <td> Pengenalan Sekolah </td>
  </tr>
  <tr>
    <td> Senarai Guru </td>
  </tr>
</table>
</body>
</html>
```

OUTPUT

GAYA PADA TABLES

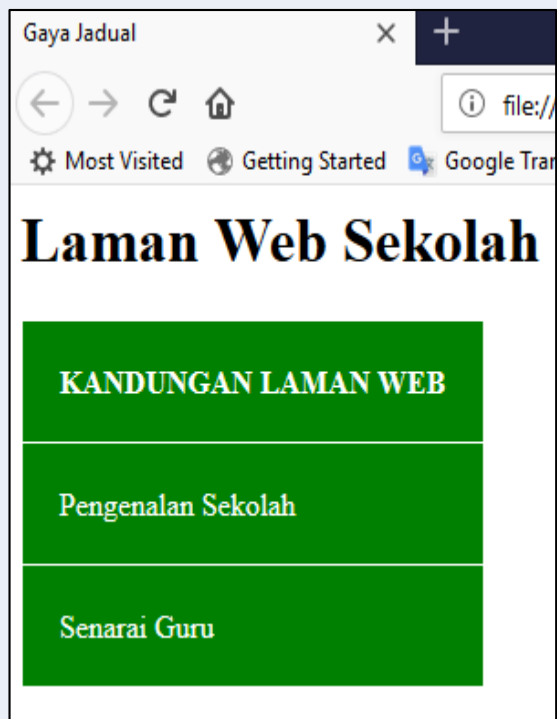
MELETAKKAN GARISAN LUAR ATAU *BORDER*
(*Menebalkan jarak di antara medan (padding) dalam table*)

ATURCARA

```
html>
<head>
  <title>Gaya Jadual </title>
</head>
<Style>
  table
  {
    border-collapse: collapse;
    background-color : green;
    color : white;
  }
  table, th, td{ border : 1px solid;}
</style>

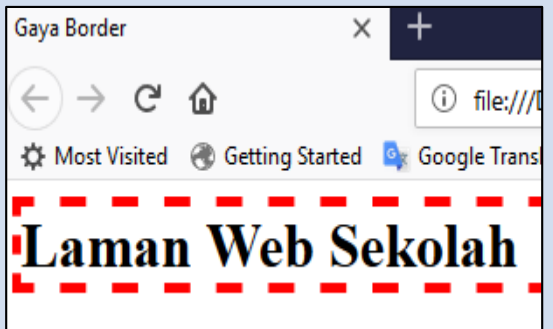
<body>
<h1>Laman Web Sekolah</h1>
<table cellpadding="20">
  <tr>
    <th> KANDUNGAN LAMAN WEB </th>
  </tr>
  <tr>
    <td> Pengenalan Sekolah </td>
  </tr>
  <tr>
    <td> Senarai Guru </td>
  </tr>
</table>
</body>
</html>
```

OUTPUT



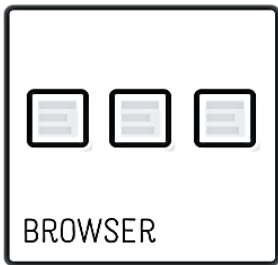
GAYA PADA BORDERS

ATURCARA	OUTPUT
<pre><html> <head> <title>Gaya Border </title> <Style> h1 { border-style: solid; border-width : 5px; } </style> </head> <body> <h1>Laman Web Sekolah</h1> </body> </html></pre>	

ATURCARA	OUTPUT
<pre><html> <head> <title>Gaya Border </title> <Style> h1 { border-style: dashed; border-width : 5px; border-color : red; } </style> </head> <body> <h1>Laman Web Sekolah</h1> </body> </html></pre>	

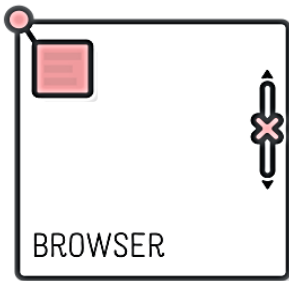
GAYA PADA *POSITION*

STATIC



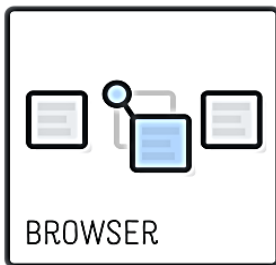
- Kedudukan lalai dalam setiap elemen HTML yang terdapat dalam dokumen.
- Kedudukan dokumen akan mengikut turutan atau susunan.
- Kedudukan elemen tidak akan berubah atau tidak akan mengikut ketetapan.

FIXED



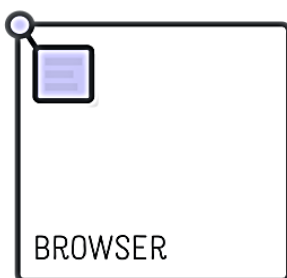
- Kedudukan elemen HTML bergantung pada *browser viewpoint*, iaitu pandangan pengguna kepada pelayar web.
- Kedudukan elemen akan tetap dikedudukan yang sama walaupun pengguna *scroll* pada halaman web.

RELATIVE



- Kedudukan elemen diubah bergantung atau berkait terhadap kedudukan normalnya atau kedudukan awalnya.

ABSOLUTE



- Kedudukan elemen tidak akan mengikut turutan atau susunan pada laman web lagi dan akan ditempatkan mengikut kedudukan yang dikehendaki.

GAYA PADA *POSITION*TANPA GAYA *POSITION*

ATURAN

```
<html>
  <head>
    <title>Tanpa Gaya Position </title>
  </head>

  <body>
    <h1>Laman Web Sekolah (Absolute)</h1>
    <h2>Sekolah 1 (Relative)</h2>
    <h3>Sekolah 2 (Fixed)</h3>
    <h4>Sekolah 3 (Static)</h4>
  </body>
</html>
```

OUTPUT

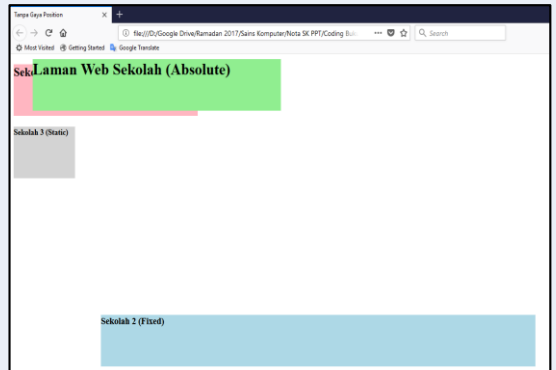
DENGAN GAYA *POSITION*

ATURAN

```
<html>
  <head>
    <title>Gaya Position </title>
    <style>
      h1 { background-color : lightgreen; position :
absolute; width : 40%; top : 10px; left : 50px; height : 100px;
      }
      h2 { background-color : lightpink; position :
relative; width : 30%; top : 70px; left : 300px; height :
100px; }
      h3 { background-color : lightblue; position :
fixed; width : 70%; bottom : 10px; left : 200px; height :
100px; }
      h4 { background-color : lightgrey; position : static;
width : 10%; bottom : 70px; left : 200px; height : 100px; }
    </style>
  </head>

  <body>
    <h1>Laman Web Sekolah (Absolute)</h1>
    <h2>Sekolah 1 (Relative)</h2>
    <h3>Sekolah 2 (Fixed)</h3>
    <h4>Sekolah 3 (Static)</h4>
  </body>
</html>
```

OUTPUT



FITUR PENGESAHAN DATA

PENGESAHAN DATA

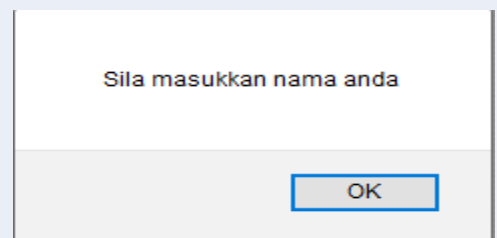
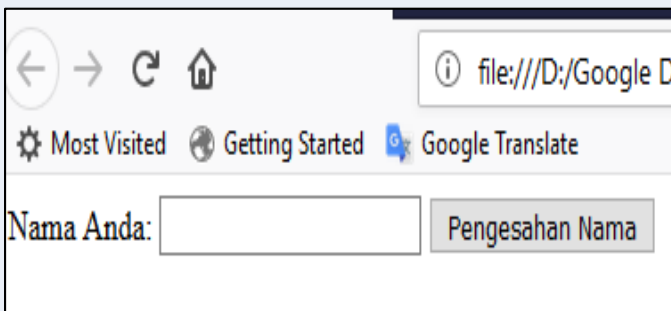
ATURCARA

```
<html>
  <body>
    Nama Anda:
    <input id = "n1">
    <button type = "button" onclick = "semakData()"> Pengesahan Nama</button>

    <script>
    function semakData(){
      var teksnama;
      teksnama = document.getElementById("n1").value;

      if (teksnama = " ")
      {
        window.alert("Sila masukkan nama anda ");
      }
    }
  </script>
</body>
</html>
```

OUTPUT



FITUR PENGESAHAN DATA

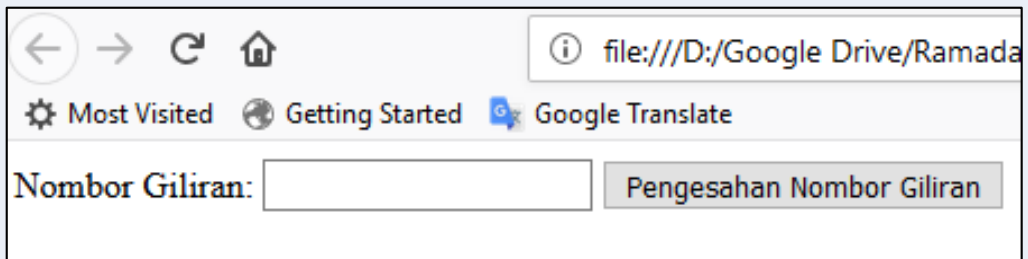
PENGESAHAN DATA

ATURCARA

```
<html>
<body>
  Nombor Giliran:
  <input id = "n2">
  <button type = "button" onclick = "semakNombor()">Pengesahan Nombor Giliran</button>
  <script>
    function semakNombor()
    {
      var teksnombor;
      teksnombor = document.getElementById("n2").value;

      if (isNaN (teksnombor))
      {
        window.alert("Nilai yang dimasukkan adalah bukan nombor. Sila masukkan nombor ");
      }
    }
  </script>
</body>
</html>
```

OUTPUT



Nilai yang dimasukkan adalah bukan nombor. Sila masukkan nombor

OK

FITUR BUTANG PILIHAN

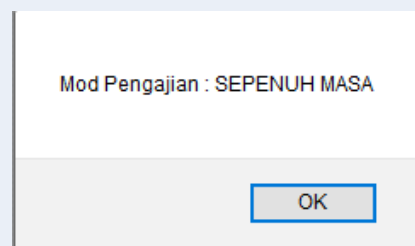
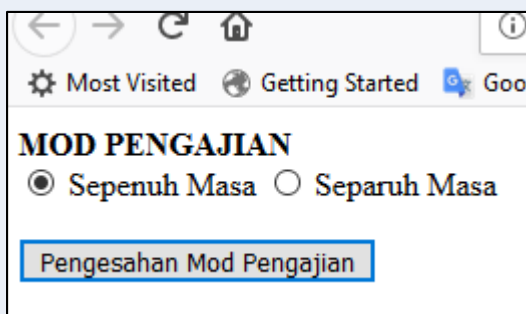
BUTANG RADIO

ATURCARA

```
<html>
  <body>
    <b> MOD PENGAJIAN </b>
    <form action = " " >
      <input type = "radio" name = "mod" id = "sepenuh"> Sepenuh Masa
      <input type = "radio" name = "mod" id = "separuh"> Separuh Masa
    <p>
      <button type = "button" onclick = "semakMod()">Pengesahan Mod Pengajian</button>
    </p>
    </form>

    <script>
      function semakMod()
      {
        if (document.getElementById('sepenuh').checked == true)
        {
          window.alert("Mod Pengajian : SEPENUH MASA ");
        }
        else
        {
          if (document.getElementById('separuh').checked == true)
          {
            window.alert("Mod Pengajian : SEPARUH MASA ");
          }
          document.getElementById('sepenuh').checked == false;
          document.getElementById('separuh').checked == false;
        }
      }
    </script>
  </body>
</html>
```

OUTPUT



FITUR BUTANG PILIHAN

BUTANG CHECKBOX

ATURCARA

```
<html>
  <body>

    <b> SUBJEK-SUBJEK YANG INGIN DIDAFTAR </b>
    <form action = " " name = "MyForm">
      <input type = "checkbox" name = "sub" id = "Matematik"> Matematik
      <input type = "checkbox" name = "sub" id = "English"> English
      <input type = "checkbox" name = "sub" id = "Sains"> Sains
      <input type = "checkbox" name = "sub" id = "Bahasa Malaysia"> Bahasa Malaysia

    <p>
    <button type = "button" onclick = "semakSubjek()">Pengesahan Subjek</button>
    </p>

    <p>
    <textarea id = "SenaraiSubjek" rows = "10" cols = "30" value = "b"></textarea>
    </p>

    <p>
    <button type = "button" onclick = "resetSubjek()">Reset Subjek</button>
    </p>

    </form>
    <script>
      function semakSubjek()
      {
        var elLength = document.MyForm.elements.length;
        for (i=0;i<elLength; i++)
        var type = MyForm.elements[i].type;
        if (type=="checkbox" && MyForm.elements[i].checked)
        {
          document.getElementById("SenaraiSubjek").value += MyForm.elements[i].id+ '\n';
        }
      }

      function resetSubjek()
      {
        var elLength = document.MyForm.elements.length;
        document.getElementById("SenaraiSubjek").value ="";
        for (i=0;i<elLength; i++)
        {
          MyForm.elements [i].checked=false;
        }
      }

    </script>
  </body>
</html>
```

FITUR BUTANG PILIHAN

BUTANG CHECKBOX

OUTPUT

 Most Visited  Getting Started  Google Translate**SUBJEK-SUBJEK YANG INGIN DIDAFAR**☐ Matematik ☒ English ☒ Sains ☐ Bahasa Malaysia

Pengesahan Subjek

 Most Visited  Getting Started  Google Translate**SUBJEK-SUBJEK YANG INGIN DIDAFAR**☐ Matematik ☒ English ☒ Sains ☐ Bahasa Malaysia

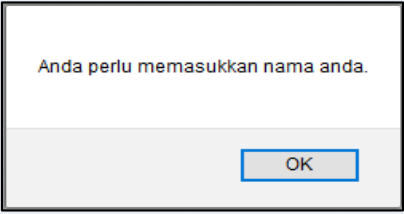
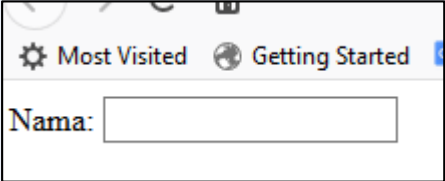
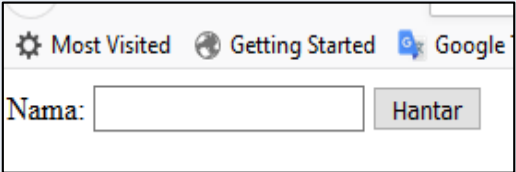
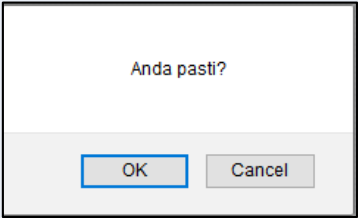
Pengesahan Subjek

English
Sains

Reset Subjek

Reset Subjek

FITUR KOTAK *POPUP*

KOTAK ALERT	
ATURCARA	OUTPUT
<pre> <html> <body> Nama: <input type = "text" > <script> alertMasuk(); function alertMasuk() { window.alert ("Anda perlu memasukkan nama anda."); } </script> </body> </html> </pre>	 
KOTAK CONFIRM	
ATURCARA	OUTPUT
<pre> <html> <body> Nama: <input type = "text" > <button onclick = "pastiHantar()">Hantar</button> <script> function pastiHantar() { if (window.confirm("Anda pasti?")==true) { document.write("Maklumat telah dihantar."); } } </script> </body> </html> </pre>	 

FITUR KOTAK *POPUP*KOTAK *PROMPT*

ATURCARA

OUTPUT

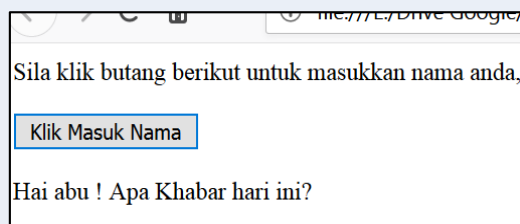
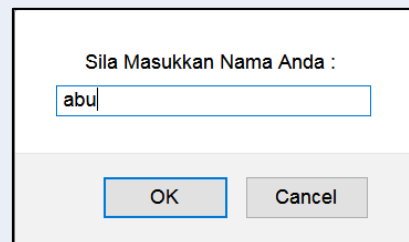
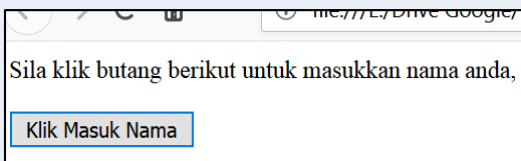
```
<html>
  <body>
    <p>Sila klik butang berikut untuk masukkan nama anda,</p>
    <button onclick = "masukNama()">Klik Masuk Nama</button>
    <p id = "maklumat"> </p>

    <script>

    function masukNama()
    {
    var pengguna = window.prompt ("Sila Masukkan Nama Anda : ", "");

    if (pengguna != null)
    {
    document.getElementById ("maklumat").innerHTML = "Hai " + pengguna + " ! Apa Khabar hari
ini? ";
    }
    }
    </script>
  </body>
</html>
```

OUTPUT



LAMAN WEB INTERAKTIF

- Laman web interaktif bermaksud suatu laman web yang mempunyai kandungan yang dinamik dan sesuai atas sesuatu carian atau permintaan daripada pengguna.
- Kandungan laman web tersebut boleh diubah daripada laman web tertentu atau hasil daripada pemprosesan komputer pelayan yang dipaparkan berbeza mengikut permintaan dari pengguna.
- Data yang disimpan dalam pangkalan data dengan menggunakan MySQL boleh dicapai dan dikemas kini melalui laman web.
- Untuk tujuan itu, sambungan kepada pangkalan data perlu dilakukan terlebih dahulu sebelum data dapat dicapai dan dikemaskini.



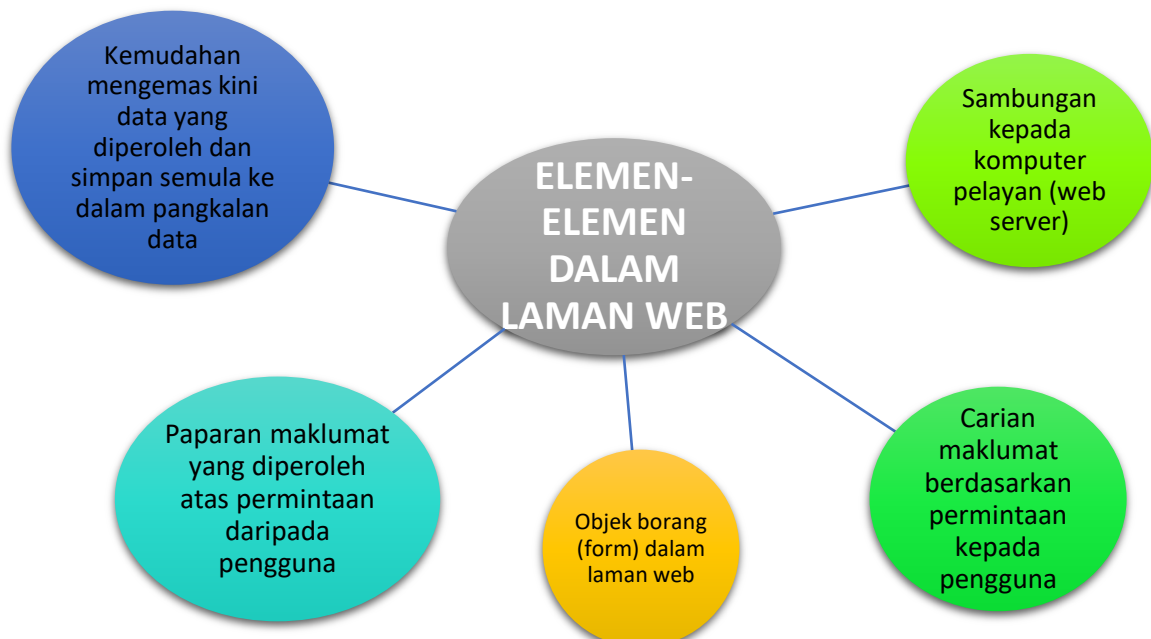
Menggunakan Bahasa pengaturcaraan web seperti PHP, ASP dan lain-lain.

Mempunyai capaian kepada pangkalan data.

Kandungan laman web berubah mengikut permintaan pengguna.

Kandungan laman web sering dikemaskini.

CIRI-CIRI LAMAN WEB INTERAKTIF



3.3

3.3.6

MENGGUNAKAN BAHASA PENSKRIPAN PELAYAN
UNTUK MEMBINA LAMAN WEB YANG BOLEH
MENCAPAI DAN MENGEMAS KINI DATA DALAM
PANGKALAN DATA

MEMBINA LAMAN WEB

CONTOH

~Rujuk Buku Teks~ ms 343 - 349

3.3

3.3.7

MENGHASILKAN SEBUAH LAMAN WEB INTERAKTIF
UNTUK KEGUNAAN PENGGUNA DAN PENTADBIR
SISTEM BAGI MENYELESAIKAN MASALAH

PENGGUNA DAN PENTADBIR SISTEM

- Sesetengah laman web mempunyai antara muka yang berbeza, untuk pengguna biasa dan pentadbir sistem.
- **LAMAN WEB UNTUK PENTADBIR** : membolehkan pentadbir sistem mendaftarkan nama pengguna yang boleh mencapai data dalam pangkalan data.
- **LAMAN WEB PENGGUNA** : membenarkan pengguna tersebut melihat maklumat yang telah ditetapkan sahaja.
- Cara untuk membezakan jenis pengguna laman web sama ada pengguna biasa atau pentadbir sistem dapat dilakukan dengan menyediakan laman web log masuk.



MEMBINA LAMAN WEB

CONTOH

~Rujuk Buku Teks~ ms 350 - 353

