

SPM2102

BAHASA PENGATURCARAAN 1

Tajuk:

**PENGENALAN KEPADA BAHASA
PENGATURCARAAN C++**

Disediakan oleh:

**NORASYKIN MOHD ZAID
NORAH MD NOOR**



Objektif

Di akhir pengajaran ini anda akan mempelajari:

- Pengenalan kepada C++
- Persekitaran Bahasa C++
- Jenis data C++
- Struktur Atur Cara C++

```
#include <stdio.h>
int main(void)
{
    printf("Hello World!\n");
    return 0;
}
```

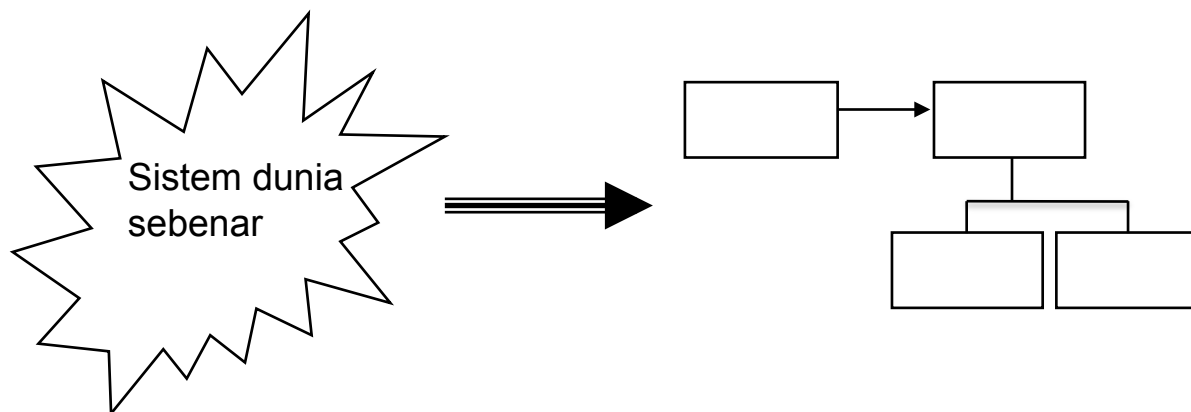
C/C++

Pengenalan Kepada C++

- C++ dicipta pada **tahun 1979** oleh **Bjarne Stroustrup** di makmal Bell New Jersey – 10 tahun selepas kelahiran bahasa C
- C++ mengandungi **semua ciri-ciri C** dengan tambahan ciri-ciri baru – bertujuan untuk menghapuskan kekurangan yang ada pada bahasa C
- C menekankan kaedah pengaturcaraan **berstruktur** manakala C++ lebih menekankan kaedah pengaturcaraan **berorientasikan objek**.

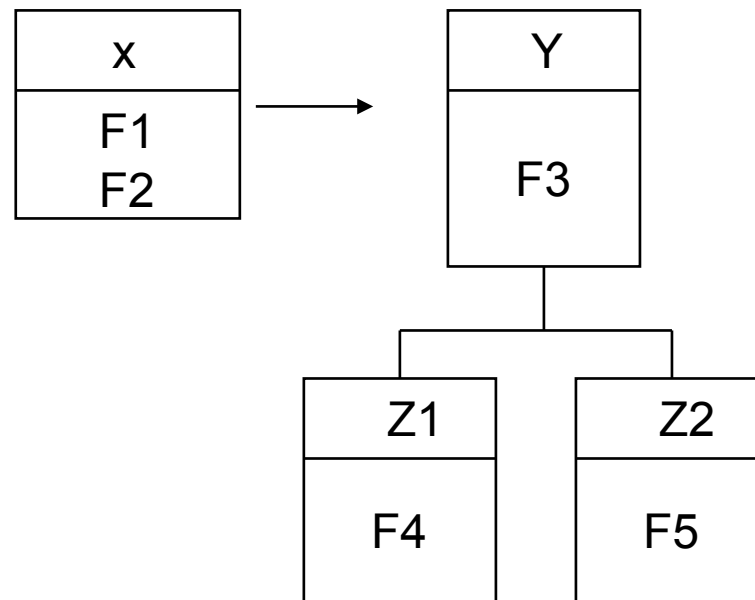
Pengenalan Kepada C++

- Apa itu Pengaturcaraan Berorientasikan Objek?
- Menggabungkan struktur data dan operasi dalam pengaturcaraan
- Metodologi pengaturcaraan berorientasikan objek secara umumnya boleh digambarkan seperti berikut:
 - a. Kenalpasti objek daripada sistem dunia sebenar



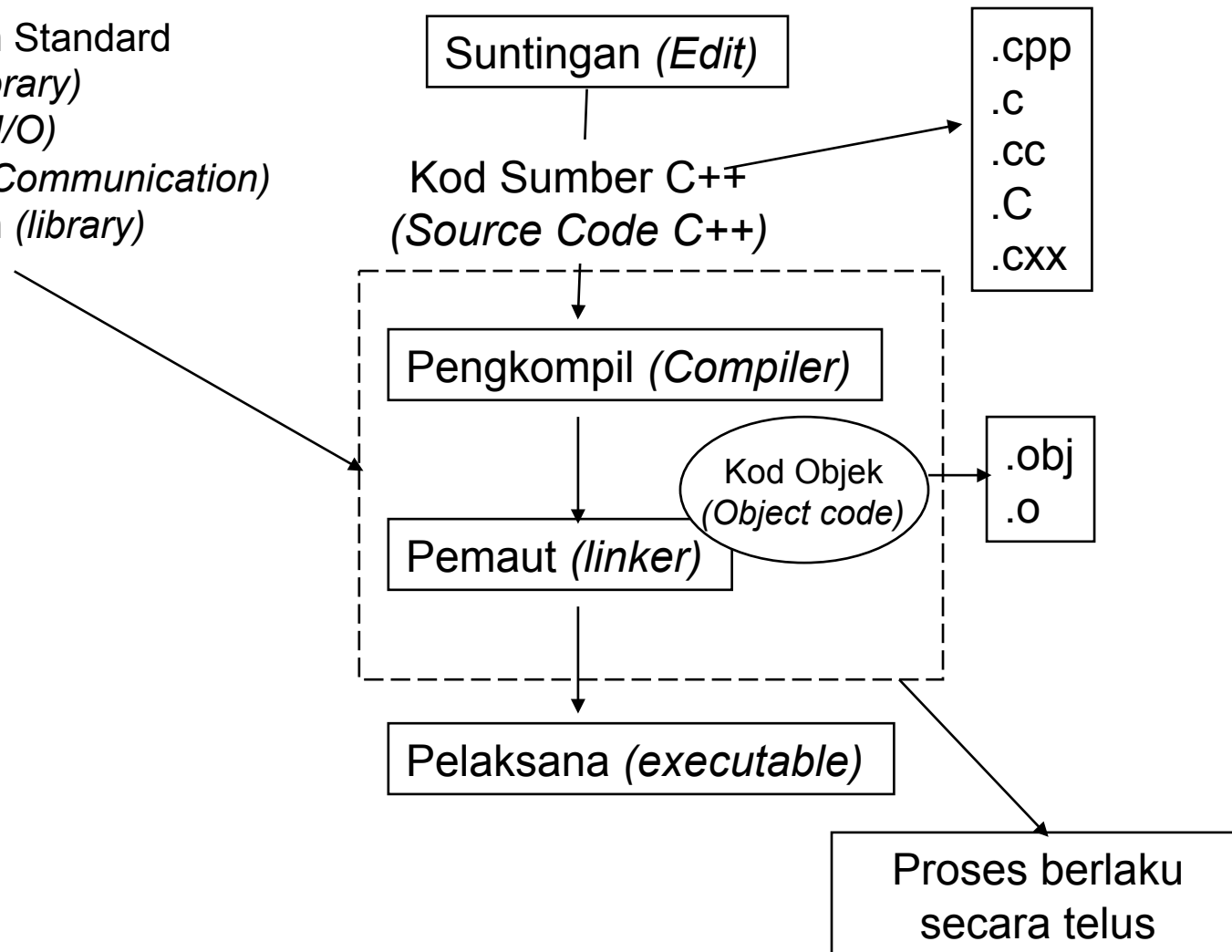
Pengenalan Kepada C++

b. Tentukan operasi, struktur data antara objek-objek



Pengenalan Kepada C++

- Perpustakaan Standard (*Standard library*)
- Input output (*I/O*)
- Komunikasi (*Communication*)
- Perpustakaan (*library*)



Persekitaran Bahasa C++

- Terdapat beberapa istilah penting yang mempunyai fungsi-fungsi tertentu dalam persekitaran bahasa C++:
 - Penyunting teks
 - Pengkompil
 - Debugger
 - Pemaut
 - Make
 - Integrated Development Environment (IDE)



Pengenalan Kepada C++

- **Penyunting teks/teks editor**
 - Membenarkan aktiviti penulisan dan penyuntingan kod aturcara C++
 - Notepad (simple editor), emacs (UNIX), pico
- **Pengkompil**
 - Menukarkan source code kepada object code yang dapat difahami oleh CPU
 - DOS/Windows
 - Borland C/C++
 - Microsoft Visual C/C++
 - UNIX
 - GNU C/C++ compiler

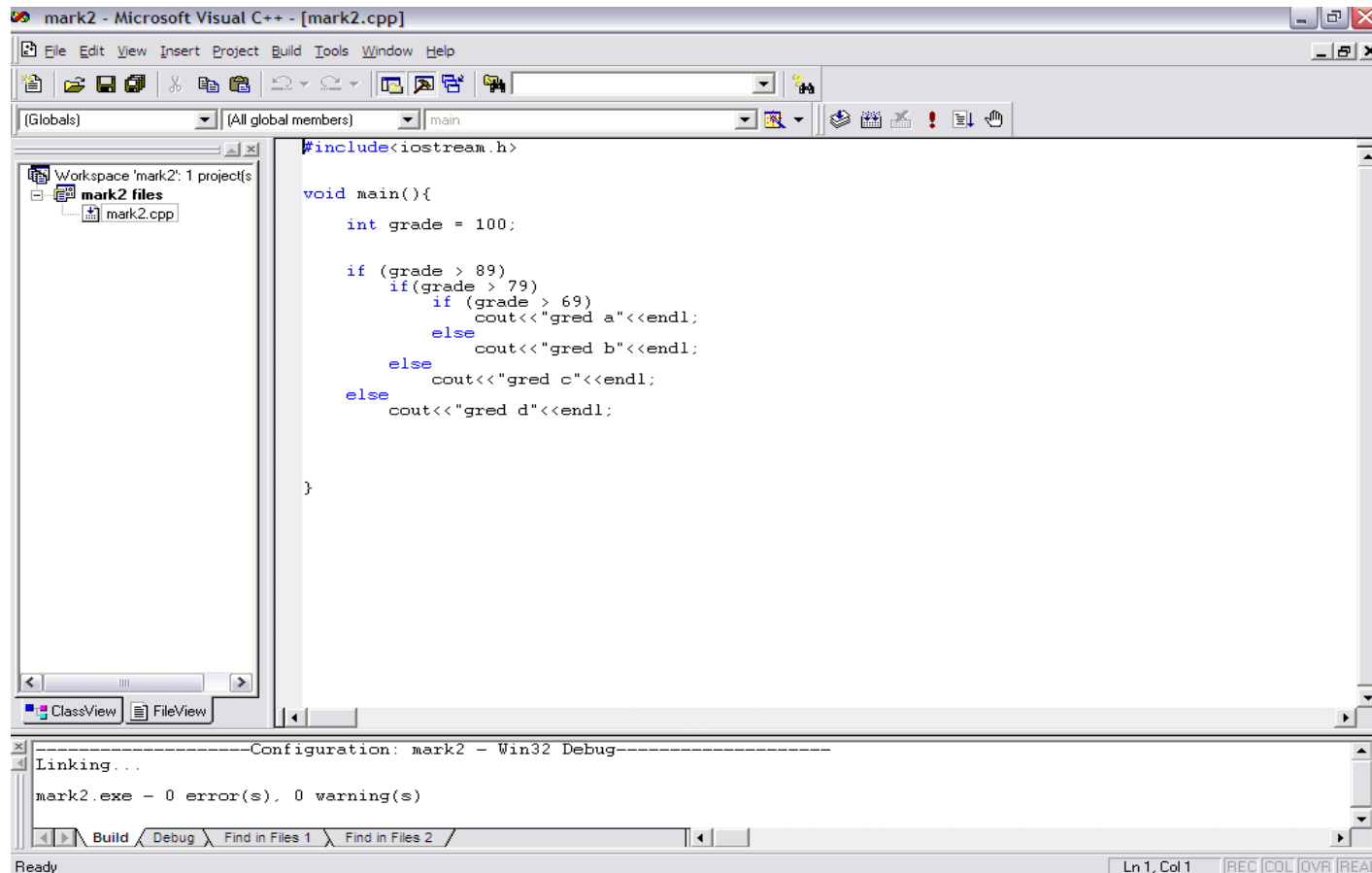
Pengenalan Kepada C++

- **Pemaut/linker**
 - Menukar object code ke bentuk fail jenis .exe.
 - menggabungkan kesemua bahagian yang diperlukan (seperti fail perpustakaan) oleh aturcara untuk menghasilkan kod pelaksanaan terakhir dalam bentuk Fail .exe yang dapat dilarikan
- **Debugger**
 - Perisian yang digunakan untuk menganalisa aturcara yang dibangunkan
 - Mengenalpasti ralat dan kesilapan dalam program

Pengenalan Kepada C++

- **Make**
 - Satu program utiliti yang digunakan dalam pembangunan projek C/C++
- **Integrated Development Environment (IDE)**
 - Mengabungkan aktiviti penyuntingan, pengkompilan, debugging & pengujian dalam satu persekitaran
 - Memudahkan pengendalian projek pengaturcaraan seperti Turbo C++

Persekitaran Bahasa C++



The screenshot displays the Microsoft Visual C++ IDE interface. The main window shows a C++ source file named 'mark2.cpp' with the following code:

```
#include<iostream.h>

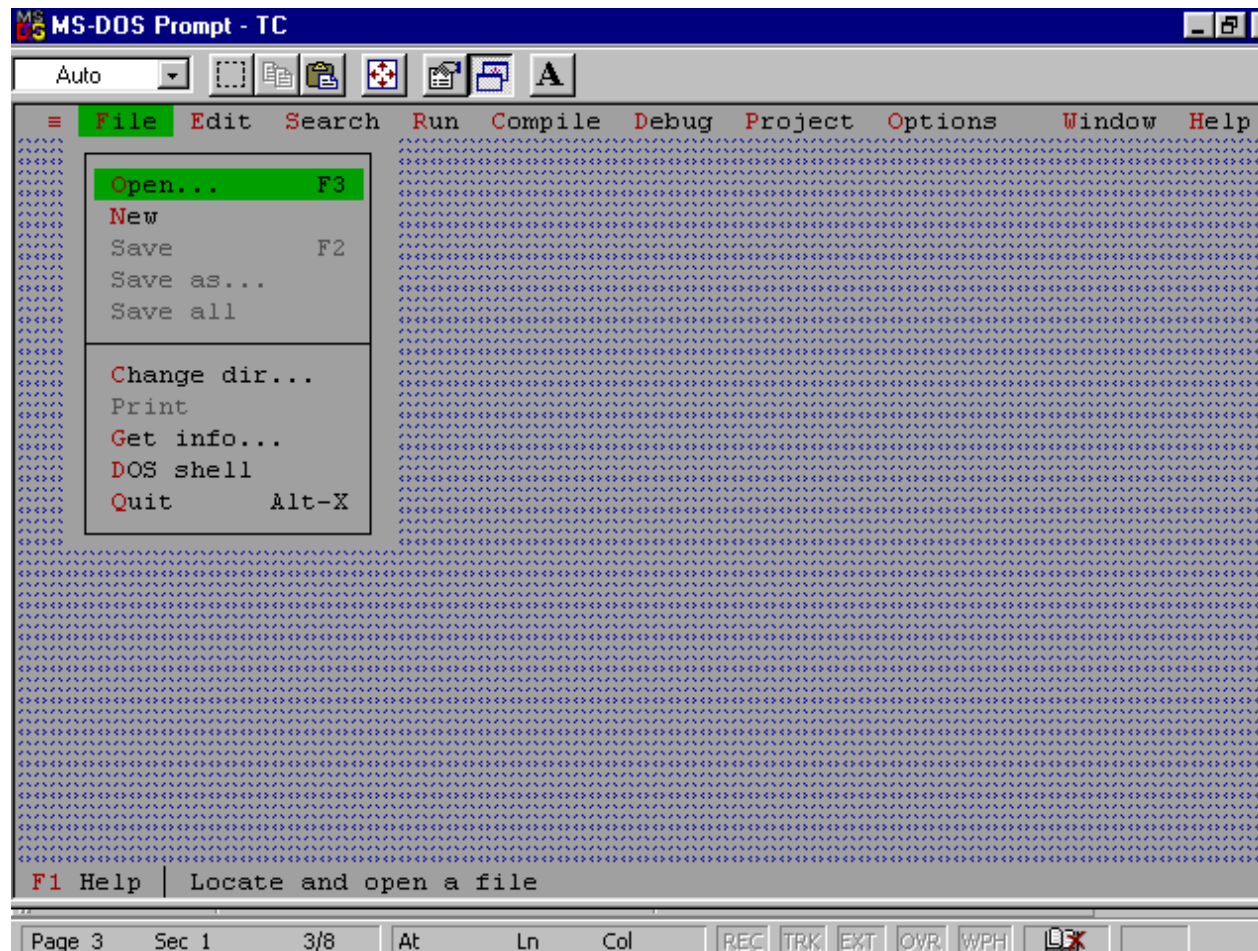
void main(){
    int grade = 100;

    if (grade > 89)
        if(grade > 79)
            if (grade > 69)
                cout<<"gred a"<<endl;
            else
                cout<<"gred b"<<endl;
        else
            cout<<"gred c"<<endl;
    else
        cout<<"gred d"<<endl;
}
```

The IDE also shows a workspace named 'mark2' containing the 'mark2 files' folder with 'mark2.cpp'. The bottom status bar indicates the configuration is 'mark2 - Win32 Debug' and the output window shows 'Linking...' and 'mark2.exe - 0 error(s), 0 warning(s)'. The status bar at the bottom of the IDE shows 'Ready' and 'Ln 1, Col 1'.

Contoh IDE Microsoft Visual C++

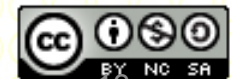
Persekitaran Bahasa C++



Contoh IDE Turbo C++

Ciri-ciri bahasa C++

- Tidak bersandar kepada sesuatu sistem pengoperasian
- Kecil dan mudah dipelajari
- Mirip kepada Pascal- jenis data dan aliran kawalan
- Direka bentuk untuk pengaturcaraan atas-bawah ·
- Membenarkan reka bentuk modular
- Bahasa yang berkesan · Mudah alih dan mudah lentur ·
- Kawalan komputer



Jenis Data C++

- Data terbahagi kepada dua kelas
 - **Data asas/primitif**
 - Data yang mempunyai hanya satu nilai (*single value*)
 - **Data berstruktur *array***
 - Data yang mempunyai nilai lebih daripada satu
 - membenarkan sebilangan data yang berkaitan dikumpul ke dalam suatu struktur
- Data Asas terdiri daripada dua jenis:
 - **Aksara**
 - Diwakili jenis data *char* (*character*)
 - **Numerik**
 - Terdiri daripada jenis data *int* (*integer*), *float* (*floating point*) dan *double* (*double precision*)

Jenis Data C++

- Data **char**
 - Digunakan bagi pengisytiharaan pembolehubah berjenis aksara
 - Contoh: **char** jantina
 - Pengisytiharaan pembolehubah jantina menyebabkan satu ruang ingatan bagi data jenis **char** diperuntukkan
 - Pembolehubah char menyimpan sebarang aksara bercetak dan tidak bercetak (huruf besar dan kecil (a-z), digit desimal (0-9), aksara khas dan aksara lepasan)

Jenis Data C++

- Data **int**
 - Digunakan bagi pengisytiharan pembolehubah berjenis integer negatif atau positif
 - Contoh: **int** jumlah
 - Pengisytiharaan pembolehubah jantina menyebabkan satu ruang ingatan bagi data jenis **int** diperuntukkan
 - Jenis data integer boleh dikalkifikasikan sebagai bertanda dan tidak bertanda
 - Bertanda-nilai positif dan negatif
 - Tidak bertanda-nilai positif sahaja

Jenis Data C++

- Data **float**
 - Digunakan bagi pengisytiharaan pembolehubah berjenis **titik apungan** atau **mempunyai titik perpuluhan** dalam ingatan komputer
 - Contoh: **float** pecahan
 - Pengisytiharaan pembolehubah pecahan menyebabkan satu ruang ingatan bagi data jenis **float** diperuntukkan
 - Data float boleh menyimpan nombor titik apungan atau perpuluhan dengan ketepatan **enam tempat perpuluhan**

Jenis Data C++

- Data **double**
 - Digunakan bagi pengisytiharan pembolehubah jenis perpuluhan juga
 - Contoh: **double** pi
 - Data double boleh menyimpan nombor titik apungan atau perpuluhan dengan ketepatan **15 titik perpuluhan**

Jenis Data C++

Jenis data	Bit	Julat	Contoh
char	8	0 hingga 255	A, b, @
int	16	-32768 hingga 32767	30566 , -2323
float	32	6 angka bernilai	2.333456
double	64	15 angka bernilai	3.142857142

Sintaks

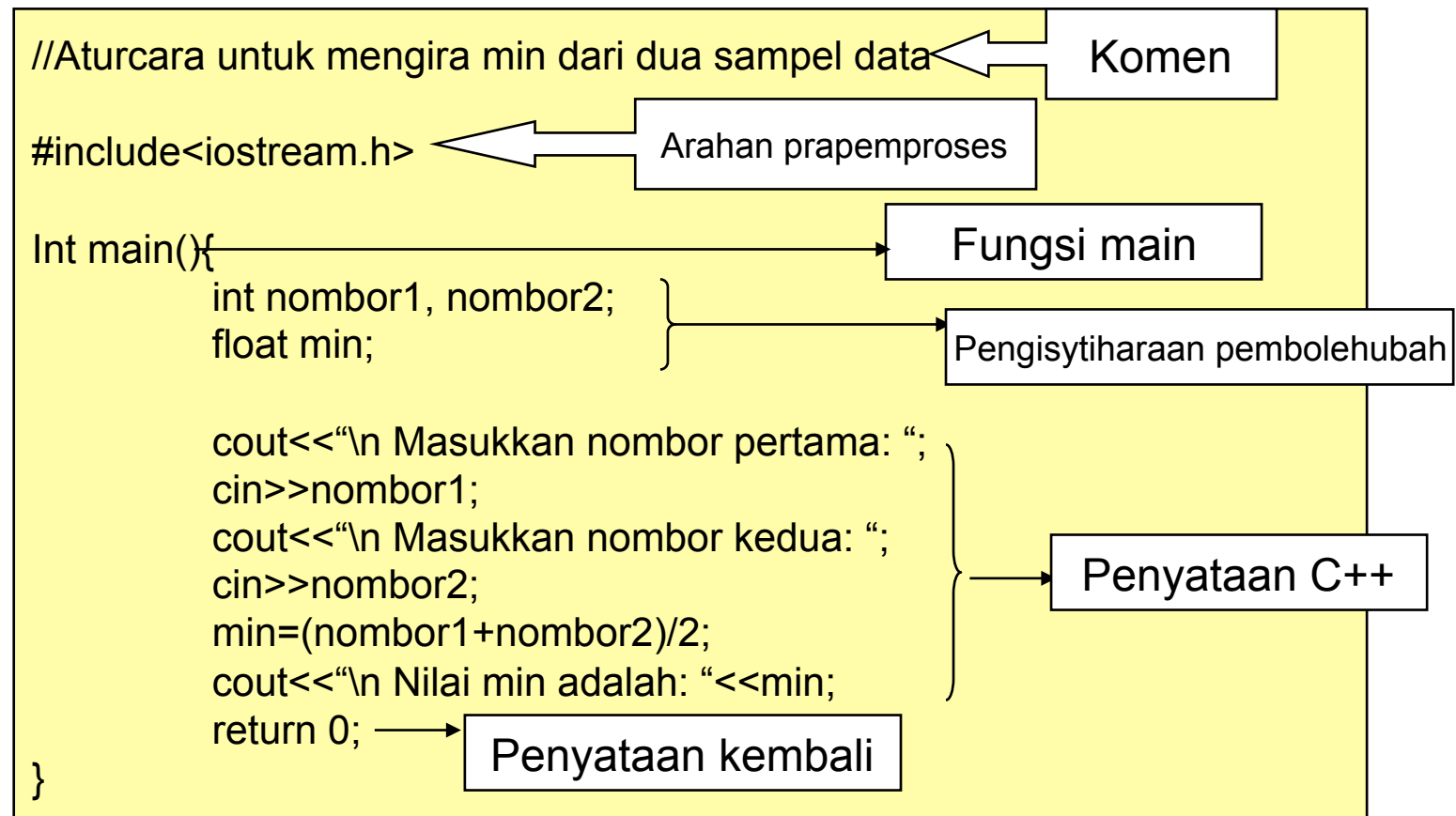
- Kata kunci merupakan perkataan-perkataan simpanan dan tidak boleh digunakan sebagai **pencam** yang ditakrifkan oleh pengguna. Ia mempunyai makna tertentu kepada pengkompil
- Pencam (identifier) ialah nama unik dalam skop yang diberikan kepada satu pemboleh ubah , pemalar , fungsi dan sebagainya.
- Contoh :-
- *If, else, do, while, for, int, long, double, unsigned, struct, short, goto, class, private, operator* dan sebagainya.22

Struktur Aturcara C++

- Suatu aturcara C++ mempunyai struktur asas seperti berikut:
 - Komen aturcara
 - Arahan prapemproses *preprocessor directives*
 - Fungsi utama *main function*
 - Pengisytiharaan pembolehubah
 - Penyataan C++
 - Penyataan kembali *return statement*

Struktur Aturcara C++

- Contoh aturcara 1:



Struktur Aturcara C++

- **Komen aturcara**
 - Boleh ditulis di mana-mana bahagian aturcara
 - Tidak menyebabkan komputer mengambil sebarang tindakan (Tidak akan diproses oleh kompiler)
 - Digunakan untuk menjadikan aturcara mudah difahami atau dibaca. Digunakan bagi tujuan penerangan sesuatu bahagian aturcara dan dokumentasi
 - Ditulis diantara `/*` dan `*/` atau selepas `//` sebagaimana contoh dibawah
 - Tanda `/*...*/`
 - cth: `/*Aturcara saya yang pertama*/`
 - Tanda `//`
 - cth: `//Aturcara saya yang pertama`

Struktur Aturcara C++

- **Arahan prapemproses**
 - Dimulakan dengan tanda #
 - Digunakan untuk memasukkan fail kepala (*header file*)
 - Bentuk arahan prapemproses adalah seperti berikut:
 - `#include<fail kepala>`
 - Arahan `#include<iostream.h>` adalah arahan untuk memasukkan fail kepala **stream input-output** yang mengandungi definisi bagi **cout** dan **cin**

Struktur Aturcara C++

- **Fungsi Main ()**
 - Setiap aturcara C++ mesti mempunyai satu fungsi **main()** yang **melaksanakan sesuatu tugas utama**
 - Terdiri daripada bahagian **kepala** dan **badan**
 - Bahagian kepala mengandungi takrifan-takrifan dan arahan-arahan prapemproses
 - Bahagian badan mengandungi kod-kod aturcara bagi fungsi main()
 - Menentukan apakah sebenarnya yang dilakukan oleh fungsi berkenaan

Struktur Bahasa C++

- Bentuk fungsi main() bagi sesuatu aturcara C++ seperti berikut:

– *Jenis fungsi* main()

```
{  
  Pernyataan C++...;  
}
```

```
int main()  
{  
    cout<<"Hello";  
    return 0;  
}
```

Contoh:

```
void main()  
{  
    cout<<"Hello";  
}
```

- Fungsi selain daripada main() boleh dibina mengikut keperluan
* akan dibincangkan pada topik seterusnya

Struktur Bahasa C++

- **Pernyataan Kembali *return***
 - Ditulis pada akhir aturcara di mana ianya akan mengalihkan kawalan dari aturcara kepada sistem pengoperasian
 - Return 0, membawa maksud aturcara dapat dilarikan tanpa sebarang ralat
 - Bagi fungsi yang menggunakan *void*, tiada nilai akan dikembalikan ke sistem pengoperasian

Struktur Bahasa C++

- **Kenyataan C++ *statements***
 - Mengarahkan komputer melakukan tindakan
 - Terdapat dua jenis kenyataan C++
 - Kenyataan Ungkapan
 - Mewakili data seperti nombor atau aksara atau satu entiti seperti gabungan pemalar dan pembolehubah
 - Cth:
$$\text{Jum_gaji} = \text{jum_jam} * \text{kadar_gaji}$$
 - Kenyataan Kawalan
 - Terdiri daripada kenyataan jujukan, pilihan dan pengulangan

Struktur Bahasa C++

- **Penamat kenyataan C++**
 - Setiap kenyataan C++ mesti diakhiri dengan semicolon ;
 - Tanda semicolon berfungsi sebagai penamat
 - Jikalau tiada semicolon, kompiler akan memberitahu proses kompiling mengandungi ralat
 - Arahan prapemproses tidak memerlukan penamat (;)

Struktur Aturcara C++

- **Pengisytiharaan Pencam / Pembolehubah**
 - Pembolehubah adalah **nama** yang **mewakili lokasi storan** di ingatan komputer yang ditakrifkan oleh pengguna
 - Maklumat yang disimpan di dalam lokasi adalah merupakan **nilai pembolehubah** berkenaan
 - Setiap pembolehubah yang digunakan perlu diisytiharkan terlebih dahulu

Struktur Aturcara C++

- **Pengisytiharaan Pencam / Pembolehubah**
 - Nama pencam mesti unik di dalam skop dan hanya dibenarkan bermula dengan huruf (A-Z atau a-z) dan underscore (_)
 - Boleh gabungan huruf dan angka tetapi memenuhi syarat di atas
 - Tiada tempat kosong dalam penamaan pengecam
 - Panjang pengecam biasanya 8 tetapi boleh sehingga 21 aksara (8 yang pertama diterima sebagai sama)
 - Katasimpanan dan sintaks tidak boleh digunakan sebagai pembolehubah.
 - contoh → nilai, nombor, bil, _pendapatan, sumber, total, bil_lelaki, bil_perempuan ...

Struktur Aturcara C++

- **Pengisytiharaan Pencam / Pembolehubah**
 - Pengisytiharaan pembolehubah adalah seperti berikut:
 - *<jenis data>* <nama pembolehubah>
 - Cth:
 - int nombor
 - **int** adalah jenis data integer manakala nombor adalah nama bagi pembolehubah tersebut

Struktur Aturcara C++

- **Pengisytiharaan Pencam / Pembolehubah**
 - Pembolehubah perlu diisytiharkan sebelum ianya digunakan
 - Contoh:

```
Void main(){  
int num  
cout<<"Masukkan nombor pertama: ";  
cout<<endl;  
cin>>num  
}
```

Aturcara yang tidak mempunyai ralat sintak

```
Void main(){  
cout<<"Masukkan nombor pertama: ";  
cout<<endl;  
cin>>num  
int num  
}
```

Aturcara yang mempunyai ralat sintak

Rujukan

- Jabatan Multimedia Pendidikan (2004), *Pengenalan kepada Bahasa Pengaturcaraan dalam Konteks Bahasa C++* , Faculti Pendidikan, UTM (unpublished)
- Abdullah Zawawi Talib, Ahamad Tajudin Khader, Maziani Sabudin, Wahidah Husain (2002), *Prinsip-Prinsip Pengaturcaraan Menggunakan C++ : Edisi kedua*, Pearson Malaysia, ISBN 983-9236-87-3
- Bakar, M. A et al (2002). Pengaturcaraan C. Malaysia. Prentice Hall
- Deitel, P. J., & Deitel, H. M. (2005). *C++ How to Program (5th Edition)*, Prentice Hall