



DASAR PEMROGRAMAN

- **Bahasa Pemrograman**
- **Sejarah singkat C++**
- **Struktur program C++**



Yoannita

Bahasa Pemrograman

- Apa yang dimaksud bahasa pemrograman?
“Alat” yang digunakan programmer untuk memberikan instruksi ke komputer agar dapat memberikan suatu layanan atau melaksanakan suatu tujuan untuk penggunaanya.
- Program dapat diibaratkan sebagai resep masakan, yang menggambarkan langkah-langkah yang diperlukan untuk membuat sesuatu atau untuk melaksanakan suatu tugas.

Bahasa Pemrograman

- Sebagai contoh, ketika hendak membuat kue coklat, ada beberapa langkah yang harus diikuti :
 - Campurkan telur, mentega, dan gula pada mangkok
 - Tambahkan tepung, baking soda dan vanili
 - Kocok sampai adonan mengembang
 - Tambahkan chocolate chips
 - Panggang di oven
- Bagi seseorang yang pernah membuat kue sebelumnya dan sudah tahu jumlah setiap bahan yang diperlukan, resep ini sudah cukup jelas, akan tetapi tidak cukup jelas untuk seseorang yang belum pernah membuat kue.

Bahasa pemrograman

- Orang tersebut memerlukan resep sbb:
 - Taruh 2 butir telur ke mangkok
 - Tambahkan 1.5 sendok gula ke telur tersebut
 - ...
 - Panggang adonan selama 10-12 menit pada suhu 375 K atau sampai terlihat kecoklatan
- Walaupun begitu, masih ada masalah pada resep di atas. Instruksi pertama menyebutkan untuk menaruh 2 butir telur ke dalam mangkok, namun tidak dijelaskan untuk membuang kulitnya terlebih dahulu!

Bahasa pemrograman

- Walaupun hal tersebut adalah hal yang umum dilakukan, namun hal ini menggambarkan suatu konsep yang fundamental : komputer akan melakukan TEPAT apa yang diminta untuk dikerjakan, tidak lebih, dan tidak kurang.
- Ketika menulis suatu program, programmer harus menginstruksikan dengan jelas setiap langkah-langkah yang mungkin dan memikirkan skenario yang dapat terjadi.

Bahasa pemrograman

- Bahasa pemrograman pertama adalah bahasa assembly. Bahasa tersebut merupakan kumpulan instruksi yang diperuntukkan kepada prosesor dan sangat sulit dipahami.
- Sebagai contoh, jika kita hendak menambahkan dua angka (misalnya 2 dan 5) dan mendapatkan hasil penambahan tersebut :

Pada C++:

```
int a = 3 + 4;
```

Pada assembly:

```
ldl 3, R1
```

```
ldl 4, R2
```

```
addl R1, R2, R3
```

Bahasa Pemrograman

- C++ merupakan bahasa pemrograman tingkat tinggi. Programmer menulis kodenya lalu menggunakan **compiler** untuk menerjemahkan kode tersebut ke dalam bahasa assembly yang kemudian diterjemahkan ke dalam bahasa mesin yang akan dijalankan pada mesin/komputer yang digunakan.

Sejarah C/C++

- C++ merupakan perkembangan lebih lanjut dari C. Bahasa C dikembangkan tahun 1969-1973 oleh *Dennis Richie* di Bell Telephone laboratories di masa yang sama ketika UNIX OS juga dikembangkan di sana.
- Oleh karena C tidak memiliki kemampuan OOP (Object Oriented Programming), dikembangkanlah C++ yang merupakan versi C yang mendukung OOP
- C++ mulai dikembangkan tahun 1979 oleh Bjarne Stroustrup di Bell Labs sebagai kelanjutan dari bahasa C.

Sejarah

- C++ dikenalkan oleh Bjarne Stroustrup pada tahun 1980, dikembangkan dari bahasa C yang telah dikenalkan oleh Dennis Ritchie pada tahun 1972. Nama C digunakan sebagai penerus bahasa B yang dikenalkan Ken Thompson yang merupakan penerus dari bahasa BPLC (Basic Combined Programming Language) yang dikenalkan Martin Richard pada tahun 1967. Nama C++ menunjukkan adanya penambahan, yaitu class, tetapi pada awalnya dinamai dengan “C with class” yang dapat memfasilitasi pemrograman berorientasi objek.

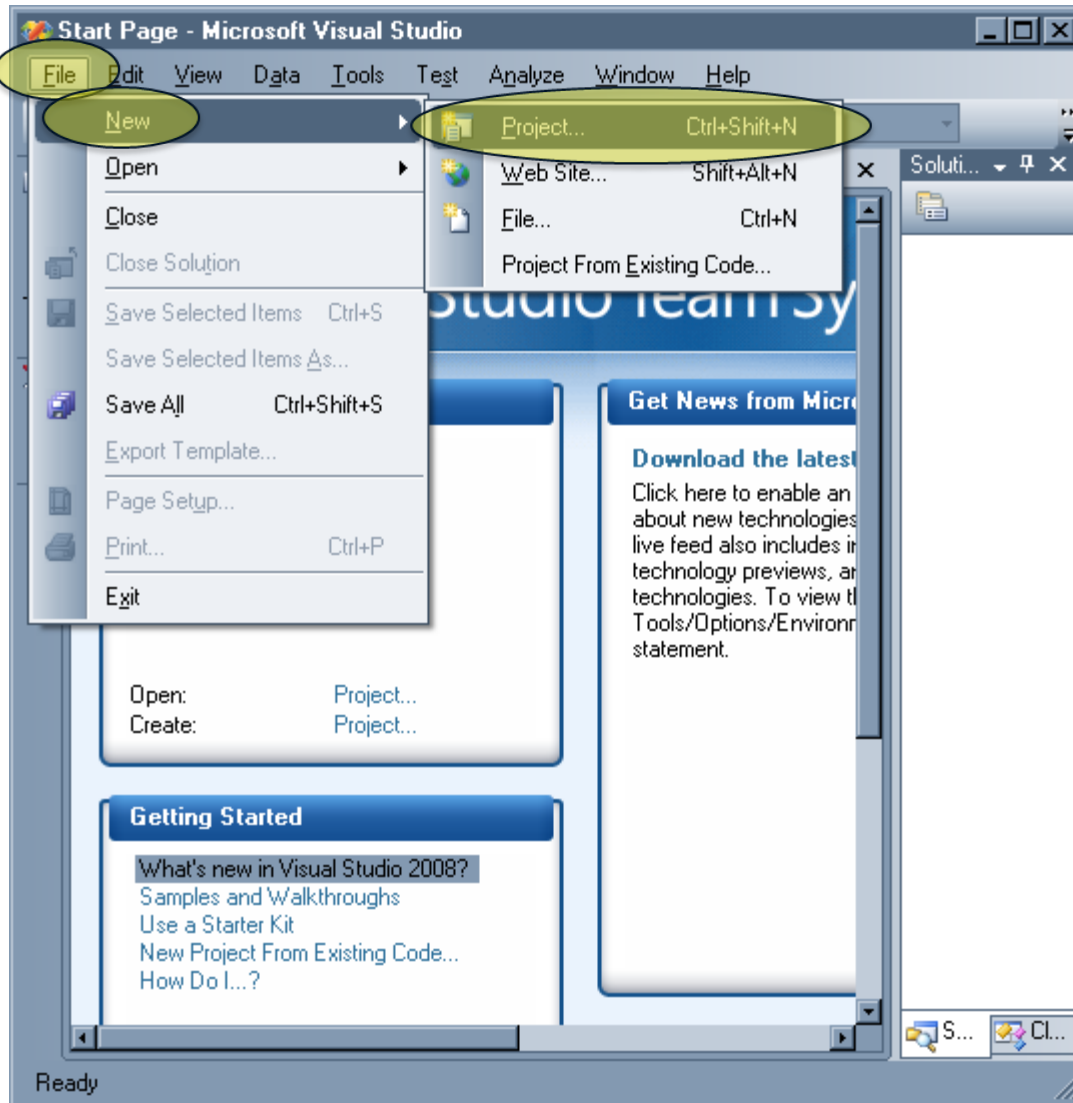
Perkembangan Bahasa C++

Nama Bahasa	Penemu	Keterangan
BCPL	Martin Richards	Basic Combined Programming Language
B	Ken Thompson	Suksesor BCPL
C	Dennis Ritchie	Suksesor Bahasa B
C++	Bjarne Stroustrup 	C dengan class

Mengapa C++ ?

- C++ diciptakan untuk mendukung pemrograman berorientasi objek. Konsep OOP ini muncul sebagai kebutuhan akan perangkat-perangkat lunak berukuran besar.
- Dikarenakan OOP memungkinkan pewarisan, jika ada perbaikan untuk suatu program yang relatif sama dengan program yang pernah dibuat sebelumnya, tidak perlu dicoding ulang dari awal; dengan cara memanfaatkan program yang sudah ada dengan tambahan-tambahan seperlunya.

Visual Studio 2008

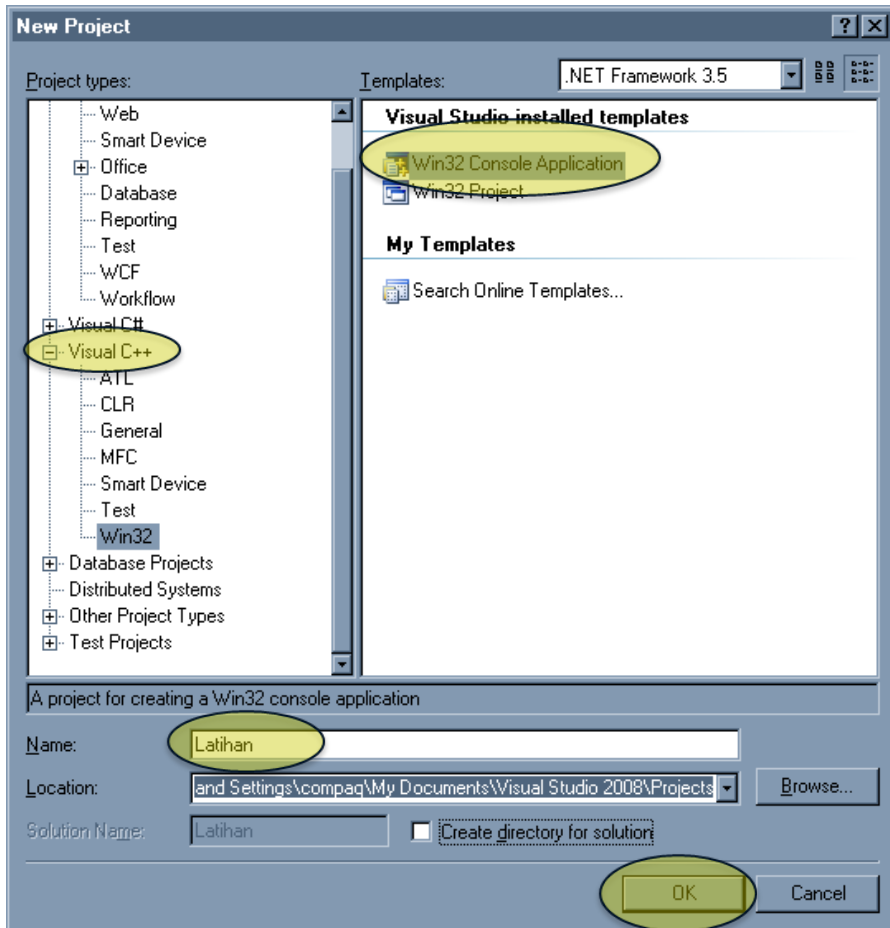


Memulai Project :

File → New →
Project

(atau tekan
CTRL + Shift +N)

Visual Studio 2008

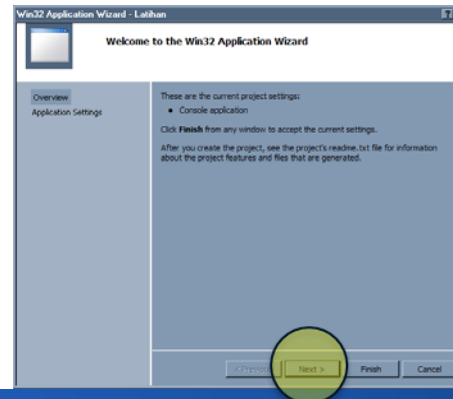


Project types :

- Pilih Visual C++ → Win32

Templates :

- Pilih Win32 Console Application
- Tulis nama Project
- Optional : Lepaskan tanda centang pada *Create Directory for Solution*



Klik OK,

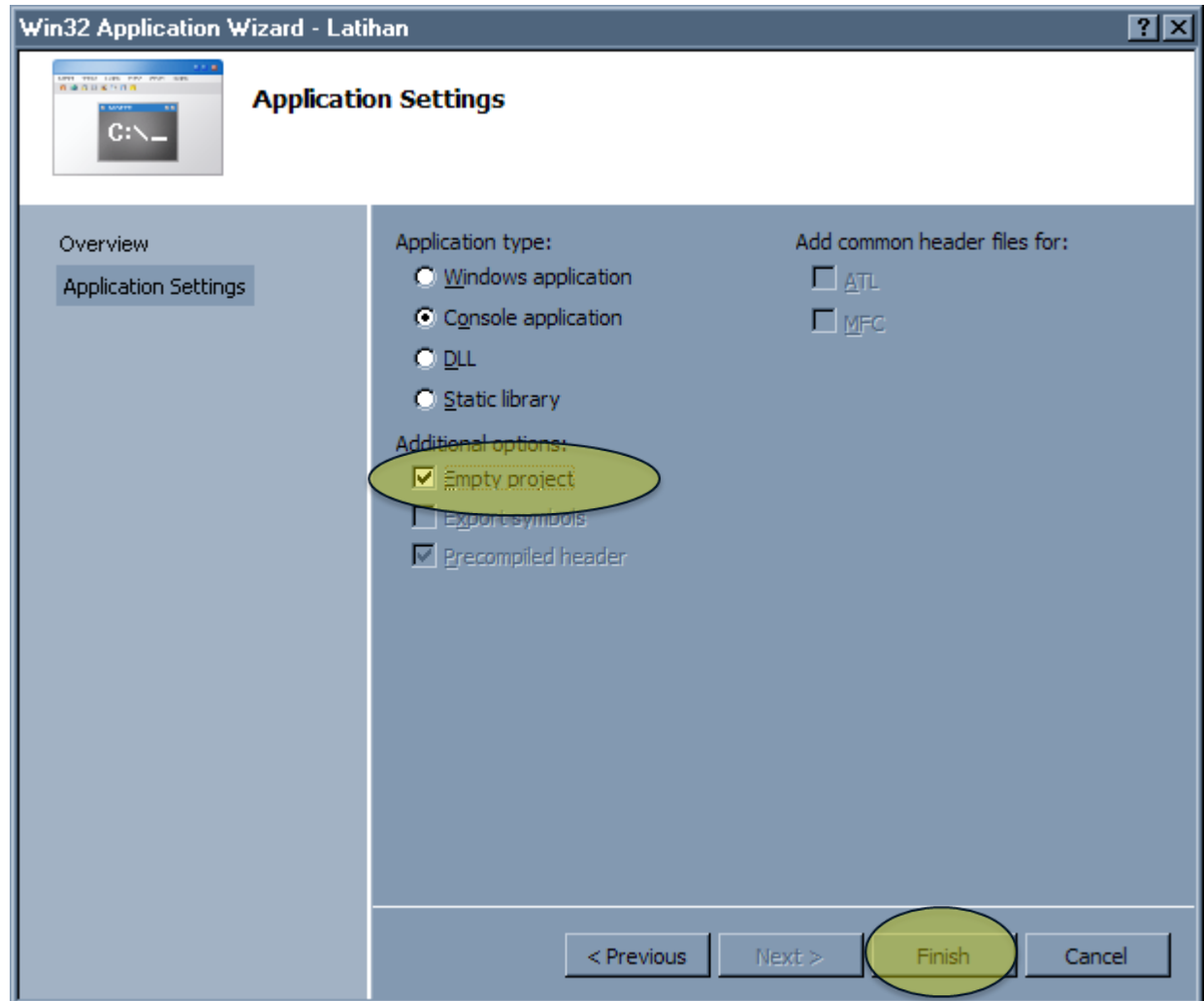
Akan tampil Win32 Application Wizard, Klik **Next**

Application type :

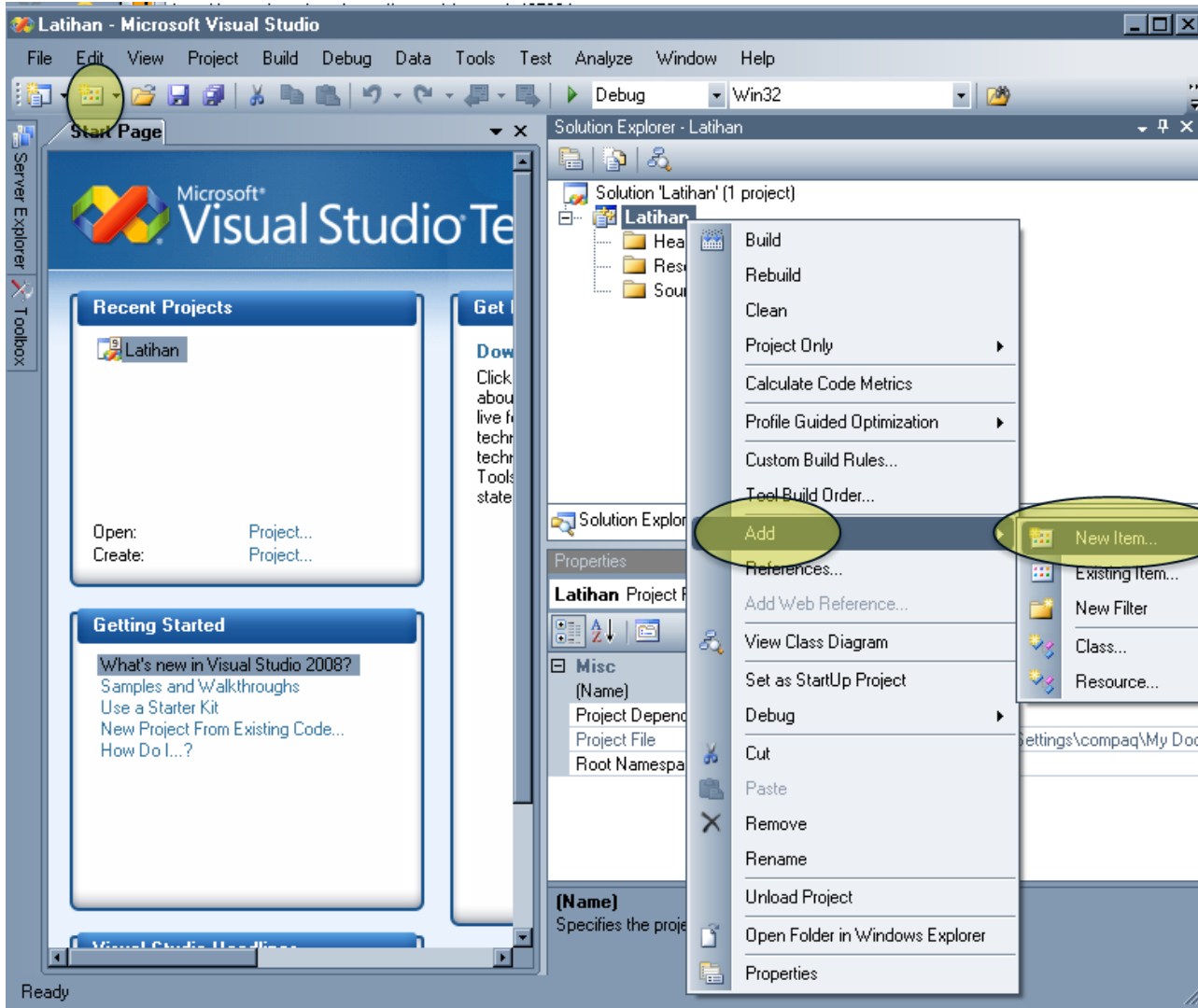
- pilih *Console application*

Additional options :

- Pilih *Empty Project*
- Klik *Finish*

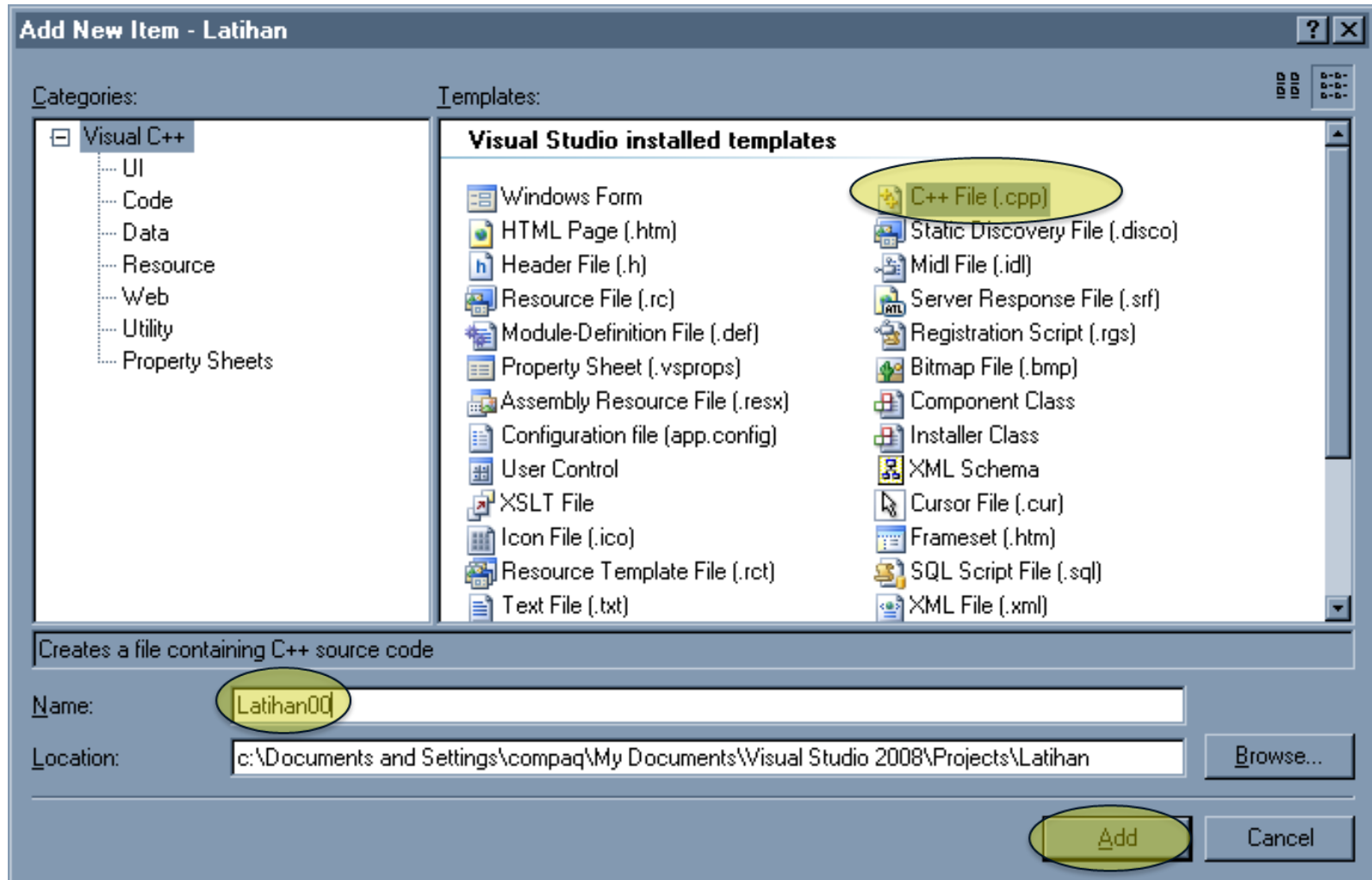


Visual Studio 2008



- Pada Solution Explorer, klik kanan nama Project → pilih Add → New Item

Visual Studio 2008



Visual Studio 2008

The screenshot shows the Visual Studio 2008 interface. The main window is titled 'Latihan - Microsoft Visual Studio'. The menu bar includes File, Edit, View, Project, Build, Debug, Data, Tools, Test, Analyze, Window, and Help. The toolbar contains various icons for file operations and development. The Solution Explorer on the right shows a project named 'Latihan' with subfolders for Header Files, Resource Files, and Source Files, and a file named 'Latihan00.cpp'. The Properties window at the bottom right shows the file properties for 'Latihan00.cpp', including its name, content type (False), file type (C/C++ Code), and full path. The main code editor is currently blank and shows '(Global Scope)'. A blue arrow points to the editor with the text 'Ketikkan kode program di sini'.

Ketikkan kode program di sini

Program C++

- Program C++ terdiri dari objek, fungsi, variabel, dan bagian-bagian lain.
- Contoh sederhana program C++ (1) :

```
1: #include <iostream>
2: using namespace std;
3: void main()
4: {
5:     cout << "Hello World!";
6: }
```

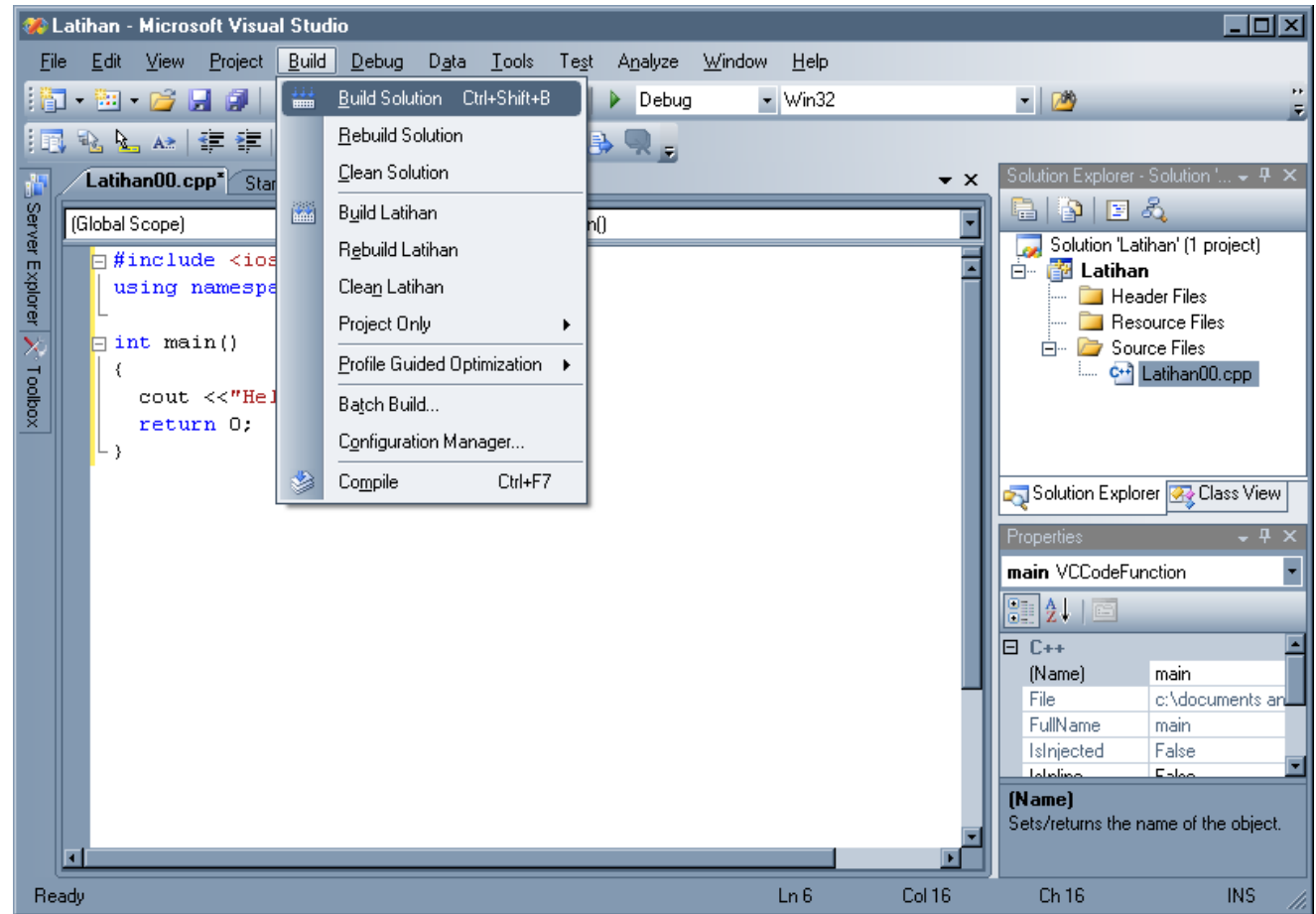
Hello World!

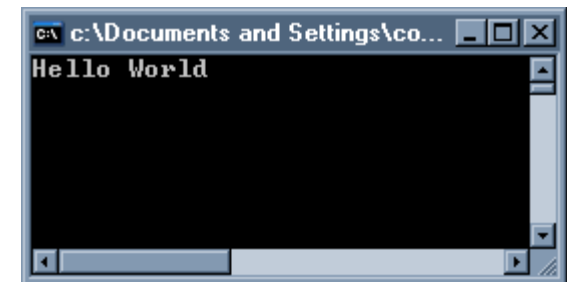
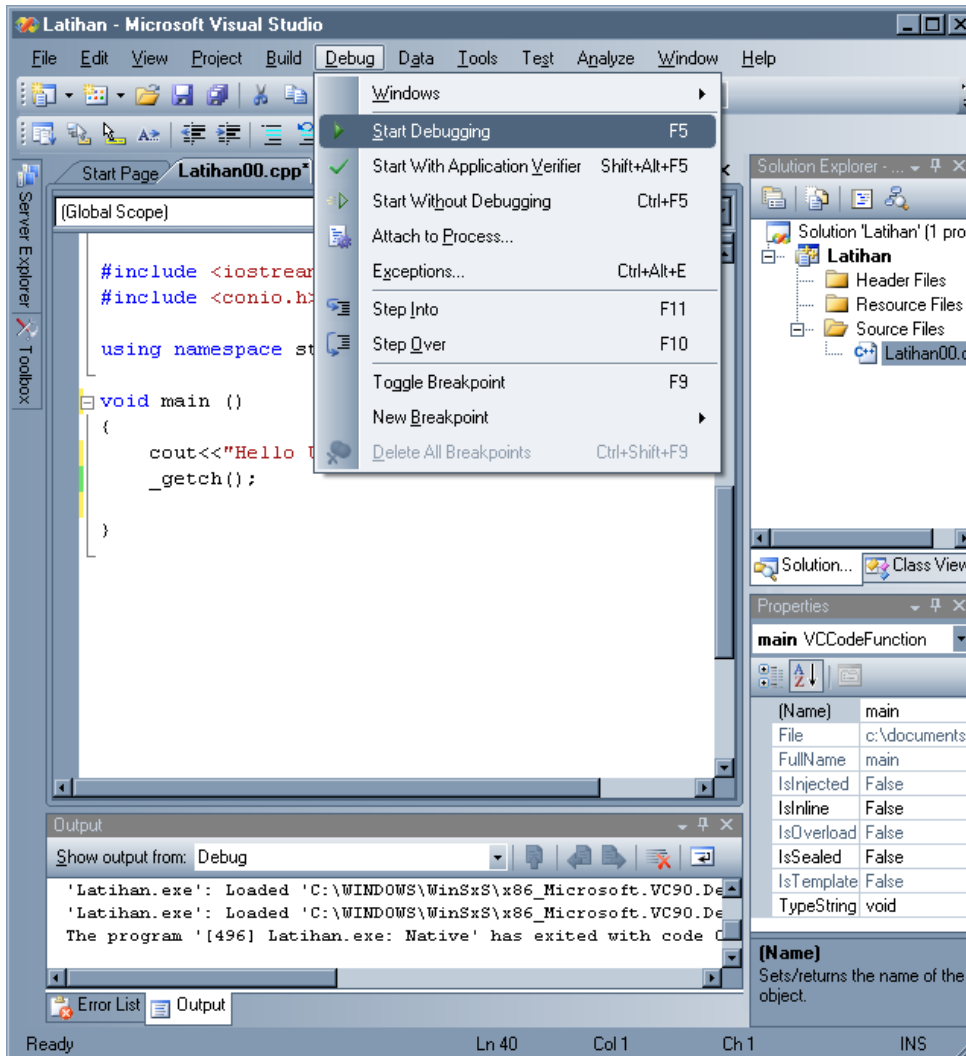
Program tersebut akan mencetak tulisan Hello World! ke layar output, namun layar output akan segera menutup.

Untuk dapat melihat hasil output sisipkan `system("pause")` atau `getch()` sebelum akhir program (lihat Contoh sederhana program C++ (2)).

Menjalankan program

- Pilih menu Build → Build Solution kemudian
- Tekan F5 (atau pilih menu Debug → Start Debugging)





Contoh sederhana program C++ (2) :

```

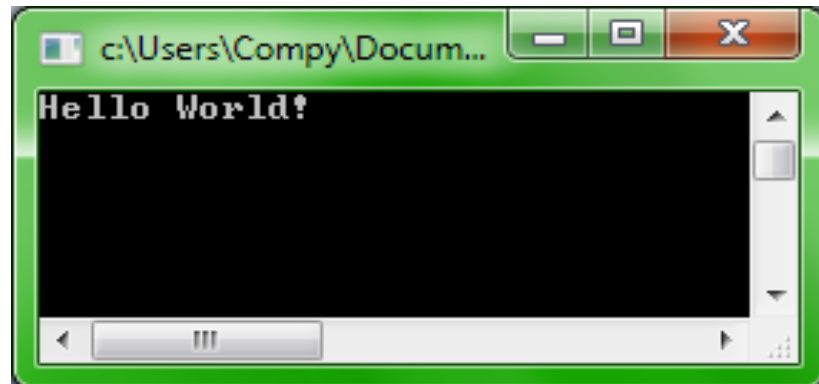
1: #include <iostream>
2: #include <conio.h>
3: using namespace std;
4: void main()
5: {
6:     cout << "Hello World!";
7:     _getch();
8: }
    
```

atau

```

1: #include <iostream>
2: using namespace std;
3:
4: void main()
5: {
6:     cout << "Hello\nWorld!";
7:     system("pause");
8: }
    
```

Program tersebut akan mencetak



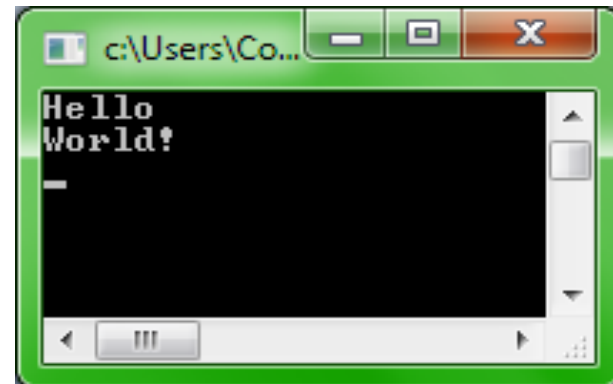
Contoh sederhana program C++ (3) :

- Untuk dapat mencetak baris baru, gunakan tanda `\n` (newline) di dalam kedua tanda petik dua, atau gunakan keyword `endl` (endline) pada `cout`.

```
1: #include <iostream>
2: #include <conio.h>
3: using namespace std;
4: void main()
5: {
6:     cout << "Hello\nWorld!";
7:     _getch();
8: }
```

Hello World!

Program tersebut akan mencetak



ke layar output

- \n digunakan agar cursor berpindah ke baris baru
- \n dapat diganti dengan `cout <<endl;`
- Contoh :
 - `cout<<"hello"<<endl<<"world!";`
 - `cout<<"hello"; cout<<endl<<"world!";`
 - `cout<<"hello"; cout<<endl; cout<<"world!";`

#include <iostream>

- Baris yang dimulai dengan tanda (#) merupakan **preprocessor**.
- Kode tersebut mengindikasikan *compiler's preprocessor*.
- Bentuk umum preprocessor bisa berupa :
 - 1) #include <namafile>
 - 2) #include "namafile"
- Bentuk pertama mengisyaratkan bahwa pencarian file dilakukan pada direktori khusus. Sedangkan bentuk kedua menyatakan bahwa pencarian file (*namafile*) dilakukan pertama kali pada direktori kerja.
- Jika hendak menggunakan file header yang disediakan oleh C++ maka gunakan bentuk pertama. Sedangkan jika memakai file header yang diciptakan sendiri maka akan menggunakan bentuk #include yang kedua

#include <iostream>

- Preprocessor disebut juga pengarah kompiler yang berfungsi mengatur proses kompilasi.
- Dalam contoh #include <iostream> dan #include <conio.h>, preprocessor tersebut menginstruksikan kompiler untuk membaca kode program yang ada dalam file `iostream` dan `conio`.
- #include <iostream> meminta preprocessor untuk menyertakan *iostream standard file*.
 - file (`iostream`) tersebut mencakup deklarasi-deklarasi standar untuk **input-output** library pada C++.
- file (`iostream`) disertakan karena fungsi-fungsi dalam file (`iostream`) tersebut akan digunakan kemudian di program.

using namespace std;

- Semua elemen dari library standar C++ dideklarasikan dengan **namespace std**.
- Baris tersebut sangat sering dicantumkan di program-program C++ karena kita membutuhkan fungsi-fungsi dari library standar tersebut.

main ()

- Baris ini berhubungan dengan definisi awal dari main function pada program.
- Main function merupakan titik awal bagi semua program C++ untuk mengeksekusi programnya.
- Kata main diikuti dengan kurung buka dan kurung tutup () yang merupakan indikasi bahwa main adalah sebuah fungsi.

(suatu fungsi dalam C++ harus mempunyai tanda “()”)

- Suatu fungsi jg diikuti dengan tanda { }, sebagai awal dan akhir suatu fungsi.

Apa yang terdapat dalam tanda { } tersebut adalah statement-statement yang akan dieksekusi ketika fungsi tersebut dipanggil.

main()

void main()

- Tidak memiliki nilai balik

```
void main ()  
{  
    //statements  
}
```

int main()

- Memiliki nilai balik, pernyataan return harus ditambahkan, untuk menyatakan nilai balik

```
int main ()  
{  
    //statements  
    return 0 ;  
}
```

cout << "Hello World!";

- Baris tersebut merupakan statement C++.
- Suatu statement adalah ekspresi sederhana maupun kompleks yang dapat menghasilkan suatu efek.
- Cout merepresentasikan standar output stream pada C++, dan arti dari `cout << "Hello World!";` tersebut adalah untuk memasukkan rangkaian karakter (dalam hal ini tulisan Hello World!) ke dalam standard output stream (biasanya berupa layar).
- `cout` dideklarasikan dalam `iostream` standard file termasuk juga `std` namespace, oleh karena itulah kita harus meng-`#include` file tersebut dan mendeklarasikan namespace tersebut di awal kode.

```
cout << "Hello World!";
```

- statement tersebut diakhiri dengan karakter semicolon (;).
- Karakter tersebut digunakan untuk menandai akhir dari suatu statement dan HARUS selalu digunakan di akhir semua *expression statement* di program C++.
- Kesalahan programmer pemula yang paling umum adalah lupa menyertakan karakter ‘;’ tersebut di akhir suatu baris.

return 0;

- Return merupakan contoh statement yang menginstruksikan agar suatu fungsi mengembalikan nilai balik tertentu.
- Return membuat suatu fungsi berakhir.
- Return dapat diikuti dengan suatu *return code* . Seperti contoh, return code 0 untuk fungsi main() mempunyai arti bahwa program bekerja seperti yang diharapkan, tanpa error selama dieksekusi.
- Biasanya return 0; digunakan untuk mengakhiri suatu C++ console program.