

Pengenalan

Lingkungan Visual C++



Umi Sa'adah, S.Kom & Ira Prasetyaningrum

Politeknik Elektronika Negeri Surabaya

e-mail : ira@eepis-its.edu

Peraturan



- ⌘ Tata tertib pratikum :
 - ☒ Selalu menyimpan (save) yang dikerjakan dalam pratikum.
 - ☒ Menggunakan jas lab.
 - ☒ Toleransi keterlambatan 10 menit (untuk mahasiswa)
 - ☒ Ada tugas resmi setelah pratikum.
 - ☒ Merapikan kursi.
 - ☒ Selalu matikan komputer setelah pratikum.

- ⌘ Sopan santun :
 - ☒ Menggunakan baju sopan.
 - ☒ Dilarang menggunakan sandal.
 - ☒ Dilarang bicara kotor.
 - ☒ Dilarang makan dan minum dikelas.
 - ☒ Dilarang merokok dikawasan kampus.
- ⌘ Ada posttest minimal 4 kali.
- ⌘ Nilai :
 - posttest : 50%
 - tugas resmi : 25 %
 - Percobaan di kelas : 25%

PERSIAPAN : Membuat Folder untuk Menyimpan

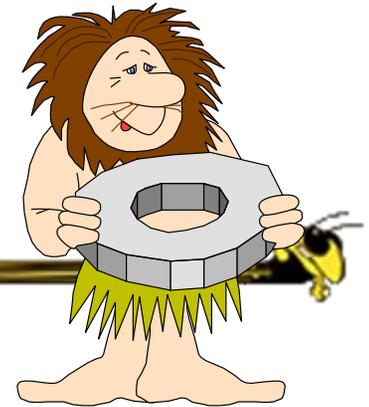


Membuat folder baru di C dengan nama

“ **C:\Prakt\ProgC\nrp sudara** ”

- ⌘ Klik kanan mouse pada tombol **Start** pada *taskbar* Windows
- ⌘ Pilih menu : **Explore** untuk masuk ke Windows Explorer
- ⌘ Klik pada drive yang diinginkan, misalnya : **C**
- ⌘ Pilih menu **File** -> **New** -> **Folder**
- ⌘ Ketikkan nama folder : **Prakt**
- ⌘ Klik mouse dua kali pada folder Prakt yg baru dibuat
- ⌘ Ulangi langkah 3 – 5 untuk nama folder : **ProgC**
- 1. Ulangi langkah 3 – 5 untuk nama folder : ***nrpAnda*** (misalnya : 7411030001)
- 2. Folder siap digunakan pada posisi : **C:\Prakt\ProgC\7411030001**

Memulai Visual C++



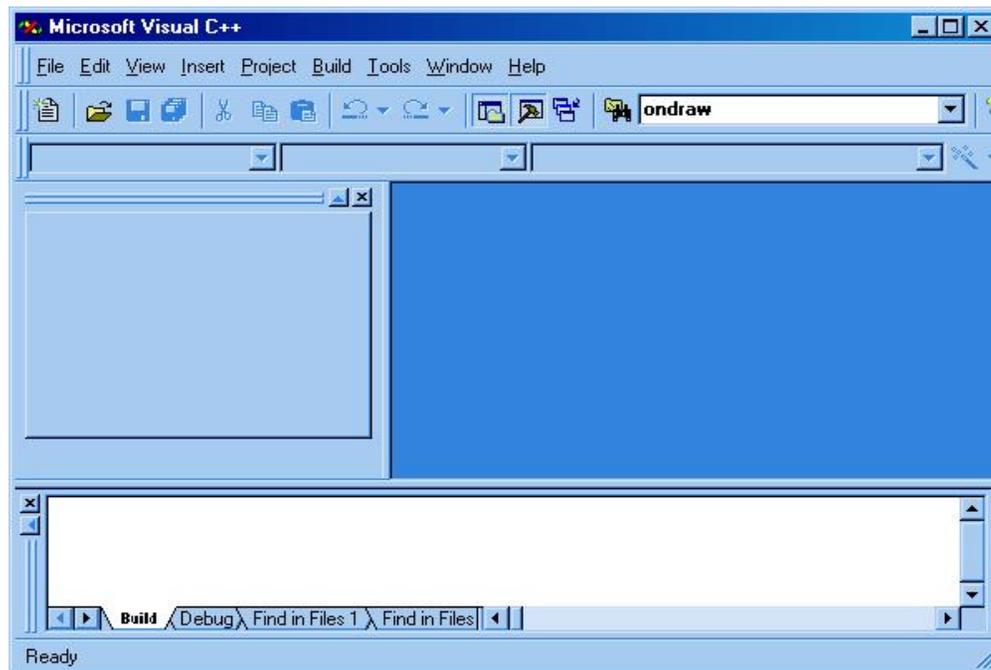
1. Klik tombol **Start** pada *taskbar* Windows
2. Klik **Programs** dari Start menu
3. Klik **Microsoft Visual C++ 6.0**

(Jika Visual C++ di-install sebagai produk yang *standalone*, maka akan terlihat Microsoft Visual C++ 6.0 sebagai salah satu pilihan dalam menu Programs. Namun jika di-install sebagai bagian dari Microsoft Visual Studio, maka Visual C++ akan tercantum dalam menu di bawah Microsoft Visual Studio 6.0 dalam menu Programs).

Memulai Visual C++ (2)



Tampilan awal dari Visual C++ 6.0



Project dan Workspace



- ❖ **Workspace** adalah ruang yang digunakan untuk menempatkan **project**.
- ❖ Dalam satu workspace dimungkinkan diisi lebih dari satu project.
- ❖ Workspace akan disimpan dengan ekstensi **.dsw** (*Developer Studio Workspace*)

Project dan Workspace(2)

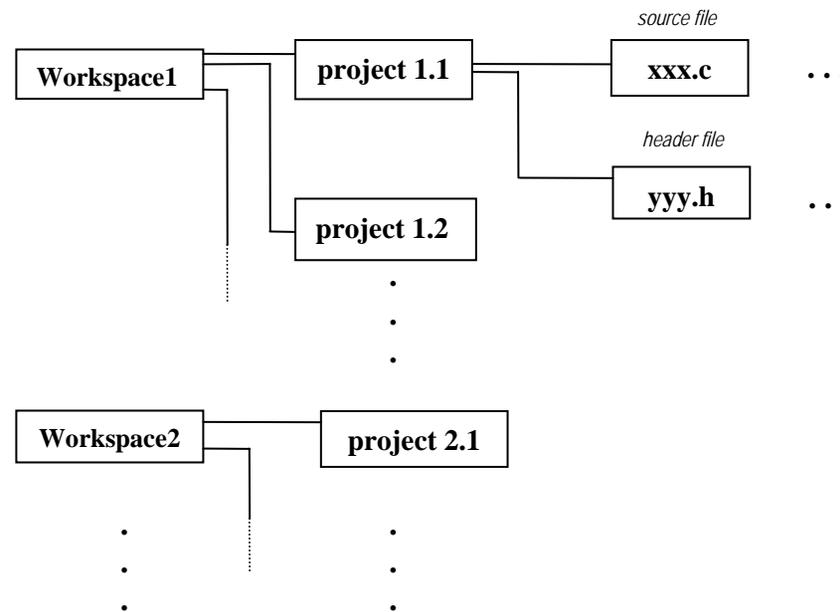


- ❖ Setiap kali memulai membuat sebuah program, terlebih dahulu harus membuat sebuah project.
- ❖ File project ini menyimpan berbagai informasi, di antaranya tentang *source code* (kode sumber) mana yang akan dipakai dalam program.
- ❖ Project disimpan dengan ekstensi **.dsp** (*Developer Studio Project*).

Project dan Workspace(3)



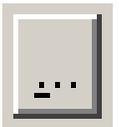
Diagram workspace, project dan file



Membuat Workspace Baru



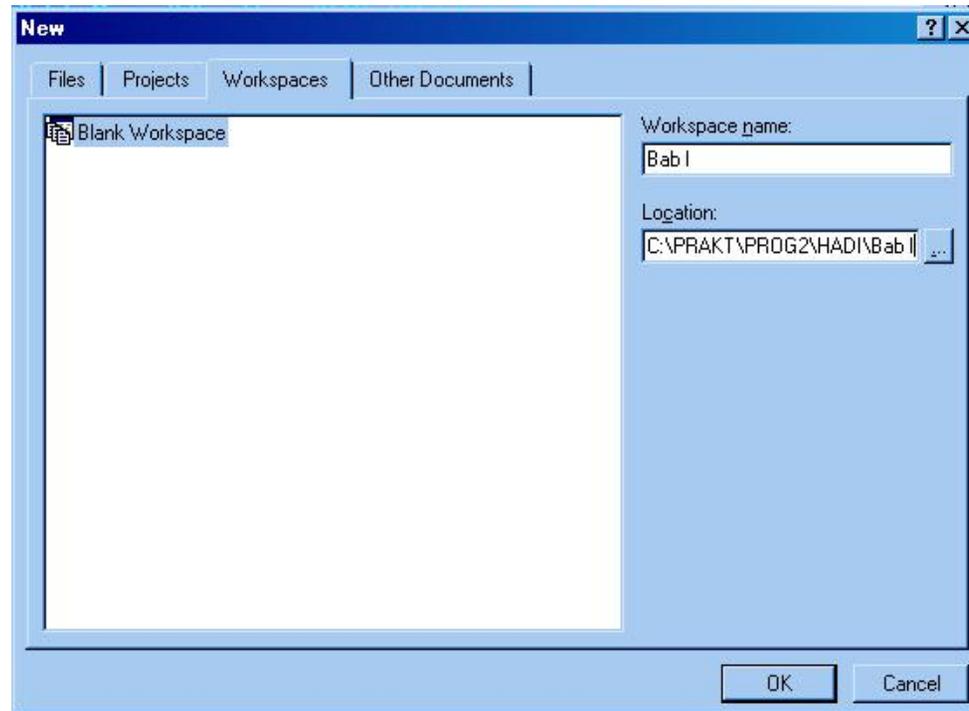
1. Klik **New** pada File menu
 2. Dalam kotak dialog New, klik **Workspaces**
 3. Ketiklah nama *workspace* yang hendak dibuat. Dalam praktikum ini, nama *workspace* mewakili nama bab bahasan, misalnya : **Bab I**
 4. Pastikan lokasi tempat *workspace* adalah :
C:\Prakt\ProgC\user_nrp)** (misalnya
C:\Prakt\ProgC\7411030001)
 5. Klik **OK**
- ***) Untuk menempatkan *workspace* klik : **Browse**



Membuat Workspace Baru(2)



Kotak dialog untuk New Workspaces



Membuat Project Baru

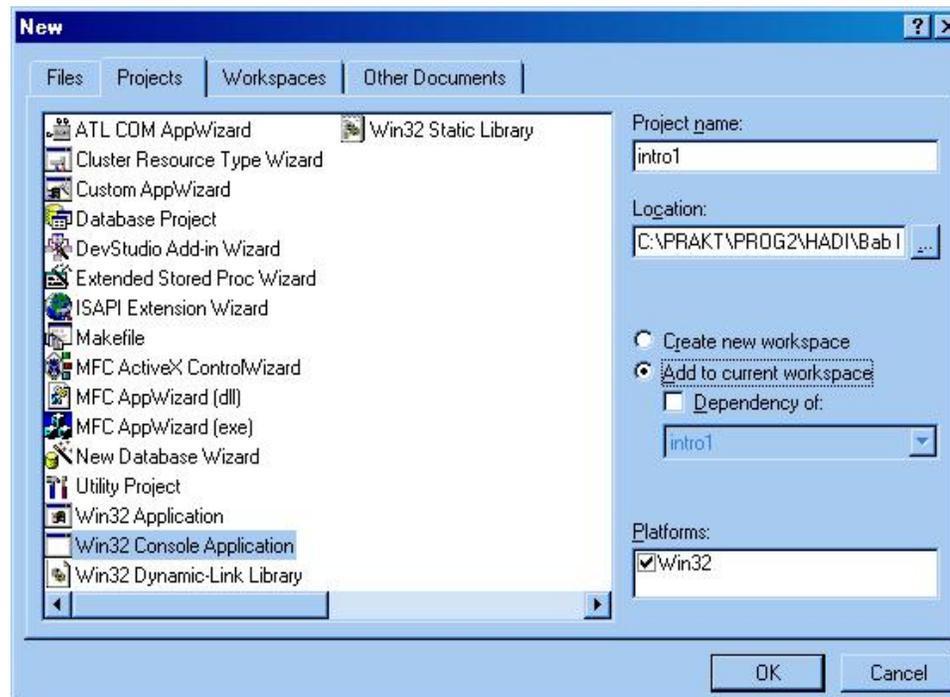


1. Klik **New** pada File menu
2. Dalam kotak dialog New, klik **Projects**
3. Klik **Win32 Console Application**
4. Ketikkan nama *project* yang dikehendaki, dalam hal ini misalnya : **intro1**
5. Pilihlah opsi : **Add to current workspace**, agar *project* yang baru dibuat disimpan di bawah *workspace* Bab I yang telah dibuat sebelumnya
6. Pastikan **Win32** telah terpilih pada kotak **Platforms**
7. Klik **OK**

Membuat Project Baru(2)



Kotak dialog untuk New Projects



Membuat Project Baru(3)

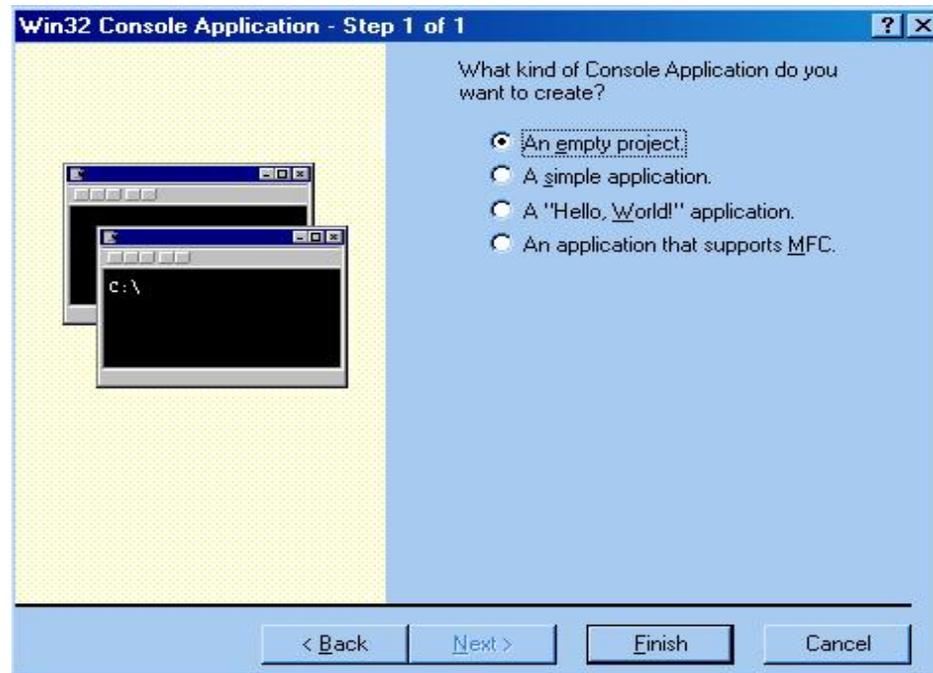


- ⌘ Ikutilah instruksi dalam kotak dialog Wizard yang akan muncul setelah kota dialog New.
- ⌘ Untuk sebuah *Win32 console application* hanya ada satu kotak di bawah ini :
 1. Dalam kotak dialog **Step 1 of 1** pilihlah opsi : **An empty project** (gambar 1.5)
 2. Klik **Finish**
 3. Dalam kotak dialog **New Project Information**, telitilah informasi yang diberikan untuk memastikan apa yang hendak dibuat. Berikutnya klik **OK**.

Membuat Project Baru(4)



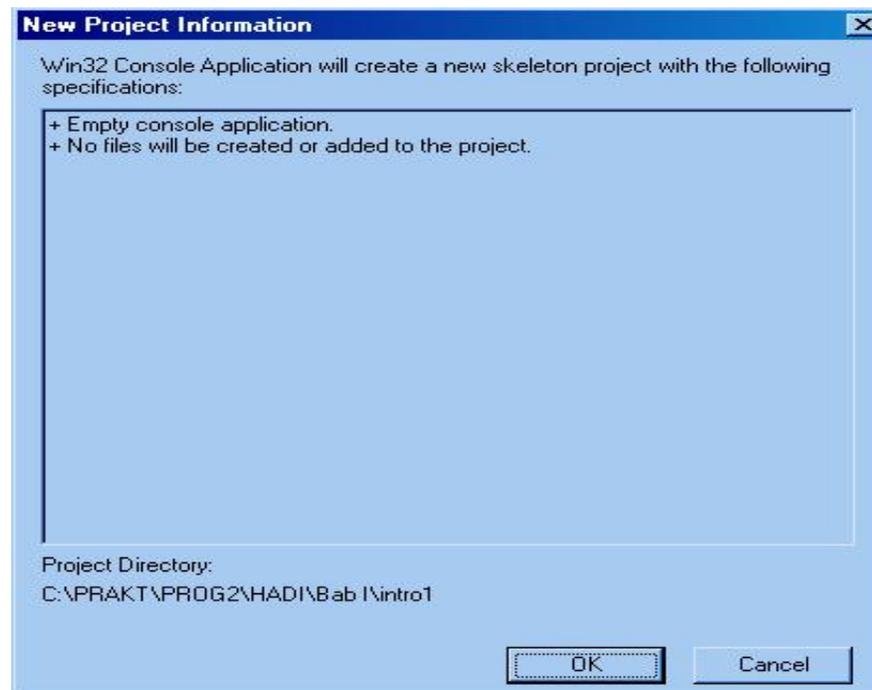
Kotak dialog Step 1 of 1



Membuat Project Baru(5)



Kotak dialog New Project Information



Membuat File Baru

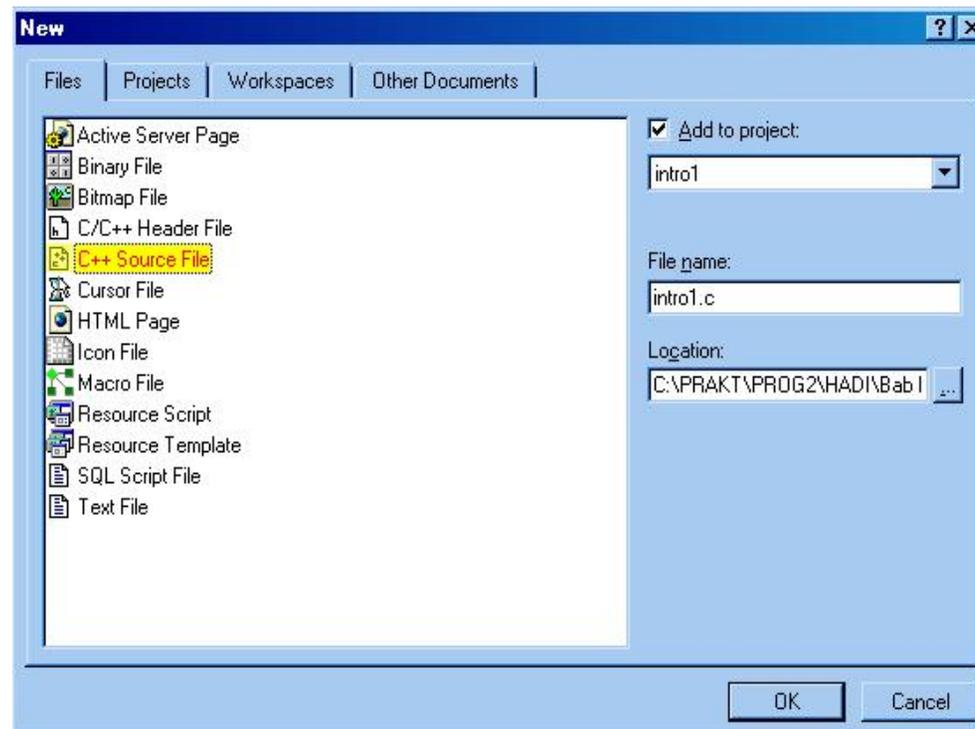


1. Klik **New** pada File menu
2. Dalam kotak dialog New, klik **Files**
3. Klik **C++ Source File**
4. Ketikkan nama file yang dikehendaki, dalam hal ini misalnya : **intro1.c** (jangan lupa untuk selalu menambahkan ekstensi **.c**. Bila user hanya menyetikkan nama file tanpa ekstensi, dalam hal ini akan dianggap sebagai file dengan ekstensi **.cpp** /C++)
5. Pastikan **Add to project** telah terpilih
6. Pastikan *project* tempat file akan diletakkan dan lokasi penyimpanan file (dalam hal ini adalah : **C:\Prakt\progC\7411030001\Bab I\intro1**).
7. Klik **OK**

Membuat File Baru(2)



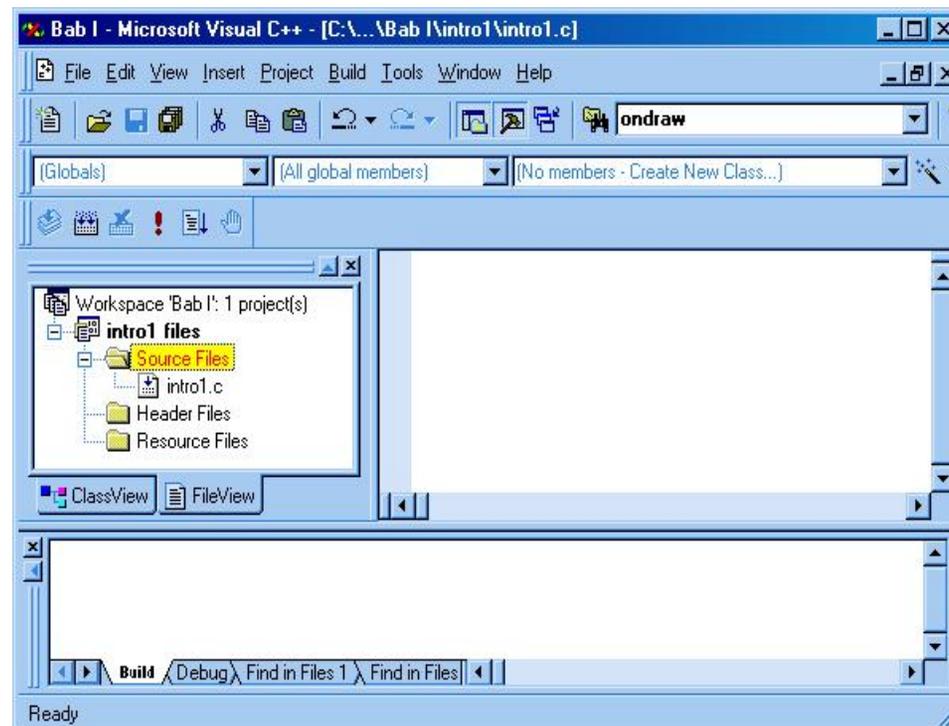
Kotak dialog untuk New Files



Hasilnya.....



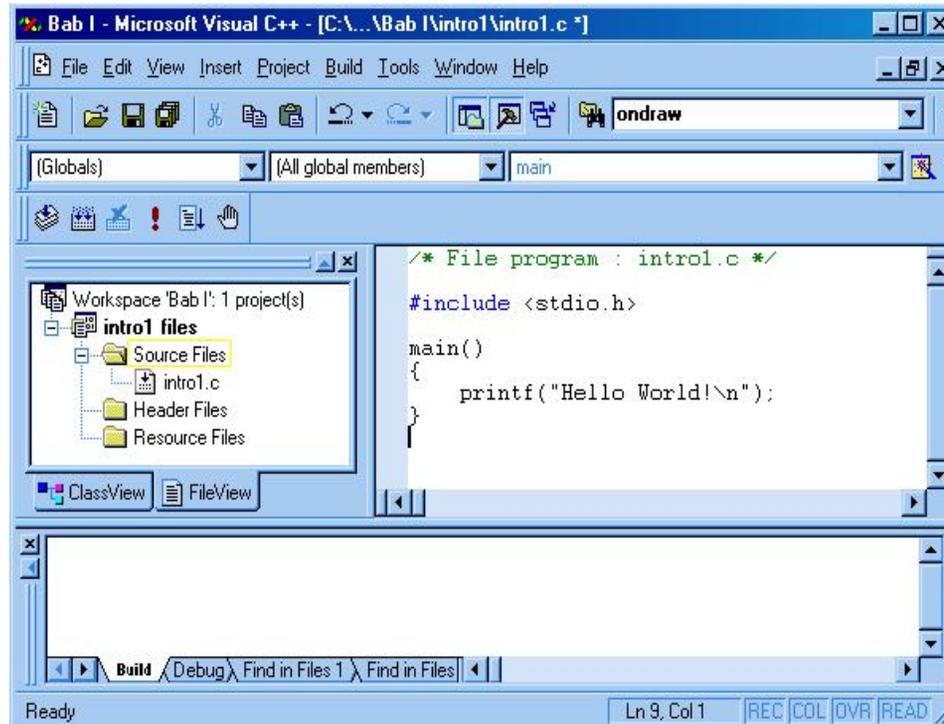
Tampilan jendela workspace setelah proses pembuatan workspace, project dan file baru



Pembuatan Program



Pembuatan source program intro1.c



```
/* File program : intro1.c */
#include <stdio.h>

main()
{
    printf("Hello World!\n");
}
```

coding



```
⌘ #include <stdio.h>
```

```
⌘ main()
```

```
⌘ {
```

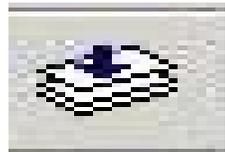
```
⌘     printf(" Hello World! \n");
```

```
⌘ }
```

Proses Kompilasi



1. Klik menu **Build**
2. Klik opsi **Compile intro1.c** atau secara singkat dengan cara menekan **Ctrl+F7** atau tekan icon bergambar

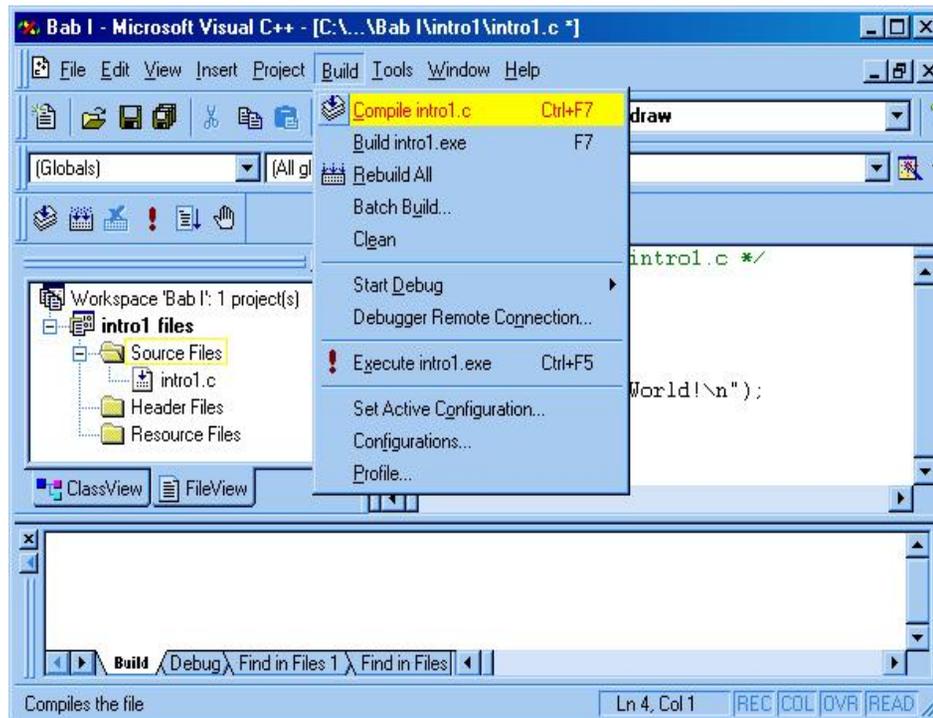


Jika program tersebut tidak mengandung kesalahan sintaks sama sekali, maka proses kompilasi akan memberikan hasil berupa pesan:
intro1.obj – 0 error(s), 0 warning(s)

Proses Kompilasi(2)



Menu pull-down untuk melakukan proses kompilasi



Proses Kompilasi(3)



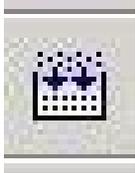
**Tampilan Pesan
ketika proses kompilasi sukses dilakukan**

```
-----Configuration: intro1 - Win32 Debug-----  
Compiling...  
intro1.c  
  
intro1.obj - 0 error(s), 0 warning(s)
```

The screenshot shows a standard Windows-style window with a title bar that reads "-----Configuration: intro1 - Win32 Debug-----". The main text area contains the output of a compilation process: "Compiling...", "intro1.c", and "intro1.obj - 0 error(s), 0 warning(s)". Below the text area is a menu bar with options: "Build", "Debug", "Find in Files 1", "Find in Files 2", and "Results". At the bottom of the window, there is a status bar that says "Ready" on the left and "Ln 6, Col 1" followed by keyboard shortcuts "REC", "COL", "OVR", and "READ" on the right.

Proses Linking dan Eksekusi



- ❖ Setelah program berhasil dikompilasi, agar bisa dieksekusi, program harus di-*link* terlebih dahulu.
- ❖ Proses *linking* dilakukan dilakukan dengan cara:
 1. meng-klik menu **Build intro1.exe** atau tekan tombol **F7** atau tekan icon bergambar 
 2. Jika proses *linking* juga sukses, maka akan didapatkan hasil berupa pesan :
intro1.exe – 0 error(s), 0 warning(s)

Proses Linking dan Eksekusi(2)



Tampilan pesan
ketika proses linking sukses dilakukan

```
-----Configuration: intro1 - Win32 Debug-----  
Linking...  
intro1.exe - 0 error(s), 0 warning(s)
```

The screenshot shows a console window with a blue border. The title bar reads 'Configuration: intro1 - Win32 Debug'. The main text area contains the output of a linking process: 'Linking...' followed by 'intro1.exe - 0 error(s), 0 warning(s)'. At the bottom of the window, there is a status bar with 'Ready' on the left and 'Ln 8, Col 2' followed by keyboard shortcuts 'REC', 'COL', 'OVR', and 'READ' on the right.

Mengeksekusi File



1. Klik menu **Build**
2. Klik opsi **Execute intro1.exe** atau tekan **Ctrl+F5** atau tekan icon bergambar



maka akan didapatkan tampilan sebagaimana terlihat berikut ini.....

Mengeksekusi File(2)



Hasil eksekusi file intro1.exe

```
MS-DOS intro1
Auto
Hello World!
Press any key to continue_
```

Membuka Workspace dan menambahkan project baru



1. Masuk ke Microsoft Visual C++
2. Pilih menu : **File -> Open Workspace**
3. Pilih workspace yg diinginkan (cari file dengan ekstensi **.dsw**), misal : **Bab I.dsw**
4. Ulangi langkah-langkah untuk membuat project baru dan file baru untuk setiap latihan dalam Bab I

coding



```
⌘ /* file program : intro1.c */
```

```
⌘ #include <stdio.h>
```

```
⌘ main()
```

```
⌘ {
```

```
⌘     printf(" Hello World! \n");
```

```
⌘ }
```

Pengenalan Praprosesor

#include



- ⌘ *#include* merupakan salah satu jenis pengarah praprosesor (*preprocessor directive*).
- ⌘ Pengarah praprosesor ini dipakai untuk membaca file yang di antaranya berisi deklarasi fungsi dan definisi konstanta.
- ⌘ Beberapa file judul disediakan dalam C.
- ⌘ File-file ini mempunyai ciri yaitu namanya diakhiri dengan ekstensi **.h**. Misalnya pada program `#include <stdio.h>` menyatakan pada kompiler agar membaca file bernama *stdio.h* saat pelaksanaan kompilasi.
- ⌘ Bentuk umum `#include`:
`#include "namafile"`

Komentar dalam Program



⌘ Untuk keperluan dokumentasi dengan maksud agar program mudah dipahami di suatu saat lain, biasanya pada program disertakan komentar atau keterangan mengenai program. Dalam C, suatu komentar ditulis dengan diawali dengan tanda `/*` dan diakhiri dengan tanda `*/`.

Fungsi *main()*



- ⌘ Fungsi *main()* harus ada pada program, sebab fungsi inilah yang menjadi titik awal dan titik akhir eksekusi program.
- ⌘ Tanda { di awal fungsi menyatakan awal tubuh fungsi dan sekaligus awal eksekusi program, sedangkan tanda } di akhir fungsi merupakan akhir tubuh fungsi dan sekaligus adalah akhir eksekusi program.
- ⌘ Jika program terdiri atas lebih dari satu fungsi, fungsi *main()* biasa ditempatkan pada posisi yang paling atas dalam pendefinisian fungsi.
- ⌘ Hal ini hanya merupakan kebiasaan. Tujuannya untuk memudahkan pencarian terhadap program utama bagi pemrogram. Jadi bukanlah merupakan suatu keharusan.

Fungsi *main()*



⌘ main()

⌘ {

⌘ statemen-statemen;
 fungsi utama

⌘ }

⌘ fungsi_fungsi-_lain()

⌘ {

⌘ statemen-statemen;
 fungsi-fungsi lain yang

⌘ }

pemrogram

ditulis oleh

Fungsi *printf()*



- ⌘ Fungsi *printf()* merupakan fungsi yang umum dipakai untuk menampilkan suatu keluaran pada layar peraga.
- ⌘ Untuk menampilkan tulisan Selamat belajar bahasa C
- ⌘ misalnya, pernyataan yang diperlukan berupa:
- ⌘ `printf("Selamat belajar bahasa C");`
- ⌘ Pernyataan di atas berupa pemanggilan fungsi *printf()* dengan argumen atau parameter berupa string.
- ⌘ Dalam C suatu konstanta string ditulis dengan diawali dan diakhiri tanda petik-ganda (").
- ⌘ Perlu juga diketahui pernyataan dalam C selalu diakhiri dengan tanda titik koma (;).
- ⌘ Tanda titik koma dipakai sebagai tanda pemberhentian sebuah pernyataan dan bukanlah sebagai pemisah antara dua pernyataan.

Fungsi *printf()*



⌘ Tanda \ pada string yang dilewatkan sebagai argumen *printf()* mempunyai makna yang khusus. Tanda ini bisa digunakan untuk menyatakan karakter khusus seperti karakter baris-baru ataupun karakter *backslash* (miring kiri). Jadi karakter seperti \n sebenarnya menyatakan sebuah karakter. Contoh karakter yang ditulis dengan diawali tanda \ adalah:

- ⌘ \ " menyatakan karakter petik-ganda
- ⌘ \\ menyatakan karakter backslash
- ⌘ \t menyatakan karakter tab

Fungsi *printf()*



- ⌘ `printf("string kontrol", daftar argumen);`
- ⌘ dengan string kontrol dapat berupa satu atau sejumlah karakter yang akan ditampilkan ataupun berupa penentu format yang akan mengatur penampilan dari argumen yang terletak pada daftar argumen. Mengenai penentu format di antaranya berupa:
 - ⌘ `%d` untuk menampilkan bilangan bulat (integer)
 - ⌘ `%f` untuk menampilkan bilangan titik-mengambang (pecahan)
 - ⌘ `%c` untuk menampilkan sebuah karakter
 - ⌘ `%s` untuk menampilkan sebuah string

contoh



```
#include <stdio.h>

main()
{
    printf("No    : %d\n",10);
    printf("Nama  : %s\n","Ali");
    printf("Nilai : %f\n",80.5);
    printf("Huruf : %c\n",'A');
}
```

output



```
C:\ "C:\PraktProgC\ira\bab2\intro2\Debug\intro2.exe"
No : 10
Nama : Ali
Nilai : 80.500000
Huruf : A
Press any key to continue
```

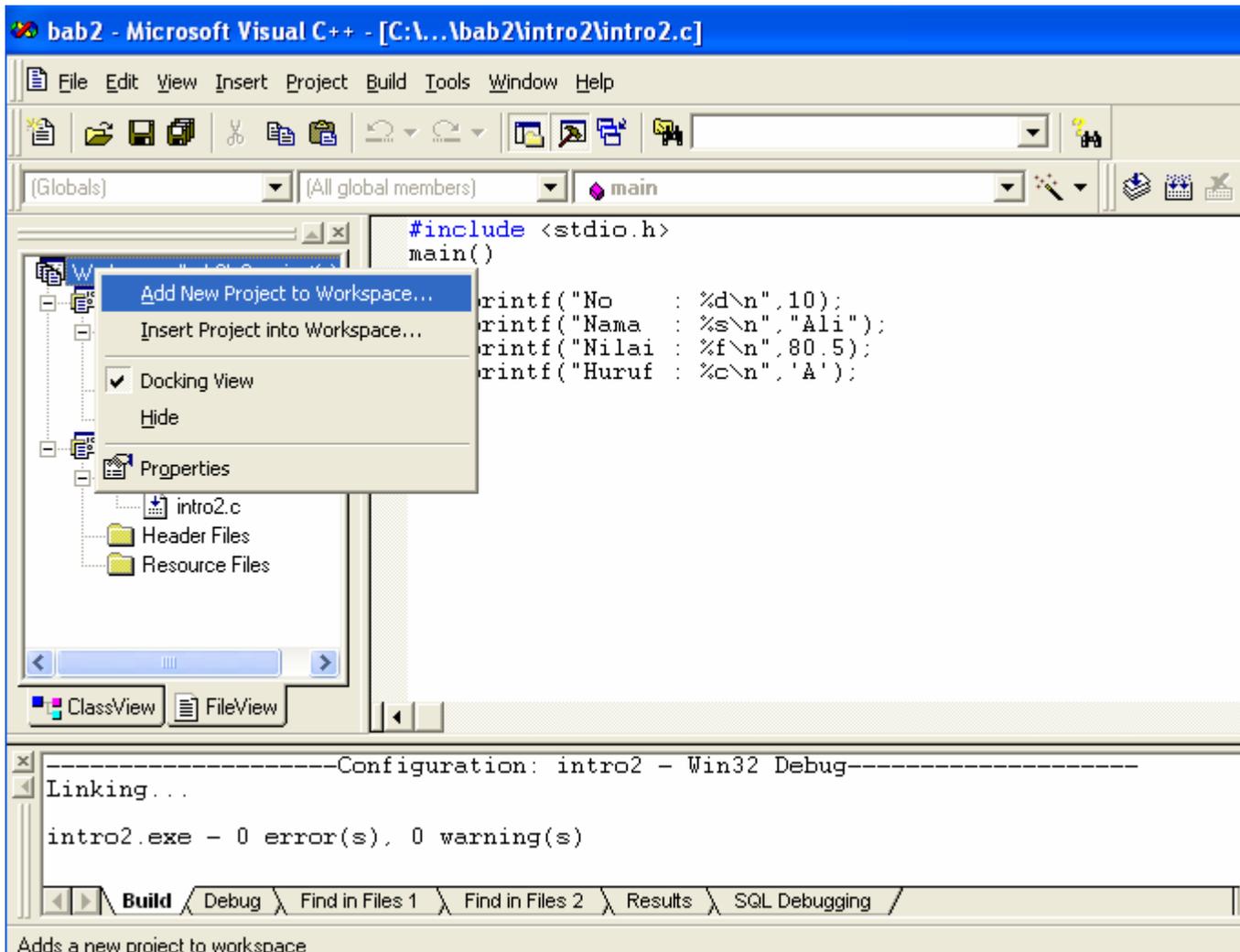
Buat project baru di workspace yang sama



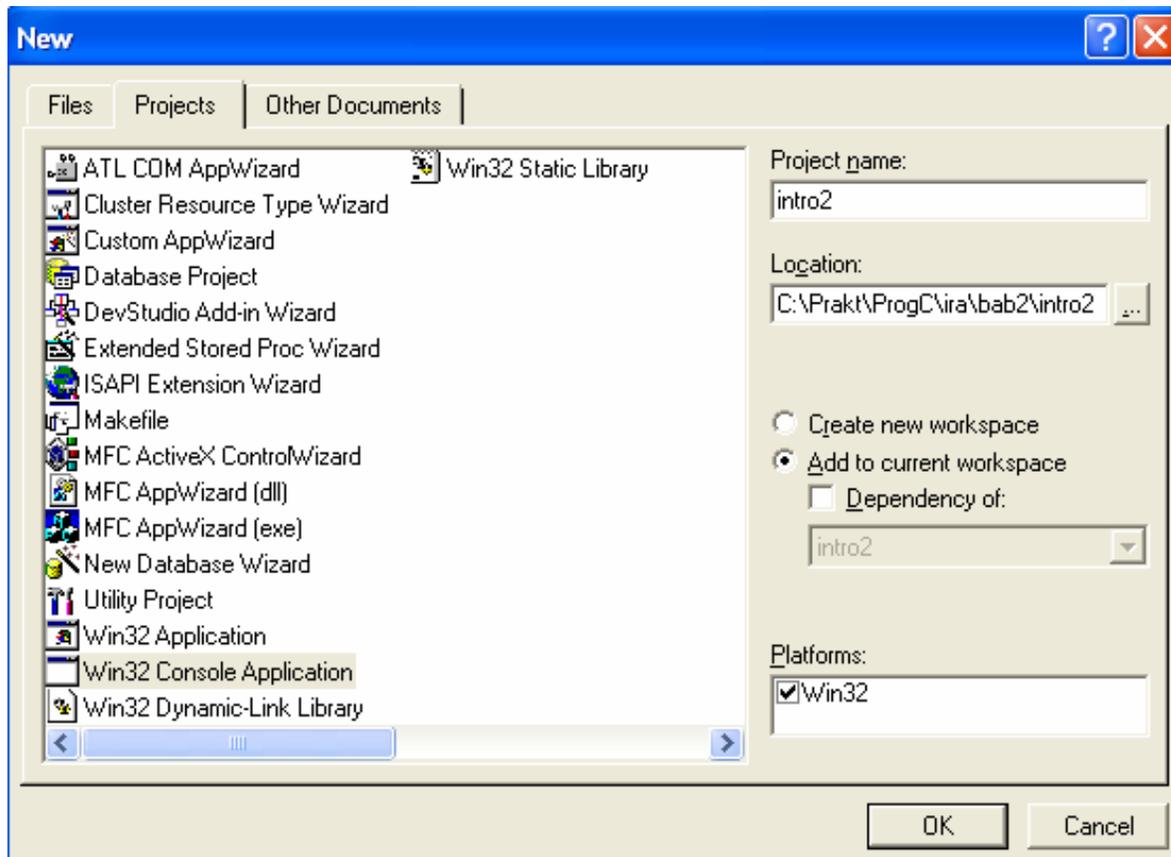
⌘ Nama project : intro2

⌘ Nama file : intro2.c

Add project dalam workspace yang sama



Add project



Add project



Win32 Console Application - Step 1 of 1

What kind of Console Application do you want to create?

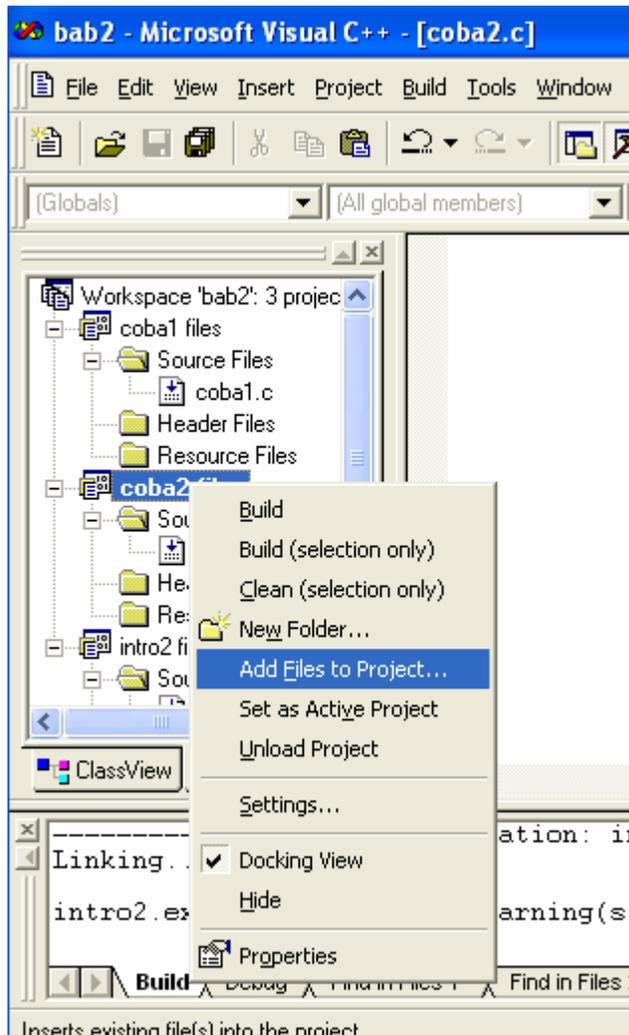
- An empty project.
- A simple application.
- A "Hello, World!" application.
- An application that supports MFC.

Project Directory:
C:\Prakt\ProgC\ira\bab2\coba2

Win32 Console Application will create a new skeleton project with the following specifications:

- + Empty console application.
- + No files will be created or added to the project.

Add file



Add file



The screenshot shows the Microsoft Visual C++ IDE with the 'Insert Files into Project' dialog box open. The dialog box is titled 'Insert Files into Project' and has a 'Look in:' field set to 'coba1'. The file list shows a folder named 'Debug' and a file named 'coba1.c'. The 'File name:' field is set to 'coba2.c'. The 'Files of type:' dropdown is set to 'C++ Files (.c;.cpp;.cxx;.tli;.h;.tlh;.inl;.rc)'. The 'OK' and 'Cancel' buttons are visible. Below the dialog box, the 'ClassView' and 'FileView' tabs are visible. The 'ClassView' tab shows a project structure with three sub-projects: 'coba1 files', 'coba2 files', and 'intro2 files'. The 'coba2 files' folder is selected. The 'FileView' tab shows the 'coba2 files' folder containing 'Source Files', 'Header Files', and 'Resource Files'. The 'Source Files' folder contains 'coba2.c'. The 'intro2 files' folder contains 'Source Files' and 'intro2.c'. The 'intro2.c' file is selected. The 'Linking...' window is open at the bottom, showing the output: 'intro2.exe - 0 error(s), 0 warning(s)'. The status bar at the bottom shows 'Ready' and 'Build'.

Mencetak kalimat dalam beberapa baris.



```
⌘ /* File program : intro2.c */
⌘ #include <stdio.h>
⌘ main()
⌘ {
⌘ printf("Pemrograman adl sesuatu yg
menyenangkan\n");
⌘     printf("Pemrograman bahasa C adl lebih
menyenangkan\n");
⌘     printf("Testing....\n..1\n...2\n....3\n");
⌘ }
```

tugas



- ⌘ Cari materi tentang macam-macam nama file (yang ditulis setelah #include).
- ⌘ Pelajari tentang tipe data.
- ⌘ Kerjakan program-program dislide selanjutnya. Dan buat flowchartnya.

Menghitung penjumlahan dan mencetak hasilnya



```
⌘ /* File program : intro3.c */  
⌘ #include <stdio.h>  
⌘ main()  
⌘ {  
⌘     int jumlah;  
⌘     jumlah = 1024 + 4096;  
⌘     printf("Jumlah dari 1024 dengan 4096 adalah %d\n",  
⌘     jumlah);  
⌘ }
```

Mengisi nilai variabel int, penjumlahan isi variabel dan mencetak hasilnya



```
⌘ /* File program : intro4.c */  
⌘ #include <stdio.h>  
⌘ main()  
⌘ {  
⌘     int nilai1, nilai2, jumlah;  
⌘     nilai1 = 1024;  
⌘     nilai2 = 4096;  
⌘     jumlah = nilai1 + nilai2;  
⌘  
⌘     printf("Jumlah dari %d dengan %d adalah %d\n",  
⌘           nilai1, nilai2, jumlah);  
⌘ }
```

mengisi variabel float, mengalihkan isi variabel dan mencetak hasilnya



```
⌘ /* File program : intro5.c */  
⌘ #include <stdio.h>  
⌘ main()  
⌘ {  
⌘     float harga, total;  
⌘     int jumlah;  
⌘     harga = 132.055f;  
⌘     jumlah = 50;  
⌘     total = harga * jumlah;  
⌘     printf("Harga total = %f\n", total);  
⌘ }
```

Memeriksa hasil cetakan
dengan lebar medan (field)
yang tepat versi 1



```
⌘ /* File program : intro6.c */  
⌘ #include <stdio.h>  
⌘ main()  
⌘ {  
⌘     float bil = 2.5f, nomor = 33.567f;  
⌘     printf("Bilangan = %f\n", bil);  
⌘     printf("Nomor     = %f\n", nomor);  
⌘ }
```

Mencetak hasil cetakan dengan lebar medan (field) yang tepat versi 2



```
⌘ /* File program : intro7.c */  
⌘ #include <stdio.h>  
⌘ main()  
⌘ {  
⌘     float bil = 2.5f, nomor = 33.567f;  
⌘     printf("Bilangan = %10.2f\n", bil);  
⌘     printf("Nomor     = %10.2f\n", nomor);  
⌘ }
```



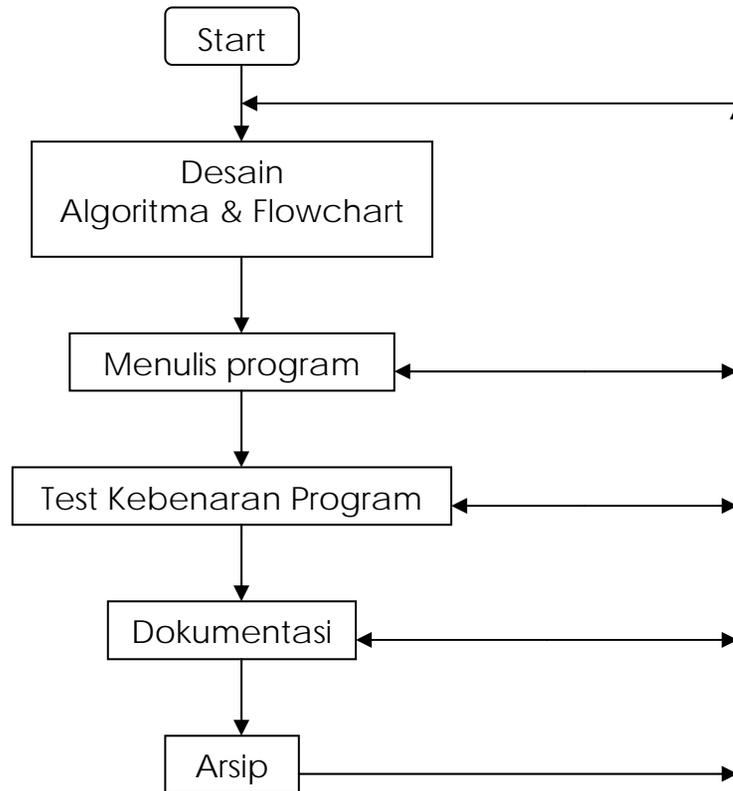
Beberapa langkah dalam proses pembuatan suatu program atau software

Mendefinisikan masalah dan menganalisisnya



- ⌘ Langkah awal yang harus dilakukan adalah mengidentifikasi masalah antara lain tujuan dari pembuatan program, parameter-parameter yang digunakan, fasilitas apa saja yang akan disediakan oleh program. Kemudian menentukan metode atau algoritma apa yang akan diterapkan untuk menyelesaikan masalah tersebut dan terakhir menentukan bahasa program yang digunakan untuk pembuatan program.

Merealisasikan dengan langkah-langkah berikut :

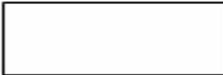


Algoritma



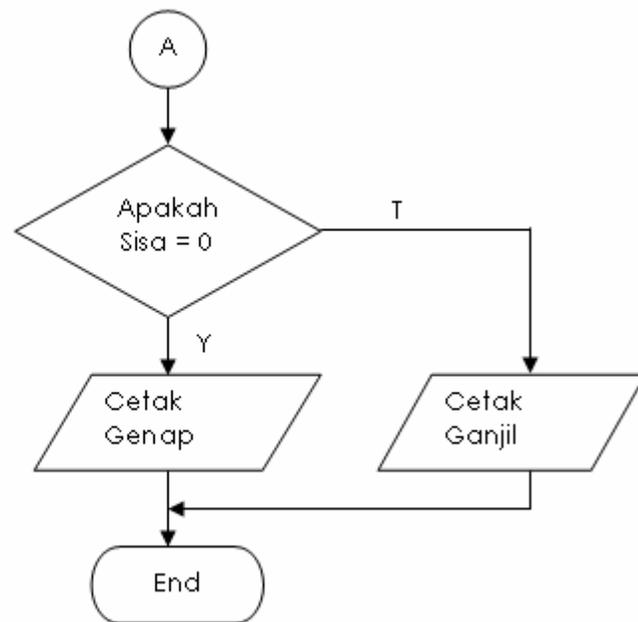
- ⌘ Algoritma adalah urutan langkah-langkah logika yang menyatakan suatu tugas dalam menyelesaikan suatu masalah atau problem.
- ⌘ Contoh : Buat algoritma untuk menentukan apakah suatu bilangan merupakan bilangan ganjil atau bilangan genap.
- ⌘ Algoritmanya :
 - ⌘ Masukkan sebuah bilangan sembarang
 - ⌘ Bagi bilangan tersebut dengan bilangan 2
 - ⌘ Hitung sisa hasil bagi pada langkah 2.
 - ⌘ Bila sisa hasil bagi sama dengan 0 maka bilangan itu adalah bilangan genap tetapi bila sisa hasil bagi sama dengan 1 maka bilangan itu adalah bilangan ganjil.



SIMBOL	NAMA	FUNGSI
	TERMINATOR	Permulaan/akhir program
	GARIS ALIR (FLOW LINE)	Arah aliran program
	PREPARATION	Proses inisialisasi/ pemberian harga awal
	PROSES	Proses perhitungan/ proses pengolahan data
	INPUT/OUTPUT DATA	Proses input/output data, parameter, informasi
	PREDEFINED PROCESS (SUB PROGRAM)	Permulaan sub program/ proses menjalankan sub program
	DECISION	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya



SIMBOL	NAMA	FUNGSI
	ON PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	OFF PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda





Selamat !!!

**Anda telah berhasil
membuat program
di lingkungan Visual C++**