

**PRAKTIKUM PEMROGRAMAN KOMPUTER
MINGGU 1
Bahasa C**

A. Tujuan praktikum:

- Mahasiswa mengetahui lingkungan dasar pemrograman turbo C.
- Mahasiswa mampu meng-*compile* dan *run* program dalam bahasa C
- Mahasiswa mampu melakukan debug program

B. Dasar Teori

B.1. Apa itu bahasa C ?

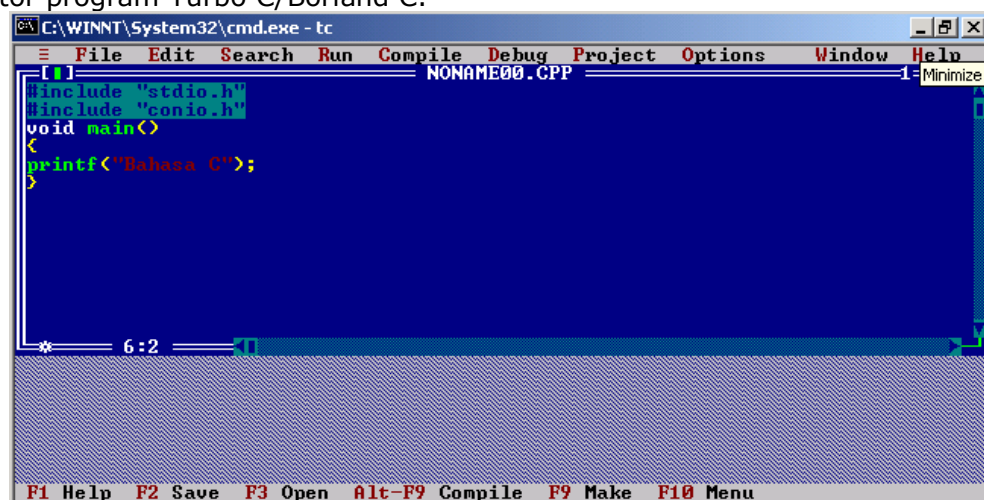
C merupakan bahasa pemrograman yang berkembang dari versi awal yang disebut B. B sendiri merupakan versi perkembangan dari BPCL (Basic Combined Programming Language). Meskipun awalnya C dikembangkan untuk membantu programmer dalam mengelola *hardware*, tetapi C menjadi populer karena kecepatan dan *portability* nya. C merupakan bahasa pemrograman procedural yang kemudian berkembang menjadi bahasa pemrograman berorientasi obyek. Tahun 1981 Bjarne Stroustrup mengembangkan C++ dan dilanjutkan hingga sekarang menjadi Visual C#.

B.2. Compiler

Sebuah *compiler* akan menerjemahkan *source code* menjadi bentuk file sementara. Langkah ini disebut *compiling* dan proses *compile* ini akan menghasilkan sebuah file *object*. Program *linking* (disebut *linker*) akan mengkombinasikan file *object* menjadi file *executable*. File *executable* berisi kode mesin yang dapat dijalankan oleh system operasi tanpa harus memiliki bahasa pemrograman C.

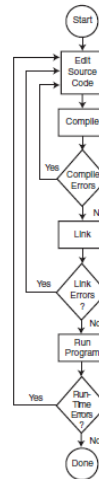
B.3. Membuat program bahasa C

Kita dapat membuat program bahasa C dengan menuliskan statemen perintah melalui NotePad atau program pengolah kata lain kemudian disimpan dengan extension file .c atau .cpp. Tetapi kita juga dapat menuliskan statemen perintah di editor program Turbo C/Borland C.



Gambar 3.1. Editor Turbo C

Setelah program ditulis, maka kita dapat melakukan proses *compiling* (dengan perintah **Compile**). Dalam proses compile ini akan dihasilkan file bertipe .obj. Jika saat proses compile terjadi kesalahan program, maka akan dimunculkan baris program yang salah. Jika program sudah diperbaiki, maka lakukan proses *linking*. Dan terakhir lakukan **Run** untuk menjalankan program yang telah dibuat. Gambar 3.2. menjelaskan langkah-langkah untuk menjalankan program. Dalam Turbo C (yang akan dipakai praktikum), kita bisa langsung melakukan perintah Run saja untuk proses *compiling*, *linking* dan *run* secara berurutan



Gambar 3.2. Langkah menjalankan program

B.4. Struktur program bahasa C

Program dalam bahasa C terdiri dari :

1. Header file
Berisi *preprocessor directive* yang akan memanggil file program bahasa C yang mempersiapkan program sehingga program kita dapat menjalankan perintah-perintah. Misal untuk menjalankan perintah `cout` kita membutuhkan `iostream.h`
2. Function `main()`
Setiap bahasa C harus memiliki `main()` karena program akan mulai dijalankan dari function ini.
3. Contoh program sederhana untuk mencetak "Bahasa C" ke layar.

```

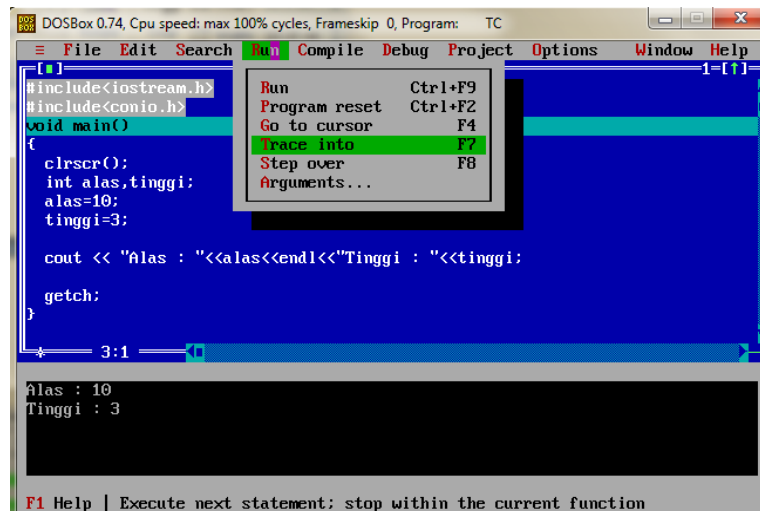
#include <iostream.h>
void main()
{
    cout << "Bahasa C\n";
}

```

B.5. Debugging

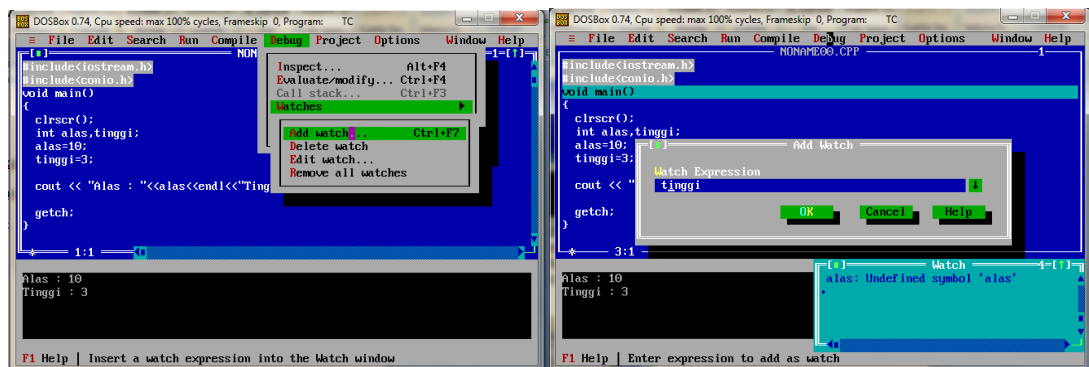
Kita dapat melakukan debugging yang berfungsi untuk merunut program sehingga mengetahui kesalahan pada program.

Cara untuk melakukan debugging adalah dengan menekan Run=> Trace Into (atau tekan F7)



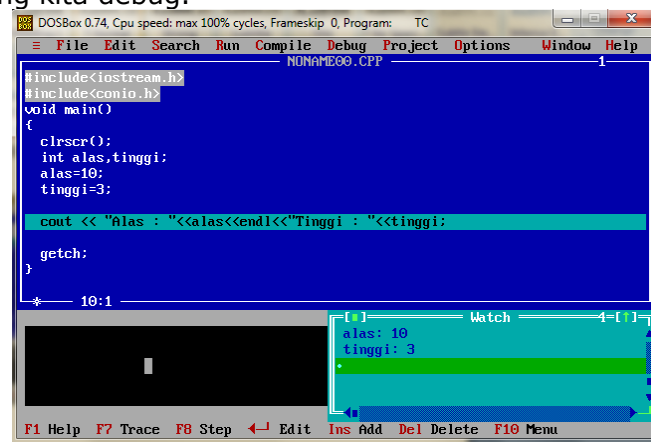
Gambar 3.3. Langkah Trace

Dengan setiap kali menekan F7, maka langkah demi langkah program dijalankan ditandai dengan baris berwarna hijau. Pada saat melakukan tracing program kita dapat mengevaluasi setiap variable program dengan menggunakan Debug => Watches => Add Watch



Gambar 3.4. Langkah untuk melakukan watch

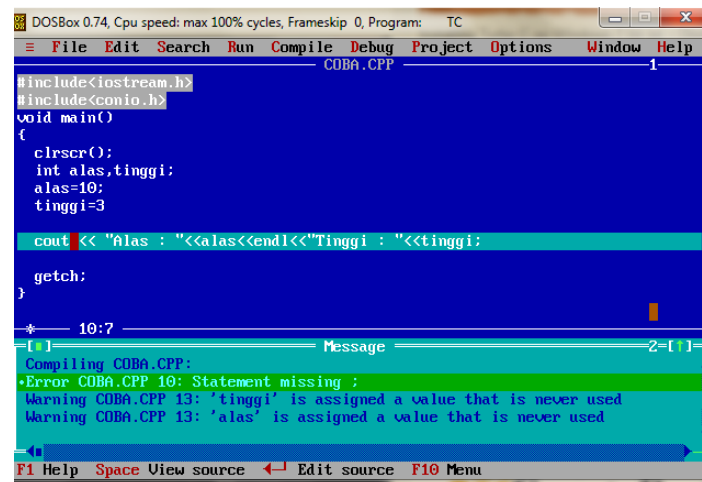
Sehingga saat tracing program dijalankan setelah melewati variable akan terlihat nilai variable yang kita debug.



Gambar 3.5. Langkah untuk melihat hasil dari watch

Jika kita melakukan kesalahan pada koding program kita, juga akan terlihat letak kesalahan program tersebut. Kesalahan dapat terjadi pada baris yang ditunjuk

atau baris sebelumnya. Seperti pada gambar 3.6 kesalahan terjadi bukan pada baris 10 seperti yang ditunjukkan, tetapi pada baris sebelum yaitu baris 8.



Gambar 3.6. Contoh melihat kesalahan program

C. Tugas I

- I. Ketiklah program berikut dalam layar Turbo C.
 - II. Simpanlah program dengan menggunakan perintah File => Save
 - III. Compile program dengan menggunakan perintah Compile => Compile
 - IV. Jalankan program dengan menggunakan perintah Run => Run
 - V. Amati apakah program sudah dapat dijalankan dengan benar.
 - VI. Amati hasil pada layar output. Jika layar output tidak ada, munculkan dengan menu Window => Output.
1.

```
#include <iostream.h>
int main()
{
    cout<<"Tampil hasilnya";
}
```
 2.

```
#include <iostream.h>
int main()
{
    cout<<"Tampil\n hasilnya";
}
```
 3.

```
#include <iostream.h>
void main()
{
    cout<<"Tampil \n hasilnya";
}
```
 4.

```
#include <iostream.h>
#include<conio.h>
void main()
{
    cout<<"Tampil \n hasilnya";
    getch();
}
```

5.

```
#include <iostream.h>
#include<conio.h>
void main()
{
    clrscr();
    cout<<"Tampil \n hasilnya";
    getch();
}
```
6.

```
#include <iostream.h>
#include<conio.h>
void main()
{
    clrscr();
    cout<<"Tampil \n"<<" hasilnya";
    getch()
}
```
7.

```
#include <iostream.h>
#include<conio.h>
void main()
{
    clrscr();
    int satu=1;
    cout<<"Tampil\\n"<<satu;
    getch();
}
```
8.

```
#include <iostream.h>
#include<conio.h>
void main()
{
    clrscr();
    //ini dipakai untuk keterangan satu baris
    int satu=1;
    /* dipakai untuk keterangan
    lebih dari satu baris*/
    cout<<"Tampil\\n"<<satu;
    getch();
}
```
9.

```
#include <iostream.h>
#include <conio.h>
void main()
{
    clrscr();
    int alas, tinggi;
    alas = 10;
    tinggi = 3;

    cout << "Alas : " << alas << endl <<"Tinggi : " << tinggi;

    getch();
}
```

10.

```
#include <iostream.h>
#include <conio.h>
void main()
{
    clrscr();
    int alas, tinggi, luas;
    alas = 10;
    tinggi = 3;

    luas=0.5*alas*tinggi;
    cout << "Alas : " << alas << endl <<"Tinggi : " << tinggi;
    cout << "\nLuas adalah : " << luas;

    getch();
}
```
11.

```
#include <iostream.h>
#include <conio.h>

void main()
{
    clrscr();
    int alas, tinggi;
    alas = 10;
    tinggi = 3;

    cout << "\nLuas segitiga dengan alas : "<< alas<<" dan tinggi :
"<<tinggi << " adalah : " << 0.5*alas*tinggi;

    getch();
}
```
12.

```
#include <iostream.h>
#include <conio.h>

void main()
{
    clrscr();
    float bil_1, bil_2;
    int bil_3;
    bil_1=2.3;
    bil_2=3.0;
    bil_3=2.3;
    cout << "Bilangan real pertama adalah : "<< bil_1;
    cout << "\nBilangan real kedua adalah : "<< bil_2;
    cout << "\nBilangan real ketiga adalah : "<< bil_3; //jelaskan
    hasilnya
    getch();
}
```
13.

```
#include <iostream.h>
#include <conio.h>
void main()
{
    clrscr();
    char karakter;
    karakter = 'a';
```

```

        cout << "Tipe data karakter hanya dapat menyimpan satu huruf
saja"<<endl<<"Contoh karakter adalah : "<<karakter;

```

```

        getch();
    }

```

14.

```

#include <iostream.h>
#include <conio.h>
void main()
{
    clrscr();
    float bil_1,bil_2;
    bil_1=2.3;
    bil_2=3.0;
    cout << "'bil_1 \' * 'bil_2\' adalah : "<< bil_1*bil_2;
    getch();
}

```
15.

```

#include <iostream.h>
#include <conio.h>
void main()
{
    clrscr();
    float bil_1,hasil_2;
    int bil_2, hasil_1;
    bil_1=2.7;
    bil_2=2.2;
    hasil_1=bil_1*bil_2;
    hasil_2=bil_1*bil_2;
    cout << "Hasil perkalian float ("<<bil_1<<") dan int
("<<bil_2<<") disimpan dalam variabel int "<< hasil_1<<endl;
    cout << "Hasil perkalian float ("<<bil_1<<") dan int
("<<bil_2<<") disimpan dalam variabel float "<< hasil_2;
    getch();
}

```
16.

```

#include <iostream.h>
#include <conio.h>

void main()
{
    clrscr();
    int bil_1,bil_2;

    cout << "Masukkan bilangan 1 : ";
    cin >> bil_1;
    cout << "Masukkan bilangan 2 : ";
    cin >> bil_2;
    cout<< "Bilangan yang anda masukkan adalah "<<bil_1<<" dan
"<<bil_2;
    getch();
}

```
17.

```

#include <iostream.h>
#include <conio.h>
void main()
{

```

