



Input Data di C

Umumnya suatu program mempunyai proses pemasukan data. Dalam program dengan bahasa C, pemasukan data dapat dilakukan dengan perintah `scanf`. Fungsi **`scanf`** merupakan fungsi yang dapat digunakan untuk memasukan berbagai jenis data, tergantung dengan format penentunya. Format-format penentu tipe data yang umum dipakai adalah :

Tabel 4.1 Format umum tipe data di C

| Format | Kegunaan |
|--------------------------------------|---|
| <code>%c</code> | Digunakan untuk pemasukan data bertipe char |
| <code>%i</code> atau <code>%d</code> | Digunakan untuk pemasukan data bertipe int, char. |
| <code>%f</code> | Digunakan untuk pemasukan data berupa bilangan pecahan |
| <code>%u</code> | Digunakan untuk pemasukan data berupa unsigned int atau unsigned char |
| <code>%s</code> | Digunakan untuk pemasukan data berupa string |

Bentuk umum penggunaan fungsi **`scanf`** adalah :

```
scanf("format",&nama_variabel);  
    dan untuk string  
scanf("format",nama_variabel);
```

Contoh Program :

```
1  /*  
2   Program 4-1  
3   Nama File : Program 4-1.c  
4   Programmer : Eko Budi Setiawan  
5  */  
6  
7  #include <stdio.h>  
8  #include <stdlib.h>  
9  
10 int main(int argc, char *argv[])
```

```

11 {
12     int a,jam,menit,detik;
13     unsigned int b;
14     float c;
15     char nama[50];
16     scanf("%i",&a);
17     scanf("%u",&b);
18     scanf("%f",&c);
19     scanf("%i %i %d",&jam,&menit,&detik);
20     fflush(stdin);scanf("%s",nama);
21     system("PAUSE");
22     return 0;
23 }

```

Fungsi **scanf()** kurang begitu bagus jika dipakai untuk pembacaan string. Karena data yang tersimpan adalah hanya sampai menemukan spasi, maksudnya jika kita mengisi 2 buah kata dengan pemisah spasi, maka data yang masuk ke variabel tersebut hanyalah kata yang pertama. Oleh karena itu, pembacaan data bertipe string biasanya menggunakan perintah **gets()** yang bentuk umumnya adalah :

```
gets(nama_variabel);
```

Contoh :

```

gets(nama);
gets(alamat);

```

Selain fungsi **scanf()** dan fungsi **gets()**, ada fungsi yang bisa digunakan untuk menginputkan data karakter. Fungsinya antara lain:

1. **getch()**

Membaca tipe karakter tanpa menampilkan karakter tersebut ke layar.

2. **getche()**

Membaca tipe karakter dengan menampilkan karakter yang diinputkan ke layar.

Contoh Program :

```

1  /*
2   Program 4-2
3   Nama File : Program 4-2.c
4   Programmer : Eko Budi Setiawan
5  */

```

```

6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int main(int argc, char *argv[])
11 {
12     char a,b;
13     printf("Masukan Huruf pertama : ");
14     a=getch();
15     printf("\nMasukan Huruf kedua : ");
16     b=getche();
17     printf("\nData yang dimasukan adalah %c dan %c\n",a,b);
18     getch();
19     return 0;
20 }

```

Hasil eksekusi program diatas adalah sebagai berikut :



Gambar 4.1 Tampilan hasil eksekusi program 4-1.c

Input Data di C++

Fungsi pada bahasa C++ yang digunakan untuk menginputkan data adalah fungsi cin. Bentuk umum dari fungsi cin adalah sebagai berikut:

```
cin >> nama_variabel;
```

Untuk menggunakan fungsi ini, kita tidak perlu memasukkan format tipe data dari variabel yang digunakan. Tentunya hal ini akan memudahkan bagi pengguna fungsi tersebut.

Contoh Program :

```

1  /*
2   Program 4-1
3   Nama File : Program 4-1.cpp
4   Programmer : Eko Budi Setiawan
5  */
6
7  using namespace std;

```

```

8  int main(int argc, char *argv[])
9  {
10     int a,jam,menit,detik;
11     unsigned int b;
12     float c;
13     char nama[50];
14     cin>>a;
15     cin>>b;
16     cin>>c;
17     cin>>jam>>menit>>detik;
18     cin>>nama;
19     system("PAUSE");
20     return EXIT_SUCCESS;
21 }

```

Sama halnya dengan **scanf()**, fungsi cin hanya bisa menyimpan data string sampai menemukan spasi saja. Apabila ingin menginputkan data string yang mengandung spasi maka fungsi yang kita gunakan adalah fungsi cin.get. bentuk umum dari fungsi cin.get adalah sebagai berikut:

```
cin.get(nama_variabel,jumlah_karater_input);
```

Output Data di C

Untuk output data, perintah yang bisa dipakai adalah **printf()**. Untuk menampilkan data dengan fungsi **printf()**, kita harus mengatur format tampilannya, dengan format-format penentu.

Untuk menentukan panjang medan yang disediakan bagi tampilan data, maka sesudah tanda % dalam penentu format dapat disisipi dengan bilangan bulat yang menyatakan panjang medan.

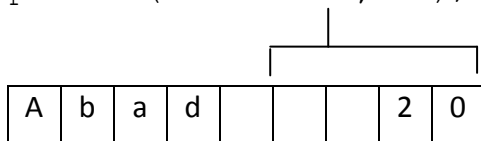
Untuk data yang berupa bilangan bulat, misalnya pada :

```
printf ("Abad %4d", 20);
```

Ket :

%4d menyatakan medan untuk menampilkan bilangan 20 adalah sepanjang 4 karakter.

```
printf ("Abad %4d", 20);
```



Untuk data yang berupa bilangan real, spesifikasi medannya berupa

m.n

Ket :

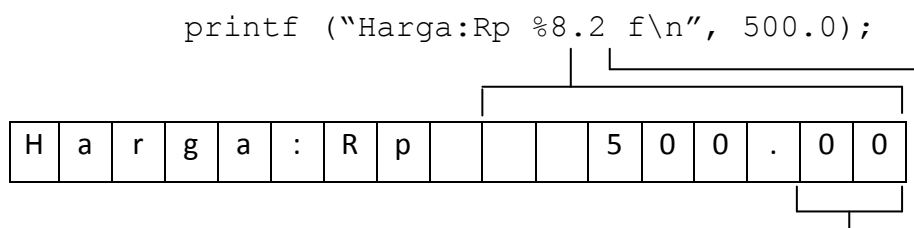
m = panjang medan

n = jumlah digit pecahan

Contoh :

```
printf ("Harga:Rp %8.2 f\n", 500.0);
```

%8.2f menyatakan panjang medan dari bilangan real yang akan ditampilkan adalah 8 karakter dengan jumlah digit pecahan 2 buah.



Jika hanya jumlah digit pecahan yang perlu ditentukan, panjang medan tak perlu disertakan, misal :

```
printf ("%2f\n", 600.0);
```

```
printf ("%2f\n", 7500.25);
```

hasilnya :

600.00

7500.25

Untuk data yang berupa string, contoh :

```
printf ("%12s", "Bahasa C");
```

maka akan ditampilkan sebagai berikut

| | | | | | | | | | | | |
|--|--|--|--|---|---|---|---|---|---|--|---|
| | | | | B | a | h | a | s | a | | C |
|--|--|--|--|---|---|---|---|---|---|--|---|

Tampak dalam berbagai jenis data diatas, penentu format yang mengandung panjang medan, secara default akan menampilkan data dalam bentuk rata kanan terhadap panjang medan yang diberikan. Untuk data string yang biasanya dikehendaki untuk ditampilkan dalam

bentuk rata kiri, maka sesudah tanda % pada penentu format %s perlu disisipkan tanda – (minus), contoh :

```
printf ("% -12s", "Bahasa C");
```

menyatakan bahwa string akan ditampilkan dalam medan dengan panjang 12 karakter dan diatur rata kiri. Sehingga tampilan diatas berubah menjadi :

| | | | | | | | | | | | |
|---|---|---|---|---|---|--|---|--|--|--|--|
| B | a | h | a | s | a | | C | | | | |
|---|---|---|---|---|---|--|---|--|--|--|--|

Contoh Program ;

```
1  /*
2   Program 4-3
3   Nama File : Program 4-3.c
4   Programmer : Eko Budi Setiawan
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int main(int argc, char *argv[])
11 {
12     int a=25000;
13     unsigned int b=50000;
14     float c=12345.678;
15     char nama[50]="Universitas Komputer Indonesia";
16     char alamat[10]="Bandung";
17     printf("Penampilan data tanpa di format\n");
18     printf("Nilai a : %d\n",a);
19     printf("Nilai b : %u\n",b);
20     printf("Nilai c : %f\n",c);
21     printf("String : %s %s\n",nama,alamat);//rata kanan
22     printf("\nPenampilan data setelah di format\n");
23     printf("\n");
24     printf("Nilai a : %8d\n",a);
25     printf("Nilai b : %8u\n",b);
26     printf("Nilai c : %11.2f\n",c);
27     printf("String 1: %40s %10s\n",nama,alamat);//rata kanan
28     printf("String 2: %-40s %-10s\n",nama,alamat);//rata kiri
29     getch();
30     return 0;
31 }
```

Maka hasil eksekusi dari program diatas adalah sebagai berikut :



```
Penampilan data tanpa di format
Nilai a : 25000
Nilai b : 50000
Nilai c : 12345.677734
String : Universitas Komputer Indonesia Bandung

Penampilan data setelah di format
Nilai a : 25000
Nilai b : 50000
Nilai c : 12345.68
String 1: Universitas Komputer Indonesia Bandung
String 2: Universitas Komputer Indonesia Bandung
```

Gambar 4.2 Tampilan Program 4-3.c

Output Data di C++

Untuk output data dalam bahasa C++ digunakan fungsi cout. Bentuk umum dari fungsi cout adalah sebagai berikut:

```
cout<<nama_variabel;
```

Contoh Program :

```
1  /*
2   Program 4-3
3   Nama File : Program 4-3.cpp
4   Programmer : Eko Budi Setiawan
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9  #include <iomanip>
10
11 using namespace std;
12
13 int main(int argc, char *argv[])
14 {
15     int x=25000;
16     unsigned int y=5000;
17     float z=12345.678;
18     char univ[35]="UNIVERSITAS KOMPUTER INDONESIA";
19     char kota[10]="BANDUNG";
20     system("cls"); //untuk mengosongkan layar
21     cout<<"Tampilan Data Tanpa Diformat"<<endl;
22     cout<<"Nilai x : "<<x<<endl;
23     cout<<"Nilai y : "<<y<<endl;
```

```

24     cout<<"Nilai z : "<<z<<endl;
25     cout<<"String : "<<univ<<" "<<kota<<"\n\n";
26     cout<<"Tampilan Data Sudah Diformat"<<endl;
27     cout<<"Nilai x : "<<setw(10)<<x<<endl;
28     cout<<"Nilai y : "<<setw(10)<<y<<endl;
29     cout<<"Nilai z : "<<setw(10)<<z<<endl;
30     cout<<"String 1: "<<setw(40)<<univ
31         <<" "<<setw(10)<<kota<<endl;
32     cout<<"String 2: "<<setw(-40)<<univ<<" "
33         <<setw(-10)<<kota<<"\n\n";
34     system("PAUSE");
35     return EXIT_SUCCESS;
36 }

```

Maka hasil eksekusi programnya adalah sebagai berikut :

```

E:\Eko Budi Setiawan\Dosen @UNIKOM\Semester Ganjil 2010-2011\SEMESTER 3\Praktikum Pemro...
Tampilan Data Tanpa Diformat
Nilai x :25000
Nilai y :5000
Nilai z :12345.7
String :UNIVERSITAS KOMPUTER INDONESIA BANDUNG

Tampilan Data Sudah Diformat
Nilai x :      25000
Nilai y :      5000
Nilai z :    12345.7
String 1:          UNIVERSITAS KOMPUTER INDONESIA      BANDUNG
String 2:UNIVERSITAS KOMPUTER INDONESIA BANDUNG

Press any key to continue . . .

```

Gambar 4.3 Tampilan hasil eksekusi

Note



Apabila nilai pada fungsi setw berupa angka positif maka tampilan teks akan rata kanan. Sebaliknya, apabila nilai pada fungsi setw berupa angka negatif maka tampilan teks akan rata kiri.

Contoh Program

Contoh kasus :

Di suatu perusahaan, data penggajian dihitung dengan ketentuan sebagai berikut :

Gaji Pokok : Rp. 5000000
Gaji Lembur/jam : Rp. 5000
Total Gaji Lembur : Lama Lembur * Gaji Lembur/jam
Gaji Kotor : Gaji Pokok + Total Gaji Lembur
Pajak : 10% * Gaji Kotor
Gaji Bersih : Gaji Kotor - Pajak

Data yang diinputkan adalah : Nama Pegawai, Lama Lembur.

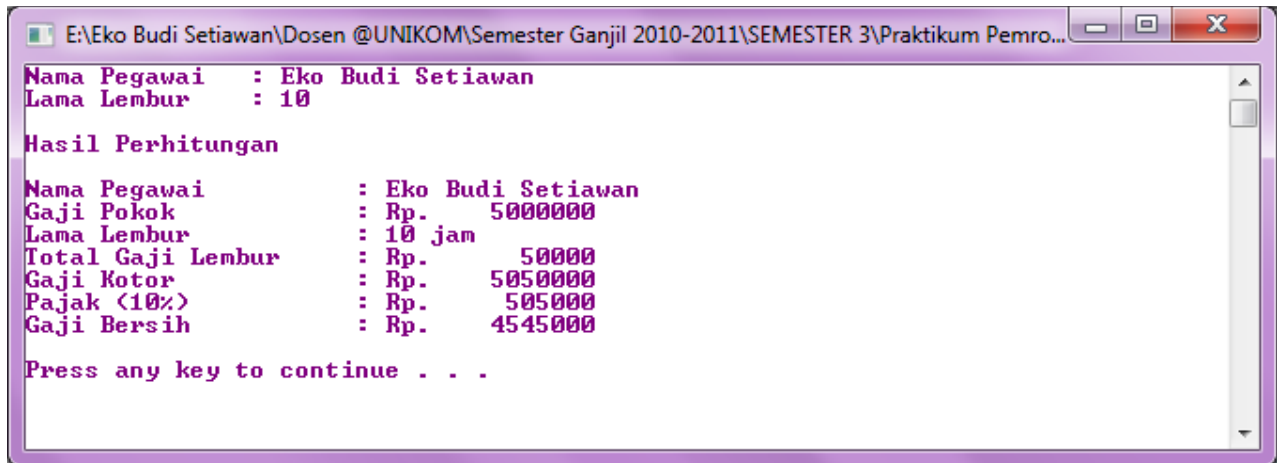
Penyelesaian :

```
1  /*
2   Program 4-4
3   Nama File : Program 4-4.c
4   Programmer : Eko Budi Setiawan
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int main(int argc, char *argv[])
11 {
12     int jamlembur;
13     long int gajipokok=5000000,gajikotor,totalgajilembur;
14     float pajak,gajibersih;
15     char nama[50];
16     printf("Nama Pegawai  : ");gets(nama);
17     printf("Lama Lembur   : ");scanf("%i",&jamlembur);
18     totalgajilembur=(long int)5000*jamlembur;
19     gajikotor=gajipokok+totalgajilembur;
20     pajak=0.1*gajikotor;
21     gajibersih=gajikotor-pajak;
22     printf("\nHasil Perhitungan\n\n");
23     printf("Nama Pegawai      : %s\n",nama);
24     printf("Gaji Pokok         : Rp. %10li\n",gajipokok);
25     printf("Lama Lembur        : %i jam\n",jamlembur);
26     printf("Total Gaji Lembur   : Rp. %10li\n",totalgajilembur);
27     printf("Gaji Kotor          : Rp. %10li\n",gajikotor);
28     printf("Pajak (10%%)        : Rp. %10.0f\n",pajak);
29     printf("Gaji Bersih         : Rp. %10.0f\n",gajibersih);
30     printf("\n");
31     system("PAUSE");
```

| | |
|----|-----------|
| 32 | return 0; |
| 33 | } |

| | |
|----|---|
| 1 | /* |
| 2 | Program 4-4 |
| 3 | Nama File : Program 4-4.cpp |
| 4 | Programmer : Eko Budi Setiawan |
| 5 | */ |
| 6 | |
| 7 | #include <cstdlib> |
| 8 | #include <iostream> |
| 9 | #include <iomanip> |
| 10 | |
| 11 | using namespace std; |
| 12 | |
| 13 | int main(int argc, char *argv[]) |
| 14 | { |
| 15 | int jam_lembur; |
| 16 | long int gaji_pokok=5000000,gaji_kotor, total_gaji_lembur; |
| 17 | float pajak,gaji_bersih; |
| 18 | char nama[31]; |
| 19 | system("cls"); |
| 20 | cout<<"Nama Pegawai : ";cin.get(nama,30); |
| 21 | cout<<"Lama Lembur : ";cin>>jam_lembur; |
| 22 | total_gaji_lembur=(long int)5000*jam_lembur; //konversi tipe data |
| 23 | gaji_kotor=gaji_pokok+total_gaji_lembur; |
| 24 | pajak=0.1*gaji_kotor; |
| 25 | gaji_bersih=gaji_kotor-pajak; |
| 26 | system("cls"); |
| 27 | cout<<"Hasil Perhitungan"<<endl; |
| 28 | cout<<"Nama Pegawai : "<<nama<<endl; |
| 29 | cout<<"Gaji Pokok : " <<gaji_pokok<<endl; |
| 30 | cout<<"Lama Lembur : "<<jam_lembur <<" jam"<<endl; |
| 31 | cout<<"Total Gaji Lembur : Rp. " <<setw(10)<<total_gaji_lembur<<endl; |
| 32 | cout<<"Gaji Kotor : Rp. " <<setw(10)<<gaji_kotor<<endl; |
| 33 | cout<<setiosflags(ios::fixed); //mengatur bilangan pecahan |
| 34 | cout<<"Pajak (10%) : Rp. " <<setw(10)<<setprecision(2)<<pajak<<endl; |
| 35 | cout<<"Gaji Bersih : Rp. " <<setw(10)<<setprecision(2)<<gaji_bersih |
| 36 | <<endl<<endl; |
| 37 | system("PAUSE"); |
| 38 | return EXIT_SUCCESS; |
| 39 | } |

Maka hasil eksekusi program diatas adalah sebagai berikut :

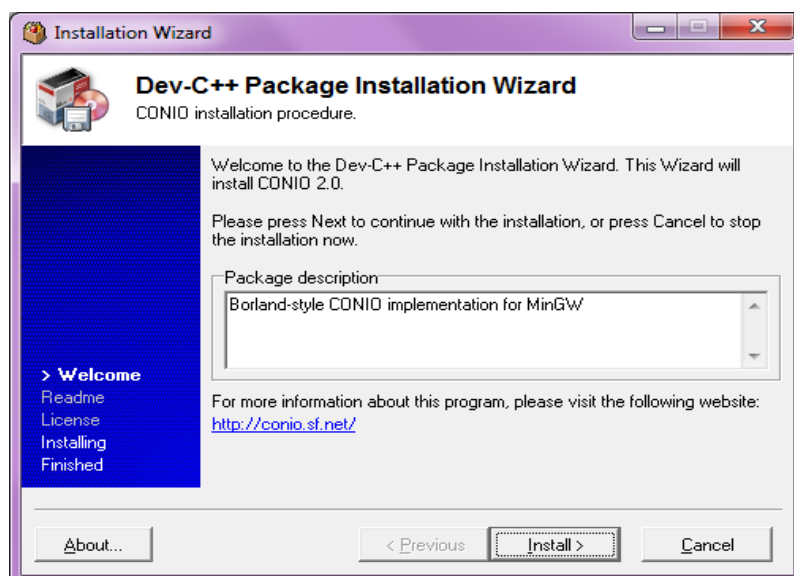


Gambar 4.4 Tampilan program 4.4

Penggunaan Warna di C/C++

Dalam pemrograman C/C++, secara default hasil eksekusi program memiliki settingan warna sesuai dengan pengaturan pada Console Application (DOS). Tetapi Anda juga dapat memberikan warna pada teks yang akan ditampilkan di program hasil eksekusi. Jika menggunakan development tools nya Turbo C dan Borland, anda dapat menggunakan file header conio.h. Tetapi untuk dev-c++ tidak bisa menggunakan conio.h dan harus menggunakan library tambahan yang bisa anda download di internet dengan nama file conio-2.0-1mol.Devpak. Langkah-langkah cara penginstalasiannya adalah sebagai berikut :

1. Klik file instalasi conio-2.0-1mol.Devpak
2. Klik instal pada form berikut ini :



Gambar 4.5 instalasi conio.h

3. Tunggu proses instalasinya
4. Klik Finish
5. Setelah selesai, maka akan tampil Package manager yang didalamnya terdapat beberapa library yang sudah diinstallkan ke Dev-C++. Silahkan anda close saja tampilan tersebut.

Contoh Program :

```

1  /*
2   Program 4-5
3   Nama File : Program 4-5.c
4   Programmer : Eko Budi Setiawan
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9  #include <conio2.h> //file header tambahan
10
11 int main(int argc, char *argv[])
12 {
13     textbackground(WHITE); //memberi background pada teks dan layar
14     clrscr(); //pengganti system("cls");
15     textcolor(MAGENTA); //warna harus dalam huruf besar
16     printf("Tulisan ini akan berwarna UNGU\n");
17     textcolor(RED); //tidak semua warna bisa
18     printf("Tulisan ini akan berwarna MERAH\n");
19     textcolor(BLUE);
20     printf("Tulisan ini akan berwarna BIRU\n");
21     textcolor(YELLOW);
22     textbackground(GREEN);
23     printf("Tulisan ini akan berwarna KUNING dan berlatar HIJAU\n\n");
24     textcolor(BLACK);
25     textbackground(WHITE);
26     gotoxy(25,10); //mengatur letak dengan aturan gotoxy(kolom,baris)
27     system("PAUSE");
28     return 0;
29 }

```

Sedangkan dalam bahasa C++ adalah sebagai berikut :

```

1  /*
2   Program 4-5
3   Nama File : Program 4-5.cpp
4   Programmer : Eko Budi Setiawan
5  */
6
7  #include <cstdlib>

```

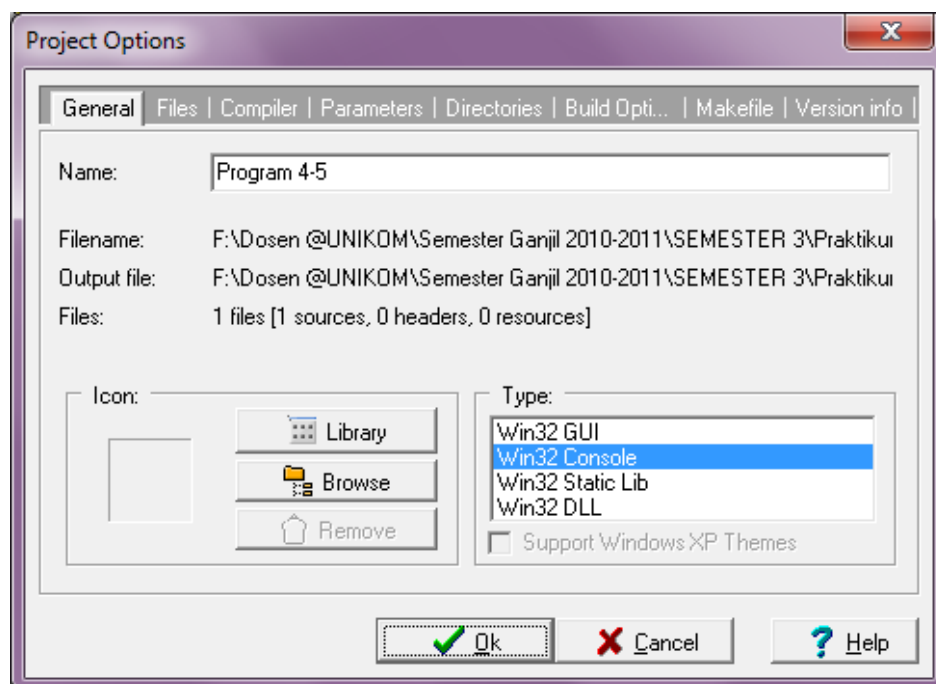
```

8  #include <iostream>
9  #include <conio2.h>
10
11  using namespace std;
12
13  int main(int argc, char *argv[])
14  {
15      textbackground(WHITE); //memberi background pada teks dan layar
16      clrscr(); //pengganti system("cls");
17      textcolor(GREEN); //warna harus dalam huruf besar
18      cout<<"Tulisan ini akan berwarna hijau tua"<<endl;
19      textcolor(LIGHTBLUE); //tidak semua warna bisa
20      cout<<"Tulisan ini akan berwarna biru terang"<<endl;
21      textcolor(RED);
22      cout<<"Tulisan ini akan berwarna merah"<<endl;
23      textcolor(10);
24      textbackground(BLACK);
25      cout<<"Tulisan ini akan berwarna hijau muda dan berlatar hitam"<<endl<<endl;
26      textcolor(BLACK);
27      textbackground(WHITE);
28      gotoxy(25,10); //mengatur letak dengan aturan gotoxy(kolom,baris)
29      system("PAUSE");
30      return EXIT_SUCCESS;
31  }

```

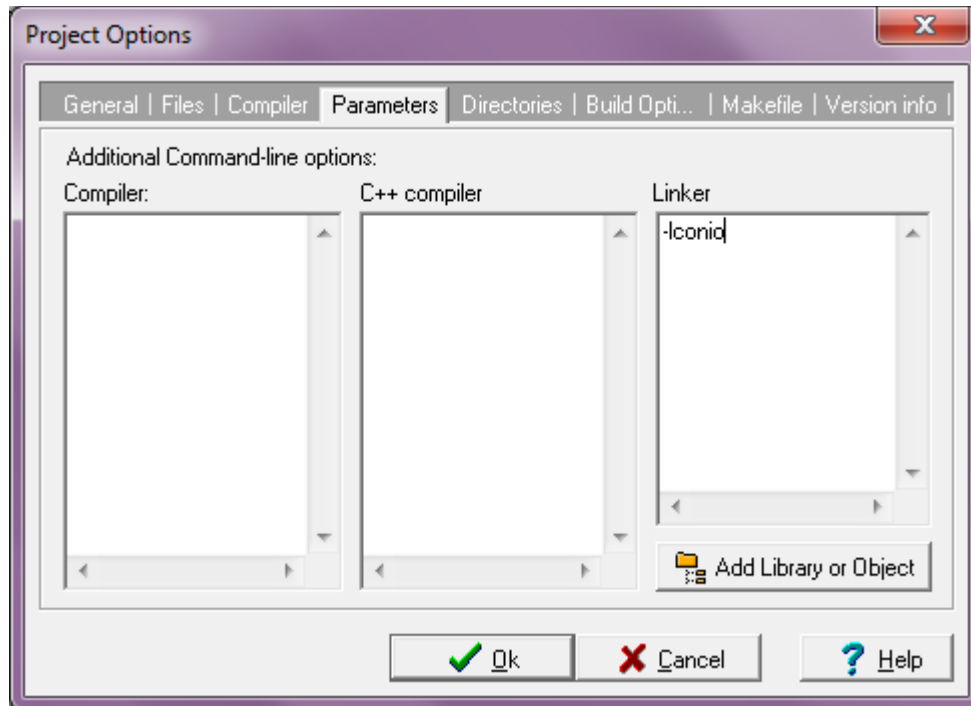
Setelah anda mengetikkan syntax diatas, maka sebelum di Compile Anda harus melakukan langkah-langkah seperti berikut :

1. Klik Project → Project Options. Maka akan tampil tampilan seperti berikut :



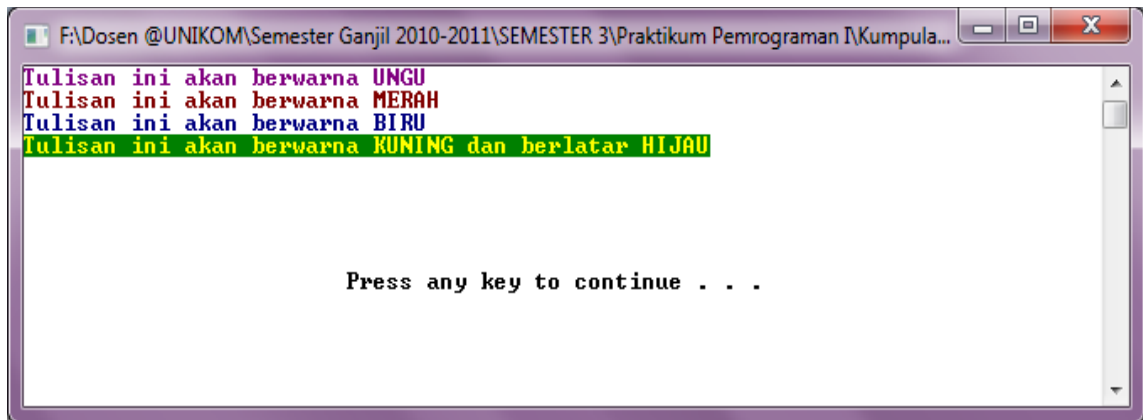
Gambar 4.6 Tampilan Form Project Option

2. Klik tab Parameters, kemudian isikan teks **-lconio** di bagian Linker. Kemudian Klik OK.



Gambar 4.7 Tampilan Linker

3. Setelah itu maka Program yang Anda buat sudah bisa di **Compile** dan **Run**. Sehingga hasil eksekusinya adalah sebagai berikut :



Gambar 4.8 Tampilan Program 4-5