|  |  |  |
| --- | --- | --- |
| logounikombw | **Diktat Mata Kuliah Pemrograman I** | IF |
| BAB VI  PERULANGAN (LOOP) |

Pendahuluan

Untuk memahami mengenai fungsi perulangan, coba lihatlah kasus sebagai berikut :

Buatlah suatu program untuk menampilkan angka dari 1 sampai dengan 5. Maka untuk kasus tersebut program yang buat adalah sebagai berikut :

|  |  |
| --- | --- |
| #include <stdio.h>  #include <conio.h>  main()  {   |  | | --- | | printf(“1\n”);  printf(“2\n”);  printf(“3\n”);  printf(“4\n”);  printf(“5\n”);  getch();  return 0; |   } |

Program di atas telah memenuhi yang diinginkan, tetapi jika bilangan yang akan ditampilkan misalkan dari 1 sampai 1000, maka sangatlah merepotkan jika harus menulis angka 1 sampai dengan 1000 secara manual. Oleh karena itu, di semua bahasa pemrograman terdapat suatu mekanisme yang bernama loop (perulangan).

Ada beberapa jenis perulangan yang dapat dilakukan oleh bahasa pemrograman C, yaitu :

* For
* While
* Do While
* Label

Perulangan Dengan Perintah *for*

Perulangan for mempunyai bentuk umum seperti berikut :

|  |
| --- |
| for(inisialisasi counter; kondisi perulangan; statement)  {  statement;  } |

Contoh berikut akan menampilkan angka 1 sampai 100, kemudian menampilkan angka 10 turun sampai 0 dengan perubahan nilainya adalah setengah (0.5).

|  |  |
| --- | --- |
| #include <stdio.h>  #include <conio.h>  main()  {   |  | | --- | | int i;  int f;  for (i=1;i<=1000;i++)  printf(“%i\n”,i);  for (f=10;f>=0;f-=0.5)  printf(“%6.2f\n”,f);  getch();  return 0; |   } |

Dalam perulangan yang menggunakan for, perulangan dilakukan hanya jika kondisi perulangannya mempunyai nilai true (tidak 0).

Perulangan Dengan Perintah *while*

Bentuk umum dari while adalah seperti berikut :

while (kondisi)

{

perintah;

perintah;

}

Cara kerja dari perulangan while mirip dengan perulangan for. Tetapi dalam perulangan while ini, tidak ada jaminan bahwa program akan masuk ke dalam perulangan. Ini dikarenakan pemeriksaan kondisinya dilakukan di awal perulangan.

|  |  |
| --- | --- |
| #include <stdio.h>  #include <conio.h>  main()  {   |  | | --- | | int i;  float f;  i=1;  while (i<=1000)  {  printf(“%i\n”,i);  i++;  }  f=10;  while (f>=0)  {  printf(“%6.2f\n”,f);  f=f-0.5;  }  getch();  return 0; |   } |

Perulangan Dengan Perintah *do while*

Bentuk umum dari do while adalah seperti berikut :

do

{

perintah;

perintah;

} while (kondisi);

Cara kerja dari perulangan do while mirip dengan perulangan while. Tetapi dalam perulangan do while ini, pengecekan kondisi dilakukan di akhir loop. Sehingga program **pasti** dapat masuk ke perulangan ini minimal 1 kali.

|  |  |
| --- | --- |
| #include <stdio.h>  #include <conio.h>  main()  {   |  | | --- | | int i;  float f;  i=1;  do  {  printf(“%i\n”,i);  i++;  } while (i<=1000);  f=10;  do  {  printf(“%6.2f\n”,f);  f=f-0.5;  } while (f>=0);  getch();  return 0; |   } |

Perulangan Dengan Menggunakan label

Perulangan dengan menggunakan teknik label, merupakan teknik perulangan yang paling awal dikenal, biasanya ada dalam pemrograman berbahasa assembly. Tetapi perulangan seperti ini **tidak dianjurkan untuk dipakai** karena bisa membuat struktur program menjadi acak-acakan.

Untuk lebih jelas, perhatikan contoh program di bawah ini.

|  |
| --- |
| #include <stdio.h>  #include <conio.h>  main()  {  int i;  i=0;  awal:  i=i+1;  printf("%i\n",i);  if(i<10)  goto awal;  else  goto akhir;  printf("Perintah ini tak akan dieksekusi\n");  printf("Perintah ini juga tak akan dieksekusi\n");  akhir:  getch();  return 0;  } |

Program di atas akan menampilkan angka 1 sampai 10. Perhatikan perintah di bawah ini :

printf("Perintah ini tak akan dieksekusi\n");

printf("Perintah ini juga tak akan dieksekusi\n");

Perintah tersebut tidak akan pernah dieksekusi, karena ketika program telah mencapai nilai 10 maka akan melewatkan perintah tersebut dan langsung loncat (goto) ke bagian akhir yang ditandai dengan perintah **akhir:**.

**Tugas :**

1. Buat program untuk menampilkan angka dari nilai awal k dengan keliapatan m dan jumlah angka sebanyak n. Contoh masukan dan keluaran:

**k = 10**

**m = 3**

**n = 4**

**10 13 16 19**

1. Buat program untuk menampilkan deret fibonanci yang sesuai dengan banyaknya angka yang ingin ditampilkan. Kemudian lakukan validasi untuk angka yang dimasukkan. Angka yang dimasukan harus antara 2 sampai 10. Bila angka yang dimasukan adalah 0, maka program selesai.

Contoh masukan dan keluaran :

**Deret Fibonanci**

**Masukan banyaknya angka yang diinginkan [2..10] : 8**

**Deret Fibonanci : 1, 1, 2, 3, 5, 8, 13, 21**