



BAB VIII

PERULANGAN “WHILE”

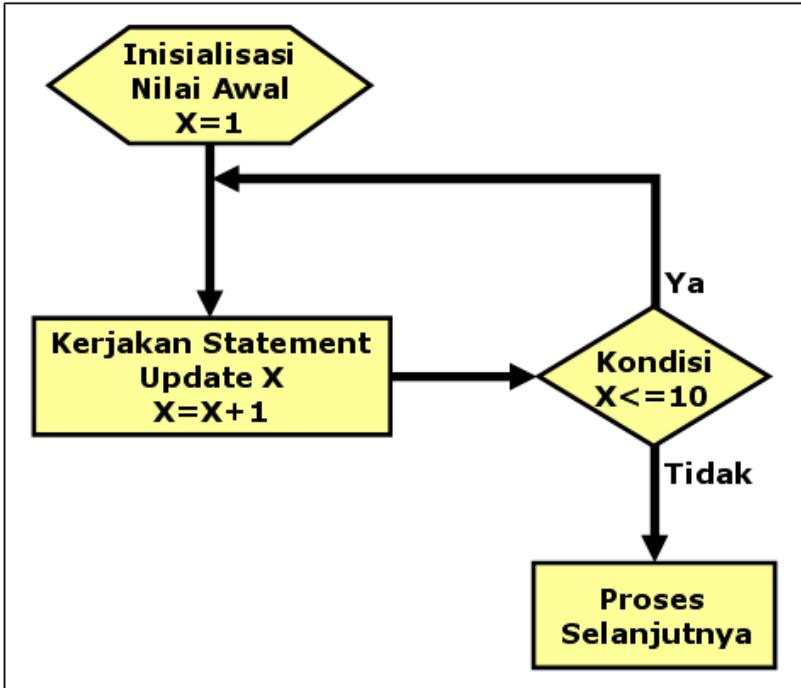
8.1 STRUKTUR PERULANGAN “DO-WHILE”

Berbeda dengan perulangan for, perulangan do-while memiliki karakteristik yang berbeda. Perulangan do-while, digunakan bila jumlah perulangannya belum diketahui.

Pada badan perulangan do-while, dapat berisi statement tunggal atau banyak statement. Pada pernyataan **do**, mula-mula statement dijalankan, selanjutnya kondisi diuji pada pernyataan **while**.

Kejadian ini berbeda dengan perulangan for, dimana pada perulangan for, kondisi diuji terlebih dahulu, jika sesuai maka dilaksanakan. Artinya pada perulangan for, jika kondisi tidak sesuai maka statement tidak dilaksanakan.

Pada perulangan do-while, statement dilaksanakan terlebih dahulu, kemudian setelahnya kondisi diuji. Jika kondisi bernilai benar, maka perulangan dilanjutkan, jika bernilai salah perulangan dihentikan.



Gambar 8.1 Struktur Perulangan Do-While

Contoh :

```

#include <stdio.h>
#include <conio.h>

main()
{
    int x;

    clrscr();
    x = 1;
    do
    {
        printf("%d BAHASA C\n", x);
        x=x+1;
    }
    while(x<=10);
  
```

```

getch();
}

```

Program 8.1 Contoh Perulangan Do-While

Pada perulangan while di atas, proses atau perintah mencetak kata-kata "BAHASA C" akan terus dilakukan selama variabel x masih kurang atau sama dengan 10. Setiap kali melakukan perulangan, nilai dari variabel x bertambah 1 angka.

Kita akan coba membuktikan apakah benar, perulangan do-while selalu melaksanakan statementnya terlebih dahulu. Artinya meskipun kondisi sejak awal bernilai salah, namun statement tetap akan dilaksanakan minimal satu kali.

Contoh :

```

#include <stdio.h>
#include <conio.h>

main()
{
    int x;

    clrscr();
    x = 5;    //Sejak Awal Nilai x Salah
    do
    {
        printf("%d BAHASA C\n", x);
        x=x+1;
    }
    while(x<=1);

    getch();
}

```

Program 8.2 Contoh Perulangan Do-While dengan Kondisi Salah

Pada program 8.2 kita dapat melihat bahwa, sejak awal nilai x tidak sesuai dengan kondisi `while`, namun statement tetap dilaksanakan. Hal ini terjadi karena pernyataan `do` berada sebelum `while`.

8.2 PERULANGAN “DO-WHILE” NEGATIF

Perulangan `do-while`, pun dapat kita jadikan menjadi perulangan `do-while` negatif, artinya nilai variabelnya akan berkurang setiap terjadi perulangan.

Perulangan `do-while` negatif dapat dilakukan jika statement update variabel melakukan pengurangan, dan nilai awal variabel lebih besar dibandingkan kondisi `while`.

Contoh :

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

main()
{
    int x;

    clrscr();
    x = 9;
    do
    {
        printf("%d BAHASA C\n", x);
        x=x-1;
    }
    while (x>=1);

    getch();
}
```

Program 8.3 Contoh Perulangan Do-While Negatif

8.3 PERULANGAN “DO-WHILE” TERSARANG

Menggunakan perulangan do-while tersarang tidak jauh berbeda dengan menggunakan perulangan for tersarang. Hanya saja, dilihat dari segi penulisan terlihat sedikit rumit. Yang perlu diingat pada perulangan do-while, bahwa pemberian nilai awal pada variabel, harus dilakukan sebelum pernyataan do.

Contoh :

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

main()
{
    int x,y;

    clrscr();
    x=1;
    do
    {
        y=1;
        do
        {
            printf(" *");
            y=y+1;
        }
        while(y<=x);

        printf("\n", x);
        x=x+1;
    }
    while(x<=10);

    getch();
}
```

Program 8.4 Contoh Perulangan Do-While Tersarang



Try And Error..

L A T I H A N - 8

Lat 8.1 :

Apa perbedaan mendasar dari perulangan for dengan perulangan while ?

Lat 8.2 :

Buatlah sebuah program yang dapat mendemonstrasikan bahwa didalam perulangan for terdapat perulangan while ?

Lat 2.3 :

Apa yang terjadi jika program berikut ini di running ?

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

main()
{
    int x;

    clrscr();
    x = 1;
    do
    {
        printf("%i \n", x);
    }
    while(x<=10);

    getch();
}
```