

MODUL 6

Perulangan (Loop) I

TUJUAN

1. Memberikan pemahaman kepada praktikan tentang konsep perulangan / loop.
2. Memberikan contoh implementasi teori perulangan pada sebuah program sederhana.

TEORI

Salah satu kelebihan komputer dibandingkan dengan manusia adalah kemampuannya untuk melakukan suatu perintah yang berulang kali tanpa mengenal lelah dan bosan. Perulangan adalah suatu bentuk kegiatan mengulang suatu statement sampai batas yang diinginkan selama kondisi masih terpenuhi. Dalam pemrograman, perulangan sangat berguna untuk menyederhanakan suatu perintah yang dijalankan berulang-ulang tanpa menulis sintaks tersebut berulang-ulang.

Contoh penggunaan sederhana perulangan adalah, jika kita ingin menampilkan tulisan “Hello World” sebanyak 50 kali, kita tidak perlu menulis `printf("Hello World")` sebanyak 50 kali, tetapi cukup dengan penambahan beberapa perintah saja. Seperti contoh berikut:



```
for(i=0 ; i<50 ; i++){  
    printf("Hello World");  
}
```

LIMA KOMPONEN LOOP

Loop atau perulangan mempunyai komponen yang perlu ada dalam strukturnya, yaitu :

1. **Inisialisasi** : Pengesetan awal variabel kontrol.

Variabel kontrol adalah variabel yang akan digunakan sebagai pengontrol perulangan, dan harus dimulai dari suatu nilai tertentu.

2. **Kondisi** : ekspresi Boolean (kondisi terhadap variabel kontrol berupa **TRUE/FALSE**) yang menentukan berakhirnya loop. Loop akan tetap dijalankan selama kondisi masih **TRUE**.

3. **Inkremen/dekremen** : Peningkatan/penurunan variabel kontrol.

Variabel kontrol ini akan di-inkremen/di-dekremen setiap kali loop terjadi.

Inkremen : Proses penambahan nilai variabel kontrol, dimana nilai tersebut terus bertambah setiap kali proses perulangan berjalan.

Contoh :

- $i++$, artinya adalah inkremen dengan 1, yaitu setara dengan $i=i+1$;
- $i+=2$, artinya adalah inkremen dengan 2, yaitu setara dengan $i=i+2$;; dst.

Dekremen : Proses pengurangan nilai variabel kontrol, dimana nilai tersebut terus berkurang setiap kali proses perulangan berjalan.

Contoh :

- $i--$, artinya adalah dekremen dengan 1, yaitu setara dengan $i=i-1$;
- $i-=2$, artinya adalah dekremen dengan 2, yaitu setara dengan $i=i-2$;; dst.

4. **Statement** : Perintah yang diloop (perintah yang akan diulang-ulang). Komponen ini berisi perintah yang ingin kita lakukan perulangan.

5. **Terminasi** : Statement pengakhir, yang menyatakan/mengkonfirmasi akhir loop (optional).

STRUKTUR LOOP

Dalam bahasa C, terdapat 3 buah struktur perulangan, yaitu for, while dan do-while. Setiap struktur yang ada memiliki aturannya sendiri. Kita harus bisa menentukan kapan sebaiknya menggunakan struktur perulangan for, while atau do-while.

1. Struktur For (**Counted Loop**)

Struktur for digunakan untuk perulangan dimana telah diketahui jumlah perulangannya. Pada struktur for, kita harus melakukan inisialisasi nilai awal terlebih dahulu, menulis kondisi untuk menghentikan proses perulangan dan menyatakan apakah perulangan bersifat inkremen atau dekremen.

Strukturnya secara umum adalah sebagai berikut :

```
for( inisialisasi; kondisi; inkremen/dekremen )
{
    statement_1;
    statement_2;
    ...
}
terminasi;
```

Contoh source code:

```
for(i=0 ; i<4 ; i++)
{
    printf("UAJY");
}
printf("Loop Selesai");
```

Alur program :

1 – 2 (0 < 4) → kondisi terpenuhi
4 – 3 – 2 (1 < 4) → kondisi terpenuhi
4 – 3 – 2 (2 < 4) → kondisi terpenuhi
4 – 3 – 2 (3 < 4) → kondisi terpenuhi
4 – 3 – 2 (4 < 4) → kondisi tidak terpenuhi
5 → terminasi

2. Struktur While (**Uncounted Loop**)

Struktur while digunakan bila jumlah perulangannya belum diketahui, dimana kondisi akan diperiksa di bagian awal. Hal ini memungkinkan saat pengecekan kondisi bernilai salah, maka tidak akan masuk ke dalam proses perulangan. Sementara jika kondisi bernilai benar maka proses perulangan akan terus berlanjut selama kondisinya benar dan akan berhenti bila kondisinya bernilai salah.

Strukturnya secara umum adalah sebagai berikut :

```
inisialisasi;
while(kondisi)
{
    statement_1;
    statement_2;
    ...
    inkremen/dekremen;
}
terminasi;
```

Contoh source code:

```
i=10; ❶
while(i > 0) ❷
{
    printf("%d ", i); ❸
    i-=2; ❹
}
printf("Loop Selesai"); ❺
```

Alur program :

1 – 2 (10 > 0) → kondisi terpenuhi
3 – 4 – 2 (8 > 0) → kondisi terpenuhi
3 – 4 – 2 (6 > 0) → kondisi terpenuhi
3 – 4 – 2 (4 > 0) → kondisi terpenuhi
3 – 4 – 2 (2 > 0) → kondisi terpenuhi
3 – 4 – 2 (0 > 0) → kondisi tidak terpenuhi
5 → terminasi

3. Struktur Do-While (**Uncounted Loop**)

Struktur do-while hampir sama dengan struktur perulangan while. Perbedaannya adalah jika kondisi while melakukan pengecekan kondisi di awal, pada perulangan do while pengecekan kondisi dilakukan di akhir. Jadi perulangan akan terjadi terlebih dahulu baru dicek kondisinya. Sehingga pada perulangan do while akan terjadi perulangan minimal 1x.

Strukturnya secara umum adalah sebagai berikut :

```
inisialisasi;
do
{
    statement_1;
    statement_2;
    ...
    inkremen/dekremen;
}while(kondisi);
terminasi;
```

Contoh source code:

```
i=4; ①
do ②
{
    printf("%d ",i);
    ③ i+=4;
}while(i<=12); ④
printf("Loop Selesai"); ⑤
```

Alur program :

1 – 2 – 3 – 4 (8 <= 12) → kondisi terpenuhi
5 – 2 – 3 – 4 (12 <= 12) → kondisi terpenuhi
5 – 2 – 3 – 4 (16 <= 12) → kondisi tidak terpenuhi
6 → terminasi

F.A.Q (Frequently Asked Questions) :

1. Apakah nama variabel pada perulangan harus selalu i ?

Jawab: Tidak, nama variabel bebas, namun diusahakan tidak membingungkan agar dalam pembuatan code tidak merugikan kita dalam mengenal variabel.

2. Kapan menggunakan for loop atau do while?

Jawab: For loop digunakan pada saat kita sudah mengetahui berapa banyak perulangan yang dibutuhkan (Counted Loop), sedangkan untuk do while digunakan untuk perulangan yang belum jelas berapa kali dia perlu diulang (Uncounted Loop).



3. Apakah bisa memberhentikan loop secara paksa ?

Jawab : Bisa, menggunakan syntax bernama **break**, sebagai contoh saya ingin menghentikan for loop pada saat variabel i bernilai 5 maka code akan berbentuk seperti ini:

```
for(i=0; i<10; i++){  
    printf("%d", i);  
    if(i==5){  
        break;  
    }  
}
```

GUIDED

Buatlah sebuah program menjumlahkan bilangan, menampilkan alfabet dan data diri dibawah ini
+ Pelajari mengenai bagaimana menggunakan rand() pada bahasa C untuk mencari sebuah angka acak ! (Berguna bagi UGD Anda kedepan) :

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
typedef char string[50];

int main(int argc, char *argv[]) {
    int menu,bil,umur,temp=0,random=0;
    string nama;
    srand(time(NULL)); // inisialisasi pertama agar angka selalu random setiap kali program di run
    char huruf;

    do
    {
        system("cls");
        printf("===== MENU GUIDED =====");
        printf("\n[1] Menu menjumlahkan bilangan");
        printf("\n[2] Menu menampilkan huruf A-Z");
        printf("\n[3] Menu input data diri");
        printf("\n[4] Menu cari angka random (20-80)");
        printf("\n-----");
        printf("\nMasukkan pilihan menu      : ");scanf("%d",&menu);
        switch (menu)
        {
            case 1:
                printf("\nMasukkan bilangan      : ");scanf("%d",&bil);
                while (bil>0)
                {
                    temp = temp +bil; // melakukan penambahan bilangan
                    printf("\nTemp = %d",temp);
                    bil--;
                }
                printf("\nJumlah bilangannya : %d",temp);

                break;
            case 2:
                printf("\nAlphabet : ");
                for (huruf='A';huruf<='Z';huruf++)
                // inisialisasi huruf mulai dari A dan akan berhenti jika sudah sampai pada huruf Z
                {
                    printf("%c",huruf);// melakukan output tiap huruf
                }
                break;
        }
    }
}
```

```

case 3:
do
{
    printf("\nNama      : ");fflush(stdin);gets(nama);// meminta inputan nama kembali jika input kosong
} while (strlen(nama)<1);

do
{
    printf("\nUmur      : ");scanf("%d",&umur);
    // meminta inputan nama kembali jika input kosong atau negatif
} while (umur<=0);
printf("\nNama      : %s",nama);
printf("\nUmur      : %d",umur);
break;

case 4:
printf("\nBerhentikan Perulangan disaat angka menunjukan (20-80) :");scanf("%d",&bil);

do{
    random = (rand() % (80 - 20 + 1)) + 20; // format (rand() % (angkaMax - angkaMin + 1)) + angkaMin
    printf("\n Random : %d",random);
} while (random !=bil);

break;

case 0: printf("\nTerima kasih :) ");break;

default:printf("\n[!] Menu hanya ada 1-4");
break;
}getch();
} while (menu!=0);

return 0;
}

```

Code komentar tidak harus dibuat ya teman-teman 😊
hanya sebagai penjelas code.

Ketentuan Pengerjaan:

1. Pastikan ekstensi program file adalah .c bukan .cpp
 2. Guided dikerjakan dalam folder terpisah lalu di zip (GD5_X_YYYYYY.zip)
- X = Kelas
YYYYYY = 5 digit terakhir NPM