

MODUL & GUIDED PERULANGAN 2

NESTED LOOP

Tujuan :

1. Memberikan pemahaman kepada praktikan tentang konsep perulangan di dalam perulangan (nested loop).
2. Memberikan pengalaman mengimplementasikan ke dalam program sederhana konsep loop kepada praktikan.

A. Nested Loop

Suatu proses yang dijalankan loop dapat berupa sebuah loop juga. Dalam hal ini, loop di dalam loop atau bisa disebut nested loop. Contoh di dalam for terdapat for lain atau juga bisa terdapat while (bisa dipelajari lebih lanjut untuk pretest). Untuk lebih jelasnya, jalankan contoh-contoh dibawah dan pahami cara kerja dari perulangan tersebut, perhatikan nilai i dan j.

Hasil run contoh 1.1

```
1 //Contoh 1.1
2 int main(int argc, char *argv[]) {
3     int i, j;
4
5     //Perulangan Pertama
6     for ( i = 1; i ≤ 5; i++) {
7         //Perulangan Kedua
8         for ( j = 1; j ≤ 5; j++ ){
9             printf("%d - %d \n", i, j);
10        }
11        printf("-----\n");
12    }
13    return 0;
14 }
```

```
1 - 1
1 - 2
1 - 3
1 - 4
1 - 5
-----
2 - 1
2 - 2
2 - 3
2 - 4
2 - 5
-----
3 - 1
3 - 2
3 - 3
3 - 4
3 - 5
-----
4 - 1
4 - 2
4 - 3
4 - 4
4 - 5
-----
5 - 1
5 - 2
5 - 3
5 - 4
5 - 5
-----
```

```

1 int main(int argc, char *argv[]) {
2     int i, j, nilai;
3
4     //Menerima Input
5     printf("Masukkan Nilai : "); scanf("%d", &nilai);
6
7     //Perulangan Pertama
8     for ( i = 1; i ≤ nilai; i++) {
9         //Perulangan Kedua
10        for ( j = 1; j ≤ i; j++ ){
11            printf("*");
12        }
13        printf("\n");
14    }
15    return 0;
16 }

```

```

Masukkan Nilai : 6
*
**
***
****
*****
*****

```

Hasil run contoh 2.2

B. Break Statement

Selain digunakan pada kasus percabangan, **break** juga berguna untuk melakukan penghentian suatu perulangan pada suatu keadaan tertentu. Penggunaannya dapat dilihat pada **contoh 2.1**. Di dalam loop dari titik dimana break berada, artinya ada keadaan yang tidak diharapkan untuk dijalankan loop secara normal. Secara umum pernyataan break dapat dieliminasi dengan penambahan kondisi, sehingga loop berhenti normal.

```

1 //contoh 2.1
2 int main(int argc, char *argv[]) {
3     int i=0, nilai;
4
5     //Menerima Input
6     printf("Masukkan Nilai : "); scanf("%d", &nilai);
7
8     //Mulai Perulangan
9     do {
10        i++;
11        printf("%d\n", i);
12        if (i == nilai) {
13            break; //Pengaplikasian Break
14        }
15    } while (i≠10);
16    return 0;
17 }

```

Hasil run contoh 2.1

```

Masukkan Nilai : 8
1
2
3
4
5
6
7
8

```

```

1 //contoh 2.2
2 int main(int argc, char *argv[]) {
3     int nilai=1;
4
5     //Mulai perulangan
6     while (nilai !=0){
7         printf("Masukkan Nilai : "); scanf("%d", &nilai);
8         if (nilai == 0){
9             printf("STOP");
10            //jika nilai yang dimasukkan 0 maka program akan berhenti
11        }
12        //Program akan meminta inputan kembali sebelum nilai bernilai 0
13    }
14    return 0;
15 }

```

Hasil run contoh 2.2

```

Masukkan Nilai : 5
Masukkan Nilai : 3
Masukkan Nilai : 2
Masukkan Nilai : 5
Masukkan Nilai : 0
STOP_

```

C. Perumpamaan Matriks

Jika nested loop diumpamakan dalam matriks, maka i sebagai baris dan j sebagai kolom sebuah matriks.

```

1 //Contoh 3.1
2 int main(int argc, char *argv[]) {
3     int i, j, nilai;
4
5     //Menerima Input
6     printf("Masukkan Nilai : "); scanf("%d", &nilai);
7
8     //Perulangan Pertama
9     for( i = 1; i ≤ nilai; i++) {
10        //Perulangan Kedua
11        for( j = 1; j ≤ nilai ; j++) {
12            printf("* ");
13        }
14        printf("\n");
15    }
16    return 0;
17 }

```

Hasil run contoh 3.1

```

Masukkan Nilai : 6
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *

```

```

1 //Contoh 3.2
2 int main(int argc, char *argv[]) {
3     int baris, kolom;
4
5     for ( baris = 1; baris ≤ 3; baris++) {
6         for ( kolom = 1; kolom ≤ 5; kolom++) {
7             printf("\t%d", kolom);
8         }
9         printf("\n");
10    }
11    return 0;
12 }

```

Hasil run contoh 3.2

1	2	3	4	5
1	2	3	4	5
1	2	3	4	5

Guided

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(int argc, char *argv[]) {
5     int i,j,n, menu, cekMenu=0;
6
7     do{
8         system("cls");
9         printf("\n\tGuided Perulangan 2");
10        printf("\n[1] Masukkan jumlah baris");
11        printf("\n[2] Show ");
12        printf("\n[0] Keluar");
13        printf("\n==>");scanf("%d", &menu);
14
15        switch (menu){
16
17            case 1:
18                do{
19                    printf("Masukkan jumlah baris: "); scanf("%d",&n);
20                    /*Ketika user memasukkan data,inputan user akan di cek terlebih dahulu*/
21                    if(n<1){
22                        printf("\nInputan tidak boleh dibawah 1");
23                    }else{
24                        printf("\nBerhasil menginput data");
25                        cekMenu++;
26                    }
27                }while(n<1);
28                break; //Pengaplikasian break
29
30            case 2:
31                if(cekMenu==0){
32                    printf("\nInputkan jumlah baris terlebih dahulu");
33                }else{
34                    printf("\nTampilan Bentuk dari jumlah baris-%d\n", n);
35                    char huruf = 'A'; /* Pendeklarasian char A, digunakan untuk memberikan
36                    inputan pada perulangan kedua dibawah, agar bisa dimulai dari huruf A*/
37
38                    //Perulangan pertama
39                    for(i = 1; i ≤ n+2; i++){
40                        // Perulangan kedua
41                        for(j = 1; j ≤ n+2; j++){
42                            if(j = 1||i = 1||j = n+2||i = n+2){
43                                printf("* ");
44                            } else {
45                                printf("%c ",huruf++);/*char akan terus bertambah pada setiap
46                                perulangannya karena adanya inkremen*/
47                            }
48
49                            if(huruf > 'Z') {
50                                huruf='A';
51                                /*Ketika char huruf sudah diakhir alfabet, maka program akan
52                                mengulangi dari huruf A*/
53                            }
54                            printf("\n");
55                        }
56                    }
57                    break;
58
59            case 0 :
60                printf("Guided Perulangan 2 - Nama/NPM/Kelas");
61                break;
62
63            default:
64                printf("Pilihan menu tidak tersedia");
65                break;
66
67        } getch();
68    } while (menu≠0);
69    return 0;
70 }
```

Note : Komentar pada code tidak perlu dibuat

Contoh hasil output Guided

```
Tampilan Bentuk dari jumlah baris-7
* * * * *
* A B C D E F G *
* H I J K L M N *
* O P Q R S T U *
* V W X Y Z A B *
* C D E F G H I *
* J K L M N O P *
* Q R S T U V W *
* * * * *
```

Format Pengumpulan Guided

Guided dikumpulkan pada folder dengan format penamaan **GD6_X_YYYY** (Pastikan ekstensi program file adalah .c bukan .cpp), kemudian folder di .zip

X = Kelas, YYYYYY = 5 digit terakhir NPM