PENYELEKSIAN KONDISI

1. STRUKTUR KONDISI "IF…."

Struktur if dibentuk dari pernyataan if dan sering digunakan untuk menyeleksi suatu kondisi tunggal. Bila proses yang diseleksi terpenuhi atau bernilai benar, maka pernyataan yang ada di dalam blok if akan diproses dan dikerjakan. Bentuk umum struktur kondisi if adalah :

if(kondisi)
pernyataan;

Contoh Program 1 :

```
/* Program struktur kondisi if untuk memeriksa suatu kondisi */
    #include "stdio.h"
    #include "conio.h"
    void main()
    { float nilai;
    scanf("%f", &nilai);
    if(nilai > 65)
    printf("\n ANDA LULUS !!!!\n");
    getch();
    }
```

2. STRUKTUR KONDISI "IF.....ELSE…."

Dalam struktur kondisi if.....else minimal terdapat dua pernyataan. Jika kondisi yang diperiksa bernilai benar atau terpenuhi maka pernyataan pertama yang dilaksanakan dan jika kondisi yang diperiksa bernilai salah maka pernyataan yang kedua yang dilaksanakan. Bentuk umumnya adalah sebagai berikut :

if(kondisi)
pernyataan-1
else
pernyataan-2

Contoh Program :

```
    #include "stdio.h"
    #include "conio.h"
    void main()
    { float nilai;
    clrscr();
    printf("Masukan nilai yang didapat : ");
    scanf("%f", &nilai);
    if (nilai > 65)
    printf("\n LULUS !!!\n");
    else
    printf("\n TIDAK LULUS !!!\n");
    getch();
    }
```

3. STRUKTUR KONDISI "SWITCH....CASE....DEFAULT…"

Struktur kondisi switch....case....default digunakan untuk penyeleksian kondisi dengan kemungkinan yang terjadi cukup banyak. Struktur ini akan melaksanakan salah satu dari beberapa pernyataan 'case' tergantung nilai kondisi yang ada di dalam switch. Selanjutnya proses diteruskan hingga ditemukan pernyataan 'break'. Jika tidak ada nilai pada case yang sesuai dengan nilai kondisi, maka proses akan diteruskan kepada pernyataan yang ada di bawah 'default'.
Bentuk umum dari struktur kondisi ini adalah :

```
switch(kondisi)
{
case 1 : pernyataan-1;
break;
case 2 : pernyataan-2;
break;
.....
.....
case n : pernyataan-n;
break;
default : pernyataan-m
}
```

Contoh Program :

```
/* Program menentukan nama hari berdasarkan inputan */
#include "stdio.h"
#include "conio.h"
void main()
{ clrscr();
int hari;
puts("Menentukan nama hari\n");
puts("1 = Senin 2 = Selasa 3 = Rabu 4 = Kamis");
puts("5 = Jum'at 6 = Sabtu 7 = Minggu");
printf("\nMasukan kode hari( 1-7) : ");
scanf("%d", &hari);
switch(hari)
{ case 1 : puts("Hari Senin"); /* kemungkinan pertama */
break;
case 2 : puts("Hari Selasa"); /* kemungkinan kedua */
break;
case 3 : puts("Hari Rabu"); /* kemungkinan ketiga */
break;
case 4 : puts("Hari Kamis"); /* kemungkinan keempat */
break;
case 5 : puts("Hari Jum'at"); /* kemungkinan kelima */
break;
case 6 : puts("Hari Sabtu"); /* kemungkinan keenam */
break;
case 7 : puts("Hari Minggu"); /* kemungkinan ketujuh */
break;
default : puts("Kode hari yang Anda masukan SALAH");}
getch();}
```

PERULANGAN

1. STRUKTUR PERULANGAN " WHILE"

Perulangan WHILE banyak digunakan pada program yang terstruktur. Perulangan ini banyak digunakan bila jumlah perulangannya belum diketahui. Proses perulangan akan terus berlanjut selama kondisinya bernilai benar (true) dan akan berhenti bila kondisinya bernilai salah.

Contoh Program 1 :

```
/* Program Perulangan menggunakan while */
    #include "stdio.h"
    #include "conio.h"
    void main()
    { int x;
    x = 1; /* awal variabel */
    while (x <= 10) /* Batas akhir perulangan */
    { printf("%d BAHASA C\n", x);
    x ++; /* variabel x ditambah dengan 1 */
    }
    getch();
    }
```

2. STRUKTUR PERULANGAN "DO.....WHILE…"

Pada dasarnya struktur perulangan do....while sama saja dengan struktur while, hanya saja pada proses perulangan dengan while, seleksi berada di while yang letaknya di atas sementara pada perulangan do....while, seleksi while berada di bawah batas perulangan. Jadi dengan menggunakan struktur do…while sekurang-kurangnya akan terjadi satu kali perulangan.

Contoh Program :

```
    #include "stdio.h"
    #include "conio.h"
    void main()
    { int x;
    x = 1;
    do
    { printf("%d BAHASA C\n", x);
    x ++;
    }
    while(x <= 10);
    getch(); }
```

3. STRUKTUR PERULANGAN "FOR"

Struktur perulangan for biasa digunakan untuk mengulang suatu proses yang telah diketahui jumlah perulangannya. Dari segi penulisannya, struktur perulangan for tampaknya lebih efisien karena susunannya lebih simpel dan sederhana. Bentuk umum perulangan for adalah sebagai berikut :

for(inisialisasi; syarat; penambahan)
pernyataan;

Keterangan :

- ***Inisialisasi*** : pernyataan untuk menyatakan keadaan awal dari variabel kontrol.
- ***syarat*** : ekspresi relasi yang menyatakan kondisi untuk keluar dari perulangan.
- ***penambahan*** : pengatur perubahan nilai variabel kontrol.

Contoh Program 1 :

```
/* Program perulangan menggunakan for */
#include “stdio.h”
#include “conio.h”
void main()
{ int x;
for(x = 1; x<= 10; x++)
{ printf(“%d BAHASA C\n”, x); }
getch();
}
```

ARRAY (LARIK)

1. ARRAY DIMENSI SATU

- Setiap elemen array dapat diakses melalui indeks.
- Indeks array secara default dimulai dari 0.
- Deklarasi Array

Bentuk umum :

Tipe_array nama_array[ukuran];

Contoh :

Nilai[0] Nilai[1] Nilai[2] Nilai[3] Nilai[4]
int Nilai[5];
70 80 82 60 75

Contoh Program :

```
/* Program untuk menginput nilai mahasiswa ke dalam array satu dimensi */
#include “stdio.h”
#include “conio.h”
void main();
{ int index, nilai[10];
clrscr();
/* input nilai mahasiswa */
printf(“Input nilai 10 mahasiswa : “);
for(index=0; index < 10; index++)
{ printf(“Mahasiswa %i : “, index+1);
scanf(“%i”, &nilai[index]);
}
/* tampilkan nilai mahasiswa */
printf(“Nilai mahasiswa yang telah diinput”);
for(index=0; index < 10; index++)
{ printf(“%5.0i”, nilai[index]);
}
getch();
}
```

2. ARRAY DIMENSI DUA

- Array dua dimensi merupakan array yang terdiri dari m buah baris dan n buah kolom.
- Bentuknya dapat berupa matriks atau tabel.

- Deklarasi array :

  Tipe_array nama_array[baris][kolom];

  Contoh :

  Int X[3][4];
  X[0][0] X[0][1] X[0][2] X[0][3]
  X[1][0] X[1][1] X[1][2] X[1][3]
  X[2][0] X[2][1] X[2][2] X[2][3]

- Cara mengakses array :
  Untuk mengakses array, misalnya kita ingin mengisi elemen array baris 2 kolom 3 dengan 10 maka perintahnya adalah sbb :

X[1][2] = 10;

- Untuk mengisi dan menampilkan isi elemen array ada dua cara yaitu :
  - Row Major Order (secara baris per baris)
  - Column Major Order (secara kolom per kolom)

/* Program menginput nilai(bilangan) ke dalam array dimensi dua dan menampilkannya */

```
#include “stdio.h”
#include “conio.h”
void main()
{ int baris, kolom, matriks[3][4];
crlscr();
// Input elemen array secara Row Major Order
printf(“Input elemen Array : \n“);
for(baris=0; baris<3; baris++)
{ for(kolom=0; kolom<4; kolom++)
{ printf(“matriks[%i][%i]”, baris+1, kolom+1);
scanf(“%i”, &matriks[baris][kolom]);
}
printf(“\n”);
}
printf(“Isi array : \n”);
for(baris=0; baris<3; baris++)
{ for(kolom=0; kolom<4; kolom++)
{ printf(“%i”, &matriks[baris][kolom]);
}
printf(“\n”);
}
getch();
}
```