

PEMROGRAMAN KOMPUTER

C Language Series

MODUL 2

(Penyeleksian Kondisi dan Perulangan)

Tim Pengajar Algoritma dan Pemrograman

Penyeleksian Kondisi dan Perulangan

Nama Mata Kuliah	:	Praktikum Pemrograman Komputer
Bobot sks	:	1 SKS
Semester	:	II (Dua)
Koordinator MK	:	I Made Widiartha, S.Si.,M.Kom
Modul Praktikum	:	Modul 2 – Penyeleksian Kondisi dan Perulangan

1. Deskripsi

Pada praktikum ini akan dipraktekan tentang penyeleksian kondisi dan perulangan pada bahasa pemrograman C yang meliputi: If-Else, Switch-Case-Default, For, While-Do, dan Do-While.

2. Tujuan Umum

Setelah mengikuti praktikum ini, mahasiswa mampu menggunakan berbagai perintah penyeleksian kondisi dan perulangan pada bahasa pemrograman C dalam memecahkan suatu permasalahan.

3. Indikator Pencapaian

Adapun indikator pencapaian dari modul praktikum ini adalah mahasiswa dapat mengimplementasikan logika penyeleksian kondisi dan perulangan yang terkait dengan permasalahan-permasalahan yang ada ke dalam sebuah program sederhana.

4. Teori

4.1. Penyeleksian Kondisi

Penyeleksian kondisi digunakan untuk mengarahkan perjalanan suatu proses. Penyeleksian kondisi dapat diibaratkan sebagai katup atau kran yang mengatur jalannya air. Bila katup terbuka maka air akan mengalir dan sebaliknya bila katup tertutup air tidak akan mengalir atau akan mengalir melalui tempat lain. Fungsi penyeleksian kondisi

Penyeleksian Kondisi dan Perulangan

penting artinya dalam penyusunan sebuah program, terutama untuk program yang kompleks.

4.1.1. 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 :

```
#include "stdio.h"
#include "conio.h"
void main()
{
    float nilai;
    clrscr();

    printf("Masukan nilai yang didapat : ");
    scanf("%f", &nilai);      /* Masukan akan disimpan dalam variable nilai */

    if (nilai > 65)
        printf("\n LULUS !!!\n");
    else
        printf("\n TIDAK LULUS !!!\n");

    getch();
}
```

Bila program tersebut dijalankan dan kita memasukan nilai 80 maka akan dicetak perkataan “LULUS !!!” namun bila kita memasukan nilai yang kurang dari 65 maka akan tercetak perkataan “TIDAK LULUS !!!”.

Penyeleksian Kondisi dan Perulangan

4.1.2. 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 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");
}
```

Penyeleksian Kondisi dan Perulangan

```

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();
}

```

Bila program tersebut dijalankan, dan kita memasukan kode hari dengan 1 maka akan tercetak “Hari Senin”, bila 2 akan tercetak “Hari Selasa” dan seterusnya.

4.2. Perulangan

Dalam bahasa pemrograman C tersedia suatu fasilitas yang digunakan untuk melakukan proses yang berulang-ulang sebanyak keinginan kita. Misalnya saja, bila kita ingin menginput dan mencetak bilangan dari 1 sampai 100 bahkan 1000, tentunya kita akan merasa kesulitan. Namun dengan struktur perulangan proses, kita tidak perlu menuliskan perintah sampai 100 atau 1000 kali, cukup dengan beberapa perintah saja. Struktur perulangan dalam bahasa pemrograman C mempunyai bentuk yang bermacam-macam yaitu `for () {...}`, `while () {...}`, dan `do {...} while ()`.

Semisal dalam suatu program kita akan menampilkan kata-kata “BAHASA C” sebanyak 10 kali seperti contoh berikut :

Penyeleksian Kondisi dan Perulangan

```
1BAHASA C
2BAHASA C
3BAHASA C
4BAHASA C
5BAHASA C
6BAHASA C
7BAHASA C
8BAHASA C
9BAHASA C
10BAHASA C
```

Maka kita dapat menggunakan fungsi-fungsi perulangan yang terdapat pada bahasa pemrograman C seperti penjelasan di bawah ini.

4.2.1. 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 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();
}
```

Penyeleksian Kondisi dan Perulangan

4.2.2. 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 kondisi di dalam `while ()` bernilai benar (true) dan akan berhenti bila kondisinya bernilai salah. Bentuk umumnya adalah sebagai berikut:

```
while(syarat) {
    pernyataan;
}
```

Contoh :

```
/* Program Perulangan menggunakan while */

#include "stdio.h"
#include "conio.h"

void main()
{
    int x;
    x = 1; // awal variable

    while (x <= 10) { // Batas akhir perulangan

        printf("%d BAHASA C\n", x);
        x++; // variabel x ditambah dengan 1
    }

    getch();
}
```

4.2.3. 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`,

Penyeleksian Kondisi dan Perulangan

seleksi while berada di bawah batas perulangan. Jadi dengan menggunakan struktur do...while sekurang-kurangnya akan terjadi satu kali perulangan.

```
do{
    pernyataan;
} while(syarat)
```

Contoh :

```
/* Program Perulangan menggunakan while */

#include "stdio.h"
#include "conio.h"

void main()
{
    int x;
    x = 1; // awal variable

    do{
        printf("%d BAHASA C\n", x);
        x++; // variabel x ditambah dengan 1
    } while (x <= 10) // Batas akhir perulangan

    getch();
}
```

5. Referensi

1. Munir, R. 1999. Algoritma dan Pemrograman Dalam Bahasa Pascal dan C. Bandung: Informatika.
2. Kadir, A dan Heriyanto. 2005. Algoritma Pemrograman Menggunakan C++. Yogyakarta: Penerbit Andi.
3. Tosin, R. 1997. Flowchart untuk Siswa dan Mahasiswa. Jakarta: DINASTINDO.

Penyeleksian Kondisi dan Perulangan

6. Pelaksanaan Praktikum

6.1. Tools

Tools yang dapat digunakan pada untuk menunjang praktikum kali ini antara lain:

1. Code :: Block IDE sebagai IDE bahasa pemrograman C

6.3. Langkah-langkah Praktikum

Berikut akan dipaparkan langkah-langkah pembuatan program sederhana dengan menggunakan bahasa pemrograman C.

Studi kasus :

Buatlah sebuah program sederhana dari proses login pada sebuah sistem. User hanya diberi 3x kesempatan untuk memasukan PIN yang dimilikinya untuk dapat login ke dalam sistem. Jika dalam 3x kesempatan user masih salah memasukan nomor PIN, maka sistem akan mem-block user tersebut sehingga tidak dapat login kembali. Asumsi PIN yang dimiliki user tersebut adalah “1234” (tanpa tanda petik).

Langkah-langkah pembuatan program :

1. Pemanggilan library-library yang dibutuhkan oleh program

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

Library di atas merupakan library yang dibutuhkan untuk standar proses input-output pada bahasa pemrograman C. Dalam praktikum kali ini, kita membutuhkan kedua library tersebut untuk memanggil fungsi-fungsi masukan dan keluaran data.

2. Pembuatan fungsi utama

```
int main()
{
    return 0;
}
```

Penyeleksian Kondisi dan Perulangan

Fungsi "main ()" ini berfungsi sebagai fungsi pertama yang akan dieksekusi oleh program, sehingga tanpa adanya fungsi "main ()" ini, maka tidak ada acuan untuk memulai eksekusi sebuah program.

3. Pendefinisian Variabel

```
int pin, i;
```

- pin : digunakan sebagai variabel yang menyimpan nomor PIN yang diinputkan oleh user.
- i : digunakan sebagai variabel increment yang membatasi kesempatan user untuk mencoba login. (dalam hal ini hanya dibatasi max 3x saja)

4. Perulangan untuk proses login

```
for (i=3; i>=1; i--) {  
}
```

User hanya dapat mencoba login sebanyak 3x saja.

5. Proses Input Data Login Oleh User

```
printf("Login Attemp : %d\n", i);  
printf("Enter Your PIN Number : ");  
scanf("%d", &pin);
```

Untuk login, user memasukan nomor PIN yang dimilikinya.

Penyeleksian Kondisi dan Perulangan

6. Proses Autentikasi User

```

if(pin == 1234) {
    printf("\n\nLogin Success!!!\n\n");
    break;
}
else if(i > 1) {
    printf("\n\nWrong PIN..\n\n");
}
else if(i == 1) {
    printf("\n\nHack Attemp!!!\n\n");
}

```

- Jika nomor PIN yang dimasukan oleh user adalah “1234”, maka user berhasil untuk login ke sistem.
- Jika nomor PIN yang dimasukan oleh user bukan “1234”, maka akan ditampilkan pesan kesalahan pada layar. → (“Wrong PIN”)
- Jika pada kesempatan ketiga user masih salah memasukan nomor PIN, maka program akan mem-blok user tersebut sehingga tidak dapat mencoba login kembali ke sistem.

7. Kode Program Keseluruhan

```

#include <stdio.h>
#include <stdlib.h>

int main()
{
    int pin,i;

    for(i=3; i>=1; i--) {
        printf("Login Attemp : %d\n",i);
        printf("Enter Your PIN Number : ");
        scanf("%d",&pin);
        if(pin == 1234) {
            printf("\n\nLogin Success!!!\n\n");
            break;
        }
        else if(i > 1) {
            printf("\n\nWrong PIN..\n\n");
        }
        else if(i == 1) {
            printf("\n\nHack Attemp!!!\n\n");
        }
    }
    return 0;
}

```

Penyeleksian Kondisi dan Perulangan

8. Ilustrasi Program

```
Login Attemp : 3
Enter Your PIN Number : 377

Wrong PIN..

Login Attemp : 2
Enter Your PIN Number : 355

Wrong PIN..

Login Attemp : 1
Enter Your PIN Number : 1235

Hack Attempt!!!
```

```
Login Attemp : 3
Enter Your PIN Number : 33

Wrong PIN..

Login Attemp : 2
Enter Your PIN Number : 344

Wrong PIN..

Login Attemp : 1
Enter Your PIN Number : 1234

Login Success!!!
```

7. Laporan

Kalian dapat memilih untuk membuat salah satu program di bawah ini:

- Buatlah sebuah program yang dapat memeriksa apakah suatu bilangan merupakan bilangan palindrom atau tidak.
- Buatlah bilangan yang dapat menampilkan deret bilangan Fibonacci berdasarkan inputan *user*.

Penyeleksian Kondisi dan Perulangan

Laporan diketik rapih pada kertas A4 dan dikumpulkan pada saat demo program.

Format Penulisan Laporan :

- Font Tulisan : Times New Roman 12 pt
- Font Source Code : Courier New 10 pt
- Line Spacing : 1.5
- Margin : 4, 3, 3, 3 (left, top, right, bottom)
- Page Numbering :
 - a). Cover = tanpa halaman
 - b). Kata Pengantar s/d sebelum BAB I = Center Bottom, angka romawi kecil (i,ii,iii)
 - c). BAB = Center Bottom, angka arab (1, 2, 3)
 - d). Bagian BAB = Top Right, angka arab (1, 2, 3)
- Isi Laporan :
 - BAB I PENDAHULUAN
(Latar Belakang, Tujuan, Manfaat)
 - BAB II LANDASAN TEORI
 - BAB III PEMBAHASAN
(Tugas Pendahuluan dan Tugas Praktikum)
 - BAB IV PENUTUP
(Kesimpulan dan Saran)