



PEMILIHAN

MODUL #4

Asisten Dasar Pemograman 2021/2022

A. Tujuan

1. Pratkan dapat memahami tentang pemilihan didalam Bahasa pemrograman C
2. Pratkan dapat mengetahui waktu yang tepat untuk menggunakan pemilihan dalam studi kasus tertentu

B. Penjelasan

Pada saat pembuatan sebuah program ada kalanya kita ingin mengeksekusi algoritma tertentu jika persyaratan yang kita tentukan terpenuhi, pada umumnya saat pembuatan program yang lebih kompleks akan sering menjumpai suatu permasalahan yang hanya dapat diselesaikan dengan menggunakan pemilihan. Pemilihan adalah suatu keadaan dimana aksi akan tereksekusi jika suatu kondisi / persyaratan yang ditentukan terpenuhi. Pada pemilihan, kondisi memiliki dua nilai yaitu bernilai *true* atau *false*, suatu aksi hanya akan dilakukan jika nilai dari kondisinya bernilai *true*. Kedua nilai ini juga mempengaruhi bagaimana nantinya program yang dibuat akan bekerja.

Operator Boolean

Operator boolean biasanya digunakan untuk membandingkan suatu kondisi / persyaratan didalam pengkondisian, hasil dari operator Boolean bernilai *true* atau *false*. Bentuk penggunaan dari operator boolean : Kondisi 1 <operator> Kondisi 2.

Operator	Nilai
==	Bernilai True jika ruas kiri dan kanan memiliki nilai yang sama Contoh: a=3, b=3 a==b → TRUE
!=	Bernilai True jika nilai di ruas kiri dan kanan berbeda Contoh: a=3, b=3 a!=b → FALSE
<	Bernilai True jika nilai di ruas kiri lebih kecil dari ruas kanan Contoh: a=3, b=7 a<b → TRUE
>	Bernilai True jika nilai di ruas kanan lebih besar dari nilai di ruas kiri Contoh: a=2, b=5

	$a > b \rightarrow \text{FALSE}$
$<=$	Bernilai True jika nilai di ruas kiri lebih kecil / sama dengan ruas kanan Contoh: $a=2, b=5, c=5$ $a <= b \rightarrow \text{TRUE}$ $c <= b \rightarrow \text{TRUE}$
$>=$	Bernilai True jika nilai di ruas kanan lebih besar / sama dengan ruas kiri Contoh: $a=2, b=5, c=5$ $a >= b \rightarrow \text{FALSE}$ $b >= c \rightarrow \text{TRUE}$
!	Bernilai True jika nilai komplemen atau kebalikan dari suatu kondisi Contoh: $a=5, b=1$ $!(b==a) \rightarrow \text{TRUE}$ $!(b < a) \rightarrow \text{FALSE}$
&& (AND)	Bernilai True jika nilai di ruas kiri dan kanan keduanya bernilai True Contoh: $a=2, b=1, c=5$ $a >= b \ \&\& \ b < c \rightarrow \text{TRUE}$ $a == c \ \&\& \ a > b \rightarrow \text{FALSE}$
 (OR)	Bernilai True jika nilai di ruas kiri dan kanan salah satunya bernilai True Contoh: $a=2, b=1$ $a > b \ \ a == b \rightarrow \text{TRUE}$ $b > a \ \ b == a \rightarrow \text{FALSE}$

Berikut ini merupakan 2 jenis pemilihan yaitu IF-ELSE dan SWITCH

A. IF-ELSE

(1) Satu Kasus

```

if (<kondisi>) {

    <aksi>;

}

```

Jadi aksi akan dieksekusi jika kondisi dapat terpenuhi, namun jika kondisi tidak terpenuhi maka aksi tidak akan tereksekusi.

(2) Dua Kasus

```
if (<kondisi>) {  
    <aksi1>;  
} else {  
    <aksi2>;  
}
```

Jadi aksi1 akan dieksekusi jika kondisi terpenuhi, sedangkan jika kondisi tidak terpenuhi maka akan masuk ke bagian else dimana aksi2 yang akan dieksekusi.

(3) Tiga kasus atau lebih

```
if (<kondisi1>) {  
    <aksi1>;  
} else if (<kondisi2>) {  
    <aksi2>;  
} else {  
    <aksi3>;  
}
```

Jadi aksi1 akan tereksekusi jika kondisi1 terpenuhi dan pemilihan akan berhenti di aksi1, namun jika kondisi1 tidak terpenuhi program akan memeriksa kondisi2 pada else if, namun jika tidak terpenuhi juga maka langsung mengakses else dan mengeksekusi aksi3.

B. SWITCH

Jenis pemilihan kedua yaitu switch case yang biasanya digunakan untuk kasus pemilihan yang lebih banyak, agar baris code yang dibuat lebih sederhana dan rapih. Switch case ini sering juga digunakan untuk membuat menu-menu pada program.

```
Switch(<variabel>){  
  
    case 1:  
  
        aksi1;  
  
        break;  
  
    case 2:  
  
        aksi2;  
  
        break;  
  
    default:  
  
        aksi3;  
  
        break;  
  
}
```

break di atas berfungsi untuk memaksa program keluar dari suatu kondisi pada pemilihan agar aksi pada kondisi selanjutnya tidak tereksekusi.

STRING

```
#include <stdio.h>  
  
#include <stdlib.h>  
  
#include <string.h>  
  
  
typedef char string[30];
```

Pada Bahasa pemrograman C tipe data *string* merupakan tipe data bentukan, dimana dibentuk dari tipe data *char* dengan mendeklarasikan typedef char string[<max karakter>] di header. Pada pemilihan akan erat kaitannya dengan suatu pengkondisian bertipe data *string*, namun untuk bisa menggunakan fungsi-fungsi yang akan membantu dalam perbandingan didalam pengkondisian ini perlu memanggil *library* #include <string.h> pada header. Didalam *library* ini terdapat beberapa fungsi yaitu strcpy, strlen, strcmp dan strcmpi.

1. STRCPY

```
strcpy(variabelA, variabelB);  
strcpy(variabelA, "kata");
```

Fungsi `strcpy` merupakan *string copy*, yang berfungsi untuk menyalin sebuah *string* kedalam suatu variabel diruas kirinya. Seperti contoh di atas menyalin nilai dari variabelB kedalam variabelA dan memasukan string “kata” kedalam variabelA. Bisa dibayangkan seperti pada variabel dengan tipe data integer yaitu $a = 5$ dimana artinya memasukan nilai 5 kedalam variabel a.

2. STRLEN

```
if(strlen(variabel) == 5){  
    <kondisi>  
}  
variabelTampung = strlen(variabel);
```

Fungsi `strlen` merupakan *string length*, yang berfungsi untuk dapat mengetahui berapa jumlah karakter yang terdapat dalam sebuah *string* atau variabel dengan tipe data *string*. Penggunaan fungsi `strlen` ini akan jauh lebih efektif jika menempel pada suatu variabel / dijadikan sebagai kondisi pada pemilihan.

```
string nama;  
int jumlahChar=0;  
strcpy(nama, "udin");  
jumlahChar = strlen(nama);
```

Seperti contoh di atas `jumlahChar` yang semula bernilai 0 menjadi bernilai 4 setelah diisi dengan `strlen(nama)` yang memiliki nilai 4 karena *value* nama adalah “udin” yang terdiri dari 4 karakter.

3. STRCMP & STRCMPI

```
string kata1, kata2;
strcpy(kata1, "PEMILIHAN");
strcpy(kata2, "pemilihan");

if(strcmpi(kata1, kata2)==0) {
    printf("Kedua kata sama");
} else {
    Printf("Kedua kata berbeda");
}
```

Fungsi strcmp dan strcmpi sama-sama berfungsi untuk membandingkan dua buah *string*. Strcmpi merupakan *string compare insensitive* dimana fungsi ini akan menganggap kedua *string* sama jika semua karakter di variabel kata1 dan kata2 identik, tidak peduli dengan huruf kapital atau tidak. Sedangkan strcmp merupakan *string compare* dengan *case sensitive*, artinya fungsi akan menganggap kedua string sama / *true* jika semua karakter benar-benar identik termasuk memperhatikan huruf kapitalnya. Kedua fungsi ini sering digunakan sebagai kondisi di perulangan. Seperti contoh diatas akan mengeluarkan output “Kedua kata sama” karena menggunakan *case insensitive* yaitu strcmpi.

```
if(strcmp(variabelA, variabelPembanding)==0)
```

Akan bernilai *True* jika variabelA dan variabelPembanding memiliki isi *string* yang sama.

```
if(strcmp(variabelA, variabelPembanding)!=0)
```

Akan bernilai *True* jika variabelA dan variabelPembanding memiliki isi string yang berbeda.

GUIDED

1. IF-ELSE

Buatlah program yang meminta inputan user berupa **nama**, **hobi** dan **umur**. Input nama dan hobi **tidak boleh kosong**, untuk inputan umur harus **lebih dari 0**. Jika user tidak menginput sesuai dengan kriteria tersebut tampilkan pesan *error*. Jika inputan user sesuai maka program akan menampilkan data-data yang diinputkan tadi. Kemudian program meminta user menginputkan **kode** A23 atau B45, jika user tidak menginputkan salah satu dari kode tersebut tampilkan *error*.

2. Switch case

Buatlah program yang memiliki 3 menu, yaitu menu login, input data dan menu tampil data. Menu pertama merupakan menu login, program ini bisa dimasuki oleh 2 user, yaitu pratikan dan admin.

Menu login **tidak bisa** diakses lagi jika user sudah login. Menu kedua yaitu menu input data, di menu ini user diminta menginputkan **nama** dan **npm**. Menu input data **tidak bisa** diakses jika user belum login atau sudah menginputkan data.

Menu ketiga merupakan menu tampil data, menu ini **hanya bisa** diakses jika user sudah login dan sudah input data. Di menu ini akan menampilkan nama dan npm yang sudah diinput oleh user, program akan meminta user memilih untuk menghapus data atau tidak. Jika user memilih menghapus data maka menu kedua bisa diakses kembali.

1.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <conio.h>
5
6  typedef char string[30];
7
8  // Nama pratikan / NPM / Kelas
9
10 int main(){
11     string nama, hobi, kode;
12     int umur;
13
14     printf("----== GD PEMILIHAN IF-ELSE ==----");
15     printf("\nNama : "); fflush(stdin); gets(nama);
16     printf("hobi : "); fflush(stdin); gets(hobi);
17     printf("Umur : "); scanf("%d", &umur);
18
19     /* fungsi strlen merupakan string length
20        digunakan untuk menghitung jumlah karakter suatu string
21        contoh: "Menggambar", jumlah karakter yaitu 10 */
22     if (strlen(nama)>0 && strlen(hobi)>0 && umur>0) {
23
24         /* parameter didalam if(<parameter>) merupakan kondisi yang harus dipenuhi
25            agar kode/aksi didalam pemilihan bisa dijalankan. */
26
27         /* operato && (and) bertanda bahwa semua kondisi harus bernilai true
28            jika salah satu bernilai false maka code didalam pemilihan
29            tidak akan dijalankan dan menjalankan code dalam else */
30
31         printf("\n\tNama : %s", nama);
32         printf("\n\tHobi : %s", hobi);
33         printf("\n\tUmur : %d", umur);
34
35         printf("\n\tKode [A23 | B45]: "); fflush(stdin); gets(kode);
36
37         /* strcmp merupakan string compare, untuk membandingkan dua string
38            jika kondisi di parameter if tidak terpenuhi maka akan
39            melanjutkan pengecekan ke else if, jika juga tidak terpenuhi
40            program akan menjalankan kode di bagian else */
41         if (strcmpi(kode, "A23")==0) {
42             printf("\n\tSelamat Kamu Menjadi gold member...");
43         } else if (strcmpi(kode, "B45")==0){
44             printf("\n\tSelamat Kamu Menjadi silver member...");
45         } else {
46             printf("\n\tKode salah [!]");
47         }
48
49     } else {
50         printf("\nSemua data tidak boleh kosong[!]");
51     }
52
53     return 0;
54 }
55
```

2.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <conio.h>
5  #include <stdbool.h>
6
7  typedef char string[30];
8
9  // Nama pratikan / NPM / Kelas
10
11 int main (int argc, char *argv[]){
12
13     string user, nama, npm, jawab;
14     int pil;
15     bool menu = true;
16
17     /* inisialisasi logika program, untuk kondisi jika belum login
18     nilai dari variabel cekLog tidak harus 0 bebas dengan nilai apapun
19     yang terpenting pengelolaan logika ketika berhasil login. */
20     int cekLog=0;
21
22     // inisialisasi logika program jika user belum input data
23     int cekInput=0;
24
25     /* while merupakan perulangan, khusus pada materi pemilihan ini digunakan
26     untuk membantu pembuatan menu-menu pada program, untuk lebih jelasnya
27     mengenai perulangan akan ada pada materi selanjutnya, jadi mohon di ikuti saja */
28
29     while(menu==true){
30
31         system("cls"); // coba jalankan program tanpa fungsi system("cls"), bandingkan.
32         printf("\n\t----- GD PEMILIHAN SWITCH CASE -----");
33         printf("\n\t1. Login");
34
35         printf("\n\t2. Input Data");
36         printf("\n\t3. Tampil Data");
37         printf("\n\t0. Keluar");
38         printf("\n\tPilih menu > "); scanf("%d", &pil);
39
40         switch(pil){
41             case 1:
42                 if( cekLog==0 ) { // kondisi jika belum login
43
44                     printf("\n\t[LOGIN]");
45                     printf("\n\tUser : "); fflush(stdin); gets(user);
46
47                     if( strcmpi(user, "(nama pratikan)")==0 || strcmpi(user, "Admin")==0 ) {
48                         // contoh strcmpi(user, "nessa")
49                         printf("\n\t\tBerhasil login...");
50
51                         // merubah logika program jika berhasil login
52                         cekLog=1;
53
54                     } else printf("\n\t\tUser tidak ditemukan[!]");
55
56                     // mengeksekusi else jika program sudah login
57                     } else printf("\n\t\tAnda Sudah Login[!]");
58
59                 break;
60
61             case 2:
62                 if(cekLog==0) printf("\n\t\tAnda harus login terlebih dahulu[!]");
63                 else if(cekInput==1) printf("\n\t\tAnda sudah input data[!]");
64                 else{
65                     printf("\n\t[Input Data]");
66                     printf("\n\tNama : "); fflush(stdin); gets(nama);
67                     printf("\n\tNPM : "); fflush(stdin); gets(npm);
```

```

67
68     printf("\n\t\tBerhasil input data...");
69
70     // merubah logika input menjadi cekInput=1 jika user sudah input data
71     cekInput=1;
72 }
73 break;
74
75 case 3:
76     if(cekLog==0) printf("\n\t\tAnda harus login terlebih dahulu[!];");
77     else if(cekInput==0) printf("\n\t\tAnda belum inputa data[!];");
78     else{
79         printf("\n\t[Tampi data]");
80         printf("\n\tNama   : %s", nama);
81         printf("\n\tNPM    : %s", npm);
82
83         printf("\n\tIngin hapus data ini? [ya|tidak] : ");
84         fflush(stdin); gets(jawab);
85
86         if(strcmpi(jawab, "ya")==0){
87             cekInput=0;
88             printf("\n\t\tData berhasil terhapus...");
89         }
90     }
91     break;
92
93 case 0:
94     printf("\n\t\tKeluar dari program...");
95     menu = false;
96     break;
97
98 default:
99     printf("\n\t\tMenu tidak ada[!];");
100    break;
101
102 }getch(); // coba jalankan program tanpa fungsi getch(), bandingkan
103
104 }
105
106 return 0;
107 }

```

Note: komentar di dalam program tidak harus dibuat

Ketentuan Pengerjaan:

1. Pastikan ekstensi program file adalah .c **bukan** .cpp
2. Guided dikerjakan dalam folder terpisah (GD4_X_YYYYYY_Z.zip) lalu di zip
3. Kedua folder zip dimasukan kedalam 1 folder yang sama (GD4_X_YYYYYY.zip) lalu di zip

Y = Kelas

YYYYYY = 5 digit terakhir NPM

Z = Nomor soal Guided