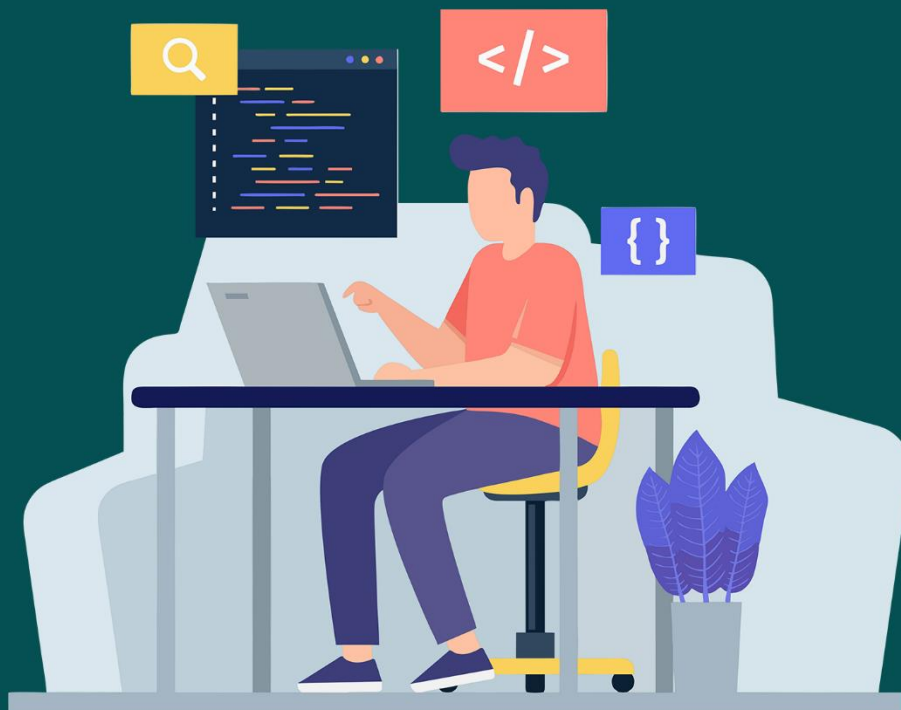


Modul Praktikum  
Algoritma dan Pemrograman I

# SWITCH CASE & TIPE DATA TERSTRUKTUR



TIM ASISTEN PEMROGRAMAN  
ANGKATAN 12

Departemen Pendidikan Ilmu Komputer  
Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam  
Universitas Pendidikan Indonesia  
2021

## SWITCH CASE

**Switch Case** atau biasa disingkat **Switch** merupakan salah satu jenis *flow control* percabangan. Dimana percabangan memungkinkan pemilihan perintah yang akan dijalankan sesuai dengan kondisi tertentu yang menentukan alur perjalanan program. Percabangan bertujuan untuk memilih atau mengondisikan apakah statement tersebut akan dijalankan atau tidak.

## KARAKTERISTIK SWITCH

- Pada **SWITCH**, kondisi yang dinyatakan harus berupa bilangan bulat, karakter, atau string. Berbeda dengan **IF-ELSE**, dimana kondisi yang dinyatakan dapat disertai dengan operator perbandingan ( **<**, **>**, **<=**, **>=**, dan **==** )
- Setiap case berupa konstanta atau nilai tetap
- Tidak ada nilai konstanta yang sama dalam sebuah SWITCH
- Angka yang digunakan dalam case berkisar antara 0-255
- Untuk mengakhiri 1 statement atau case pada SWITCH diperlukan **BREAK**
- Digunakan untuk mengecek suatu kondisi yang memiliki banyak kemungkinan nilai tunggal

## PENULISAN SWITCH CASE

```
1  #include <stdio.h>
2
3  int main(){
4      int kondisi;
5      switch (kondisi)
6      {
7          case konstanta1:
8              /* perintah yg dijalankan jika kondisi sama dgn konstanta1 */
9              break;
10         case konstanta2:
11             /* perintah yg dijalankan jika kondisi sama dgn konstanta2 */
12             break;
13         default:
14             /* perintah yang dijalankan jika tidak ada konstanta yg
15              bernilai sama dgn kondisi (tidak ada case yg memenuhi) */
16             break;
17     }
18     return 0;
19 }
```

### Keterangan

- **Kondisi** : variabel yang ingin dicek kemungkinan nilainya
- **Konstanta** : nilai tetap yang mungkin dari sebuah kondisi/variabel
- **Case** : pernyataan kasus yang mungkin terjadi pada sebuah kondisi
- **Perintah** : kode-kode yang ingin dijalankan saat ada kasus yang memenuhi kondisi
- **Break** : penutup atau penanda selesainya sebuah kasus
- **Default** : kasus general diluar seluruh kasus yang telah didefinisikan

## CONTOH PROGRAM 1

Program :

```
1  #include <stdio.h>
2
3  int main(){
4      char grade;
5      printf("ENTER YOUR GRADE HERE : ");
6      scanf(" %c", &grade);
7      switch (grade)
8      {
9          case 'A':
10             printf("PERFECT!\n");
11             break;
12          case 'B':
13             printf("GOOD!\n");
14             break;
15          case 'C':
16             printf("NOT BAD!\n");
17             break;
18          case 'D':
19             printf("TRY HARDER!\n");
20             break;
21          case 'E':
22             printf("OH MY.. LET'S TAKE A REMEDIAL\n");
23          default:
24             printf("YOUR GRADE IS UNDEFINED!\n");
25             break;
26      }
27      return 0;
28 }
```

Output :

ENTER YOUR GRADE HERE : A  
PERFECT!

ENTER YOUR GRADE HERE : D  
TRY HARDER!

ENTER YOUR GRADE HERE : Z  
YOUR GRADE IS UNDEFINED!

## CONTOH PROGRAM 2

Program :

```
1  #include <stdio.h>
2
3  int main(){
4      int bil1 = 0, bil2 = 0;
5      char operator;
6      // bil1 - operator - bil2
7      scanf("%d %c%d", &bil1, &operator, &bil2);
8      switch (operator)
9      {
10     case '+':
11         printf("= %d\n", bil1 + bil2);
12         break;
13     case '-':
14         printf("= %d\n", bil1 - bil2);
15         break;
16     case '*':
17         printf("= %d\n", bil1 * bil2);
18         break;
19     case '/':
20         printf("= %d\n", bil1 / bil2);
21         break;
22     case '%':
23         printf("= %d\n", bil1 % bil2);
24     default:
25         printf("Undefined\n");
26         break;
27     }
28     return 0;
29 }
```

Output :

5 + 4  
= 9

11 \* 2  
= 22

10 / 4  
= 2

2 - 2  
= 0

6 x 2  
Undefined

## TIPE DATA TERSTRUKTUR

Tipe data terstruktur (struct) merupakan tipe data bentukan yang di dalamnya menampung kumpulan dari beberapa variabel dengan beragam tipe data lain yang dibungkus menjadi satu.

## PENULISAN TIPE DATA TERSTRUKTUR

```
1  #include <stdio.h>
2
3  typedef struct {
4      tipeData namaVariabel;
5      tipeData namaVariabel;
6  } namaStruct;
7
8  int main(){
9      /* Code */
10     return 0;
11 }
```

```
1  #include <stdio.h>
2
3  typedef struct {
4      int bil1, bil2;
5      char operator;
6  } namaStruct;
7
8  int main(){
9      /* Code */
10     return 0;
11 }
```

Keterangan

- Pendefinisian struct diletakkan di atas blok main, di bawah pemanggilan library
- Variabel dalam struct dapat memiliki tipe data yang berbeda

## CONTOH PROGRAM 3

Program :

```
1  #include <stdio.h>
2
3  typedef struct {
4      int nim;
5      char grade;
6  } mhsAlpro;
7
8  int main(){
9      mhsAlpro mhs1;
10     mhsAlpro mhs2;
11
12     printf("Masukkan data mahasiswa 1\n");
13     printf("NIM : ");
14     scanf("%d", &mhs1.nim);
15     printf("Grade : ");
16     scanf(" %c", &mhs1.grade);
17     if(mhs1.grade > 'D') {
18         printf("Mahasiswa %d LULUS!\n", mhs1.nim);
19     } else {
20         printf("Mahasiswa %d TIDAK LULUS!\n", mhs1.nim);
21     }
22 }
```

```

23     printf("Masukkan data mahasiswa 2\n");
24     printf("NIM : ");
25     scanf("%d", &mhs2.nim);
26     printf("Grade : ");
27     scanf(" %c", &mhs2.grade);
28     if(mhs2.grade > 'D') {
29         printf("Mahasiswa %d LULUS!\n", mhs2.nim);
30     } else {
31         printf("Mahasiswa %d TIDAK LULUS!\n", mhs2.nim);
32     }
33     return 0;
34 }

```

Output :

```

Masukkan data mahasiswa 1
NIM : 2007703
Grade : A
Mahasiswa 2007703 LULUS!
Masukkan data mahasiswa 2
NIM : 2002223
Grade : D
Mahasiswa 2002223 TIDAK LULUS!

```

```

Masukkan data mahasiswa 1
NIM : 2000111
Grade : E
Mahasiswa 2000111 TIDAK LULUS!
Masukkan data mahasiswa 2
NIM : 2009121
Grade : C
Mahasiswa 2009121 LULUS!

```

## CONTOH PROGRAM 4

Program :

```

1  #include <stdio.h>
2
3  typedef struct {
4      int menuPilihan;
5      int jumlahPesan;
6      int totalHarga;
7      char member;
8  } customer;
9
10 int main(){
11     printf("=====\n");
12     printf(" SELAMAT DATANG DI EDUN RESTO!\n");
13     printf("=====\n");
14     printf("          DAFTAR MENU          \n");
15     printf("1. Sosis Bakar      : Rp5.000\n");
16     printf("2. Nasi Goreng      : Rp10.000\n");
17     printf("3. Boba             : Rp21.000\n");
18     printf("4. Es Teh           : Rp3.000\n");
19     printf("=====\n");

```

```

21     customer sekar;
22     sekar.totalHarga = 0;
23
24     printf("Masukkan pilihan menu Anda : ");
25     scanf("%d", &sekar.menuPilihan);
26
27     printf("Masukkan jumlah pesanan : ");
28     scanf("%d", &sekar.jumlahPesanan);
29
30     printf("Anda member Edun Resto? (y/n) ");
31     scanf(" %c", &sekar.member);

```

```

32
33     switch (sekar.menuPilihan) {
34     case 1:
35         sekar.totalHarga = sekar.jumlahPesanan * 5000;
36         break;
37     case 2:
38         sekar.totalHarga = sekar.jumlahPesanan * 10000;
39         break;
40     case 3:
41         sekar.totalHarga = sekar.jumlahPesanan * 21000;
42         break;
43     case 4:
44         sekar.totalHarga = sekar.jumlahPesanan * 3000;
45         break;
46     default:
47         printf("=====\n");
48         printf("      Menu tidak tersedia\n");
49         printf("=====\n");
50         break;
51     }
52
53     if((sekar.totalHarga > 0) && (sekar.menuPilihan >= 1) && (sekar.menuPilihan <= 4)) {
54         if(sekar.member == 'Y' || sekar.member == 'y') {
55             sekar.totalHarga = (sekar.totalHarga * 80) / 100;
56         }
57         printf("=====\n");
58         printf("Total bayar : Rp%d\n", sekar.totalHarga);
59         printf("=====\n");
60     }
61     return 0;
62 }

```

Output :

<pre> ===== SELAMAT DATANG DI EDUN RESTO! =====           DAFTAR MENU 1. Sosis Bakar      : Rp5.000 2. Nasi Goreng     : Rp10.000 3. Boba            : Rp21.000 4. Es Teh          : Rp3.000 ===== Masukkan pilihan menu Anda : 2 Masukkan jumlah pesanan : 2 Anda member Edun Resto? (y/n) y ===== Total bayar : Rp16000 ===== </pre>	<pre> ===== SELAMAT DATANG DI EDUN RESTO! =====           DAFTAR MENU 1. Sosis Bakar      : Rp5.000 2. Nasi Goreng     : Rp10.000 3. Boba            : Rp21.000 4. Es Teh          : Rp3.000 ===== Masukkan pilihan menu Anda : 2 Masukkan jumlah pesanan : 2 Anda member Edun Resto? (y/n) n ===== Total bayar : Rp20000 ===== </pre>	<pre> ===== SELAMAT DATANG DI EDUN RESTO! =====           DAFTAR MENU 1. Sosis Bakar      : Rp5.000 2. Nasi Goreng     : Rp10.000 3. Boba            : Rp21.000 4. Es Teh          : Rp3.000 ===== Masukkan pilihan menu Anda : 5 Masukkan jumlah pesanan : 2 Anda member Edun Resto? (y/n) y ===== Menu tidak tersedia ===== </pre>
--	--	--

## LATIHAN SOAL

Buatlah sebuah program berisi **tipe data terstruktur** atau bungkusan yang terdiri dari 3 buah variabel integer dan 1 variabel char. Nilai setiap variabel diisi oleh user.

Berdasarkan nilai variabel char akan dicek sebagai berikut :

- Jika char bernilai 'A' maka akan menghasilkan nilai penjumlahan dari hasil perkalian antara bilangan pertama dan kedua dengan hasil perkalian antara bilangan kedua dan ketiga
- Jika char bernilai 'B' maka akan menghasilkan nilai pembagian dari hasil perkalian antara bilangan pertama dan kedua dengan bilangan ketiga
- Jika char bernilai 'C' maka akan menghasilkan nilai perkalian dari hasil pengurangan antara bilangan kedua dan ketiga dengan bilangan pertama
- Jika char tidak bernilai 'A', 'B', maupun 'C' maka akan menghasilkan nilai penjumlahan bilangan pertama dan kedua yang kemudian dikurangi bilangan ketiga

Masukan 1 :

6 3 4

B

Keluaran 1 :

4

Masukan 2 :

6 3 4

A

Keluaran 2 :

30

Masukan 3 :

3 2 3

D

Keluaran 3 :

2





## PENUTUP

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada semua elemen yang telah mendukung berjalannya sesi praktikum pada mata kuliah Algoritma dan Pemrograman I ini. Semoga apa yang telah kita laksanakan dapat menjadi berkah dan memberikan manfaat bagi kita semua dikemudian hari.

## DAFTAR PUSTAKA

Tim Asisten Praktikum Algoritma dan Pemrograman I Angkatan 11. (2020). *Switch dan Tipe Data Terstruktur*. Modul Praktikum Algoritma dan Pemrograman I. Bandung: Universitas Pendidikan Indonesia.