ALGORITMA DAN PEMROGRAMAN

MATERI 3

❑STRUKTUR ALGORITMA

Oleh : Roni Sapto P., S. Kom.
E : ronisapto [at] gmail [dot] com
T : +62 821 16 75 93 57
W : mycampus.dezignwebster.com

STRUKTUR ALGORITMA

Macam-macam struktur algoritma :

- **RUNTUNAN (*SEQUENCE*)**
- **PEMILIHAN (*SELECTION*)**
- **PENGULANGAN (*REPETITION*)**

STRUKTUR ALGORITMA
RUNTUNAN

- **Runtunan** merupakan struktur algoritma paling dasar yang berisi rangkaian instruksi yang diproses secara sekuensial, satu per satu, mulai dari instruksi pertama sampai instruksi terakhir.
  - Tiap instruksi dikerjakan satu per satu.
  - Tiap instruksi dilaksanakan satu kali, tidak ada instruksi yang diulang.
  - Pelaksanaan instruksi mengikuti urutan pada algoritma.
  - Akhir dari intruksi terakhir merupakan akhir algoritma.

STRUKTUR ALGORITMA
PEMILIHAN

- **Pemilihan** memungkinkan kita melakukan aksi dengan suatu persyaratan tertentu.
  - Pemilihan dengan 1 kasus.
  - Pemilihan dengan 2 kasus.
  - Pemilihan dengan banyak kasus.

STRUKTUR ALGORITMA

PEMILIHAN DENGAN 1 KASUS

- Bila sebuah kondisi terpenuhi, maka sebuah aksi akan dieksekusi.
- Bila kondisi tersebut tidak terpenuhi, maka tidak ada aksi yang akan dieksekusi.
- Penulisan dalam algoritma :

```
IF (kondisi) THEN
  aksi
ENDIF
```


STRUKTUR ALGORITMA

PEMILIHAN DENGAN 2 KASUS

- Bila sebuah kondisi terpenuhi, maka sebuah aksi pertama akan dieksekusi.
- Bila kondisi tersebut tidak terpenuhi, maka aksi kedua yang akan dieksekusi.
- Penulisan dalam algoritma :

```
IF (kondisi) THEN
  { aksi pertama }
ELSE
  { aksi kedua }
ENDIF
```

STRUKTUR ALGORITMA
PEMILIHAN DENGAN BANYAK KASUS

- Pada pemilihan dengan banyak kasus, punya banyak pilihan kondisi yang bila sebuah kondisi terpenuhi, maka sebuah aksi akan dieksekusi dan aksi yang lain tidak akan dieksekusi.
- Prinsip dari pemilihan dengan banyak kasus sama dengan pemilihan dengan 2 kasus, namun aksi kedua diganti dengan pemilihan 2 kasus lagi, sehingga didapatkan pemilihan bertingkat.

STRUKTUR ALGORITMA
PEMILIHAN DENGAN BANYAK KASUS

- Penulisan dalam algoritma :

```
IF (kondisi_1) THEN
  { aksi pertama }
ELSE
  IF (kondisi_2) THEN
    { aksi kedua }
  ELSE
    IF (kondisi_3) THEN
      { aksi ketiga }
    ELSE
      { aksi keempat }
    ENDIF
  ENDIF
ENDIF
```

STRUKTUR ALGORITMA

PEMILIHAN - CONTOH ALGORITMA

```
ALGORITMA TAMPIL_GENAP_GANJIL
   {I.S. : masukkan s≥0 dan s≤100}
   {F.S. : menampilkan masukan dan ket. genap/ganjil}
DEKLARASI :
   s, k : integer

ALGORITMA :

INPUT(s)
k ← s MOD 2
IF ((s < 0 ) AND (s > 100)) THEN
  OUTPUT('Masukkan antara 0 - 99')
ELSE
  IF (k = 0) THEN
    OUTPUT(s,' adalah angka GENAP')
  ELSE
    OUTPUT(s,' adalah angka GANJIL')
  ENDIF
ENDIF
```


STRUKTUR ALGORITMA

PEMILIHAN DENGAN BANYAK KASUS (CASE)

- Pada pemilihan dengan banyak kasus, selain menggunakan IF..THEN.. bertingkat, dapat juga menggunakan perintah **CASE**
- Prinsip dari pemilihan dengan banyak kasus sama dengan pemilihan dengan 2 kasus, namun aksi kedua diganti dengan pemilihan 2 kasus lagi, sehingga didapatkan pemilihan bertingkat.
- Terdapat aksi alternatif bila semua kondisi tidak terpenuhi.

STRUKTUR ALGORITMA
PEMILIHAN - CONTOH ALGORITMA

```
ALGORITMA TAMPIL_GENAP_GANJIL
   {I.S. : masukkan s≥0 dan s≤100}
   {F.S. : menampilkan masukan dan ket. genap/ganjil}

DEKLARASI :
   s, k : integer

ALGORITMA :

INPUT(s)
k ← s MOD 2
CASE k
  0 : OUTPUT(s,' adalah angka GENAP')
  1 : OUTPUT(s,' adalah angka GANJIL')
  OTHERWISE : OUTPUT('Salah hitung ?')
ENDCASE
```


STRUKTUR ALGORITMA
PENGULANGAN (*REPETITION / LOOP*)

- Di dalam **Pengulangan** dapat dilakukan **sejumlah** kali, atau **selama** kondisi tertentu, atau **sampai** kondisi tertentu.
- Macam-macam pengulangan yang dapat digunakan :
  - **FOR**
  - **WHILE**
  - **REPEAT**

STRUKTUR ALGORITMA
PENGULANGAN - FOR

1. FOR TO DO

- Pengulangan dengan variabel perulangan bertipe data integer dengan nilai awal dan akhir yang harus kita tentukan pada awal pengulangan.
- Nilai awal lebih kecil dari pada nilai akhir.
- Setiap kali terjadi perulangan, nilai dari variabel perulangan akan bertambah 1.

STRUKTUR ALGORITMA
PENGULANGAN - FOR

- Penulisan dalam algoritma :

```
FOR variabel ← nilai_awal TO nilai_akhir DO
    { aksi }
ENDFOR
```

STRUKTUR ALGORITMA

PENGULANGAN - FOR - CONTOH ALGORITMA

```
ALGORITMA TAMPIL_GENAP_GANJIL
   {I.S. : nilai awal = 0, nilai akhir = 100}
   {F.S. : menampilkan angka disertai keterangan genap
           atau ganjil}
DEKLARASI :
   s, k : integer

ALGORITMA :

FOR s ← 0 TO 100 DO
   k ← s MOD 2
   IF (k = 0) THEN
        OUTPUT(s,' adalah angka GENAP')
   ELSE
        OUTPUT(s,' adalah angka GANJIL')
   ENDIF
ENDFOR
```


STRUKTUR ALGORITMA

PENGULANGAN - FOR

2. FOR DOWNTO DO

- Pengulangan dengan variabel perulangan bertipe data integer dengan nilai awal dan akhir yang harus kita tentukan pada awal pengulangan.
- Nilai awal lebih besar dari pada nilai akhir.
- Setiap kali terjadi perulangan, nilai dari variabel perulangan akan berkurang 1.

STRUKTUR ALGORITMA
PENGULANGAN - FOR

- Penulisan dalam algoritma :

```
FOR variabel ← nilai_awal DOWNTO nilai_akhir DO
   { aksi }
ENDFOR
```


STRUKTUR ALGORITMA
PENGULANGAN - FOR - CONTOH ALGORITMA

```
ALGORITMA TAMPIL_GENAP_GANJIL
   {I.S. : nilai awal = 100, nilai akhir = 0}
   {F.S. : menampilkan angka disertai keterangan genap
           atau ganjil}
DEKLARASI :
   s, k : integer

ALGORITMA :

FOR s ← 100 DOWNTO 0 DO
   k ← s MOD 2
   IF (k = 0) THEN
        OUTPUT(s,' adalah angka GENAP')
   ELSE
        OUTPUT(s,' adalah angka GANJIL')
   ENDIF
ENDFOR
```

STRUKTUR ALGORITMA
PENGULANGAN - WHILE

- **WHILE DO**
  - Pengulangan dengan sebuah kondisi sebagai parameter pengulangan.
  - Pengulangan akan terjadi **selama kondisi terpenuhi**.
  - Pengecekan kondisi dilakukan di **awal** pengulangan.
  - Ada kemungkinan aksi tidak pernah dieksekusi.
- Penulisan dalam algoritma :

```
WHILE (kondisi) DO
    { aksi }
ENDWHILE
```


STRUKTUR ALGORITMA
PENGULANGAN - WHILE - CONTOH ALGORITMA

```
ALGORITMA TAMPIL_GENAP_GANJIL
   {I.S. : nilai awal = 100, nilai akhir = 0}
   {F.S. : menampilkan angka disertai keterangan genap
           atau ganjil}
DEKLARASI :
   s, k : integer

ALGORITMA :

s ← 0
WHILE (s ≤ 100) DO
   k ← s MOD 2
   IF (k = 0) THEN
        OUTPUT(s,' adalah angka GENAP')
   ELSE
        OUTPUT(s,' adalah angka GANJIL')
   ENDIF
   S ← s + 1
ENDWHILE
```

STRUKTUR ALGORITMA
PENGULANGAN - REPEAT

- **REPEAT UNTIL**
  - Pengulangan dengan sebuah kondisi sebagai parameter pengulangan.
  - Pengulangan akan terjadi **sampai kondisi terpenuhi**.
  - Pengecekan kondisi dilakukan di **akhir** pengulangan.
  - Minimal aksi dieksekusi satu kali.
- Penulisan dalam algoritma :

```
REPEAT
    { aksi }
UNTIL (kondisi)
```


STRUKTUR ALGORITMA
PENGULANGAN - REPEAT - CONTOH ALGORITMA

```
ALGORITMA TAMPIL_GENAP_GANJIL
   {I.S. : nilai awal = 100, nilai akhir = 0}
   {F.S. : menampilkan angka disertai keterangan genap
           atau ganjil}
DEKLARASI :
   s, k : integer

ALGORITMA :

s ← 0
REPEAT
   k ← s MOD 2
   IF (k = 0) THEN
        OUTPUT(s,' adalah angka GENAP')
   ELSE
        OUTPUT(s,' adalah angka GANJIL')
   ENDIF
   S ← s + 1
UNTIL (s > 100)
```