

STRUKTUR DASAR ALGORITMA

TEMU 4



Struktur Dasar Algoritma

1. Struktur Sequence (Runtunan)
2. Struktur Selection (Pemilihan)
3. Struktur Repetition (Perulangan)

Struktur Sequence (Runtunan)

- Algoritma sekuensial atau biasa disebut algoritma runtunan merupakan salah satu struktur dasar algoritma yang bisa dikatakan cukup sederhana jika dibandingkan dengan struktur algoritma yang lain.
- Algoritma sekuensial bekerja dengan cara mengeksekusi setiap instruksi secara berurutan. Setiap instruksi akan dikerjakan satu per satu pada setiap barisnya dari awal hingga akhir, sesuai dengan urutan penulisan instruksi tersebut.

Struktur Sequence (Runtunan)

- Instruksi dikerjakan secara berurutan baris perbaris mulai dari baris pertama hingga baris terakhir, tanpa ada loncatan atau perulangan.
 - Tiap instruksi dikerjakan satu per satu.
 - Tiap instruksi dilaksanakan tepat sekali, tidak ada instruksi yang diulang.
 - Urutan instruksi yang dilaksanakan pemroses sama dengan urutan instruksi sebagai tertulis di dalam teks algoritma.
 - Akhir dari instruksi terakhir merupakan akhir algoritma.

Contoh :

- Terdapat beberapa instruksi dalam sebuah program seperti berikut :
 - instruksi 1
 - instruksi 2
 - instruksi 3
 - nstruksi ...n
- Algoritma sekuensial akan mengeksekusi instruksi 1, kemudian setelah itu instruksi 2 dan seterusnya. Setiap instruksi dikerjakan satu persatu dan hanya sekali (tidak ada instruksi yang diulang) sampai instruksi ke - n sebagai instruksi terakhir merupakan akhir dari proses algoritmanya.
- Algoritma Sekuensial biasanya digunakan untuk program yang sederhana seperti program menghitung atau program lainnya yang hanya terdiri dari proses komputer dan input/output dari user.

Algoritma Penjumlahan Dua Bilangan

- **Bahasa Natural**

Mulai

Inisialisasi variabel a, b
dan hasil

Masukan nilai a & b

Hitung $hasil = a + b$;

Tampilkan nilai hasil

Selesai

- **Flowchart**



Algoritma Penjumlahan Dua Bilangan

- **Pseudocode**

algoritma penjumlahan_dua_bilangan

deklarasi:

```
var a:b:hasil:integer;
```

Deskripsi:

```
a <- 6;
```

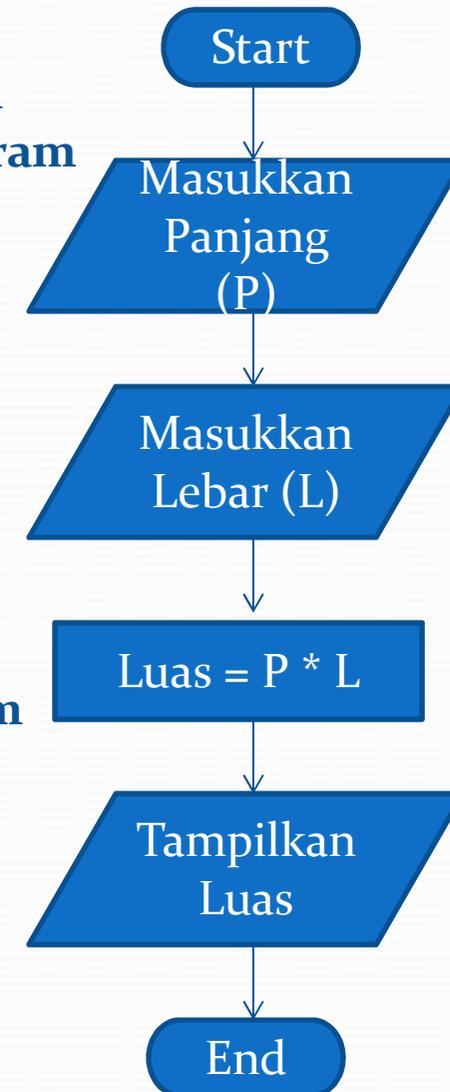
```
b <- 2;
```

```
hasil <- a+b;
```

```
write (hasil);
```

Contoh : Struktur sequence (Runtunan)

1. Program hitung_luas_persegipanjang; } Judul Program
 2. Var
 3. P, L, Luas : integer; } Deklarasi variabel
 4. Begin
 5. Write('Masukkan Panjang');
 6. Read(P);
 7. Write('Masukkan Lebar');
 8. Read(L);
 9. Luas := P * L;
 10. Write('Luas Persegi Panjang =',Luas);
 11. Readln;
 12. End.
- Badan Program



Keterangan contoh

- Keterangan :
 - Setiap baris program diatas akan diproses secara urut, mulai dari baris 1 sampai dengan baris 12
 - Setiap baris proses tidak ada yang dikerjakan lebih dari 1 kali atau tidak dikerjakan/dilewati

Contoh urutan instruksi yang berbeda tetapi tidak mempengaruhi hasil.

Deklarasi :

A, B, C, D : integer

Deskripsi :

1. Read (A, B)
2. $C \leftarrow A + B$
3. $D \leftarrow A * B$
4. Write (C, D)

Deklarasi :

A, B, C, D : integer

Deskripsi :

1. Read (A, B)
2. $D \leftarrow A * B$
3. $C \leftarrow A + B$
4. Write (C, D)

Dari kedua algoritma hasil C dan D adalah Sama

Tetapi jika algoritma di atas urutan no. 1,2 dan 3 diubah (ditukarkan) maka menghasilkan keluaran yang berbeda (tidak sama) seperti berikut :

Deklarasi :

A, B, C, D : integer

Deskripsi :

$C \leftarrow A + B$

$D \leftarrow A * B$

read (A, B)

write (C, D)

Hasil C dan D akan berbeda dengan dua algoritma sebelumnya