RUNTUNAN DAN PEMILIHAN

Runtunan adalah...

Rangkaian instruksi yang diproses secara sequential (berurutan), satu persatu mulai dari instruksi pertama sampai terakhir

Algoritma dan Runtunan

Algoritma merupakan runtunan satu atau lebih instriksi yang berarti bahwa :

- tiap instruksi dikerjakan satu persatu
- tiap instruksi hanya dijalankan satu kali tidak ada perulangan
- rrutan instruksi yang dilaksanakan pemroses sesuai dengan algoritmanya
- akhir instruksi = akhir algoritma

Contoh Runtunan : Mencetak "*Hello, World*"

```
PROGRAM Hello_world
{Program untuk mencetak "Hello, World"}

DEKLARASI
  {tidak ada}

ALGORITMA:
  write('Hello, world')
```

Menggunakan Variabel

```
PROGRAM Hello_world
{Program untuk mencetak “Hello,world”}

DEKLARASI
 pesan  : string

ALGORITMA:
 Pesan ← “Hello, world”
 write(pesan)
```

Menggunakan Konstanta

```
PROGRAM Hello_world
{Program untuk mencetak “Hello,world”}

DEKLARASI
 const pesan  = ‘Hello, world’

ALGORITMA:
 write(pesan)
```

Membaca Input

```
PROGRAM Halo_Nama
  {Mencetak string 'Halo ' dan diikuti dengan
  nama orang. Nama orang dibaca dari keyboard}
DEKLARASI
  nama : string
ALGORITMA
  read(nama)
  write('Hallo: ', nama)
```

Algoritma Pertukaran

Buatlah sebuah program yang dapat membaca nilai 2 peubah (*variable*) dan menukarkannya.

Misalkan:

a = 8 dan b = 5

Setelah proses a= 5 dan b= 8

Penyelesaian

```
PROGRAM Pertukaran
{Mempertukarkan nilai A dan B. Nilai A dan B dibaca terlebih dahulu.}

DEKLARASI
  A, B, C : integer

ALGORITMA:
   {asumsikan A dan B sudah terdefinisi dengan nilai, misalnya melalui
   pengisian langsung atau dibaca nilainya dari keyboard}
  C ← A {simpan nulai A di tempat penampungan sementara, C}
  A ← B {sekarang A dapat diisi dengan nilai B}
  B ← C {isi B dengan nilai A semula yang tadi disimpan di C}
  {Tulis nilai A dan B setelah pertukaran, jika diperlukan}
```

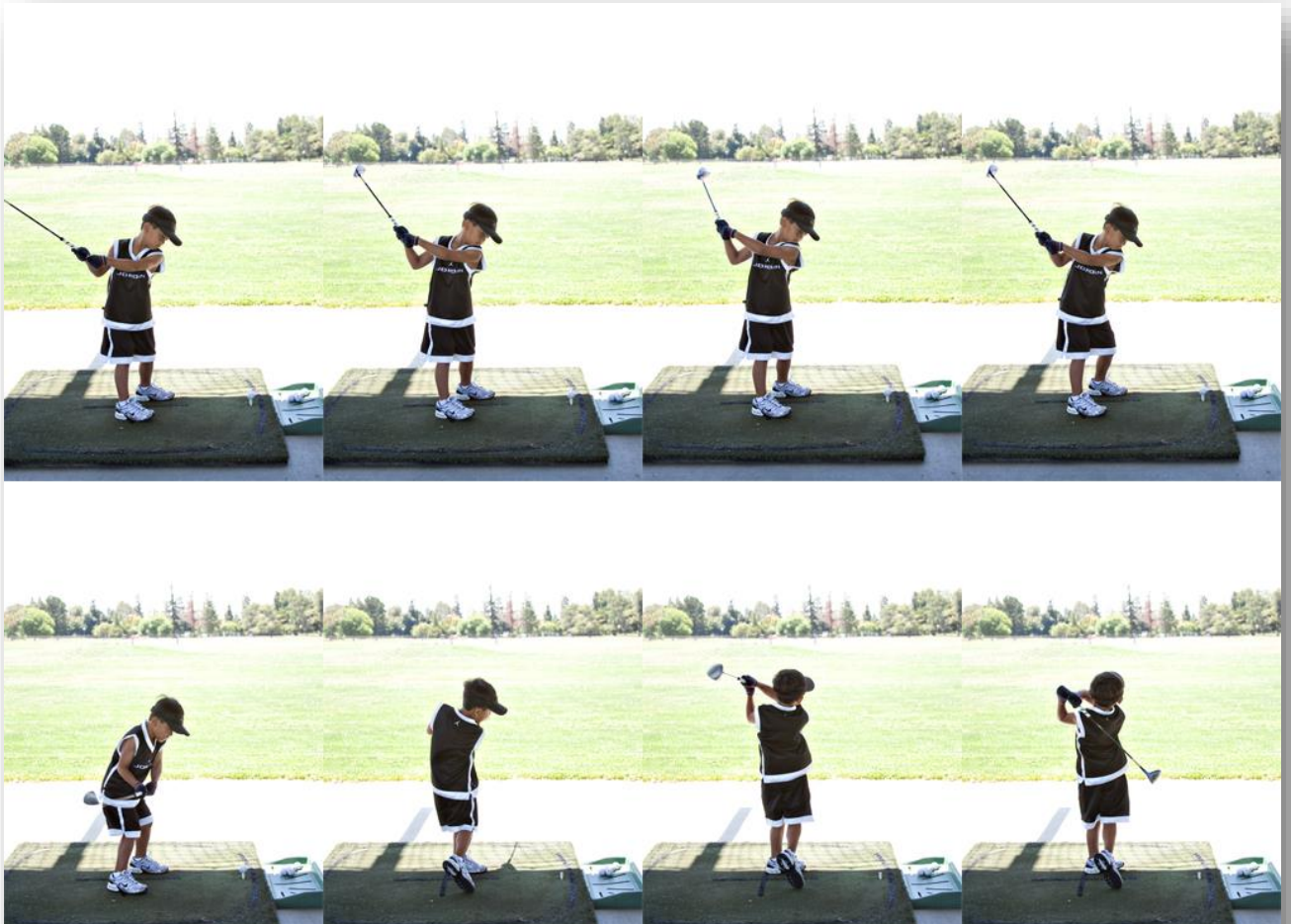
Menuliskan ke dalam Bahasa Pemrograman : PASCAL

```
program Pertukaran
{Mempertukarkan nilai A dan B. Nilai A dan B dibaca terlebih
dahulu.}
{*DEKLARASI*}
var
  A, B, C = integer;
{*ALGORITMA :*}
begin
      {baca nilai A dan B}
   write('A=?'); readln(A);
   write('B=?'); readln(B);
      {pertukarkan nilai A dan B}
   C := A;
   A := B;
   B := C;
      {tulis nilai A dan B setelah dipertukarkan}
  writeln('A= ', A);
  writeln('B= ', B);
end.
```

Menuliskan ke dalam Bahasa Pemrograman : C

```
/* PROGRAM Pertukaran */
/* Mempertukarkan nilai A dan B. Nilai A dan B dibaca terlebih
   dahulu. */
#include <stdio.h>
main()
{
/* DEKLARASI */
   int A, B, C;
/* ALGORITMA : */
        /* baca nilai A dan B */
   printf("A=?"); scanf("%d", $A);
   printf("B=?"); scanf("%d", $B);
        /* pertukarkan nilai A dan B */
   C = A;
   A = B;
   B = C;
        /* tulis nilai A dan B setelah dipertukarkan */
   printf("A= %d\n", A);
   printf("B= %d\n", B);
}
```

Contoh Algoritma Runtunan



Apa yang dilakukan oleh potongan algoritma dibawah ini?

```
ALGORITMA:
...
A ← A + B
B ← A - B
A ← A – B
...
```

Pertukaran Tanpa Peubah Bantu

```
PROGRAM Tukar

{Mempertukarkan nilai A dan B yang bertipe bilangan bulat tanpa
peubah bantu. Nilai A dan B dibaca terlebih dahulu.}


DEKLARASI
  A : integer     {nilai pertama}
      B : integer {nilai kedua}


ALGORITMA:
   read(A,B)      {baca nilai A dan B}
   write(A,B)     {tulis nilai A dan B sebelum pertukaran}

   {proses pertukaran}
   A ← A + B
   B ← A - B
   A ← A - B
  write(A,B)      {tulis nilai A dan B setelah pertukaran}
```

Menghitung Komisi yang Diterima Salesman

```
PROGRAM Komisi_Salesman
{menghitung komisi yang diterima salesman. Besar komisi adalah 5%
  dari nilai penjualan yang dicapainya. Data masukan adalah nama
  salesman dan nilai penjualannya. Keluaran algoritma adalah nama
  salesman dan besar komisi yang diterima salesman tersebut.}
DEKLARASI
   NamaSalesman   : string
   NilaiPenjualan       : real      {nilai penjualan yang dicapai}
   komisi         : real      {besar komisi}
ALGORITMA
   read(NamaSalesman, NilaiPenjualan)
   komisi ← 0.05 * NilaiPenjualan
   write(NamaSalesman, komisi)
```

Wartel

```
PROGRAM Wartel
{Menghitung biaya percakapan di warung telekomunikasi. Masukan adalah waktu
awal dan waktu selesai percakapan (hh:mm:ss). Keluaran adalah lama dan biaya
percakapan. Satu pulsa = 5 detik dan ongkos per pulsa adalah Rp. 150}


DEKLARASI
   const BiayaPerPulsa = 150   {biaya per pulsa}
   const LamaPulsa = 5         {1 pulsa = 5 detik}
   type  Jam : record  <hh:integer,  {0..23}
                       mm:integer,   {0..59}
                       ss:integer    {0..59}
                       >
   J1        : Jam {jam awal percakapan}
   J2        : Jam {jam akhir percakapan}
   J3        : Jam {lama percakapan}
   TotalDetik1, TotalDetik2 : integer {peubah bantu untuk menampung sisa
                                       pembagian}
   sisa           : integer
   durasi    : integer
   pulsa          : real
   biaya          : real
```

Wartel (lanjutan)

```
ALGORITMA :
   read(J1.hh, J1.mm, J1.ss)          {jam awal percakapan}
   read(J2.hh, J2.mm, J2.ss)          {jam selesai percakapan}

   {konversi masing-masing jam ke total detik}
   TotalDetik1 ← (J1.hh*3600) + (J1.mm*60) + J1.ss
   TotalDetik2 ← (J2.hh*3600) + (J2.mm*60) + J2.ss

   {hitung lama percakapan}
   durasi ← TotalDetik2 – TotalDetik1

   {hitung jumlah pulsa dan biaya untuk seluruh pulsa}
   Pulsa ← durasi/LamaPulsa
   Biasa ← Pulsa * BiayaPerPulsa

   {konversi durasi ke dalam jam-menit-detik}
   J3.hh ← durasi div 3600  {mendapatkan jam}
   sisa  ← durasi mod 3600
   J3.mm ← sisa div 60      {mendapatkan menit}
   J3.ss ← sisa mod 60      {mendapatkan detik}
   write(J3.hh, J3.mm, J3.ss, biaya)
```

Membaca/Menulis ke Arsip

Pembacaan tergantung pada format penyimpanan data di dalam arsip.

Data masukkan disimpan dalam program pengolah kata yang menghasilkan karakter ASCII(notepad,joe,editor text)

Contoh Program Penyimpanan Dalam Bentuk Arsip (PASCAL)

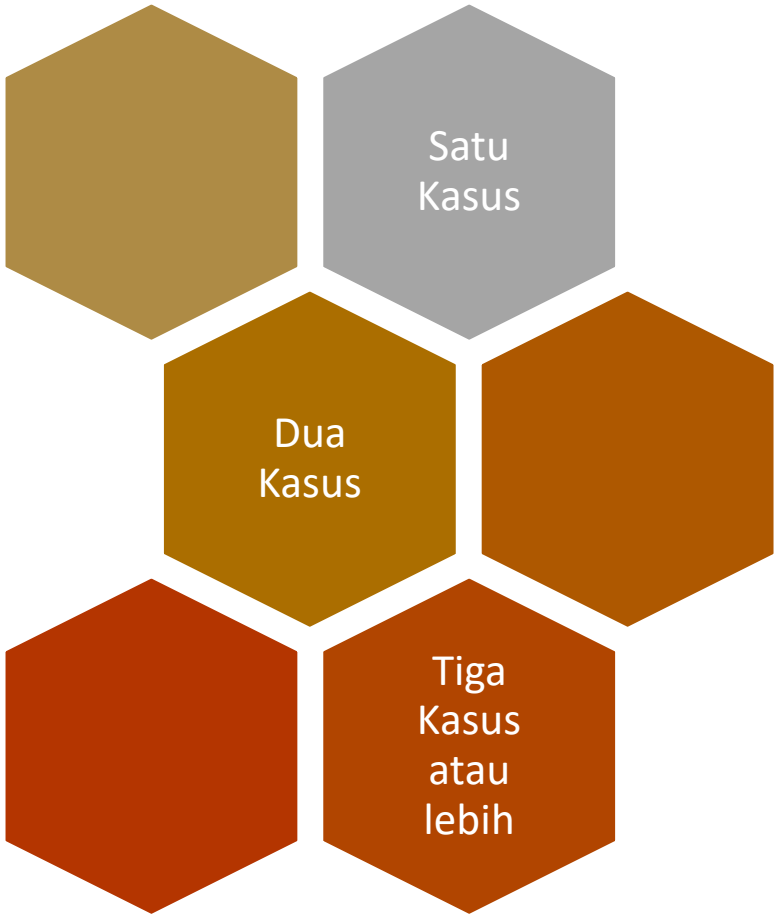
```
Program
luas_empat_persegi_panjang;
(*menghitung luas empat
persegipanjang, kemudian mencetak
nilai luas tersebut ke dalam
arsip "hasil.txt"*)


{Deklarasi}
var
   panjang,lebar,luas : real;
   Fin,Fout : text;
   NamaArsip1,NamaArsip2 :
   string[12];
{Algoritma}
begin
   write('Nama arsip masukan: ');
   readln(NamaArsip1);
   write('Nama arsip keluaran');
   readln(NamaArsip2);
   {buka arsip masukan}
   assign(Fin, NamaArsip1);
   reset(Fin)

   {buka arsip keluaran}
   assign(Fout,NamaArsip2);
   rewrite(Fout);

   {baca panjang dan lebar dari arsip Fin}
   read(Fin,panjang,lebar);
   Luas := panjang * lebar;
   writeln(Fout,'Luas segi empat
   = ',luas);

   {tutup arsip}
   close(Fin);
   close(Fout);
end.
```

Pemilihan

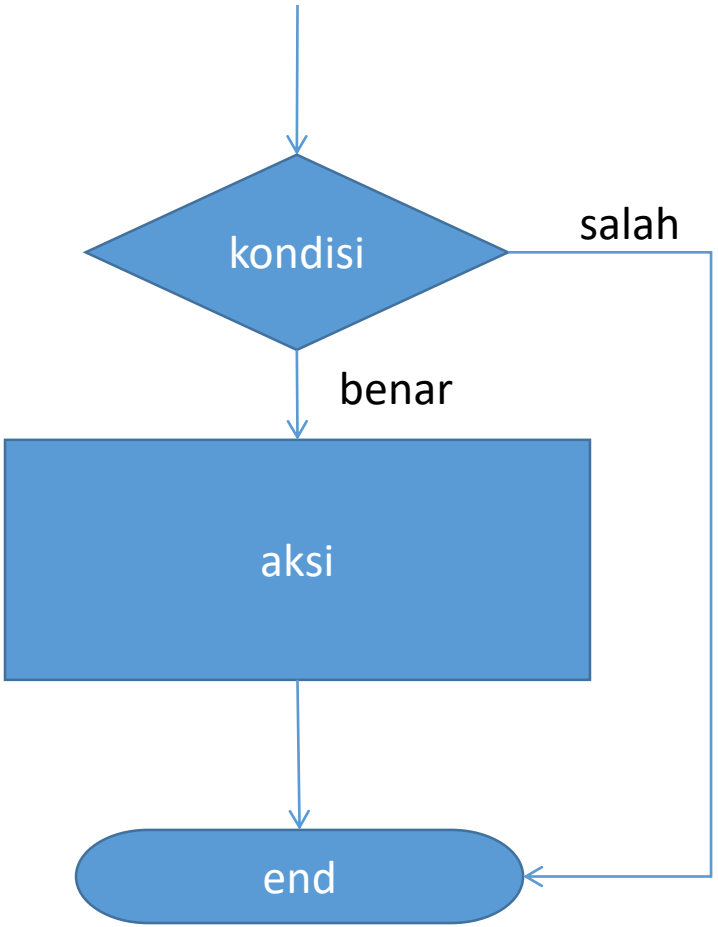
Pemilihan

Struktur pemilihan memungkinkan kita melakukan aksi jika suatu syarat dipenuhi

Analisis Kasus



Satu Kasus




```
if x > 100 then
   x ← x+ 1
endif
```

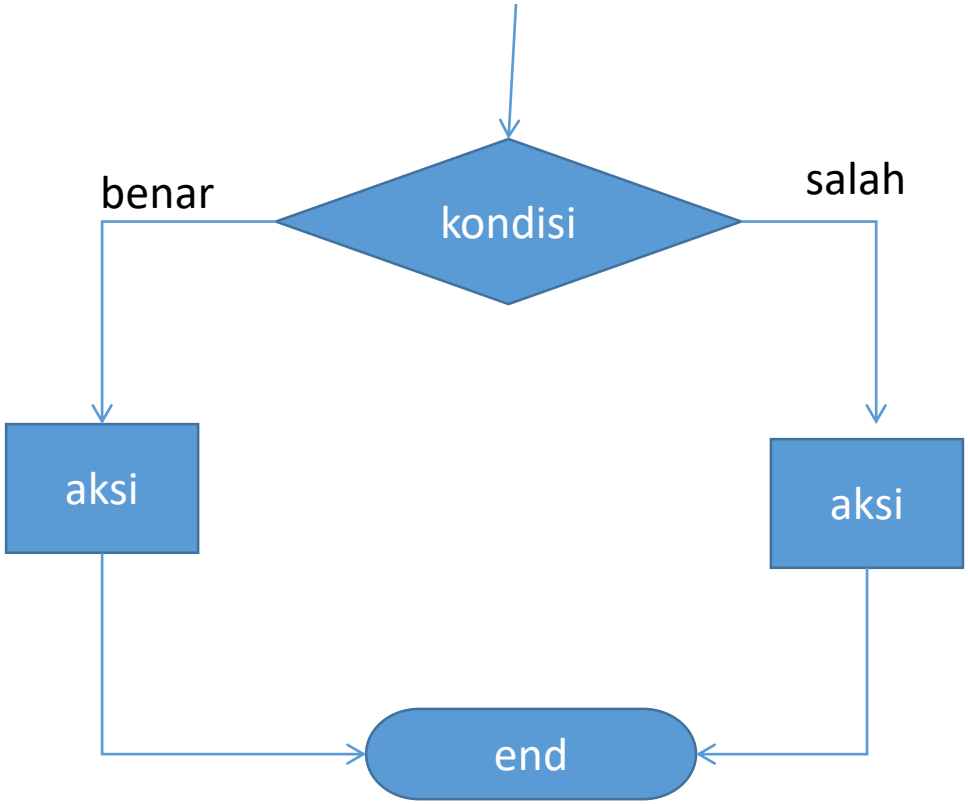
```
if kar = '*'  then
   stop ← true
endif
```

```
if (a ≠ 0) or (p=1) then
   q ← a*p
   write(q)
endif
```

Contoh Masalah dengan Satu Kasus

```
PROGRAM genap
{Mencetak pesan bilangan genap}

DEKLARASI
   x : integer

ALGORITMA:
read(x)
if (x mod 2 = 0) then
    write('genap')
endif
```

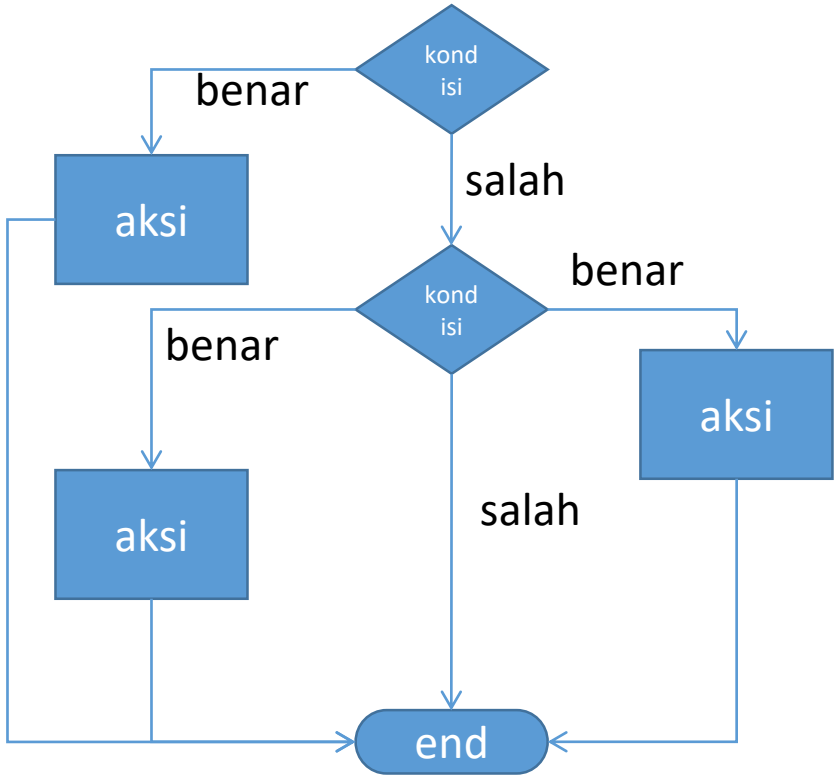
DUA KASUS



Conto 2 Kasus

```
PROGRAM genapGanjil
{Mencetak pesan bilangan genap ganjil}

DEKLARASI
x : integer

ALGORITMA
read(x)
if (x mod 2 = 0) then
    write('genap')
else
    write('ganjil)
```

Tiga Kasus atau Lebih



Contoh Algoritma 3 Kasus

```
PROGRAM BilanganBulat
{Menuliskan 'positif' bila nilai x > 0, 'negatif' bila nilai x <0
dan 'nol' bila x=0}

DEKLARASI
     x : integer

ALGORITMA:
read(x)
if x >0 then
    write('positif')
else
    if x <0 then
        write('negatif')
    else
        if x = 0 then
            write (nol)
        endif
     endif
endif
```

Struktur CASE

Struktur **CASE** dapat menyederhanakan ‘if-then-else’ yang bertingkat-tingkat

Contoh Algoritma

```
PROGRAM genapganjil
{menentukan apakah suatu bilangan termasuk bilangan
 genap atau ganjil}

DEKLARASI
    x : integer

ALGORITMA
read(x)
CASE (x mod 2)
    0 : write('genap')
    1 : write('ganjil')
endcase
```