

# ALGORITMA DAN PEMROGRAMAN

---

## DEFINISI ALGORITMA:

**ALGORITMA** adalah *suatu urutan langkah-langkah (Steps) yang disusun secara logis untuk menyelesaikan masalah dengan menggunakan komputer, dengan criteria sebagai berikut :*

1. Setiap langkah/step harus jelas dan pasti (*Definite*)
2. Diperbolehkan tanpa ada input tetapi minimal harus ada 1 output.
3. Jumlah langkah harus berhingga atau dengan kata lain harus ada *stopping criteria*.

#. Penyajian suatu algorithm:

Algorithma dapat disajikan dalam dua cara:

1. Secara tulisan
  - Struktur bahasa /diskripsi
  - Pseudocode
2. Teknik gambar (Flowchart)

Contoh:

1). Masalah menghitung Luas dan keliling lingkaran

Misal : input = R, output = L dan K,

Proses :  $L = \pi * r^2$   
dan  $K = 2 * \pi * r$

Algorithmanya:

**Secara Struktur bahasa:**

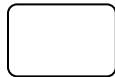
1. Tetapkan nilai  $\pi = 3.14$
2. Bacalah nilai R (jari-jari)
3. Hitung luas (L)
4. Hitung keliling (K)
5. Tulis L dan K

**Secara Pseudocode:**

1.  $\pi \leftarrow 3.14$
2. input  $\leftarrow R$
3.  $L \leftarrow \pi * r^2$
4.  $K \leftarrow 2 * \pi * r$
5. Output L dan K

**Secara Flowchart:**

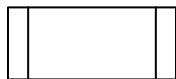
Simbol symbol dalam flowchart:



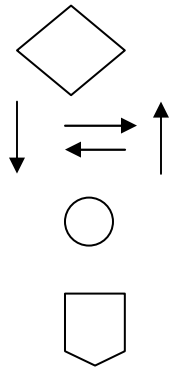
Terminal/ Proses (awal/akhir)



Input/output



Proses



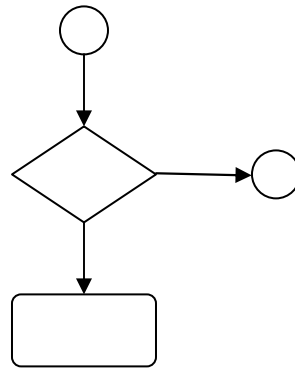
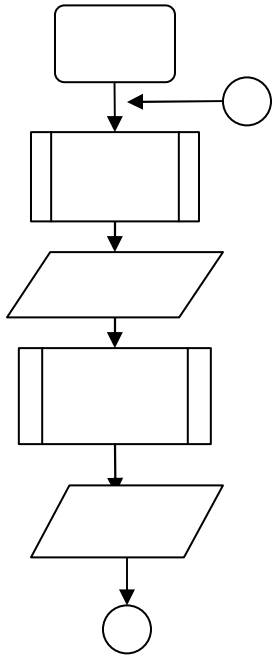
Titik keputusan

Arah arus

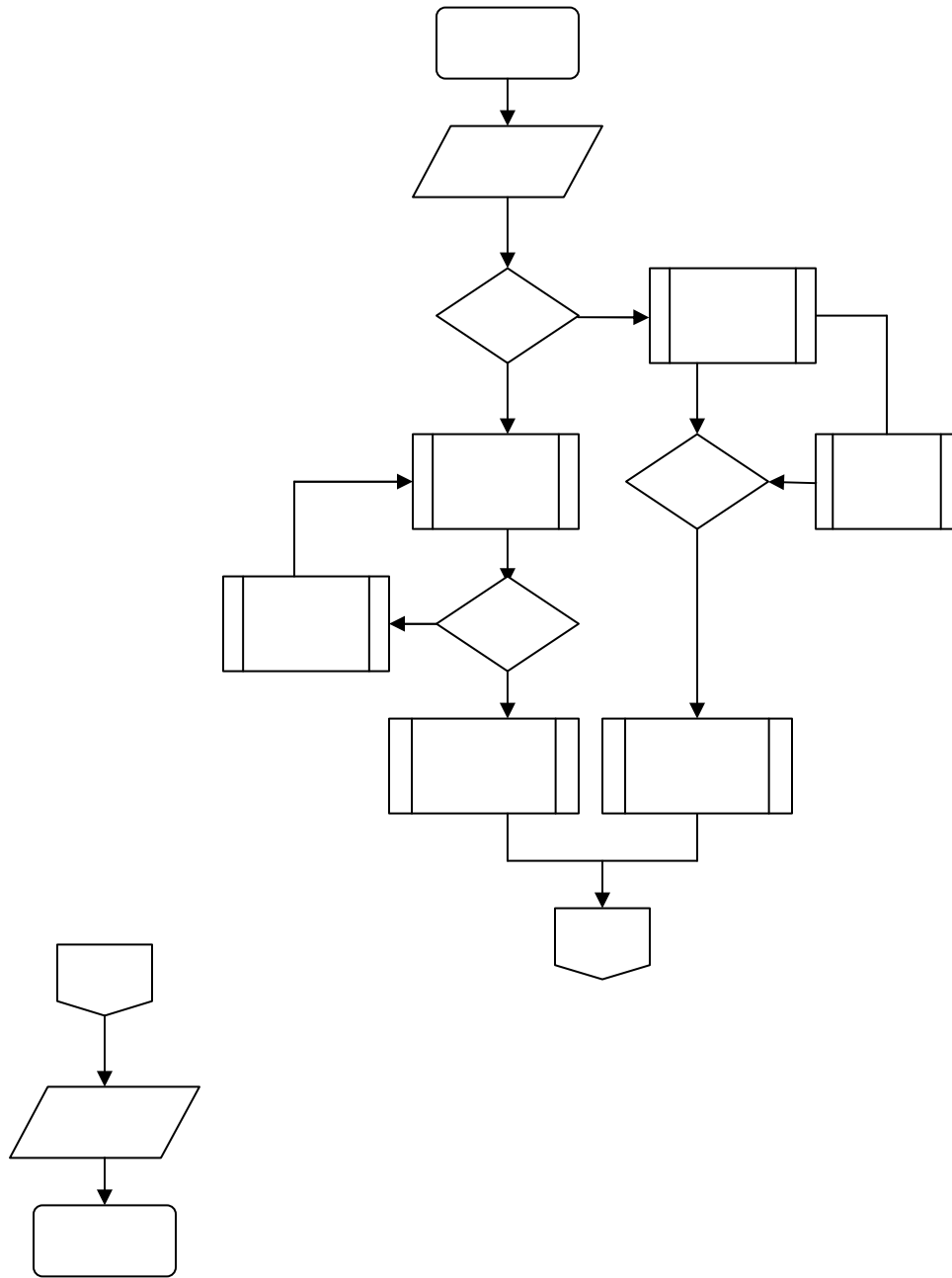
Sambungan dalam satu hal.

Sambungan lain halaman.

Dari soal diatas: **(menghitung luas dan keliling lingkaran)**



**Flow chart untuk algoritma Euclid:**



|

**Kasus Buatlah Flowchartnya!**

1).

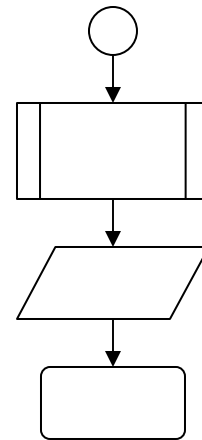
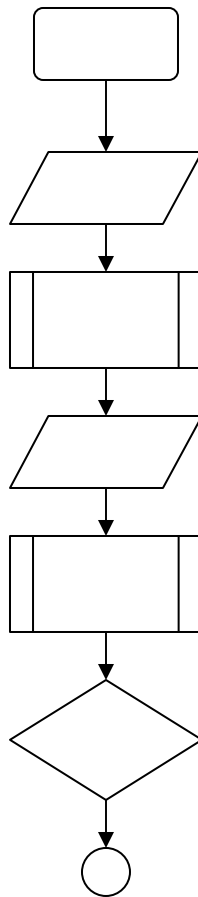
$$X = \frac{1}{N} \sum_{i=1}^N Xi = \frac{x1 + x2 + ..... + xn}{N}$$

Disini akan muncul tiga kemungkinan yaitu:

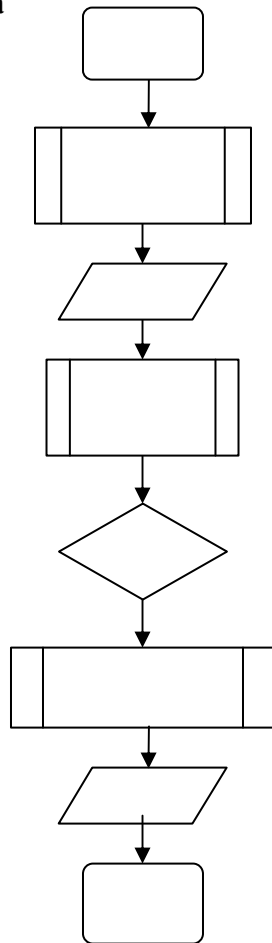
- 1).N tertentu
- 2).N dibaca
- 3).N tidak tentu

Bentuk Flowchartnya :

**\*N taktentu**



**\*N dibaca**



Contoh program untuk mencari FPB dengan menggunakan Algorithma Euclid:

**{Program : Menghitung FPB 2 biangan}**

*uses crt,dos;*

*var m,n,r : integer;*

*label awal,akhir;*

*begin*

*clrscr;*

*writeln('Masukkan nilai m dan n yang ingin dihitung!');*

*write('M =');readln(m);*

*writeln;*

*write('N =');readln(n);*

*writeln;*

*if m>n then*

*awal:*

*begin*

*r := m mod n;*

```

    writeln('r =',r);readln;
    if r =0 then
begin
    write('FPB dari', m , 'dan', n, 'adalah',n);readln;
    end else
    m:=n; n:=r;
    goto awal;end
    else
    if m<n then
    akhir:
    begin
        r:= n mod m;
        writeln('r =',r);readln;
        if r=0 then
            write('FPB dari', n , 'dan', m, 'adalah',m);readln;
            end else
            n:=m; m:=r;
            goto akhir;
        end.

```

### **Hasil Run Program:**

*masukkan nilai m dan n yang ingin dihitung!*

*M = 20*

*N = 30*

*r =10*

*r =0*

*FPB dari10dan30adalah10*

### **{Program : menghitung luas dan keliling lingkaran}**

```

uses crt,dos;
var k,l,r: real;
    kar: char;
label awal;
const pi = 3.14;
begin
    clrscr;
    awal:
    writeln;
        writeln('Untuk menghitung luas dan keliling lingkaran, maka');

```

```

    writeln('masukkan nilai r!');
writeln;
    writeln('(Lalu tekan enter untuk mulai!);writeln;
    write('R =');readln(r);
        L:= pi*r*r;
            write('Luas lingkaran =',L:2:0);
            readln;
        K:= 2* pi * r;
            write('Keliling lingkaran =',K:2:0);
            readln;
    write('Ingin menghitung lagi(y/n) ?');
    kar:= readkey;
        if upCase(kar) = 'Y' then goto awal; readln;
end.

```

### Hasil Run Program:

*Untuk menghitung luas dan keliling lingkaran, maka masukkan nilai r!*

*(Lalu tekan enter untuk mulai!)*

*R = 7*

*Luas lingkaran =154*

*Keliling lingkaran =44*

*Ingin menghitung lagi(y/n) ?y*

*Untuk menghitung luas dan keliling lingkaran, maka masukkan nilai r!*

*(Lalu tekan enter untuk mulai!)*

*R = 10*

*Luas lingkaran =314*

*Keliling lingkaran =63*

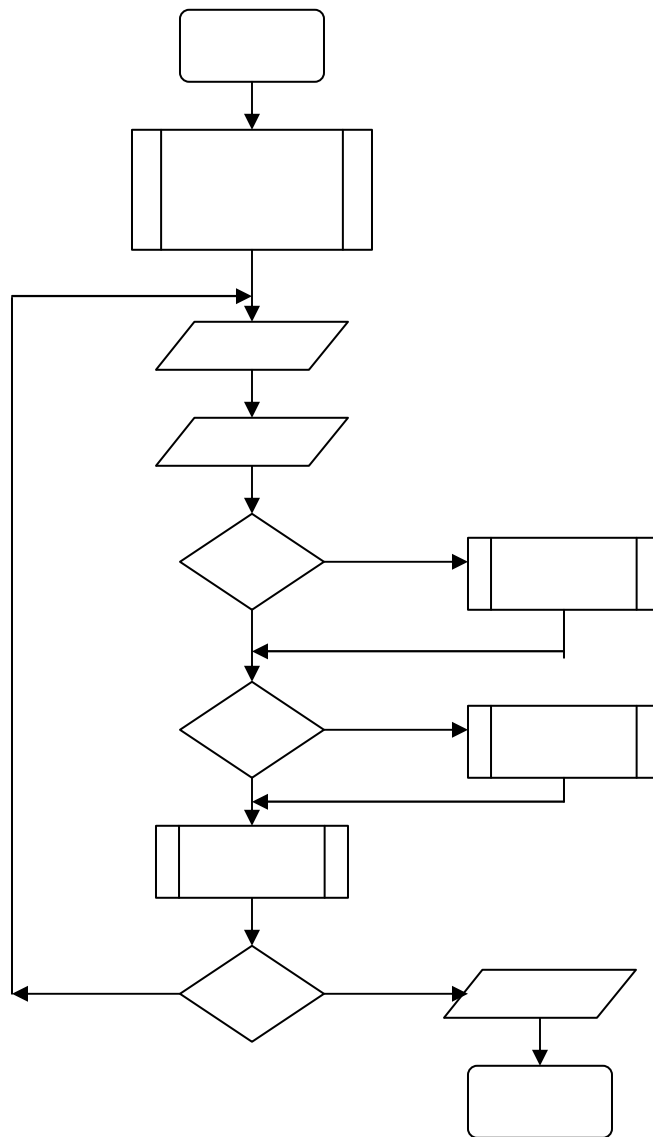
*Ingin menghitung lagi(y/n) ?n*

2). Susunlah sebuah flowchart untuk mencari bilangan terbesar dan terkecil dari sejumlah data!

Misalkan : Masukkan     N = jumlah data  
                                    $X_1, X_2, \dots, X_n = \text{data}$

Proses

Keluaran      $X_{\max}$  dan  $X_{\min}$



3). Masalah Program Selisih waktu.

**Algorithma sederhana :**

1. Baca jam Mulai
2. Baca jam selesai
3. Hitung selisih jam/ waktu

**Perbaikan algorithma :** Untuk mengatasi masalah selisih waktu maka semua nilai waktu yang diperoleh, semuanya diubah ke dalam satuan detik sehingga diharapkan tidak akan ditemui nilai negatif dalam perhitungan selisih.

**Algorithma Psedocode:**

Mulai :

1. Input( jam1, menit1, detik1)
2. Input( jam2, menit2, detik2)
3.  $\Delta I \leftarrow \text{jam1} * 3600 + \text{menit1} * 60 + \text{detik1}$

4. Delta II  $\leftarrow$  jam2\*3600 + menit2\*60 + detik2
5. Delta  $\leftarrow$  Delta II - Delta I
6. Selisih jam  $\leftarrow$  Delta **div** 3600
7. Menit  $\leftarrow$  Delta **mod** 3600
8. Selisih menit  $\leftarrow$  menit **div** 60
- Selisih detik  $\leftarrow$  menit **mod** 60
9. Output(Selisih jam, Selisih menit, Selisih detik)

\*\*\*\*\*

## PROGRAM PASCAL

### STRUKTUR PROGRAM

1. Judul program (opsional)  
Format : { **judul program** } atau **program\_nama program;**
  
2. Deklarasi program:  
**Var**                    {Variabel program}  
**Const**                {Constanta program}  
**Label**                 {Label}  
**Func**                 {Function}  
**Procedure**            {Prosedur program}

Catatan :

Deklarasi program bersifat opsional khusus, artinya jika diperlukan maka harus ada dalam program dan apabila tidak diperlukan maka boleh tidak dicantumkan. Deklarasi program termasuk **identifier** (pengenal), yaitu kata-kata yang diciptakan oleh pemrogram untuk mengidentifikasi sesuatu.

**Syarat identifier:**

- a) Harus diawali oleh alphabet
  - b) Huruf besar/ kecil yang digunakan dianggap sama
  - c) Penyusun dari sebuah identifier tidak boleh berupa: tanda baca, tanda relasi, symbol arithmatik, karakter khusus atau reserved word
  - d) Tidak boleh dipisahkan dengan spasi, apabila hendak dipisahkan maka menggunakan tanda ( \_ )
  - e) Panjang identifier tidak dibatasi tetapi sebaiknya gunakan se-efektif mungkin.
3. Program Utama  
Diawali dengan **Begin** dan diakhiri dengan **End**.

## TIPE- TIPE DATA DALAM TURBO PASCAL

Tipe data berfungsi membatasi jangkauan data yang akan dilaksanakan.

Macam-macam tipe data:

- a) Tipe data Standard, terdiri atas :
  - Ordinal
  - Non-ordinal
- b) Tipe data Terstruktur
- c) Tipe data Pointer

### Tipe Data Standard

#### I. Tipe Ordinal

Pada tipe data ini akan berlaku operasi-operasi:

- *Predecessor* (Pred), berfungsi memberikan nilai kembalian berupa nilai predecessor (urutan sebelumnya) dari suatu nilai ordinal.
- *Successor* (Succ), berfungsi memberikan nilai successor (urutan berikutnya) dari suatu nilai ordinal.

#### a) Tipe Integer

Operator-operator yang berlaku pada tipe data integer, yaitu :

- Operator aritmatik : +, -, \*, /, div, mod
- Operator logic : <, =, >, <=, >=, <>

Tipe data integer masih dibagi menjadi 5 tipe:

Shortint (1 byte bertanda)	Byte (1 byte tak bertanda)
Integer (2 byte bertanda)	Word (2 byte tak bertanda)
Longint (4 byte bertanda)	

Catatan : Semua tipe integer adalah tipe Ordinal.

#### b) Tipe Real

Dibagi menjadi 5 macam, yaitu :

Real (6 byte)	Extended (10 byte)
Single (4 byte)	Comp (8 byte)
Double (8 byte)	

#### c) Tipe Char (Karakter)

Char adalah semua tombol yang terdapat pada keyboard atau lebih lengkapnya semua karakter yang terdapat pada kode ASCII.

Catatan : Apabila char ingin dijadikan sebagai konstanta maka karakter yang dimasukkan harus diapit dengan tanda kutip satu. Dan apabila karakter itu berupa tanda kutip satu maka harus diapit dengan dua tanda kutip satu.

#### d) Tipe Boolean

Mempunyai nilai benar /salah. Operator yang penghubung untuk tipe Boolean adalah : =, <>, >, <, >=, <=, IN

Boolean akan menghasilkan False jika nilai = 0 dan akan menghasilkan True jika nilai = 1.

e) Tipe Enumerated (terbilang)

Bentuk umum :

*Type*

*Nama = (Pengenal 1, pengenal 2, ..., pengenal n);*

Contoh:

Type

Jawa Tengah = ( Semarang, Tegal, Kendal, Demak) ;

Sehingga dengan fungsi berikut :

Ord(Semarang) = 0

Ord(Tegal) = 1

Dan seterusnya.....

f) Tipe Subrange

Bentuk umum :

*Constanta1..Constanta2*

Contoh :

0..99 ;

-128..127 ;

## ALGORITHMMA EUCLID

(Digunakan untuk mencari FPB/ Faktor Persekutuan Besar)

Langkah-langkah yang digunakan dalam algorithmma Euclid adalah:

- Diberikan dua buah bilangan bulat positif misalnya  $m$  dan  $n$  dengan  $m > n$ .
- Bagilah  $m$  dengan  $n$ , misalnya hasilnya adalah  $r$ .
- Apabila  $r = 0$ , maka stop dan hasilnya adalah bahwa  $r$  itu sendiri merupakan FPB dari  $m$  dan  $n$ .
- Jika  $r \neq 0$  maka lanjutkan ke langkah selanjutnya yaitu ganti  $m$  dengan  $n$  dan  $n$  dengan  $r$ , kemudian ulangi dari langkah (b) demikian selanjutnya.

Contoh:

Carilah FPB dari 36 dan 20:

- $\frac{m}{n} = \frac{36}{20}$ , mempunyai sisa  $r = 16$ ,
  - $r = 16 \neq 0$
  - $m = 20$  dan  $n = 16$
- $\frac{m}{n} = \frac{20}{16}$  mempunyai sisa  $r = 4$ ,
  - $r = 4 \neq 0$
  - $m = 16$  dan  $n = 4$
- $\frac{m}{n} = \frac{16}{4}$  mempunyai sisa  $r = 0$ ,
  - $r = 0$ ,

Jadi FPB( 36, 20) = 4.

# STRUKTUR PEMROGRAMAN/ STRUKTUR KONTROL

Ada tiga macam struktur pemrograman yaitu :

1. Runtutan/ Sequential
2. Seleksi/ Conditional
3. Perulangan/ Repetition/ Looping

## 1). SEQUENTIAL/ RUNTUTAN

Berfungsi mengontrol program sehingga berjalan secara berurutan. Program dibaca dari mulai perintah yang pertama dituliskan kemudian turun ke perintah selanjutnya demikian seterusnya sampai pada akhir perintah. Secara umum program akan dibaca oleh compiler secara terurut.

Contoh :

```
{Program_Selisih_Jam}  
Uses crt;  
var j1,j2,m1,m2,d1,d2,B1,B2,J,M,S,Delta,sisa: integer;  
  
begin  
  clrscr;  
  write('Baca jam mulai =');readln(j1,m1,d1);  
  writeln;  
  write('Baca jam selesai =');readln(j2,m2,d2);  
  B1:= 3600 * j1 + 60 * m1 + d1;  
  B2:= 3600 * j2 + 60 * m2 + d2;  
  Delta:= B2 - B1;  
  J:= Delta div 3600;  
  sisa:= Delta mod 3600;  
  M:= sisa div 60;  
  S:= sisa mod 60;  
  write('Selisih jam =',J);  
  writeln;  
  write('Selisih menit =',M);  
  writeln;  
  write('Selisih detik =',S);readln;  
end.
```

### Hasil Run program:

*Baca jam mulai = 3 23 29*

*Baca jam selesai = 4 30 27*

*Selisih jam = 1*  
*Selisih menit = 6*  
*Selisih detik = 58*

Catatan : Tanda (*:=*) disebut sebagai **Operator Penugasan**.

## 2). SELECTION/ CONDITION

Sistem kontrol ini memungkinkan programmer untuk membuat program yang dapat memilih satu langkah di antara sejumlah langkah untuk dikerjakan.

Dalam Turbo Pascal disediakan 2 buah struktur kontrol seleksi, yaitu:

- a. Struktur **IF.....THEN.....**
- b. Struktur **CASE...OF...**

### a). Struktur **IF**

Merupakan struktur kontrol di mana suatu aksi dilaksanakan berdasarkan kondisi logikanya (benar atau salah). Bentuk umum :

**if** (*kondisi*) **then** *aksi* ;  
{Jika kondisi benar maka aksi akan dilaksanakan dan sebaliknya }

**if** (*kondisi*) **then** *aksi1* **else** *aksi2* ;  
{Jika kondisi benar maka aksi1 dilaksanakan, jika kondisi salah maka aksi2 akan dilaksanakan}

**if** (*kondisi1*) **then** *aksi1* **else**  
**if** (*kondisi2*) **then** *aksi2* **else**

.....  
**if** (*kondisi-n*) **then** *aksi-n* ;  
{Jika kondisi1 benar maka aksi1 dilaksanakan tanpa membaca kondisi2 dan seterusnya. Jika kondisi1 salah maka aksi2 dilaksanakan tanpa membaca aksi3 dan selanjutnya. Demikian seterusnya}

**if** (*kondisi1*) **then** *aksi1* ;  
**if** (*kondisi2*) **then** *aksi2* ;  
.....  
**if** (*kondisi-n*) **then** *aksi-n* ;  
{Masing-masing kondisi akan dibaca dan aksi akan tetap dilaksanakan. Jadi masing-masing struktur terpisah satu dengan yang lain}

Contoh program :

```
{ program_menghitung_akar_kwadrat}  
uses crt;  
var A,B,C:integer;  
    x1,x2,D:real;  
begin  
  clrscr;  
  write('Baca koefisien:');readln(A,B,C);writeln;  
  writeln(A,'x*x + ('B,') x +',C);  
  if A=0 then writeln('Bukan persamaan kwadrat') else  
  begin  
    D:=(B*B) - (4*A*C);
```

```

writeln('Determinannya :',D:2:2);readln;
if D>0 then
  begin
    writeln('Persamaan kwadrat mempunyai 2 akar yang berbeda');

    x1:= (-B + sqrt(D))/(2*A);
    x2:= (-B - sqrt(D))/(2*A);
    writeln('Akar-akarnya adalah:',x1:2:2,'dan',x2:2:2);
  end else
if D=0 then
  begin
    writeln('Persamaan kwadrat mempunyai akar yang sama'

    x1:= -B/(2*A);
    x2:= -B/(2*A);
    writeln('Akar-akanya adalah:',x1:2:2);
  end else
    writeln('Tidak memiliki akar riil');
end;
readln;
end.

```

**Hasil Run Program :**

*-Baca koefisien: 1 -4 4*

$$1x^2 + (-4)x + 4$$

*Determinannya :0.00*

*Persamaan kwadrat mempunyai akar yang sama*

*Akar-akanya adalah:2.00*

*-Baca koefisien: 1 -5 6*

$$1x^2 + (-5)x + 6$$

*Determinannya :1.00*

*Persamaan kwadrat mempunyai 2 akar yang berbeda*

*Akar-akarnya adalah:3.00dan2.00*

*-Baca koefisien: 1 3 4*

$$1x^2 + (3)x + 4$$

*Determinannya :-7.00*

*Tidak memiliki akar riil*

**{Program\_Konversi\_nilai}**

```
Uses Crt;  
var Nilai : Byte;  
begin  
  clrscr;  
  write('Baca nilai :');readln(Nilai);  
  if nilai>=80 then write('Nilai = A') else  
    if nilai>=65 then write('Nilai = B') else  
      if nilai>=41 then write('Nilai = C') else  
        if nilai>=26 then write('Nilai = D') else  
          write('Nilai = E');  
  readln;  
end.
```

**Hasil Run Program :**

Baca nilai : 90	Baca nilai :55
Nilai = A	Nilai = C
Baca nilai :75	Baca nilai :25
Nilai = B	Nilai = E

b). Struktur **CASE...OF..**

Merupakan peluasan dari struktur IF. Karena kalau dalam struktur IF hanya disediakan dua pilihan (berdasarkan kondisi logikanya) maka dalam struktur Case ..of dimungkinkan untuk memilih satu pilihan di antara banyak pilihan yang ada. Bentuk umumnya :

<i>Case var.pilih of</i>	<i>atau</i>	<i>Case var.pilih of</i>
<i>Pilih1 : aksi1 ;</i>		<i>pilih1 : aksi1 ;</i>
<i>Pilih2 : aksi2 ;</i>		<i>pilih2 : aksi2 ;</i>
<i>..... ;</i>		<i>..... ;</i>
<i>pilih-n : aksi-n ;</i>		<i>pilih-n : aksi n; else aksi-n+1</i>
<i>end;</i>		<i>end;</i>

Catatan : Ekspresi yang digunakan dalam statemen Case adalah yang mempunyai tipe ordinal yaitu dengan batas antara (-32768 s/d 32767). Sehingga tipe lain seperti integer yang bertipe longint, tipe string atau word tidak boleh digunakan.

Contoh program :

**{Program\_Konversi\_nilai2}**

```
Uses Crt;  
Var Nilai : integer;  
begin  
  Clrscr;  
  write('Baca nilai =');readln(Nilai);  
  Case Nilai of  
    0..25 : writeln('Nilainya = E');  
    26..39 : writeln('Nilainya = D');  
    40..64 : writeln('Nilainya = C');  
    65..79 : writeln('Nilainya = B');
```

```

      80..100: writeln('Nilainya = A');
    else
      writeln('Tidak ada nilai yang dimaksud');
    end;readln;
end.

```

Catatan : Program ini akan memberikan nilai yang sama persis dengan yang menggunakan struktur IF.

### 3). PERULANGAN/ LOOPING/ REPETITION

Dalam membuat suatu program kadang-kadang diinginkan agar program tersebut mampu memproses hal yang sama secara berulang-ulang sampai ditemukan suatu nilai tertentu yang diinginkan atau mencapai batas yang telah ditentukan. Untuk itu maka Turbo Pascal telah menyediakan suatu struktur perulangan yang memudahkan untuk melakukan proses tersebut, sehingga kode-kode dalam program menjadi lebih sederhana. Ada beberapa struktur perulangan, yaitu :

- Struktur **FOR...DO....**
- Struktur **WHILE...DO.....**
- Struktur **REPEAT...UNTIL....**

#### a). Struktur **FOR**

Ada 2 jenis struktur FOR, yaitu :

- Ascendant (Naik)
- Descendant (Turun)

##### Naik

Format : **For** *count := awal to akhir do aksi/ blok aksi ;*

Catatan :     - awal <= akhir  
                   - Variabel count akan naik satu setelah menyelesaikan aksi

##### Turun

Format : **For** *count := awal downto akhir do aksi1/ blok aksi ;*

Catatan :     - awal >= akhir  
                   - Variabel count akan turun satu setelah menyelesaikan aksi

Struktur FOR hanya dapat digunakan pada cacah perulangan yang diketahui (berapa kali perulangan tersebut akan dilakukan).

Contoh program :

#### **{Program Latihan1}**

```

uses Crt;
var N,i,j :integer;

begin
  clrscr;
  write('Baca nilai : ');readln(N);
  for i:= 1 to N do
  begin
    for j:=1 to i do

```

#### **{Program Latihan 2}**

```

uses Crt;
var N, I, j, data : integer;

begin
  clrscr;
  write('Baca nilai: ');
  readln(N);Data:=N;
  for i:=1 to N do
  begin

```

```
    write(i:3);  
    writeln;  
end;  
readln;  
end.
```

**Hasil Run Program :**

*Baca nilai : 7*

```
1  
2 2  
3 3 3  
4 4 4 4  
5 5 5 5 5  
6 6 6 6 6 6  
7 7 7 7 7 7 7
```

```
for j:=1 to N do  
write(data:3);writeln;  
data := data -1;end;  
readln;  
End.
```

**Hasi Run Program :**

*Baca nilai : 7*

```
7  
6 6  
5 5 5  
4 4 4 4  
3 3 3 3 3  
2 2 2 2 2 2  
1 1 1 1 1 1 1
```