ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/318722368

Buku Penunjang Praktek Pascal

Book · January 2006

CITATIONS
0

READS
3,687

1 author:

Jufriadif Naam
Universitas Putra Indonesia YPTK
36 PUBLICATIONS **58** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Edge Detection on Objects of Medical Image with Enhancement multiple Morphological Gradient Method View project

All content following this page was uploaded by Jufriadif Naam on 25 October 2018.

The user has requested enhancement of the downloaded file.

Panduan Praktikum Bahasa Pemrogramman Pascal (Pascal Programming language)

Founder: Niklaus Wirth (Turing award 1984)

I. Struktur Bahasa Pemrograman Pascal

Secara ringkas, struktur program Pascal terdiri dari:

```
Judul program
Bagian deklarasi
      deklarasi piranti
      deklarasi label
      deklarasi konstanta
      deklarasi tipe
      deklarasi variabel
      deklarasi prosedur
      deklarasi fungsi
Bagian pernyataan
```

Berarti para mahasiswa dapat menyimpulkan langsung bahwa bahasa Pemrograman Pascal ini mempunyai struktur dan merupakan sebuah bahasa pemrogramman yang terstruktur. Dalam aplikasinya, tidak harus semua bagian dari struktur tersebut harus ada, tetapi disesuaikan dengan kebutuhan dalam aplikasi yang dibutuhkan. Seperti tidak ada judul program, deklarasi label, konstanta, dan lain-lain.

Penulisan Program Pascal tidak mengenal aturan penulisan perintah pada kolom tertentu, tetapi dapat ditulis pada kolom keberapapun dengan ketentuaan setiap pernyataan diakhiri dengan **titik koma (;)**. Tata cara penulisan **huruf besar dan kecil diabaikan** atau **dianggap sama** oleh **Compiler Pascal**. Sebuah program yang paling sederhana dapat terdiri atas satu pernyataan saja, seperti:

```
Begin  {awal pernyataan }

End.   {akhir pernyataan dan diakhiri titik (•)}
```

Bentuk umum dari bagian pernyataan ini adalah:

```
Begin
      :
    pernyataan; {diakhiri titik koma (;)}
      :
      :
End.
```

Judul program berfungsi untuk memberi identitas sebuah program. Dalam sebuah program hanya terdiri atas satu (1) identitas atau satu kata **Program.** Setelah kata program diikuti oleh sebuah kata yang merupakan sebagai **identitas**. Identitas ini merupakan sebuah kata yang

didefinisikan oleh programmer sendiri (user definite word), dimana ketentuan definite word ini adalah:
- Tidak boleh terdiri atas **Space**
- Tidak boleh terdiri atas **simbol matematis**, seperti: +, -, /, *, dan lain-lain. Jika menggunakan simbol, maka gunakanlah simbol **non matematetis**, seperti: _
- Tidak boleh menggunakan kata-kata yang sudah didefinisikan oleh Software Pascal (Pascal definition word), seperti **Begin**, **End**, **Write**, **Read**, dan lain-lain

Bagian deklarasi tempat untuk mendefinisikan piranti (**unit**) input dan output (**I/O**) yang dibutuhkan oleh program. Beberapa jenis piranti (unit) yang telah disediakan adalah:
- **Crt,** yaitu piranti yang mengatur media monitor dan keyboard, seperti penggunaan perintah **Clrscr** untuk menghapus layar, **Gotoxy** untuk mengatur posisi tampilan di layar.
- **Printer**, untuk mengatur printer, seperti penggunaan perintah **lst**.
- **Graph**, untuk perintah yang berorientasi ke tampilan grafik.
- **Overlay**, unit pengaturan pembagian sebuah aplikasi menjadi dua atau lebih program.

II. Perintah output

Perintah (statement) output digunakan mengatur tampilan dari program pada media output komputer, seperti monitor, printer, dan lain-lain. Perintah yang digunakan adalah:

- **Write** dan **Writeln**. Perbedaan perintah ini adalah:
  - **Write**; setelah pernyataan ini dieksekusi oleh processor, maka posisi kursor berada pada baris yang sama di layar.
  - **Writeln**; setelah pernyataan ini dieksekusi oleh processor, maka posisi kursor berada pada baris berikutnya di layar.
  -

Contoh 1.

```
Program cetak1 (output);
Begin
     Write (‘Be’);
     Write (‘la’);
     Write (‘jar’);
End.
```

Contoh 2.

```
Program cetak2 (output);
Begin
     Write (‘My name is’);
     Writeln (‘Associate Professor Doktor Haji Jufriadif Na`am, Sarjana
Komputer, Master Komputer’);
     Write (‘what yours?’);
End.
```

Contoh 3.

```
Program tampil;
Uses crt;
Begin
     Clrscr;
     Write ('Nama : ');
     Writeln ('Na`am');

     Writeln ('Nama: ');
     Writeln ('Jufriadif');

     Write ('F');
     Write ('i');
     Write ('l');
     Write ('k');
     Write ('o');
     Writeln('m');
     Writeln('U');
     Writeln('P');
     Writeln('I');
     Readln;
End.
```

Contoh 4.

```
Program tampil;
Uses crt;
Const
     Kota  = 'Minang Maimbau';
Var
     Nama  : String [15] = 'Jufriadif';
     Umur  : Integer  = 51;
Begin
     Clrscr;

     Write ('Nama : ');
     Writeln (nama);

     Write ('Umur : ');
     Writeln (umur);

     Write ('Kota : ');
     Writeln (kota);

     Write ('IPK : ');
     Writeln (4);

     Readln;
End.
```

Contoh 5.

```
Program tampil;
Uses Crt;
Const
     Kota  = 'Minang Maimbau';
Var
     nama  : String   = 'Na`am';
     umur  : Integer  = 51;
Begin
     Clrscr;
     Writeln('Nama : ',nama,'. Umur : ',umur,'. Kota : ',kota,'. IPK :
',4);
     Readln;
End.
```

Contoh 6.

```
Program cetak3 (output);
Begin
     Writeln (2000);
     Writeln (2000+503);
     Writeln (1075-500);
     Writeln (14*62);
     Writeln (23 div 7);
     Writeln (18 mod 4);

End.
```

Contoh 7.

```
Program tampil;
Uses Crt;
Var
     angka_int  : Integer  = 12345;
     angka_real : real     = 1234.123456;
Begin
     Clrscr;

     Writeln (angka_int);
     Writeln (angka_int:4);
     Writeln (angka_int:8);

     Writeln (angka_real);
     Writeln (angka_real:4:2);
     Writeln (angka_real:2:4);
     Readln;
End.
```

Buatlah program dengan hasil pada layar monitor sebagai berikut:
Latihan 1.

```
*  *  *  *  *
*           *
*           *
*           *
*  *  *  *  *
```

Buatlah program dengan hasil pada layar monitor sebagai berikut:
Latihan 2.

```
       +  +  +  +  +
          x  x  x  x  x
               -   -   -  -  -
                  /   /   /  /  /
```

Buatlah program dengan hasil pada layar monitor sebagai berikut:
Latihan 3.

```
  *  *  *  *  *
     *           *
     *           *
     *  *  *  *
     *  *  *  *
     *           *
     *           *
  *  *  *  *  *
```

Buatlah program dengan hasil pada layar monitor sebagai berikut:
Latihan 4.

```
            2 + 3
      6 x   -------
            4 + 5
```

Buatlah program dengan hasil pada layar monitor sebagai berikut:
Latihan 5.

```
       42
      ---- x 52 + 6 X 3 + 4
        7
```

III. Perintah input

Perintah input berfungsi untuk memasukkan nilai dari media input komputer yang ditampung oleh variabel sebagai data penampung. Perintah yang digunakan adalah **Read** atau **Readln**. Dalam menggunakan perintah ini, maka harus dipersiapkan variabel sebagai penampung hasil inputan. Variabel ini harus memiliki tipe data yang sama dengan nilai yang akan diinput.

Perbedaan Read dengan **Readln** adalah perintah **Read** akan membaca data secara horizontal. Dimana setelah proses input selesai, maka posisi cursor akan tetap berada di baris yang sama. Penggunaan **Spasi** atau **Enter** dapat memisahkan antara satu input dengan input lainnya.

Perintah R**eadln** akan membaca data secara vertikal. Dimana setelah proses input selesai, maka posisi cursor akan pindah ke baris baru berikutnya. Penggunaan **Enter** dapat memisahkan antara satu input dengan input lainnya.

Defenisi perbedaan diatas hanya secara teori. Dalam prakteknya, kedua perintah ini akan membuat prilaku berbeda tergantung tipe data yang digunakan. Terutama untuk perintah read yang kadang berfungsi aneh. Kita akan melihat permasalahan ini melalui beberapa contoh kode program.

Contoh 8.

```
Program input;
Uses Crt;
Var
     nama,alamat: String;
     umur: Integer;
     ipk: Real;
Begin
     Clrscr;
     Writeln ('Masukkan Data Mahasiswa');
     Writeln ('======================');
     Write ('Nama   :');
     Readln (nama);
     Write ('Alamat :');
     Readln (alamat);
     Write ('Umur   :');
     Read (umur);
     Write ('IPK    :');
     Readln (ipk);
     Writeln;
     Writeln ('=========HASIL=========');
     Write ('Nama: ',nama,'. Alamat: ',alamat);
     Writeln ('. Umur: ',umur,'. IPK: ',ipk:1:2);
     Readln;
End.
```

Contoh 9.

```
Program input;
Uses Crt;
Var
     a,b,c,d    : Integer;
Begin
     Clrscr;
     Writeln ('Input 4 angka, dipisah dengan spasi:');
     Read (a);
     Read (b);
     Read (c);
     Read (d);
     Writeln;
     Writeln ('Hasil Input:');
     Writeln ('a: ',a,', b: ',b,', c: ',c,', d: ',d);
     Readln;
     Readln;
End.
```

Contoh 10.

```
Program input;
Uses Crt;
Var
     a,b,c,d    : Integer;
Begin
     Clrscr;
     Writeln('Input 4 angka, dipisah dengan spasi:');
     Read(a,b,c,d);
     Writeln;
     Writeln('Hasil Input:');
     Writeln('a: ',a,'. b: ',b,'. c: ',c,' d: ',d);
     Readln;
End.
```

Contoh 11.

```
Program input;
Uses Crt;
Var
     a,b,c,d    : Integer;
Begin
     Clrscr;
     Writeln ('Input 4 angka, dipisah dengan enter:');
     Readln (a);
     Readln (b);
     Readln (c);
     Readln (d);
     Writeln;
     Writeln ('Hasil Input:');
     Writeln ('a: ',a,'. b: ',b,'. c: ',c,' d: ',d);
     Readln;
End.
```

Contoh 12.

```
Program input;
Uses Crt;
Var
     a,b,c,d,e,f,g,h,i     : Integer;
Begin
     Clrscr;
     Writeln ('Input Matriks 3x3');
     Read (a,b,c);
     Read (d,e,f);
     Read (g,h,i);
     Writeln;
     Writeln ('Hasil Matriks:');
     Writeln (a,' ',b,' ',c);
     Writeln (d,' ',e,' ',f);
     Writeln (g,' ',h,' ',i);
     Readln;
     Readln;
End.
```

Contoh 13.

```
Pprogram input;
Uses Crt;
Var
     a,b,c,d,e  : Char;
Begin
     Clrscr;
     Writeln ('Input Teks Sembarang (5 karakter)');
     Read(a);
     Read(b);
     Read(c);
     Read(d);
     Read(e);
     Writeln;
     Writeln ('Hasil Teks:');
     Write (a,' ',b,' ',c,' ',d,' ',e);
     Readln;
     Readln;
End.
```

Contoh 14.

```
Program input;
Uses Crt;
Var
     a,b,c,d,e  : Char;
Begin
     Clrscr;
     Writeln ('Input Teks Sembarang (5 karakter) :');
     Readln (a);
     Readln (b);
     Readln (c);
     Readln (d);
     Readln (e);
     Writeln;
     Writeln ('Hasil Teks:');
     Write (a,' ',b,' ',c,' ',d,' ',e);
     Readln;
End.
```

Contoh 15.

```
Program input;
Uses Crt;
Var
     a,b,c,d    : String;
Begin
     Clrscr;
     Writeln ('Input 4 kata:');
     Read (a);
     Read (b);
     Read (c);
     Read (d);
     Writeln;
     Writeln ('Hasil Teks:');
     Writeln ('a: ',a);
     Writeln ('b: ',b);
     Writeln ('c: ',c);
     Writeln ('d: ',d);
     Readln;
     Readln;
End.
```

Contoh 16.

```
Program input;
Uses Crt;
Var
     a,b,c,d: String;
Begin
     Clrscr;
     Writeln ('Input 4 kata:');
     Readln (a);
     Readln (b);
     Readln (c);
     Readln (d);
     Writeln;
     Writeln ('Hasil Teks:');
     Writeln ('a: ',a);
     Writeln ('b: ',b);
     Writeln ('c: ',c);
     Writeln ('d: ',d);
     Readln;
End.
```

Contoh 17.

```
Program hasilakh (input,output);
Const
     jumbil= 5;
Var
     jumakh     : Integer;
     bilinput   : Integer;
     index      : Integer;
Begin
     jumakh := 0;
     Write (‘inputkan bilangannya : ‘);
     For index := 1 To jumbil Do
     Begin
          Read (bilinput);
          Write (bilinput);
          jumakh := jumakh + bilinput;
     End;
     Writeln;
     Write (‘Hasil akhir adalah: ‘,jumakh);
End.
```

Contoh 18. Lengkapi dulu….!

```
Read (nilai);
Writeln ('Nilai tersebut adalah : ', nilai);
Nilai := nilai + 5;
Writeln ('Nilai sekarang adalah : ', nilai);
For index := 4 To 7 Do
Begin
     Write (index);
     Read (next);
     Write (next);
     Nilai := nilai + next mod index;
     Writeln(nilai);
End
```

Contoh 19.

```
Program penjumlahan(input,output);
Var
     jum   : Integer;
     bil   : Integer;
     index : Integer;
     juminput   : Integer;
Begin
     Read (juminput);
     Writeln ('Ada : ',juminput,'nilai yang di inputkan');
     jum:=0;
     Write (' Input : ');
     For index := 1 To juminput Do
     Begin
          Read (bil);
          Write (bil);
          Jum := jum + bil;
     End;
     Writeln;
     Write ('jumlahnya  adalah :', jum);
End.
```

Buatlah program dengan hasil pada layar monitor sebagai berikut:

Latihan 6.

```
MENCARI KONVERSI SUHU
=====================
Input derjad Celsius = -
Derjad Fahrenheit    = -
```

Proses:
Fahrenheit = 9/5 * Celsius + 32

Buatlah program dengan hasil pada layar monitor sebagai berikut:

Latihan 7.

```
Menentukan Harga Bayar
======================
Jumlah barang    = -
Harga satuan Rp. = -
Pajak        Rp. = -

Harga Bayar Rp.  = -
```

Proses:
Harga = Jumlah x harga_satuan + pajak

Buatlah program dengan hasil pada layar monitor sebagai berikut:

Latihan 8.

```
Mencari Luas Lingkaran
======================
Input Panjang jari-jari (cm)      = -
Luas adalah  (cm2)                = -
```

Proses:
Luas = 22/7 * jari * jari

Buatlah program dengan hasil pada layar monitor sebagai berikut:

Latihan 9.

```
RUMUS PHYTAGORAS
================
Panjang sisi vertikal (m)   = -
Panjang sisi horizontal (m) = -
Panjang sisi diagonal (m)   = -
```

Proses:

$diagonal = \sqrt{vertikal^2 + horizontal^2}$

$y = x^2$ => y=SQR(x)

$y = \sqrt{x}$ => y=SQRT(x)

Buatlah program dengan hasil pada layar monitor sebagai berikut:

Latihan 10.

```
MENCARI KELILING PERSEGI PANJANG
================================
Input panjang     : -
Input lebar : -

Keliling          : -
```

Proses:
Keliling = 2*panjang + 2*lebar

Buatlah program dengan hasil pada layar monitor sebagai berikut:

Latihan 11.

```
MENCARI LUAS TRAPESIUM
======================
Input tinggi                : -
Input Panjang sisi atas     : -
Input Panjang sisi bawah    : -

Luas        : -
```

Proses:

$$tinggi = \frac{2 * luas}{atas + bawah}$$

Buatlah program dengan hasil pada layar monitor sebagai berikut:

Latihan 12.

```
MENCARI TINGGI TRAPESIUM
========================
Input luas              : -
Input Panjang sisi atas : -
Input Panjang sisi bawah    : -

Tinggi          : -
```

Proses:
tinggi = ½tinggi*(atas+Panjang)

Buatlah program dengan hasil pada layar monitor sebagai berikut:
Latihan 13.

```
Mencari Standar Deviasi (SD)
========================
Data ke x (data)        = -
Rata – rata (rata)      = -
Jumlah Data (n)         = -
Standar Deviasi (sd)    = -
```

Proses:

$$sd = \sqrt{\frac{(data - rata)^2}{n-1}}$$

Buatlah program dengan hasil pada layar monitor sebagai berikut:
Latihan 14.

```
NILAI TABEL DARI DISTRIBUSI NORMAL
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Nilai rata-rata masing-masing (x)= -
Nilai rata-rata keseluruhan (m)  = -
Data ke x (data)                 = -
Jumlah data (n)                  = -
Nilai Tabel (z)                  = -
```

Proses:

$$z = \frac{x - m}{\sqrt{\frac{(data - x)^2}{n-1}}}$$

Buatlah program dengan hasil pada layar monitor dirancang sendiri:
Latihan 15.

Proses:

$$ep = \frac{1}{2} k.x^2$$

Dimana:
- k : ketetapan gaya pegas
- x : pertambahan panjang pegas

Buatlah program dengan hasil pada layar monitor dirancang sendiri:
Latihan 16.

Proses:

$$g = k\frac{m}{r^2}$$

Dimana:
- g : percepatan grafitasi
- k : ketepatan umum grafitasi
- m : massa benda
- r : jarak titik ke benda

Buatlah program dengan hasil pada layar monitor dirancang sendiri:
Latihan 17.

Proses:

$$t = \sin(s).k$$

Dimana:
- t : tinggi pohon
- s : besar sudut pandang
- k : jarak benda

IV. Perintah perulangan (looping)

Perintah perulangan berfungsi untuk mengulang baris pernyataan dieksekusi oleh alat proses (processor) komputer. Banyak perulangan yang dilakukan tidak terbatas, sesuuai dengan kebutuhan algoritma dan processor mampu untuk mengolahnya. Jenis perintah ini terdiri atas:

FOR, melakukan proses perulangan dengan telah menetapkan batas awal dan batas akhir dari perulangan tersebut.

Bentuk perintah terdiri atas:

```
For (variabel_counter) := (nilai_awal) To (nilai_akhir) Do
Begin
    pernyataan yang ingin diproses ulang ...;
end;
```

atau

```
For (variabel_counter) := (nilai_awal) Downto (nilai_akhir) Do
Begin
    pernyataan yang ingin diproses ulang ...;
End;
```

Tipe **variabel_counter** haus bilangan bulat, seperti **Integer, Byte, Word, Shortint, atau Longint**.

Contoh 20.

```
Program cetak5(output);
Uses Crt;
Var k   : Integer;
Begin
     For k:= 1 to 5 do          {mencetak 5 kali}
     Begin
          Writeln('Tampilan ke-', k);
     End;
     Repeat Until Keypressed;
End.
```

Contoh 21.

```
Program for_1;
Uses Crt;
var
    i: Integer;
Begin
    Clrscr;
    For i := 1  To 1000 Do
    Begin
        writeln('Hello Bahasa Pemrograman Pascal');
    End;
    Readln;
End.
```

Contoh 22.

```
Program for_2;
Uses Crt;
Var
     i: Integer;
Begin
     Clrscr;
     For i := 10  Downto 0 Do
     Begin
         Writeln('Hitung mundur: ',i);
     End;
     Readln;
End.
```

Contoh 23.

```
Program graph;
Var
     gr:integer;
Procedure datar;
Var
     pjg:integer;
Begin
     For pjg := 1 To 8 Do
          Write (‘#’);
     Writeln;
End;

Procedure sisi;
Var
     ss:integer;
Begin
     Write (‘#’);
     For ss := 1 To 6 Do
          Write (‘ ‘);
     Write (‘#’);
End;

Begin
     datar;                {panggil procedure datar}
     For gr := 1 To 5 Do
          sisi;            {panggil procedure sisi}
     datar;                {panggil procedure datar}
End.
```

Latihan: Penggunaan For

Buatlah program dengan hasil pada layar monitor sebagai berikut:
Latihan 18.

```
     0  faktorial adalah        1
     1  faktorial adalah        1
     2  faktorial adalah        2
     3  faktorial adalah        6
     4  faktorial adalah        24
     5  faktorial adalah         .
     6  faktorial adalah         .
     7  faktorial adalah         .
     8  faktorial adalah         .
     9  faktorial adalah         .
     10  faktorial adalah        .
```

sebagai bantuan dan harap dilengkapi !

```
          .
          .
          .
     Writeln (0, 'Faktorial adalah', 1);
     b := 1;
     For i := 1 To 10 Do
     Begin
          Writeln (i, 'Faktorial adalah', i*b);
          B := i*b;
     end
          .
          .
          .
```

Latihan 19.

Dari Latihan 18, modifikasi programnya agar tampilan/outputnya menjadi:

```
     0!    = 1                        = 1
     1!    = 1 X 1                    = 1
     2!    = 1 X 2 X 3                = 6
     3!    = 1 X 2 X 3 X 4            = 24
     4!    = 1 X 2 X 3 X 4 X 5        = 120
     5!    = 1 X 2 X 3 X 4 X 5 X 6    = 720
      .          .                   .
      .          .                   .
      .          .                   .
     dst
```

dimana tanda ! , X , = , dan juga semua bilangan, ditampilkan !

Panduan Latihan 19.

```
Program Latihan2_for; _
Uses Crt;
Var
      i,a: Integer;
      b: Longint;
Begin
      Clrscr;
      Writeln (0, '!   = ', 1);
      b := 1;
      For i := 1 To 10 Do
      Begin
           Write (i, '!   = ');
           For a:= 1 To i Do
           Begin
                Write (a, ' x ');
           End;
           b := i * b;
           Writeln (' = ', b);
      End;
      Readln;
End.
```

Pengulangan mencetak dengan Sistem NESTING LOOP

Pengertian sederhananya dari Nesting Loop adalah **Ada Loop dalam Loop**
Contoh 24

```
Program NL (output);
Var
     index:integer;
Begin
For index := 1 To 10 Do
Begin
     For index := 1 To 8 Do
          Write (‘+’);
     Writeln;
End
End.
```

Contoh 25.

```
Program NL (output);
Var
     baris, kolom : Integer;
Begin
     For baris := 1 To 10 Do
     Begin
          For kolom := 1 To 8 Do
               Write (‘+’);
          Writeln;
     End;
End.
```

Jika ada loop/perulangan yang tidak saling berhubungan dapat digunakan variabel yang sama, seperti Contoh 25.

Contoh 26.

```
Program duasegi(output);
Var
     baris, kolom    : Integer;
Begin
     For baris := 1 To 10 Do
     Begin
          For kolom := 1 To 8 Do
               Write(‘+’);
          Write (‘  ‘);
          For kolom := 1 To 8 Do
               Write (‘+’);
          Writeln;
     End;
End.
```

Latihan 20. lengkapi

```
For foo := 1 To 7 Do
Begin
     For bar := 1 To 5 Do
          Write('ada-ada');
     Writeln ('aja');
End.
```

Latihan 21. Buat program dengan tampilan sebagai berikut:

```
+  #  #  #  #  #  #  #  #  #
+  +  #  #  #  #  #  #  #  #
+  +  +  #  #  #  #  #  #  #
+  +  +  +  #  #  #  #  #  #
+  +  +  +  +  #  #  #  #  #
+  +  +  +  +  +  #  #  #  #
+  +  +  +  +  +  +  #  #  #
+  +  +  +  +  +  +  +  #  #
+  +  +  +  +  +  +  +  +  #
+  +  +  +  +  +  +  +  +  +
```

Latihan 22. Buat program dengan tampilan sebagai berikut:

```
+
+  +
+  +  +
+  +  +  +
+  +  +  +  +

+  +  +  +  +
+  +  +  +
+  +  +
+  +
+
```

Latihan 23. Buat program dengan tampilan sebagai berikut:

```
For index := 5 to 1 do
     Write (index);
```

V. Perintah percabangan (conditional)

Perintah percabangan dibutuhkan pada saat pemilihan terhadap satu pernyataan tertentu dari beberapa pernyataan yang ada untuk dieksekusi oleh processor. Jika sebuah kondisi terpenuhi, maka processor akan mengeksekusi satu pernyataan yang sudah ditentukan sedangkan pernyataan lain tidak.

Perintah percabangan **If**

Bentuk umum pertama:

```
If (kondisi) Then pernyataan;
```

Kondisi berperan sebagai penentu dari stuktur percabangan. Jika kondisi terpenuhi (bernilai TRUE), maka pernyataan akan dieksekusi. Dan jika kondisi tidak terpenuhi (bernilai FALSE), maka pernyataan tidak dieksekusi. Kondisi terdiri atas operasi perbandingan, misalnya apakah variabel a berisi angka 10, atau variabel password berisi String ‘rahasia’.

Bagian yang ditandai dengan **Begin** dan **End;** adalah blok pernyataan yang yang dieksekusi oleh processor jika kondisi terpenuhi (TRUE). Setelah itu, processor akan melanjutkan mengeksekusi pernyataan berikutnya.

Contoh 27.

```
Program struktur_if_then;
Uses Crt;
Var
      Angka : Integer;
Begin
      Clrscr;
      angka := 10;
      If (angka > 5) Then
      Begin
            Writeln ('Variabel "angka" lebih besar dari 5');
      End;
      Writeln ('Belajar Pascal di Duniailkom');
      Readln;
End.
```

Contoh 28.

```
Program struktur_if_then;
Uses Crt;
Var
      Angka : Integer;
Begin
      Clrscr;
      angka:=10;
      If (angka > 5) Then
      Begin
            Writeln ('====================================');
            Writeln ('Variabel "angka" lebih besar dari 5');
            Writeln ('====================================');
      End;
      Writeln ('Belajar Pascal di Dunia Komputer');
      Readln;
End.
```

Contoh 29.

```
Program struktur_if_then;
Uses Crt;
Var
      Angka : Integer;
      Begin
      Clrscr;
      Angka := 5;
      If (angka > 5) Then
            Writeln ('Variabel "angka" lebih besar dari 5');
      Writeln('Belajar Pascal di Dunia Komputer');
      Readln;
End.
```

Contoh 30.

```
Program struktur_if_then;
Uses Crt;
Var
      Angka : Integer;
Begin
      Clrscr;
      Write ('Masukkan sebuah angka: ');
      Readlnv(angka);
      If (angka mod 2 = 0) Then
      Begin
            Writeln('Angka yang anda masukkan merupakan bilangan genap');
      End;
      Readln;
End.
```

Contoh 31.

```
Program kesukaan(input,output);
Const
      Bilkessukaan    = 18;
Var
      bil   : Integer;       {nilai-nilai yang diinputkan}
      index : Integer;       {banyaknya penginputan}
      juminput   : Integer; {berapa kali ingin menginputkan}
      Begin
            Read (juminput);
            Writeln ('Ada :  ' ,juminput,' nilai yang di inputkan');
            For index := 1 To juminput Do
            Begin
                  Read (bil);
                  Write (bil);
                  If (bil = bilkesukaan) Then
                      Write (' adalah bilangan kesukaan saya');
                  Writeln;
            End;
End.
```

Pernyataan yang berada diantara tanda kurung kurawal buka ({) dengan kurung kurawal tutup (}) tidak dieksekusi oleh procesoor. Pernyataan ini berfungsi sebagai komentar.

Beberapa bentuk kondisi:

```
<Yang_dibandingkan> <Operator> <Pembanding>
```

Operator	Keterangan
=	Sama dengan
<	Kecil dari
>	Besar dari
<=	Kecil sama dengan
>=	Besar sama dengan

Contoh 32.

```
Program compare(input,output);
Var
     juminput   : Integer;
     comparand  : Integer;
     bil        : Integer;
     greater    : Integer;
Begin
     Read (juminput);
     Writeln ('Ada sebanyak ', juminput, ' nilai input');
     Read (comparand);
     Wrriteln ('Kita akan check nilai-nilai yang besar dari ',
comparand);
     greater :=0;
     For counter := 1 To juminput Do
     Begin
          Read (bil);
          Write (bil);

          If (bil > comparand) Then
           Begin
               Write (' adalah lebih besar ');
               greater := greater + 1;
          End;
          Writeln;
     End;
     Writeln (' Maka ada ', greater, ' nilai lebih besar.');
     Readln;
End.
```

Bentuk umum kedua:

```
If (kondisi) Then pernyataan_pertama Else pernyataan_kedua;
```

Pernyataan_kedua dieksekusi oleh processor apabila kondisi tidak terpenuhi (False). Sebelum perintah **Else** tidak boleh ada tanda semicolon (;), karena IF-THEN-ELSE merupakan satu perintah (statement).

Contoh 33.

```
Program struktur_if_then_else;
Uses Crt;
Var
     Angka : Integer;
Begin
     Clrscr;
     angka := 4;
     If (angka > 5) Then
     Begin
          Writeln ('Variabel "angka" lebih besar dari 5');
     End
     Else
     Begin
          Writeln ('Variabel "angka" lebih kecil dari 5');
     End;
     Readln;
End.
```

Contoh 34.

```
Program struktur_if_then_else;
Uses Crt;
Var
     angka : Integer;
Begin
     Clrscr;
     Write ('Masukkan sebuah angka: ');
     Readln (angka);
     If (angka mod 2 = 0) Then
     Begin
          Writeln  ('Angka  yang  anda  masukkan  merupakan  bilangan
 genap');
     End
     Else
     Begin
          Writeln('Angka  yang  anda  masukkan  merupakan  bilangan
 ganjil');
     End;
     Readln;
End.
```

Contoh 35.

```
Program compare(input,output);
Var
     juminput   : Integer;
     comparand  : Integer;
     bil        : Integer;
     greater    : Integer;
     smaller    : Integer;
Begin
     Read (juminput);
     Writeln (‘ Ada sebanyak ‘, juminput, ‘ nilai input’);
     Read (comparand);
     Writeln  (‘Kita  akan  check  nilai-nilai  yang  besar  dari  ‘,
comparand);
     greater=0;
     For counter := 1 To juminput Do
     Begin
          Read (bil);
          Write (bil);

          If (bil > comparand) Then
          Begin
               Write (‘ adalah lebih besar ‘);
               greater := greater + 1;
          End        {jika nilai > comparand};
          Else
          Begin
               Write (‘ adalah lebih kecil ‘);
               smaller := smaller + 1;
          End  {jika nilai < comparand}
          Writeln;
     End;
     Writeln (‘ Maka ada ‘, greater, ‘ nilai lebih besar.’);
     Writeln (‘ Dan ada ‘,smaller, ‘niai yang lebih kecil.’);
      Readln;
End.
```

Di dalam pernyataan juga dapat diisi dengan:

```
If (kondisi) Then pernyataan;

Atau

If (kondisi) Then pernyataan_pertama Else pernyataan_kedua;
```

Contoh 36.

```
Program coba_else (input,output);
Var
     padabrs, padacol : Integer;
     baris, colom     : Integer;

Begin
     Read (padabrs, padacol);
     Writeln ('Posisi pada(', padabrs, ', ',padacol, ') .');
     For baris :=1 To 8 Do
     Begin
         For colom := 1 To 8 Do
         If (padabrs And padacol) THEN
             Write (' * ')
         else
             If ((baris + colom) Mod 2 = 0) Then
                 Write (' W')
             Else
                 Write (' B');
        Writeln;
      End;
End.
```

Contoh 37.

```
Program coba_else (input,output);
Var
     padabrs, padacol : Integer;
     baris, colom     : Integer;

Begin
     Read (padabrs, padacol);
     Writeln ('Posisi pada(', padabrs, ', ',padacol, ') .');
     For baris :=1 To 8 Do
     Begin
         For colom := 1 To 8 Do
         If ((padabrs=baris)  And  (padacol=colom)) THEN
             Write (' * ')
         else
             If ((baris + colom) Mod 2 = 0) Then
                 Write (' W')
             Else
                 Write (' B');
        Writeln;
      End;
End.
```

Buatlah program dengan hasil pada layar monitor sebagai berikut:
Latihan 24.

```
Mencari Standar Deviasi (SD)
============================
Data ke x (data)        = -
Rata – rata (rata)      = -
Jumlah Data (n)         = -
Standar Deviasi (sd)    = -
Keterangan data         = -
```

Proses:

$$sd = \sqrt{\frac{(data - rata)^2}{n - 1}}$$

-jika sd = 0, maka ket = Valid
-jika sd > 1, maka ket = Kurang Valid
-jika sd < 1, maka ket = Tidak Valid

Buatlah program dengan hasil pada layar monitor sebagai berikut:
Latihan 25.

```
JENIS SEGITIGA TEOREMA PHITAGORAS
=================================
Panjang sisi vertikal   : -
Panjang sisi horizontal : -
Panjang sisi miring     : -

Jenis segitiga          : -
```

Proses:

- $diagonal = \sqrt{vertikal^2 + horizontal^2}$
- Jika miring > diagonal, maka jenis = Segitiga sudut lancip
- Jika miring = diagonal, maka jenis = Segitiga siku-siku
- Jika miring < diagonal, maka jenis = Segitiga tumpul

Latihan 26. Buat program dengan tampilan sebagai berikut:

```
INFORMASI DATA NILAI
====================
Nomor BP         : -
Nama Mahasiswa   : -
Matakuliah       : -
Nilai Tugas            : -
Nilai Quiz       : -
Nilai Mid        : -
Nilai Ujian Akhir      : -
Nilai Akhir            : -
Keterangan       : -
```

Proses:

- Nilai Akhir = 10% x Nilai Tugas + 15% x Nilai Quiz + 35% x Nilai Mid + 40% x Nilai Ujian Akhir
- Jika Nilai Akhir >= 55, maka Keterangan = Lulus, kalau tidak Keterangan = Gagal

Latihan 27. Buat program dengan tampilan sebagai berikut:

```
INFORMASI BODY MASS INDEX (BMI)
===============================
Nama Mahasiswa   : -
Umur (tahun)     : -
Tinggi badan (cm)     : -
Berat badan (kg)      : -
BMI                   : -
Keterangan            : -
```

Proses:

- $bmi = \frac{berat\ badan}{tinggi\ badan : 100}$
- Jika bmi < 17, maka Keterangan = underweight
- Jika 17 < bmi < 21, maka Keterangan = ideal
- Jika bmi > 21, maka Keterangan = obesitas

Latihan 28. Buat program dengan tampilan sebagai berikut:

```
INFORMASI PENJUALAN BARANG
============================
Kode Barang      : -
Nama Barang      : -
Harga Satuan     : Rp. -
Jumlah Jual      : -
Bonus      : -
Harga Bayar      : Rp. -
```

Proses:

- Jika Jumlah Jual > 25, maka Bonus = Payung
- Jika Jumlah Jual > 10, maka Bonus = Jam Dinding
- Harga Bayar = Jumlah Jual X Harga Satuan

Latihan 29. Buat program dengan tampilan sebagai berikut:

```
Perhitungan ZAKAT
=================
Jenis Zakat [1-4]     : -
Keterangan Zakat: -
Nisab            : -
Kadar                 : -
Haul             : -
Jumlah harta          : -
Jumlah orang          : -
Jumlah Zakat          : -
```

Proses:

- Jika Jenis = 1, maka Ket = Zakat Uang, Nisab = 85 gram emas, Kadar = 2,5, Haul = 1 tahun, Jumlah Zakat = Kadar * Jumlah Harta
- Jika Jenis = 2, maka Ket = Zakat Hasil Tani, Nisab = 653 kg beras, Kadar = 5, Haul = Setiap Panen, Jumlah Zakat = Kadar * Jumlah Harta
- Jika Jenis = 3, maka Ket = Zakat Fitrah, Nisab = Hidup saat Lebaran, Kadar = 3,5, Haul = 1 tahun, Jumlah Zakat = Kadar * Jumlah orang

Latihan 30. Buat program dengan tampilan sebagai berikut:

```
Perhitungan ZAKAT
===============
Jenis Zakat [1-4]     : -
Keterangan Zakat: -
Nisab           : -
Kadar                 : -
Haul            : -
Jumlah harta          : -
Jumlah orang          : -
Jumlah Zakat          : -
```

Proses:

- Jika Jenis = 1, maka Ket = Zakat Uang, Nisab = 85 gram emas, Kadar = 2,5, Haul = 1 tahun, Jumlah Zakat = Kadar * Jumlah Harta
- Jika Jenis = 2, maka Ket = Zakat Hasil Tani, Nisab = 653 kg beras, Kadar = 5, Haul = Setiap Panen, Jumlah Zakat = Kadar * Jumlah Harta
- Jika Jenis = 3, maka Ket = Zakat Fitrah, Nisab = Hidup saat Lebaran, Kadar = 3,5, Haul = 1 tahun, Jumlah Zakat = Kadar * Jumlah orang

Latihan 31. Buat program dengan tampilan sebagai berikut:

```
Pembagian Harta Gonimah Perang
==============================
Nama Perang                     : -
Jumlah (keping emas)            : -
Jumlah Sahabat r.hum  (orang)   : -
Nilai harta rampasan       Rp.  : -
Bagian Nabi                Rp.  : -
Bagian Sahabat             Rp.  : -
```

Proses:

- 1 keping emas = 4,25 gram
- 1 gram = Rp. 667.000,-
- Bagian Rasulullah = 1/5 bagian

Latihan 32. Buat program dengan tampilan sebagai berikut:

```
PT. MAJU TERUS PANTANG MUNDUR
~~~~~~~~~~~~~~~~~~~~~~~~~~~~
NIK                         : -
Nama                        : -
Jenis Kelamin   [L/P]       : -
Status Keluarga [S/K/D/J]   : -
Golongan                    : -
Gaji Pokok                  : -
Tunjangan Anak (Rp)         : -
Tunjangan Istri (Rp)        : -
Potongan (Rp)               : -
Gaji Bersih                 : -
```

Proses :

- Jenis Kelamis (L) Laki-Laki, dan (P) Perempuan
- Status Keluarga terdiri dari S (Sendiri), K(Kawin), D(Duda), dan (J)Janda
- Untuk menentukan Tunjangan Anak(TA) dan Istri (TI) adalah sebagai berikut (dari Gaji Pokok):

Jenis Kelamis	Status Keluarga	Tunjangan Anak	Tunjangan Istri
L	S	0%	0%
	K	10%	15%
	D	10%	0%
P	S	0%	0%
	K	15%	10%
	J	15%	0%

- Setiap Pegawai dikenakan Potogan 10% dari Gaji kotor
- Total Gaji Bersih = (TA + TI + Gapok) – Potongan

Latihan 33. Buat program dengan tampilan sebagai berikut:

```
PT. MAJU TERUS PANTANG MUNDUR
==============================

NIK                   : -
Nama                  : -
Jenis Kelamin   [L/P]      : -
Status Keluarga [S/K/SP]   : -
Keterangan                 : -
Gaji Pokok            : -
Tunjangan Anak (Rp)        : -
Tunjangan Istri / Suami (Rp)     : -
Potongan (Rp)              : -
Gaji Bersih                : -
```

Proses :

- Jika Status Keluarga = S, maka
  - Tunjangan Anak = 0
  - Tunjangan Istri/Suami = 0
  - Keterangan = Single
- Jika Status Keluarga = K, maka
  - Tunjangan Anak = 10% x Gaji Pokok
  - Tunjangan Istri/Suami = 15% x Gaji Pokok
  - Keterangan = Kawin
- Jika Status Keluarga = SP, maka
  - Tunjangan Anak = 10% x Gaji Pokok
  - Tunjangan Istri/Suami = 0
  - Keterangan = Single Parent
- Potongan = 2,5% x (Gaji Pokok + Tunjangan Anak + Tunjangan Suami/Istri)
- Gaji Bersih = Gaji Pokok + Tunjangan Anak + Tunjangan Istri - Potongan

Latihan 34. Buat program dengan tampilan sebagai berikut:

```
LAPORAN STOCK BARANG Economic Order Quantity (EOQ)
=================================================
Kode Barang                 : -
Nama Barang                 : -
Satuan barang               : -
Harga Satuan                : -
Jumlah Kebutuhan (r)        : -
Biaya Pemesanan (o)         : Rp. -
Biaya Penyimpanan (c)       : Rp. -
Jumlah Stock (eoq)          : -
Total Biaya                 : Rp. -
Keterangan                  : -
Kelebihan/Kekurangan Stock  : -
```

Proses :

- $eoq = \sqrt{\frac{2.r.o}{c}}$
- Total Biaya = Jumlah Stock * Harga Satuan + Biaya Pemesanan + Biaya Penyimpanan
- Jika Jumlah Kebutuhan > Jumlah Stock, maka Keterangan = Stock Kurang; Kelebihan = Jumlah Kebutuhan – Jumlah Stock
- Jika Jumlah Kebutuhan = Jumlah Stock, maka Keterangan = Stock Cukup; Kelebihan = 0
- Jika Jumlah Kebutuhan < Jumlah Stock, maka Keterangan = Stock Berlebih; Kelebihan = Jumlah Stock – Jumlah Kebutuhan

Latihan 35. Buat program dengan tampilan sebagai berikut:

```
PENGOLAHAN DATA RETRIBUSI SAMPAH
================================

Bulan             : -
Kode Pelanggan    : -
Nama Pelanggan    : -
Tipe Pelanggan    : -
Keterangan        : -
Jumlah pengambilan / bln    : -
Biaya Administrasi (Rp)     : -
Biaya Bulanan (Rp)          : -
Total Tagihan (Rp)          : -
```

Proses :

- Jika Tipe Pelanggan = R, maka
  - Keterangan = Rumah Tangga
  - Biaya Administrasi = 10000
  - Biaya Bulanan = 15000
- Jika Tipe Pelanggan = T, maka
  - Keterangan = Toko
  - Biaya Administrasi = 20000
  - Biaya Bulanan = 35000
- Jika Tipe Pelanggan = P, maka
  - Keterangan = Perusahaan
  - Biaya Administrasi = 30000
  - Biaya Bulanan = 55000

Latihan 36 Buat program dengan tampilan sebagai berikut:

```
DATA PENJUALAN BARANG
---------------------
Kode Barang          : -
Nama Barang     : -
Satuan          : -
Permintaan      : -
Harga Dasar (Rp): -
Harga Jual      : -
```

Proses :
- Jika Permintaan > 25, maka Harga Jual = 15% x Hrg. Dasar
- Jika Permintaan > 50, maka Harga Jual = 35% x Hrg. Dasar
- Jika Permintaan > 100, maka Harga Jual = 55% x Hrg. Dasar

Latihan 37. Buat program dengan tampilan sebagai berikut:

```
Biaya Layanan RUMAH SAKIT
=========================
Kode Layanan          : -
Nama Pasien           : -
Alamat                : -
Jenis [VIP/Ekonomi]   : -
Telpon                : -
E-Mail                :-
Harga Dasar (Rp)      : -
Besar Tagihan (Rp)    : -
```

Proses :
- Jika Jenis = VIP, maka Besar Tagihan = 165% x Harga Dasar
- Jika Jenis = Ekonomi, maka Besar Tagihan = 105% Harga Dasar

Latihan 38. Buat program dengan tampilan sebagai berikut:

```
DATA PENJUALAN BARANG
=========================
Kode Barang           : -
Nama Barang           : -
Satuan Barang         : -
Harga Satuan (Rp)     : -
Jenis Kebutuhan       : -
Jumlah                : -
Keterangan            : -
Total Harga (Rp)      : -
```

Proses :
- Total Harga = Harga Satuan * Jumlah
- Jika Jenis Kebutuhan = Primer, maka Keterangan = Barang Pokok
- Jika Jenis Kebutuhan = Secunder, maka Keterangan = Barang Pendukung
- Jika Jenis Kebutuhan = Lux, maka Keterangan = Barang Mewah

Latihan 39. Buat program dengan tampilan sebagai berikut:

```
DATA PENJUALAN TICKET PESAWAT
=================================
Kode Tiket                      : -
Nama Pemesan                    : -
Tujuan                          : -
Kelas Tempat Duduk [VIP/EKO]    : -
Harga Dasar / orang (Rp)        : -
Jumlah Penumpang
     Dewasa (orang)        : -
     Anak-anak (orang)     : -
Total Harga (Rp)           : -
Keterangan                 : -
```

Proses :

- Total Harga = 115% x Dewasa x Harga Dasar + 90% x Anak-anak x Harga Dasar
- Jika Kelas = VIP, maka Keterangan = Kelas Excekutif
- Jika Kelas = EKO, maka Keterangan = Kelas Umum

Latihan 40. Buat program dengan tampilan sebagai berikut:

```
DATA TAGIHAN REKENING LISTRIK
============================
Periode Tagihan       : -
Kode Pelanggan        : -
Nama Pelanggan        : -
Alamat Pelanggan      : -
Tipe Langganan [ 1 - 4 ]   : -
Pemakaian (kWh)            : -
Besar Tagihan (Rp)         : -
```

- Jika Tipe Langganan = 1, maka Besar Tagihan = 500 x Pemakaian
- Jika Tipe Langganan = 2, maka Besar Tagihan = 600 x Pemakaian
- Jika Tipe Langganan = 3, maka Besar Tagihan = 7500 x Pemakaian
- Jika Tipe Langganan = 4, maka Besar Tagihan = 1000 x Pemakaian

Latihan 41. Buat program dengan tampilan sebagai berikut:

```
Data Persedian Obat Apotik XYZ
==============================
Kode Obat                   : -
Nama Obat                   : -
Jenis Obat                  : -
Golongan                    : -
Keterangan                  : -
Jumlah Persedian (satuan)       : -
Harga Satuan (Rp)               : -
Nilai Harga Obat (Rp)      : -
```

Proses :

- Jika Jenis Obat = A1, maka Golongan = Ringan, Keterangan = Label Hijau
- Jika Jenis Obat = A2, maka Golongan = Ringan Terbatas, Keterangan = Label Biru

- Jika Jenis Obat = A3, maka Golongan = Keras, Keterangan = Label Merah
- Nilai Harga Obat = Jumlah Persedian x Harga Satuan

Latihan 42. Buat program dengan tampilan sebagai berikut:

```
Data Pinjaman Nasabah
=====================
Kode Nasabah                      : -
Nama Nasabah                      : -
Pokok Hutang     [Rp]             : -
Jangka Waktu Pinjaman [tahun]          : -
Cicilan Pokok per Bulan (Rp)           : -
Bunga Pertahun (Rp)               : -
Total Pinjaman (Rp)                    : -
Bayar Cicilan per Bulan                : -
Biaya Administrasi (Rp)                : -
Biaya Asuransi (Rp)               : -
Total Hutang (Rp)                      : -
```

Proses :

- Cicilan Pokok per Bulan = Pokok Hutang / (Jangka Waktu Pinjaman x 12)
- Jika Jangka Waktu Pinjaman = 1, maka Bungan pertahun = 7% x Pokok Hutang
- Jika Jangka Waktu Pinjaman = 2, maka Bungan pertahun = 9% x Pokok Hutang
- Jika Jangka Waktu Pinjaman = 3, maka Bungan pertahun = 10% x Pokok Hutang
- Total Pinjaman = Pokok Hutang + (Bunga pertahun x Jangka Waktu Pinjaman)
- Total Hutang = Total Pinjaman + Biaya Administrasi + Biaya Asuransi

Latihan 43. Buat program dengan tampilan sebagai berikut:

```
     TRANSAKSI PENJUALAN PULSA ELEKTRONIK
     ====================================
     Nomor Kartu                 = -
     Nama Pemilik                = -
     Kode Operator [T/I/X] = -
          Nama Operator          = -
          Biaya SMS Operator Sama    = Rp.-
          Biaya SMS Operator Lain    = Rp.-
          Biaya percakapan per menit Operator Sama    = Rp.-
          Biaya percakapan per menit Operator Lain    = Rp.-
          Biaya Transaksi                  = Rp. –
     Jumlah Pulsa          = Rp. –
     Masa Aktif (hari)           = -
     Biaya Transaksi       = Rp. –
```

Proses :

- Jika Kode Operator = T, maka
  Nama Operator = Telkomsel
  Biaya SMS Operator Sama = 150
  Biaya SMS Operator Lain = 180
  Biaya percakapan per menit operator Sama = 850
  Biaya percakapan per menit operator Lain = 1050
- Jika Kode Operator = I, maka
  Nama Operator = Indosat
  Biaya SMS Operator Sama = 125
  Biaya SMS Operator Lain = 150
  Biaya percakapan per menit operator Sama = 900
  Biaya percakapan per menit operator Lain = 1150
- Jika Kode Operator = X, maka
  Nama Operator = XL
  Biaya SMS Operator Sama = 50
  Biaya SMS Operator Lain = 80
  Biaya percakapan per menit operator Sama = 650
  Biaya percakapan per menit operator Lain = 1250
- Jika Jumlah Pulsa = 15.000, maka Masa Aktif = 15
- Jika Jumlah Pulsa = 50.000, maka Masa Aktif = 30
- Jika Jumlah Pulsa = 100.000, maka Masa Aktif = 90
- Biaya Transaksi = Jumlah Pulsa + 2000

Latihan 44. Buat program dengan tampilan sebagai berikut:

```
BIAYA PARKIR KENDARAAN HARIAN
Call Center : 7890
================================
No. Plat Kendaraaan   : -
Tanggal Masuk    : -
Kode Lokasi Parkir    : -
Jam Masuk        : -
Jam Keluar       :-
Lama Parkir (jam)     :-
Biaya Parkir per Jam  : Rp. –
Besar Biaya Parkir    : Rp.-
Besar Pulsa yang Dipotong  : -
Kode Yang Dikirim          : -
```

Proses :

- Jika Kode Lokasi Parkir = L1, maka Biaya Parkir per Jam = 3000
- Jika Kode Lokasi Parkir = L2, maka Biaya Parkir per Jam = 3500
- Jika Kode Lokasi Parkir = L3, maka Biaya Parkir per Jam = 4000
- Jika Kode Lokasi Parkir = R1, maka Biaya Parkir per Jam = 5000
- Jika Kode Lokasi Parkir = R2, maka Biaya Parkir per Jam = 6000
- Lama Parkir = Jam Keluar – Jam Masuk
- Besar Biaya Parkir = Lama Parkir x Biaya Parkir per Jam
- Jika Besar Biaya Parkir >=3000, maka Besar Pulsa 10, Kode yang dikirim = P10
- Jika Besar Biaya Parkir >=6000, maka Besar Pulsa 20, Kode yang dikirim = P20
- Jika Besar Biaya Parkir >=10000, maka Besar Pulsa 30, Kode yang dikirim = P30
- Jika Besar Biaya Parkir >=20000, maka Besar Pulsa 50, Kode yang dikirim = P50

Latihan 45. Buat program dengan tampilan sebagai berikut:

```
INFORMASI TAGIHAN AIR
~~~~~~~~~~~~~~~~~~~~~
Identitas Pelanggan   : -
Nama Pelanggan        : -
Alamat Pelanggan      : -
Gol Tarif [ I - IV ]  : -
Bulan-Tahun Tagihan   : -
Golongan Pelanggan    : -
Harga air per kubik   Rp.   : -
Biaya Administrasi    Rp.   : -
Retribusi Sampah      Rp.   : -
Pemakaian (kubik)           : -
Denda                 Rp.   : -
Total Tagihan        Rp.    : -
```

Proses :

- Jika Gol Tarif = I, maka Gol Pelanggan = Sosial, Harga air = 1.800, Biaya Adm = 8.000
- Jika Gol Tarif = II, maka Gol Pelanggan = Rumah Tangga, Harga air = 3.100, Biaya Adm = 15.000
- Jika Gol Tarif = III, maka Gol Pelanggan = Niaga, Harga air = 4.500, Biaya Adm = 25.000
- Jika Gol Tarif = IV, maka Gol Pelanggan = Khusus, Harga air = 6.200, Biaya Adm = 100.000
- Retribusi Sampah = 5.000
- Total Tagihan = Biaya Adm + Harga air * Pemakaian + Denda + Retribusi Sampah

Latihan 46. Buat program dengan tampilan sebagai berikut:

```
Biaya Layanan RUMAH SAKIT
~~~~~~~~~~~~~~~~~~~~~~~~~
Kode Layanan          : -
Nama Pasien           : -
Alamat                : -
Jenis [VIP/Ekonomi]   : -
Lama Inap (hari)      : -
Telpon                : -
E-Mail                : -
Harga Dasar (Rp)      : -
Biaya Dokter (Rp)     : -
Biaya Obat (Rp)       : -
Biaya Kamar (Rp)      : -
Besar Tagihan (Rp)    : -
```

Proses :

- Jika Jenis = VIP, maka Biaya Kamar = 165% x Harga Dasar
- Jika Jenis = Ekonomi, maka Biaya Kamar = 105% Harga Dasar
- Besar Tagihan = Biaya Kamar + Biaya Obat + Biaya Dokter.

Latihan 47. Buat program dengan tampilan sebagai berikut:

```
Data Tagihan Listrik
~~~~~~~~~~~~~~~~~~~
No. Reg                : -
Nama Pelanggan         : -
Tipe Langganan         : -
Angka Meter Bulan Lalu     : -
Angka Meter Bulan Ini : -
Banyak Listrik Terpakai (kWh): -
Keterangan Pelanggan  : -
Biaya per kWh         Rp. -
Biaya Administrasi          Rp. -
Biaya Pemakaian Listrik     Rp. -
Pajak                       Rp. -
Besar Tagihan         Rp. -
```

Proses :

- Banyak Listrik Terpakai = Angka Meter Bulan Ini – Angka Meter Bulan Lalu
- Jika Tipe Langganan = R1, maka
  - Keterangan Pelanggan = Rumah Tangga
  - Biaya Administrasi = 15000,
  - Biaya per kWh = 1200
  - Jika Banyak Listrik Terpakai >= 300, maka
    - Biaya Pemakaian Listrik = (Biaya per kWh x 300) + ((Banyak Listrik Terpakai – 300) x 150% x Biaya per kWh) + Biaya Administrasi
  - Jika Banyak Listrik Terpakai < 300, maka
    - Biaya Pemakaian Listrik = (Biaya per kWh x Biaya per kWh) + Biaya Administrasi
- Jika Tipe Langganan = R2, maka
  - Keterangan Pelanggan = Rumah Toko
  - Biaya Administrasi = 25000
  - Biaya per kWh = 2225
  - Jika Banyak Listrik Terpakai >= 450, maka
    - Biaya Pemakaian Listrik = (Biaya per kWh x 450) + ((Banyak Listrik Terpakai – 450) x 200% x Biaya per kWh) + Biaya Administrasi
  - Jika Banyak Listrik Terpakai < 450, maka
    - Biaya Pemakaian Listrik = (Biaya per kWh x Biaya per kWh) + Biaya Administrasi
- Jika Tipe Langganan = R3, maka
  - Keterangan Pelanggan = Industri
  - Biaya Administrasi = 35000
  - Biaya per kWh = 4250
  - Jika Banyak Listrik Terpakai >= 950, maka
    - Biaya Pemakaian Listrik = (Biaya per kWh x 950) + ((Banyak Listrik Terpakai – 950) x 250% x Biaya per kWh) + Biaya Administrasi
  - Jika Banyak Listrik Terpakai < 950, maka
    - Biaya Pemakaian Listrik = (Biaya per kWh x Biaya per kWh) + Biaya Administrasi
- Pajak = 10% x Biaya Pemakaian Listrik
- Besar Tagihan = Biaya Pemakaian Listrik - Pajak

Latihan 48. Buat program dengan tampilan sebagai berikut:

```
Data Gaji Pegawai
=================
Nomor induk pegawai         : -
Nama pegawai                : -
Golongan [1-3]              : -
Jam kerja per hari          : -
Hari kerja per bulan        : -
Status pegawai         : -
Gaji pokok (Rp)        : -
Tunjangan              : -
Uang makan (Rp)        : -
Uang transportasi (Rp)      : -
Uang lembur (Rp)       : -
Bonus (Rp)             : -
Gaji Kotor (Rp)        : -
Pajak PPh (Rp)              : -
Gaji bersih (Rp)       :
```

Proses:

- Jika Golongan = 1, maka Status pegawai = Tetap
- Jika Golongan = 2, maka Status pegawai = Kontrak
- Jika Golongan = 3, maka Status pegawai = Lepas
- Jika Status pegawai = Tetap, maka Gaji pokok = 7.500.000
- Jika Status pegawai = Kontrak, maka Gaji pokok = 5.000.000
- Jika Status pegawai = Lepas, maka Gaji pokok = 3.500.000
- Jika Golongan = 1, maka Tunjangan = 15% x Gaji pokok
- Jika Golongan = 2, maka Tunjangan = 10% x Gaji pokok
- Jika Golongan = 3, maka Tunjangan = 5% x Gaji pokok
- Jika Jam kerja >=9, maka Uang makan = (2 x 12.500) x Hari kerja; kalau tidak Uang makan = 12.500 x Hari kerja
- Uang trasportasi = 7.500 x Hari kerja
- Jika Jam kerja > 8, maka Uang lembur = (Jam kerja – 8) x Hari kerja; kalau tidak Uang lembur = 0
- Jika Hari kerja >=26, maka Bonus = 5% x Gaji poko; kalau tidak Bonus = 0
- Gaji kotor = Gaji pokok + Tunjangan + Uang makan + Uang Transport + Uang lembur + Bonus
- Pajak PPh = 10% x Gaji kotor
- Gaji bersih = Gaji kotor – Pajak PPh

Latihan 49. Buat program dengan tampilan sebagai berikut:

```
INFORMASI SEWA KAMAR HOTEL
==========================
Nomor Kamar           : -
Tipe Kamar [1-3]: -
Keterangan Kamar: -
Harga Sewa / hari     : Rp. –
Nama Penyewa     : -
Lama Menginap (hari): -
Discount         : Rp.-
Besar Tagihan         : Rp. –
```

Proses:

- Jika Tipe Kamar = 1, maka Keterangan Kamar = Exlusive, Harga Sewa Perhari = 650000
- Jika Tipe Kamar = 2, maka Keterangan Kamar = VIP, Harga Sewa Perhari = 500000
- Jika Tipe Kamar = 3, maka Keterangan Kamar = Ekonomi, Harga Sewa Perhari = 350000
- Discount = 15% x (Lama Menginap x Harga Sewa Perhari)
- Besar Tagihan = Lama Menginap x Harga Sewa Perhari - Discount

Perintah loncat (jumping)

Perintah loncat dalam Pascal adalah **Goto**. Peintah ini berfungsi loncat ekseksi ke pernyataan yang dideklarasikan oleh Label.

Bentuk penulisan:

```
Goto Label;
```

Label harus berada didalam blok yang sama dengan pernyataan Goto. Hal ini tidak memungkinkan melompat keluar dari dalam procedure atau function. Suatu Label dideklarasikan pada bagian deklarasi Label.

Bentuk Penulisan:

```
Label identifier, ... identifier;
```

Contoh 38.

```
Program penggunaan_GoTo_Label;

Uses Crt;
Label atas;
Var
     nil1, nil2 : Integer;
     nilRata    : Real;
     Lagi       : Char;
Begin
     atas:
     ClrScr;
     WriteLn ('Program Hitung Nilai');
     WriteLn ('--------------------');
     WriteLn;
     Write ('Masukan Nilai 1 : ');
     ReadLn (nil1);
     Write ('Masukan Nilai 2 : ');
     ReadLn (nil2);
     nilRata := (nil1 + nil2)/2;
     WriteLn ('Nilai Rata - Rata : ',nilRata:5:2);
     WriteLn;
     Write ('Ingin Hitung Lagi[Y/T] : ');
     ReadLn (Lagi);
     If (Lagi='Y') Then
     GoTo atas;
End.
```

Contoh 39.

```
Program contohLabel;
Uses Crt;
Label menu, luasPersegi, luasSegiTiga, luasLingkaran, keluar;
Const phi = 3.14;
var
     s          : Integer;
     p,l,luasP  : Integer;
     a,t,luasS  : Real;
     r,luasL    : Real;
Begin
     Clrscr;
     Goto menu;

     menu:
          Begin
               Writeln ('Menu Menghitung Luas');
               Writeln ('=====================');
               Writeln ('1. Luas Persegi');
               Writeln ('2. Luas Segi Tiga');
               Writeln ('3. Luas Lingkaran');
               Writeln ('4. Keluar');
               Write ('Masukkan Pilihan Menu [1-4] : '); Readln(s);
               Writeln;
               If (s = 1) Then
                    Goto luasPersegi;
               If (s = 2) Then
                    Goto luasSegiTiga;
               If (s = 3) Then
                    Goto luasLingkaran;
               If (s = 4) Then
                    Goto keluar;
          End;

     luasPersegi:
          Begin
               Writeln ('Menghitung Luas Persegi');
               Writeln ('======================');
               Write ('Masukkan Panjang     : '); Read(p);
               Write ('Masukkan Lebar       : '); Read(l);
               luasP := p * l;
               Write ('Luas Persegi Panjang : ', luasP);
               Writeln;
               Goto menu;
          End;

     luasSegiTiga:
          Begin
               Writeln ('Menghitung Luas Segi Tiga');
               Writeln ('=========================');
               Write ('Masukkan Alas   : '); Read(a);
               Write ('Masukkan Tinggi : '); Read(t);
               luasS := 0.5 * a * t;
               Write ('Luas Segitiga   : ', luasS:3:2);
               Writeln;
               Goto menu;
          End;
```

```
     luasLingkaran:
          Begin
               Writeln ('Menghitung Luas Lingkaran');
               Writeln (‘=========================');
               Write('Masukkan Jari-jari : ');read(r);
               luasL := phi * r * r;
               Write('Luas Lingkaran     : ', luasL:3:2);
               Writeln;
               Goto menu;
          End;

     keluar:
End.
```

VII. Larik (Array)

Array adalah tipe data bentukan yang terdiri dari kumpulan tipe data lain yang diwakili oleh 1 (satu) nama variabel. Dalam satu nama variabel ini dapat menampung banyak nilai (value) dengan tipe dan panjang (type & width) datanya (field) sama. Seperti contoh berikut:

Contoh 40.

```
Program tipe_array;
Uses Crt;
Var
     nilai1, nilai2, nilai3, nilai4   : Integer;
Begin
     Clrscr;
     nilai1 := 17;
     nilai2 := 12;
     nilai3 := 5;
     nilai4 := 40;

     Writeln ('nilai1: ', nilai1);
     Writeln ('nilai2: ', nilai2);
     Writeln ('nilai3: ', nilai3);
     Writeln ('nilai4: ', nilai4);

     Readln;
End.
```

Tipe data nilai pada Program 40. Dirobah kedalam bentuk Array.

Contoh 41.

```
Program tipe_array;
Uses Crt;
Var
     Nilai      : Array [1..4] Of Integer;
Begin
     Clrscr;
     Nilai [1] := 17;
     Nilai [2] := 12;
     Nilai [3] := 5;
     Nilai [4] := 40;

     Writeln ('nilai1: ', nilai [1]);
     Writeln ('nilai2: ', nilai [2]);
     Writeln ('nilai3: ', nilai [3]);
     Writeln ('nilai4: ', nilai [4]);

     Readln;
End.
```

Pengisian dan menampilkan data dapat dilakukan secara acak, tetapi harus disesuaikan dengan batasan dan ketetapan yang ada. Lakukan Latihan 42. dibawah ini:

Contoh 42.

```
Program tipe_array;
Uses Crt;
Var
     Nilai      : Array [1..4] Of Integer;
Begin
     Clrscr;
     Nilai [3] := 17;
     Nilai [4] := 12;
     Nilai [2] := 5;
     Nilai [1] := 40;

     Writeln ('nilai1: ', nilai [3]);
     Writeln ('nilai2: ', nilai [4]);
     Writeln ('nilai3: ', nilai [2]);
     Writeln ('nilai4: ', nilai [1]);

     Writeln ('nilai1: ', nilai [1]);
     Writeln ('nilai2: ', nilai [2]);
     Writeln ('nilai3: ', nilai [3]);
     Writeln ('nilai4: ', nilai [4]);

     Readln;
End.
```

Seluruh tipe data dapat digunakan dalam Array dan jumlah indek Arraynya harus sesuai dengan kemampuan memory komputer. Secara standar hanya mampu menampung 255 buah indek yang dimulai dari 0.

Contoh 43.

```
Program tipe_array;
Uses Crt;
Var
     Kata  : Array[0..255] Of String[20];
Begin
     Clrscr;

     Kata [24] := 'Sedang ';
     Kata [25] := 'belajar Pascal ';
     Kata [26] := 'di ';
     Kata [27] := 'Komputer';

     Write (kata [24]);
     Write (kata [25]);
     Write (kata [26]);
     Write (kata [27]);

     Readln;
End.
```

Yang perlu diperhatikan adalah:
- Setiap variabel yang dinyatakan sebagai ARRAY harus dideklarasikan pada blok VAR

- Hal yang perlu dideklarasikan adalah nama variabel, indeks variabel dan tipe data variabel
- Deklarasi ini berfungsi untuk memesanan tempat/memori dari setiap variabel ARRAY tersebut.

Contoh 44.

```
Program mhs (input, output);
Uses Crt;
Var
     Nama  : Array [1..10] Of String[15];
     nobp  : Array [1..10] Of Longint;
     quiz,mid,akhir,rata   : Array [1..10] Of Real;
     indeks: Array[1..10] of String[1];
     i,n,k : Integer;
Begin
     Clrscr;
     Write (‘Jumlah Mahasiswa    : ‘); Readln(n);
     For i := 1 To n Do
     Begin
           clrscr;
           Gotoxy (10, 5); Write (‘Nama     : ‘);
           Gotoxy (10, 6); Write (‘Nobp          : ‘);
           Gotoxy (10, 7); Write (‘Nilai Quiz    : ‘);
           Gotoxy (10, 8); Write (‘Nilai Mid     : ‘);
           Gotoxy (10, 9); write (‘Nilai Akhir   : ‘);
           Gotoxy (30, 5); Readln (nama [j]);
           Gotoxy (30, 6); Readln (nobp [j]);
           Gotoxy (30, 7); Readln (quiz [j]);
           Gotoxy (30, 8); Readln (mid [j]);
           Gotoxy (30, 9); Readln (akhir [j]);
     End;

     For k := 1 To n Do
     Begin
           Rata [k] := 0.25 * quiz [k] + 0.25 * mid [k] + 0.5 * akhir
[k];
           Case rata[k] Of
                  0..39    : indeks [k] :=’E’;
                  40..54   : indeks [k] :=’D’;
                  55..64   : indeks [k] :=’C’;
                  65..79   : indeks  [k] :=’B’;
                  80..100 : indeks [k] :=’A’;
           End;
     End;
     Clrscr;
     Gotoxy (2,2); Write(‘Nama   Nobp       Rata-rata  Indeks’);
     Gotoxy (2,3); Write(‘~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~’);
     {                    234567890123456789012345678901234567890
     {                              1            2        3   }
    For i := 1 To n Do
    Begin
         Clrscr;
         Gotoxy (2,  3+i); Write (nama [i]);
         Gotoxy (11, 3+i); Write (nobp [i]);
         Gotoxy (22, 3+i); write (rata [i]);
         Gotoxy (33, 3+i); write (indeks [i]);
    End;
```

```
    Gotoxy (2,4+i); Write('~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~');
End.
```

Data Array juga dapat meampung data lebih dari satu indek. Indek ini disebut dengan dimensi. Berikut ini adalah latihan penggunaan Aray lebih dari 1 dimensi.

Contoh 45.

```
Program tipe_array;
Uses Crt;
var
     nilai : Array [0..1,0..2] Of Integer;
Begin
     Clrscr;

     nilai [0,0] := 1;
     nilai [0,1] := 2;
     nilai [0,2] := 3;
     nilai [1,0] := 4;
     nilai [1,1] := 5;
     nilai [1,2] := 6;

     Writeln ('nilai 0,0: ',nilai [0,0]);
     Writeln ('nilai 0,1: ',nilai [0,1]);
     Writeln ('nilai 0,2: ',nilai [0,2]);
     Writeln ('nilai 1,0: ',nilai [1,0]);
     Writeln ('nilai 1,1: ',nilai [1,1]);
     Writeln ('nilai 1,2: ',nilai [1,2]);

     Readln;
End.
```

Contoh 46.

```
Program tipe_array;
Uses Crt;
var
     nilai : Array [0..1,0..2] Of Integer;
Begin
     Clrscr;

     nilai [0,0] := 1;
     nilai [0,1] := 2;
     nilai [0,2] := 3;
     nilai [1,0] := 4;
     nilai [1,1] := 5;
     nilai [1,2] := 6;

     Write ('nilai 0,0: ',nilai [0,0]);
     Write ('nilai 0,1: ',nilai [0,1]);
     Writeln ('nilai 0,2: ',nilai [0,2]);
     Write ('nilai 1,0: ',nilai [1,0]);
     Write ('nilai 1,1: ',nilai [1,1]);
     Writeln ('nilai 1,2: ',nilai [1,2]);

     Readln;
End.
```

Contoh 47.

```
Program tipe_array;
Uses Crt;
Var
     nilai : Array [0..1,0..2,0..3] Of Integer;
Begin
     Clrscr;

     nilai [0,0,2] := 2;
     Writeln (nilai[0,0,2]);

     nilai [1,2,3] := 999;
     Writeln (nilai[1,2,3]);

     Readln;
End.
```

Contoh 48.

```
Program magicSquare;
Uses Crt;
Label mulai, kosong, isi, hasil, selesai;
Var
      kotak     : Array [1..25, 1..25] Of Integer;
      b,k,n,i   : Integer;
Begin
     Clrscr;
     Writeln ('Program Kotak Ajaib (Magic Square');
     Writeln ('--------------------------------');
     mulai:
           Write ('Input banyak baris/kolom kotak (harus ganji) : ');
           Readln (n);
           if (n mod 2 = 0) Then
              Goto mulai;
           Goto kosong;
     kosong:
            For b := 1 To n Do
            Begin
                 For  k := 1 To n Do
                 Begin
                      kotak [b, k] := 0;
                 end;
            end;
            Goto isi;
     isi:
         b := 1;
         k := n div 2 + 1;
         For i := 1 To n*n Do
         Begin
             kotak [b, k] := i;
             b := b - 1;
             if (b < 1) Then
                b := b + n;
             k := k + 1;
             If (k > n) Then
                k := k - n;
             If (kotak [b, k] <> 0) Then
             Begin
                  b := b + 2;
                  If (b > n) Then
                     b := b -n;
                  k := k - 1;
                  If (k < 1) Then
                     k := k + n;
             end;
         End;
     hasil:
           Writeln;
           For b :=  1 to n Do
           Begin
                For k := 1 To n Do
                     Write (kotak [b, k]:5);
                Writeln;
           End;
     selesai:
```

```
     Readln;
End.
```

Contoh 49.

```
Program tipe_array;
Uses Crt;
Type
     nama_hari  = (senin,selasa,rabu,kamis,jumat,sabtu,minggu);
     usia       = 0..99;
Var
     hari: Array [0..9] Of nama_hari;
     umur: Array [0..9] Of usia;
Begin
     Clrscr;

     hari [1] := senin;
     hari [9] := sabtu;

     umur [3] := 17;
     umur [8] := 80;

     Writeln ('hari [1]: ', hari[1]);
     Writeln ('hari [9]: ', hari[9]);
     Writeln ('umur [3]: ', umur[3]);
     Writeln ('umur [8]: ', umur[8]);

     Readln;
End.
```

Contoh 50.

```
Program exArrays;
Var
   N  : Array [1..10] Of Integer;   (* n is an array of 10 integers *)
   i, j    : Integer;

Begin
   (* initialize elements of array n to 0 *)
   For i := 1 To 10 Do
       n[ i ] := i + 100;   (* set element at location i to i + 100 *)
    (* output each array element's value *)

   For j := 1 To 10 Do
      Writeln ('Element[', j, '] = ', n[j] );
End.
```

Latihan 50.

```
DATA TAGIHAN REKENING LISTRIK
============================
Periode Tagihan       : -
Propinsi              : -
Kabupaten / Kota      : -
-----------------------------------------------------
Nama Pelanggan        : -
Alamat Pelanggan      : -
Pemakaian (kWh)       : -
Ada data lagi [y/t]        : -
-----------------------------------------------------
Jumlah Pemakaian (kWh)     : -
-----------------------------------------------------
Nama Pemakaian Tertinggi   : -
Alamat Pemakaian Tertinggi : -
Pemakaian (kWh) Tertinggi  : -
-----------------------------------------------------
```

Proses :

- Jika ada data lagi = y, maka ulangi mengisi data Nama Pelanggan, Alamat Pelangan, Pemakaian(ARRAY)
- Cari Jumlah pemakaian
- Cari Nama, Alamat dan Pemakaian Tertinggi

Latihan 51.

```
DATA PENJUALAN TICKET PESAWAT
============================
Nama Maskapai Penerbangan  : -
Nomor Flight                    : -
Bendara Berangkat               : -
Bendara Tujuan             : -
-----------------------------------------------------------
Kode Tiket                 : -
Nama Penumpang             : -
Harga Tiket (Rp)           : -
Ada data lagi [y/t]             : -
-----------------------------------------------------------
Jumlah  Harga (Rp)              : -
-----------------------------------------------------------
Kode Tiket termurah        : -
Nama Penumpang             : -
Harga Tiket (Rp)           : -
-----------------------------------------------------------
```

Proses :

- Jika ada data lagi = y, maka ulangi mengisi Kode Tiket, Nama Penumpang dan Harga Tiket(ARRAY)
- Cari Jumlah Harga Tiket keseluruhan
- Cari Kode Tiket, Nama Penumpang dan Harga Tiket termurah

Latihan 52.

```
DATA PENJUALAN BARANG
=============================
Nama Toko        : -
Pemilik Toko          : -
Alamat Toko           : -
-------------------------------------------------
Nama Barang           : -
Satuan Barang    : -
Harga Satuan (Rp)     : -
Ada data lagi [y/t]   : -
-------------------------------------------------
Jumlah Harga keseluruhan (rp)    : -
Rata-rata harga satuan (rp)      : -
-------------------------------------------------
Nama Barang Termahal       : -
Satuan Barang Termahal          : -
Harga Satuan Termahal (Rp) : -
-------------------------------------------------
```

Proses :

- Jika Ada data lagi = y, maka ulangi mengisi Nama Barnag, Satuan Barang dan Harga Satuan(ARRAY)
- Cari Jumlah Harga Keseluruhan dan Rata-rata harga satuan
- Cari Nama Barang, Satuan Barang dan Harga Satuan Termahal

Latihan 53.

```
Biaya Layanan RUMAH SAKIT
================================
     Nama Rumah Sakit: -
Alamat Rumah Sakit: -
-------------------------------------------------------------
     Nama Pasien           : -
Jenis Penyakit  : -
Lama inap (hari): -
Biaya Tagihan (Rp)    : -
Ada data lagi [y/t]   : -
-------------------------------------------------------------
Jumlah Biaya Tagihan (Rp)        : -
Rata-rata Biaya Tagihan (Rp)     : -
-------------------------------------------------------------
Nama Pasien Biaya Tagihan Terbesar    : -
Biaya Tagihan                    : -
-------------------------------------------------------------
```

Proses :

- Jika Ada data lagi = y, maka ulangi input data Nama pasien, Jenis Penyakit, Lama Inap dan Biaya Tagihan (ARRAY)
- Cari Jumlah Biaya Tagihan dan Rata-rata Biaya Tagihan
- Cari Nama Pasien dengan Biaya Tagihan Terbesar dan besar Biaya Tagihannya.

ARRAY terhadap RECORD

Ketentuan:

- Record merupakan type data terstruktur yang harus dideklarasikan sendiri
- Record terdiri dari beberapa field dimana setiap field harus dinyatakan namanya dan type dari field tersebut
- Deklarasi dari record dapat dinyatakan pada bagian blok TYPE dan pada blok VAR
- Untuk melakukan pengolahan terhadap field dapat dilakukan dengan 2 cara:
  - perintah titik
  - statement WITH

- Dibawah ini adalah contoh deklarasi record:

```
TYPE   mahasiswa  =  RECORD
           Nama  :  string [15] ;
           BP    :  string [8] ;
           Umur  :  integer ;
       END;

VAR  mhs  :  mahasiswa;        { hanya untuk 1 orang mahasiswa }
```

Membaca field dengan menggunakan perintah titik:

```
Write ('Nama     : '); readln (mhs.nama);
Write ('Nobp     : '); readln (mhs.nobp);
Write ('Umur     : '); readln (mhs.umur);
```

Dapat juga menggunakan statement WITH:

```
  WITH mhs DO
  BEGIN
     Write ('Nama     : '); readln (nama);
     Write ('Nobp     : '); readln (nobp);
     Write ('Umur     : '); readln (umur);
  END ;
```

Untuk menampung data Record juga dapat menggunakan Array, seperti:
Contoh 51.

```
Program arrayRecord;
Uses Crt;
TYPE   mahasiswa  =  RECORD
           Nama  :  string [15] ;
           BP    :  string [8] ;
           Umur  :  integer ;
END;

VAR
```

```
     mhs  :  array [1..5] of mahasiswa;
          i       :  byte;
BEGIN
     For  i := 1 to 5 do
        BEGIN
        WITH mhs[ i ]   DO
        BEGIN
           Write ('Nama    : '); readln (nama);
           Write ('Nobp    : '); readln (nobp);
           Write ('Umur    : '); readln (umur);
        END ;

        For  i := 1 to 5 do
        BEGIN
           WITH mhs[ i ]   DO
           BEGIN
                Write ('Nama    : '); writeln (nama);
                Write ('Nobp    : '); writeln (nobp);
                Write ('Umur    : '); writeln (umur);
           END;
        END;
END.
```

VIII. Function dan Procedure

Procedure dan Function adalah suatu program terpisah dalam blok sendiri yang berfungsi sebagai sub-program (modul program) yang merupakan sebuah program kecil untuk memproses sebagian dari pekerjaan program utama.

Prosedur diawali dengan kata cadangan Procedure di dalam bagian deklarasi prosedure. Prosedure dipanggil dan digunakan di dalam blok program yang lainnya dengan menyebutkan judul prosedurnya.

Prosedur banyak digunakan pada program yang terstruktur, karena :

- Merupakan penerapan konsep program modular, yaitu memecah-mecah program yang rumit menjadi program-program bagian yang lebih sederhana dalam bentuk prosedur-prosedur.
- Untuk hal-hal yang sering dilakukan berulang-ulang, cukup dituliskan sekali saja dalam prosedur dan dapat dipanggil atau dipergunakan sewaktu-waktu bila diperlukan.

Sebagaimana halnya sebuah program, suatu procedure juga memiliki header dan block. Perbedaan bentuknya dengan program hanyalah pada bagian header-nya saja. Bentuk Umum header suatu procedure adalah :

PROCEDURE nama;

Atau

PROCEDURE nama (formal parameter : jenis);

Jika kita menggunakan procedure dalam suatu program, maka procedure tersebut harus dituliskan pada bagian deklarasi.

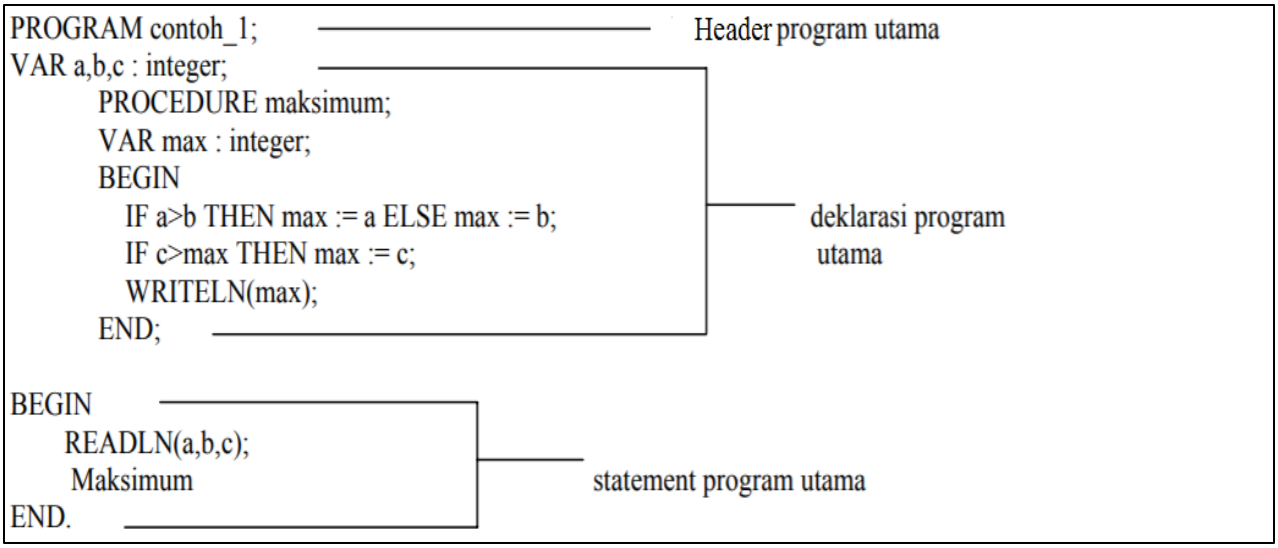
```
Procedure nama_procedure (parameter);
Deklarasi variabel,label,…dll;
Begin
      Statement procedure;
      …….
      …….
End;
```

Contoh untuk mebuat suatu procedure untuk menentukan bilangan bulat terbesar diantara tiga bilangan bulat, maka procedure tersebut adalah sebagai berikut :

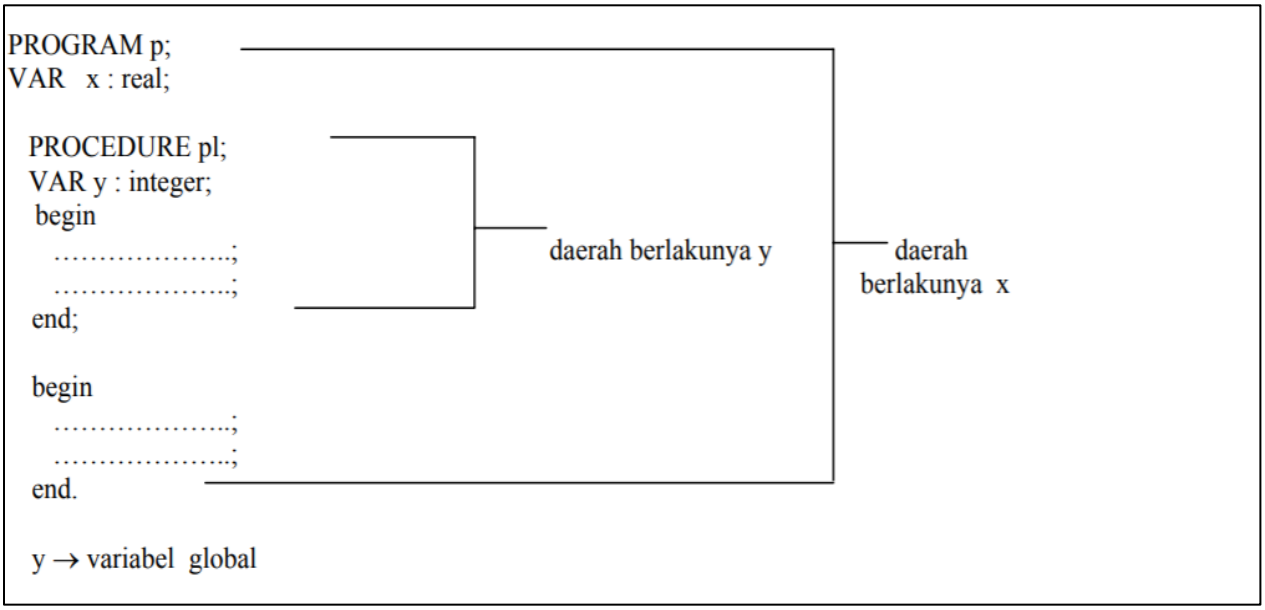
Contoh 52.

```
PROCEDURE maksimum;
VAR max : integer;
BEGIN
      IF a > b THEN max := a ELSE max := b;
      IF c > max THEN max := c;
      WRITELN(max);
END.
```

Keterangan:



Jangkauan identifier yang dideklarasikan dalam suatu blok program hanya berlaku pada blok dimana identifier tersebut didefinisikan.



```
Program P ;
Var x,y : real;  ──────────────────────  y (real) berlaku disemua blok
……                                      program P, kecuali di P1
 ……                                     x (real) berlaku disemua
                                        blok P, kecuali di P2
Procedure Pl ;
Var y : integer;  ─────────────────────  hanya berlaku di P1 saja (lokal)
……..
Begin
    ……
    ……
End;

Procedure P2;
Var x : char;  ────────────────────────  hanya berlaku di P2 saja (lokal)
……
Begin
    ……..
    ……..
End;

Begin
    ……
    x := …….  ┐
    y := …….  ┘──  x dan y yang dimaksud adalah x dan y real
    ……              (variabel global)
End.
```

Nilai parameter dalam suatu modul program Pascal sifatnya adalah lokal, artinya hanya dapat digunakan pada modul atau unit program y.

Contoh 53.

```
Prosedur Tanya_hitung;
Var X,Y :real;
Begin
        Write ('Nilai X ?');
        Readln(X);
        Y:=X*X;
        Writeln('Nilai Y = ',Y:6:2);
End;

Begin
        Tanya_Hitung;
End.
```

Keterangan : Variabel X dan Y sifatnya adalah lokal untuk prosedur Tanya_hitung, artinya hanya dapat digunakan pada modul itu saja, Pada modul yang lain tidak dapat digunakan. Kerjakan Latihan 53.

Contoh 54.

```
Prosedur Tanya_hitung;
Var X,Y :real;
Begin
        Write ('Nilai X ?');
        Readln(X);
        Y:=X*X;
End;

Begin
        Tanya_Hitung;
        Writeln('Nilai Y = ',Y:6:2);
End.
```

Agar nilai variabel dapat digunakan di modul lainnya, maka dapat dilakukan dengan cara dideklarasikan secara global, yaitu dideklarasikan di atas modul yang menggunakannya.

Struktur 1.

```
Procedure kesatu;
Begin
        .......
        ........
End; (*akhir dari procedure kesatu……*)

Var
        A,B : word;

Procedure kedua;
Begin
        ........
        ........
End; (*akhir dari procedure kedua*)

Procedure ketiga;
Begin
        ........
        ........
End; (*akhir dari procedure ketiga*)

Begin
        ........
        ........
End. (*akhir dari modul utama*)
```

Pada Struktur 1 diatas, variabel A dab B bersifat global untuk prosedur kedua, ketiga dan utama, tetapi tidak bersifat global untuk prosedur

kesatu, sehingga prosedur kesatu tidak dapat menggunakan variabel-variabel tersebut.

Struktur 2.

```
Var
        A,B : real;

Procedure kesatu;
Begin
        .......
        ........
End; (*akhir dari procedure kesatu……*)

Procedure kedua;
Begin
        ........
        ........
End; (*akhir dari procedure kedua*)

Procedure ketiga;
Begin
        ........
        ........
End; (*akhir dari procedure ketiga*)

Begin
        ........
        ........
End. (*akhir dari modul utama*)
```

Pada Struktur 2 diatas, variabel A dan B bersifat global untuk semua modul.

Parameter yang dikirim dari modul utama ke modul prosedur disebut actual parameter, dan parameter yang ada dan dituliskan pada judul prosedur disebut formal parameter.

Parameter yang dikirimkan secara nilai, maka parameter formal yang ada di prosedur akan berisi nilai yang dikirimkan yang kemudian bersifat lokal di prosedur.

```
Procedure konversi;
begin
  f := (5/9) * c + 32;   ————————  f dan c variabel global, c sebagai input dan
  writeln(c,f);                    f sebagai output.
end;
```

Procedure di atas dapat dipanggil dengan variabel yang berbeda, tetapi penulisannya harus diubah dengan menggunakan parameter sbb:

```
Procedure konversi (var f : real; c : real);
Begin
 F := (5/9) * c + 32;
 Writeln(c,f);
End;
```

disebut formal parameter

Selanjutnya procedure di atas dapat dipanggil dengan parameter lain, misalnya :

```
konversi (x,y) ;
```

actual parameter

x dan y disebut sebagai actual parameter.
Pada eksekusinya x akan menggantikan c dan y akan menggantikan f.
x dan y ini dapat berupa :

- konstanta
- variabel
- procedure, atau
- fungsi

Contoh 55.

```
Procedure Hitung(A,B : integer);
Var C : integer;
Begin
        C := A + B;
        Writeln('Nilai C = ',C)
End;

Var X,Y : integer;
Begin
        Write('Nilai X ? ');
        Readln(X);
        Write('Nilai Y ?');
        Readln(Y);
        Hitung(X,Y);
End.
```

Bila pengiriman parameter secara acuan (by reference), maka perubahan-perubahan yang terjadi pada nilai parameter formal di prosedur akan mempengaruhi nilai actual parameter.

```
Procedure hitung(Var A,B,C : integer);
```

Menunjukkan pengiriman parameter secara acuan.

Contoh 56.

```
Procedure Hitung(Var A,B,C : integer);
Begin
        C := A + B;
End;

Var  X,Y,Z : integer;
Begin
        X := 2;  Y:= 3;
        Hitung(X,Y,Z);
        Writeln('X = ',X,' Y = ',Y,' Z = ',Z);
End.
```

Acuan Forward digunakan untuk mendeklarasikan dimuka judul prosedur terpisah dari bloknya.

Contoh 57.

```
Procedure pro1(var I : integer); Forward;
Procedure pro2(var I : integer);
Begin
        Writeln('prosedur pro', I);
End;

Procedure pro1;
Begin
        Writeln('prosedur pro',I);
End;

Var I : integer;
Begin
        I := 1;  pro1(I);
        I := 2;  pro2(I);
End.
```

Prosedur Standar yang disediakan oleh Turbo Pascal :

- Prosedur standar **EXIT**
  Digunakan untuk keluar dari suatu blok.
- Prosedur standar **HALT**
  Digunakan untuk menghentikan proses program baik di program bagian maupun di program utama.
- Prosedur standar **MOVE**
  Bentuk umum : MOVE (Var source,dest; count : word);
  Digunakan untuk menyalin suatu blok sebanyak count byte memori dari blok dimulai byte pertama source dan disalinkan ke byte pertama dest.
- Prosedur standar **FILLCHAR**
  Digunakan untuk mengisi sejumlah byte nilai ke dalam suatu variabel, sebagai berikut:

FillChar(x;count :word;ch);
X adalah variabel yang dapat bertipe apapun yang akan diisi dengan nilai tipe ordinal Ch sebanyak count byte.

Contoh 58.

```
Contoh program;
Uses Crt;
Procedure hitungluas(p,l:integer);
Var
     Ls    :Integer;
Begin
     Ls := p * l;
     Writeln ('Luas: ', ls);
End;

Procedure hitkeliling (pp, ll    : Integer);
Var
     Kel   : Integer;
Begin
     Kel := (2 * pp) + (2 * ll);
     Writeln ('Keliling: ', kel);
End;

Var
     Pj,lbr : Integer;
Begin
     Clrscr;
     Write ('Masukan panjang : '); Readln(pj);
     Write ('Masukan lebar : '); Readln(lbr);
     Writeln;
     Hitungluas (pj, lbr);
     hitkeliling(pj, lbr);
     Readln;
End.
```

Function hampir sama dengan blok pada procedure, hanya pada function harus dideklarasikan dengan tipe dari function tersebut yang merupakan tipe hasil dari function itu sendiri. Sehingga dikatakan function dapat mengembalikan nilai.
Perbedaan fungsi dengan prosedur adalah:

- Pada fungsi, nilai yang dikirimkan balik terdapat pada nama fungsinya(kalau pada prosedur parameter yang dikirimkan secara acuan).
- Karena nilai balik berada di nama fungsi tersebut, maka fungsi tersebut dapat langsung digunakan untuk dicetak hasilnya. Atau nilai fungsi tersebut dapat juga langsung dipindahkan ke pengenal variabel yang lainnya.
- Pada prosedur, nama prosedur tidak dapat digunakan langsung, yang dapat langsung digunakan adalah parameternya yang mengandung nilai balik.

Hal ini dapat dilihat dari bentuk header-nya yang menyebutkan jenis data dari kuantitas yang dihasilkan.Secara umum bentuk header suatu function adalah :

```
          FUNCTION nama : jenis hasil;
Atau      FUNCTION nama (formal parameter : jenis ) : jenis_hasil;
```

Pada Latihan 58. Adalah sebuah fungsi dengan nama MAX yang dapat menentukan integer terbesar di antara dua integer.

Contoh 59.

```
Function max (x,y : Integer) : Integer;
Begin
     If x < y Then
        max := y ;
     Else
        max := x;
End;
```

Selanjutnya kita dapat menggunakan fungsi di atas dalam suatu program, misalnya dengan menyatakan sebagai berikut :

```
p := max(a,b);
z := max(a+b,a*b);
q := max(MAX(a,b),c);
……
dsb.
```

Contoh 60.

```
Function log (x : Real) : Real;
Begin
     log := Ln (x) / Ln (10.0);
End;
```

Contoh 61.

```
Function power (x,y : Real) : Real;
Begin
 power := Exp (y * Ln (X))
End;
```

Contoh 62.

```
Function hitung(Var a, b : Integer): Integer;
Begin
     hitung := a + b;
End;
Var x, y : Integer;
Begin
     Write ('Nilai X ? ');
     Readln (X);
     Write ('Nilai Y ? ');
     Readln (y);
     Writeln;
     Writeln (X,' + ',Y,' = ',Hitung(x,y));
End.
```

Perbedaan fungsi dengan prosedur adalah:

1. Pada fungsi, nilai yang dikirimkan balik terdapat pada nama fungsinya (kalau pada prosedur pada parameter yang dikirimkan secara acuan). Pada contoh, nama fungsi tersebut adalah Hitung dan nilai yang dikirim balik berada pada nama fungsi tersebut. Sehingga nama fungsi ini harus digunakan untuk menampung hasil yang akan dikirimkan dari fungsi, sebagai berikut :

```
Hitung := A + B;
   └──────────── Nama fungsi yang berisi nilai yang akan dikirimkan
```

2. Karena nilai balik berada di nama fungsi tersebut, maka fungsi tersebut dapat langsung digunakan untuk dicetak hasilnya, sebagai berikut :

```
Writeln(X,' + ',Y,' = ',Hitung(X,Y));
                        └──────── Nama fungsi yang langsung digunakan
                                  untuk ditampilkan hasilnya.
```

Atau nilai fungsi tersebut yang dapat juga dipindahkan langsung ke pengenal variabel yang lainnya, seperti berikut :

```
     Hasil := Hitung(X,Y);
     Writeln(X,' + ',Y, ' + ',Hasil)
```

Sedangkan pada prosedur, nama prosedur tersebut tidak dapat digunakan langsung, yang dapat langsung digunakan adalah parameternya yang mengandung nilai balik.

Contoh 63.

```
Program contoh;
Uses Crt;
Function faktor(bilangan : Integer) : Real;
Begin
     If (bilangan = 0) Then
          faktor := 1
     Else
          faktor := faktor (bilangan - 1) * bilangan;
End;

Var
     n    : Integer;
Begin
     Clrscr;
     Write (‘Berapa Faktorial Dari = ‘); Readln(n);
     Writeln (n,’ faktorial = ‘, faktor (n) :9:0);
     Readln;
End.
```

Contoh 64.

```
Program hitung_lingkaran;

Uses Crt;

Type
     warna=(merah, kuning, biru);

Const
     pi    = 3.14;

Var
     jari2 : integer=7;
     warna_lingkaran  : warna=merah;

Function luas_lingkaran    : real;
Begin
   luas_lingkaran := pi * jari2 * jari2;
End;

Procedure kel_lingkaran (jari2 : integer);
Begin
     Write ('Keliling Lingkaran = ');
     Writeln(PI*(jari2+jari2):4:2, ' cm');
End;

Begin
     Clrscr;
     Writeln ('PROGRAM MENGHITUNG LUAS LINGKARAN');
     Writeln ('--------------------------------');
     Writeln ('Diketahui: jari-jari lingkaran = ', jari2,' cm');
     Writeln ('Warna Lingkaran = ', warna_lingkaran);
     Writeln ('Luas Lingkaran = ', luas_lingkaran:4:2, ' cm');
     kel_lingkaran (jari2);
     Readln;
End.
```

Contoh 65.

```
Program exFunction;
Var
   a, b, ret    : Integer;

(*function definition *)
Function max (num1, num2: integer)    : Integer;
Var
   (* local variable declaration *)
   Result  : Integer;

Begin
   If (num1 > num2) Then
      result := num1

   Else
      result := num2;
   max := result;
End;

Begin
   a := 100;
   b := 200;
   (* calling a function to get max value *)
   ret := max(a, b);

   Writeln( 'Max value is : ', ret );
End.
```

Contoh 66.

```
Program exProcedure;
Var
   a, b, c,  min: Integer;

Procedure findMin (x, y, z  : Integer; Var m: Integer);
(* Finds the minimum of the 3 values *)

Begin
   If x < y Then
      m:= x
   Else
      m:= y;

   If z < m Then
      m:= z;
End; { end of procedure findMin }

Begin
   Writeln (' Enter three numbers: ');
   Readln ( a, b, c);
   findMin (a, b, c, min); (* Procedure call *)

   Writeln (' Minimum: ', min);
End.
```

Rekursif adalah suatu fungsi atau prosedur tersebut dapat memanggil dirinya sendiri. Berikut ini sebuah contoh fungsi dan prosedur yang rekursif.

Contoh 67.

```
Program exRecursion;
Var
   num, f  : Integer;
Function fact (x : Integer): Integer;
(* calculates factorial of x - x! *)

Begin
     If x=0 Then
          fact := 1
     Else
          fact := x * fact(x-1);
(* recursive call *)
End;
{ end of function fact}

Begin
   Writeln (' Enter a number: ');
   Readln (num);
   f := fact (num);

   Writeln (' Factorial ', num, ' is: ' , f);
End.
```

Contoh 68.

```
Program recursiveFibonacci;
Var
   I  : Integer;
Function fibonacci(n  : Integer) : Integer;

Begin
   If n = 1 Then
      fibonacci := 0

   Else
      If n=2 Then
      fibonacci := 1

   Else
      fibonacci := fibonacci(n-1) + fibonacci(n-2);
End;

Begin
   For i:= 1 To 10 Do

   Write (fibonacci (i), '  ');
End.
```

IX. Latihan Program Algoritma (Pemantapan)

Contoh 69.

```
PROGRAM binary_search;

{Program to search a number using binary search}

USES crt;

TYPE index=1..100;
VAR  arr:ARRAY[1..100] OF index;
VAR  mid,low,high,search:integer;
     i,n:index;
     found:boolean;

BEGIN
     clrscr;
     writeln('BINARY SEARCH');
     writeln('Enter the array size');
     readln(n);
     writeln('Enter the array elements');
     FOR i:=1 TO n DO
     BEGIN
          readln(arr[i]);
     END;
     writeln('Enter the search element');
     readln(search);
     low:=1;
     high:=n;
     found:=false;
     REPEAT
           mid:=trunc(low+high) DIV 2;
           IF (search<arr[mid]) THEN
           high:=mid-1;
           IF (search>arr[mid]) THEN
           low:=mid+1;
           IF (search=arr[mid]) THEN
           found:=true
           ELSE
           found:=false;
     UNTIL ((found=true) OR (high<low));
     IF found=true THEN writeln('ELEMENT FOUND')
     ELSE writeln('ELEMENT NOT FOUND');
END.
```

Contoh 70.

```
PROGRAM linear_search;

{Program to search a number using linear search}

USES crt;

TYPE index=1..100;

VAR n,searchkey,i:integer;
    found:boolean;
    arr:ARRAY[1..100] OF index;

BEGIN
     writeln('LINEAR SEARCH');
     writeln('Enter the boundary of the array');
     readln(n);
     writeln('Enter the array elements');
     FOR i:=1 TO n DO
     BEGIN
          readln(arr[i]);
     END;
     i:=1;
     found:=false;
     writeln('Enter the search element');
     readln(searchkey);
     WHILE ((i<=n) AND (found=false)) DO
     BEGIN
          IF arr[i]=searchkey THEN
          found:=true
          ELSE found:=false;
          i:=i+1;
     END;
     IF found=true THEN
     writeln('ELEMENT FOUND')
     ELSE
     writeln('ELEMENT NOT FOUND');
END.
```

Contoh 71.

```
PROGRAM bubble_sort;

CONST items=100;

VAR n,temp,pass,index:integer;
    sorted:boolean;
    vector:ARRAY[1..items] of integer;

PROCEDURE sort;

BEGIN
     pass:=1;
     REPEAT
           sorted:=true;
           FOR index:=1 TO items-pass DO
           BEGIN
                IF vector[index]>vector[index+1] THEN
                BEGIN
                     sorted:=false;
                     temp:=vector[index];
                     vector[index]:=vector[index+1];
                     vector[index+1]:=temp;
                END;
           END;
           pass:=pass+1;
     UNTIL sorted;
END;

BEGIN
     writeln('How many elements');
     readln(n);
     writeln('Enter the unsorted elements');
     index:=1;
     REPEAT
     readln(vector[index]);
     index:=index+1;
     UNTIL index=n+1;
     sort;
     writeln('Sorted Data');
     FOR index:=1 TO items DO
     BEGIN
          IF ((index-1) MOD 10)=0 THEN writeln;
          writeln(vector[index]:4);
     END;
     writeln('Total number of passes=> ',pass);
     writeln;
END.
```

Contoh 72.

```
PROGRAM insertion_sort;

{Program to sort the given nos using insertion sort}

USES crt;

VAR a:ARRAY[1..100] of real;

VAR temp:real;
    i,j,n:integer;

BEGIN
     clrscr;
     writeln('Enter the boundary of the array');
     readln(n);
     writeln('Enter the elements of the array');
     FOR i:=1 TO n DO
     BEGIN
          readln(a[i]);
     END;
     FOR i:=2 TO n DO
     BEGIN
          j:=i-1;
          WHILE ((j>=1) AND (a[j+1]<a[j])) DO
          BEGIN
               temp:=a[j];
               a[j]:=a[j+1];
               a[j+1]:=temp;
               j:=j-1;
          END;
     END;
     writeln('The sorted elements are as follows');
     FOR i:=1 TO n DO
     writeln(a[i]);
END.
```

Contoh 73.

```
Program QSort;
{$R-,S-}
uses Crt;

{ This program demonstrates the quicksort algorithm, which      }
{ provides an extremely efficient method of sorting arrays in   }
{ memory. The program generates a list of 1000 random numbers   }
{ between 0 and 29999, and then sorts them using the QUICKSORT  }
{ procedure. Finally, the sorted list is output on the screen.  }
{ Note that stack and range checks are turned off (through the  }
{ compiler directive above) to optimize execution speed.        }

const
  Max = 1000;

type
  List = array[1..Max] of Integer;

var
  Data: List;
  I: Integer;

{ QUICKSORT sorts elements in the array A with indices between  }
{ LO and HI (both inclusive). Note that the QUICKSORT proce-    }
{ dure provides only an "interface" to the program. The actual  }
{ processing takes place in the SORT procedure, which executes  }
{ itself recursively.                                           }

procedure QuickSort(var A: List; Lo, Hi: Integer);

procedure Sort(l, r: Integer);
var
  i, j, x, y: integer;
begin
  i := l; j := r; x := a[(l+r) DIV 2];
  repeat
    while a[i] < x do i := i + 1;
    while x < a[j] do j := j - 1;
    if i <= j then
    begin
      y := a[i]; a[i] := a[j]; a[j] := y;
      i := i + 1; j := j - 1;
    end;
  until i > j;
  if l < j then Sort(l, j);
  if i < r then Sort(i, r);
end;

begin {QuickSort};
  Sort(Lo,Hi);
end;

begin {QSort}
  Write('Now generating 1000 random numbers...');
  Randomize;
  for i := 1 to Max do Data[i] := Random(30000);
  Writeln;
```

```
  Write('Now sorting random numbers...');
  QuickSort(Data, 1, Max);
  Writeln;
  for i := 1 to 1000 do Write(Data[i]:8);
end.
```

Contoh 74.

```
PROGRAM backwards;

{This program reads a line of text and writes it out in a reverse
order}

USES crt;

PROCEDURE flipit;

{Reads single characters recursively and then writes them out}

VAR c:char;

{The procedure flipit is the key thing in this program.First it reads
a single character and then makes sure (checks) that it is not an end
of line,and if this condition satisfies then it once again goes to
the
procedure flipit and reads the next single character.This process
continues
until the end of line is detected,after which the computer writes out
the
output in the order of the most recent character written first (i.e.,
the
character where the end of line was encountered) and the first
character
written last.Hence we get a line of text written in a reverse order in
the
output.}

BEGIN
     read(c);
     IF NOT eoln THEN flipit;
     write(c)
END;

BEGIN
     clrscr;
     writeln('Enter a line of text');
     writeln;
     flipit;
END.
```

Contoh 75.

```
 PROGRAM factorial;

{Program to calculate the factorial of a number using recursive
function}

VAR x:integer;

FUNCTION fact(n:integer):integer;

BEGIN
     IF n<=1 THEN fact:=1
     ELSE fact:=n*fact(n-1);
END;

BEGIN
     writeln('Enter any integer');
     read(x);
     writeln('The factorial is ',fact(x))
END.
```

Contoh 76.

```
PROGRAM fibonacci_series;

(*Program to find the fibonacci series upto a given number*)

VAR a,b,j,n:integer;

PROCEDURE fib(a,b,j:integer);

BEGIN
     IF j>0 THEN
     BEGIN
          WHILE j<>a DO
          BEGIN
          writeln(a:1,' ');
          fib(b,a+b,j-1);
          END;
     END;
END;
BEGIN
     writeln('FIBONACCI SERIES');
     writeln;
     writeln('Enter any number');
     readln(n);
     writeln;
     IF n<=0 THEN writeln('Invalid Entry,please try again!')
     ELSE
     fib(0,1,n);
END.
```

Contoh 77.

```
PROGRAM gcd_recursion;

{Program to calculate the GCD of 2 nos using recursive function}

USES crt;

VAR a,b:integer;

FUNCTION gcd(p,q:integer):integer;

BEGIN
     IF p<q THEN
        BEGIN
             gcd:=gcd(q,p);
        END
     ELSE
        IF q=0 THEN
           BEGIN
                gcd:=p;
           END
        ELSE
            gcd:=gcd(q,p MOD q);
END;

BEGIN
     clrscr;
     writeln('Enter any two elements');
     readln(a,b);
     gcd(a,b);
     writeln('The gcd of two numbers is ',gcd(a,b));
END.
```

Contoh 78.

```
PROGRAM fact1;
{Factorial of a number}
USES crt;
VAR n:integer;
FUNCTION fact(i:integer):integer;
VAR prod1:integer;
BEGIN
     BEGIN
          prod1:=1;
          REPEAT
          prod1:=prod1*i;
          i:=i-1;
          UNTIL i = 1;
     END;
writeln('The factorial of ',n,' is ',prod1)
END;
BEGIN
     clrscr;
     writeln('Enter any number');
     read(n);
     fact(n);
END.
```

Contoh 79.

```
PROGRAM gcd;

(*program to find the gcd of two numbers*)

USES crt;

VAR a,b,c,d,i:integer;

BEGIN
     clrscr;
     writeln('Enter any two integers');
     readln(a,b);
     IF a<=b THEN c:=a;
     c:=b;
     FOR i:=1 TO c DO

         BEGIN

              IF (a MOD i=0)AND(b MOD i=0) THEN
              d:=i;
         END;

     writeln('The GCD of two numbers is ',d);
END.
```

Contoh 80.

```
PROGRAM file_create;

TYPE student=RECORD
                  name:string[20];
                  rollno,marks:integer;
             END;
VAR n,i:integer;
    data:student;
    file1:FILE OF student;

BEGIN
     writeln('Program to create a sequential file of student data');
     assign(file1,'file1.dat');
     rewrite(file1);
     REPEAT
           write('Enter the number of students:');
           readln(n);
     UNTIL n>0;
     FOR i:=1 TO n DO
         WITH data DO
         BEGIN
              write('NAME : ');
              readln(name);
              write('ROLL NO : ');
              readln(rollno);
              write('MARKS : ');
              readln(marks);
              write(file1,data);
         END;
         reset(file1);
         writeln('The data file contains the following information:
');
         writeln('NAME':15,'':12,'ROLL NO.':8,'MARKS':10);
         WHILE (NOT eof(file1)) DO
         BEGIN
              read(file1,data);
              WITH data DO
                   writeln(name:20,rollno:12,marks:12);
         END;
END.
```

Contoh 81.

```
PROGRAM function_procedure_parameter;

VAR x:real;

          FUNCTION f1(a:real):real;

          BEGIN
                f1:=sqr(a);
          END;

    PROCEDURE p(x:real);
                                    {A function is declared within a
Procedure}
          FUNCTION f(x:real):real;

          VAR y:real;

          BEGIN
               y:=f(x);
               writeln('The output is...');
               writeln(y);
          END;

BEGIN             {main program statements}
     x:=9;
     p(x,f1);
END.
```

Contoh 82.

```
PROGRAM towers_of_hanoi;

{This program solves a well known game using recursive procedures
calls
and user defined data}

TYPE poles=(left,centre,right);
     disks=0..maxint;

VAR n:disks;

    PROCEDURE transfer(n:disks;origin,destination,other:poles);

    {Note that origin,destination and other are formal parameters for
the
    procedure transfer,they are supposed to be replaced by the actual
parameter
    left,centre,right in the procedure reference in the main program}

    {Transfer n disks from the origin to the destination}

              PROCEDURE diskmove(origin,destination:poles);

              {Move a single disk from the origin to the destination}
              BEGIN
                   write('Move ');
                   CASE origin OF
                        left    :IF destination=centre
                                 THEN writeln('left to centre')
                                 ELSE writeln('left to right');
                        centre  :IF destination=left
                                 THEN writeln('centre to left')
                                 ELSE writeln('centre to right');
                        right   :IF destination=centre
                                 THEN writeln('right to centre')
                                 ELSE writeln('right to left');
                   END;          {End case}
              END;               {End diskmove}
    BEGIN     {Transfer}
              IF n>0 THEN BEGIN
                          transfer(n-1,origin,other,destination);
                          diskmove(origin,destination);
                          transfer(n-1,other,destination,origin);
                          END;
    END;                  {End Transfer}

BEGIN                     {Main action block}
     write('Enter the number of disks->');
     readln(n);
     writeln;
     transfer(n,left,right,centre);  {Transfer n disk from left to
right}
END.
```

Contoh 83.

```
PROGRAM matrix1;

{Program that declares a integer matrix and initializes it to 1's
on the diagonal and 0's elsewhere}

VAR arr:ARRAY[1..100,1..100] OF integer;
    i,j,index,m,n:integer;

BEGIN
     writeln;
     writeln('Enter the number of rows and columns');
     readln(m,n);
     FOR i:=1 TO m DO
     BEGIN
          FOR j:=1 TO n DO
          BEGIN
               IF i=j THEN
               arr[i,j]:=1
               ELSE
               arr[i,j]:=0;
               IF ((j-1) MOD n)=0 THEN writeln;
               write(' ',arr[i,j],' ');
          END;
     END;

END.
```

Contoh 84.

```
PROGRAM power;

{Program that will allow an integer type number to be raised to an
integer type power}

USES crt;

VAR x,y:integer;

PROCEDURE pow(a,b:integer);

{Procedure to calculate x^y}

VAR count,expo:integer;

BEGIN
     count:=1;
     expo:=1;
     FOR count:=1 TO b DO
         BEGIN
              expo:=expo*a;
         END;
         writeln('The answer is ',expo);
END;

BEGIN
     clrscr;
     writeln('Program to calculate "x to the power of y" ');
     writeln;
     writeln('Enter any two numbers x & y');
     readln(x,y);
     pow(x,y);
END.
```

Contoh 85.

```
PROGRAM prime_check;

VAR n,i,s:integer;
    flag:boolean;

BEGIN
     writeln('Enter any number');
     readln(n);
     flag:=false;
     s:=trunc(sqrt(n*1.0));
     FOR i:=2 TO s DO
     IF ((n MOD i)=0) THEN
     flag:=false
     ELSE
     flag:=true;
     IF flag=true THEN writeln('It is a Prime number')
     ELSE
     writeln('Not a prime number');
END.
```

Contoh 86.

```
PROGRAM prime_generation;

VAR i,j,k,n:integer;

BEGIN
     writeln('Enter any number');
     readln(n);
     FOR i:=2 TO n DO
     BEGIN
          k:=0;
          FOR j:=1 TO n DO
          IF i MOD j =0 THEN k:=k+1;
          IF k<=2 THEN writeln(i);
     END;
END.
```

Contoh 87.

```
PROGRAM graph;

CONST scale=30;
      centre=40;
      increment=15;
      PI=3.14159;

VAR i,angle:integer;

    FUNCTION sinetrace:integer;
    {Evaluate position on the screen for plotting sine wave}
    BEGIN
         sinetrace:=trunc(centre-scale*sin(angle*PI/100));
    END;

BEGIN
     angle:=0;
     WHILE angle>=0 DO
     BEGIN
          FOR i:=1 TO sinetrace DO
               write(' ');
               writeln('sine');
               angle:=angle+increment;
     END;
END.
```

Contoh 88.

```
PROGRAM vowels;

USES crt;

{Program that counts the number of vowels in a sentence}

CONST space=' ';
      maxchar=80;

TYPE vowel=(a,e,i,o,u);

VAR buffer:ARRAY[1..maxchar] of char;
    vowelcount:ARRAY[vowel] of integer;

PROCEDURE initialize;

VAR ch:vowel;

BEGIN
     FOR ch:=a TO u DO
     BEGIN
          vowelcount[ch]:=0;
     END;
END;

PROCEDURE textinput;

VAR index:integer;

BEGIN
     writeln('Input a sentence');
     FOR index:=1 TO maxchar DO
         IF eoln THEN buffer[index]:=space
         ELSE read(buffer[index]);
         readln;
END;

PROCEDURE analysis;

VAR index:integer;
    ch:vowel;

BEGIN
     index:=1;
     WHILE index<>maxchar+1 DO
     BEGIN
          IF buffer[index] IN ['a','e','i','o','u'] THEN
          BEGIN
               CASE buffer[index] OF
               'a':ch:=a;
               'e':ch:=e;
               'i':ch:=i;
               'o':ch:=o;
               'u':ch:=u;
               END;
               vowelcount[ch]:=vowelcount[ch]+1;
          END;
```

```
          index:=index+1;
     END;
END;

PROCEDURE vowelout;

VAR ch:vowel;

BEGIN
     clrscr;
     writeln;
     writeln('   a   e   i   o   u');
     FOR ch:=a TO u DO
     write(vowelcount[ch]:4);
     writeln;
END;

BEGIN
     initialize;
     textinput;
     analysis;
     vowelout;
END.
```

Contoh 89.

```
PROGRAM no_of_words;

{Program to count the number of words in a sentence}

USES crt;

CONST space=' ';

VAR nextchar:char;
    words:integer;

BEGIN
     words:=1;
     clrscr;
     writeln('Input sentence-terminate with return');
     WHILE not eoln DO
     BEGIN
          read(nextchar);       {If readln was used instead of
                                 read then the program would not
work}
          IF nextchar=space THEN
          words:=words+1;
     END;
     writeln('Number of words in the sentence => ',words);
END.
```

Contoh 90.

```
PROGRAM makelist;

TYPE link=^personal;
     personal=RECORD
                name:PACKED ARRAY[1..30] OF char;
                next:link
              END;

VAR item,pointer:link;

PROCEDURE readname(VAR newname:link);
{This procedure reads a name into the computer}

VAR count:0..40;

BEGIN
     FOR count:=1 TO 40 DO
     newname^.name[count]:=' ';
     write('New name: ');
     count:=0;
     WHILE NOT eoln DO
     BEGIN
          count:=count+1;
          read(newname^.name[count])
     END;
     readln
END;

BEGIN
     BEGIN
          new(item);
          readname(item);
          item^.next:=NIL;
          pointer:=item;
          WHILE NOT ((item^.name[1] IN ['E','e'])
                 AND (item^.name[2] IN ['N','n'])
                 AND (item^.name[3] IN ['D','d'])) DO
                 BEGIN
                      new(item);
                      readname(item);
                      item^.next:=pointer;
                      pointer:=item;
                 END;
          pointer:=item^.next
     END;
     BEGIN
          writeln;
          WHILE pointer<>NIL DO
          BEGIN
               item:=pointer;
               writeln(item^.name);
               pointer:=item^.next
          END;
     END;
END.
```

Contoh 91.

```
Procedure BubbleSort(numbers : Array of Integer; size : Integer);
Var i, j, temp : Integer;


Begin
 For i := size-1 DownTo 1 do
  For j := 2 to i do
   If (numbers[j-1] > numbers[j]) then
    Begin
     temp := numbers[j-1];
     numbers[j-1] := numbers[j];
     numbers[j] := temp;
    End;

End.
```

Contoh 92.

```
Procedure InsertionSort(numbers : Array of Integer; size : Integer);
Var i, j, index : Integer


Begin
 For i := 2 to size-1 do
  Begin
   index := numbers[i];
   j := i;
   While ((j > 1) AND (numbers[j-1] > index)) do
    Begin
     numbers[j] := numbers[j-1];
     j := j - 1;
    End;
   numbers[j] := index;
  End;

End.
```

Contoh 93.

```
Procedure QSort(numbers : Array of Integer;
                left : Integer; right : Integer);
Var pivot, l_ptr, r_ptr : Integer;

Begin
 l_ptr := left;
 r_ptr := right;
 pivot := numbers[left];
 While (left < right) do
  Begin
   While ((numbers[right] >= pivot) AND (left < right)) do
    right := right - 1;
   If (left <> right) then
    Begin
     numbers[left] := numbers[right];
     left := left + 1;
    End;
   While ((numbers[left] <= pivot) AND (left < right)) do
    left := left + 1;
   If (left <> right) then
    Begin
     numbers[right] := numbers[left];
     right := right - 1;
    End;
  End;
 numbers[left] := pivot;
 pivot := left;
 left := l_ptr;
 right := r_ptr;
 If (left < pivot) then
  QSort(numbers, left, pivot-1);
 If (right > pivot) then
  QSort(numbers, pivot+1, right);
End;

Procedure QuickSort(numbers : Array of Integer; size : Integer);
Begin
 QSort(numbers, 0, size-1);
End;
```


View publication stats