



# **BONUS MATERI PEMROGRAMAN**

## **INFORMATICS OLYMPIAD(I/O) 2021**

## Pengantar

Dalam kehidupan sehari-hari permasalahan selalu muncul kapan saja. Mulai dari yang sederhana hingga yang kompleks. Permasalahan bukan hanya sesuatu yang negatif namun bisa menjadi sesuatu yang positif. Misal saat kita melihat sebuah tumpukan kartu kemudian ingin mengurutkannya, hal tersebut memberikan permasalahan kepada kita yaitu cara mengurutkan kartu tersebut.

Sebenarnya banyak hal yang tanpa kita sadari telah memberikan permasalahan kepada kita namun tanpa sadar pula kita juga bisa menyelesaikannya. Misalkan lagi saat kita ingin membuat kopi, maka masalah akan muncul kepada kita. Bagaimana membuat kopi yang enak?

Dengan masalah masalah tersebut maka munculah sebuah cabang ilmu baru yang berfungsi sebagai alat pemecah permasalahan yaitu Algoritma.

## Algoritma

Algoritma adalah langkah-langkah yang logis, sistematis dan efisien untuk menyelesaikan masalah. Langkah langkah disini dapat didefinisikan dengan berbagai hal. Misal saat ingin membuat kopi maka kita dapat membuat algoritma pembuatan kopi sebagai berikut:

1. Masak air panas
2. Ambil cangkir
3. Masukkan kopi ke cangkir 2,5 sendok makan
4. Jika ingin manis maka tambahkan gula sesuai selera
5. Jika tidak maka biarkan saja
6. Tuang air yang telah mendidih ke cangkir hingga penuh
7. Aduk hingga merata
8. Kopi siap disajikan

Apabila kita lihat terdapat 8 langkah untuk membuat kopi terasa enak. 8 langkah tersebut merupakan algoritma yang dibuat untuk membuat kopi.

Lalu apa hubungannya langkah langkah atau algoritma dengan komputer?

Perlu diketahui bahwa cara berpikir komputer dengan kita manusia berbeda. Manusia dapat berpikir dan melakukan langkah langkah untuk menyelesaikan permasalahan tanpa disadari. Namun komputer tidak, komputer butuh langkah langkah yang jelas untuk

menyelesaikan permasalahan yang ada. Kita akan masuk kedalam contoh lagi untuk lebih jelasnya.

Misalkan Andi memiliki 2 ember terisi penuh. Ember A terisi penuh dengan minyak dan satunya Ember B terisi penuh dengan air. Namun Andi baru sadar bahwa iya tertukar dalam menaruh kedua cairan tersebut. Andi ingin menukar supaya ember A terisi penuh dengan air dan ember B terisi penuh dengan minyak. Bagaimanakah cara supaya Andi dapat menukar kedua cairan tersebut?

**Jawaban:**

Sebenarnya caranya cukup sederhana, namun terdapat langkah-langkah yang harus dilakukan Andi yaitu:

1. Ambil satu lagi ember kosong dengan label C
2. Masukkan isi ember A ke dalam Ember kosong C
3. Kemudian Masukkan isi ember B kedalam ember A
4. Terakhir masukan isi ember C ke dalam ember B
5. Selesai, cairan telah tertukar

Permasalahan yang dihadapi andi sebenarnya merupakan sebuah analogi saat kita ingin menukar isi 2 variabel. Misal

$A = 2$

$B = 4$

Bagaimana menukar variabel A dan B? Jika manusia maka tinggal tulis aja langsung  $A = 4$  dan  $B = 2$ . Tapi komputer tidak bisa melakukan itu. Komputer harus melakukan langkah langkah yang lebih jelas untuk menukar variabel itu.

1. Buat satu variabel baru yaitu C
2. Masukkan nilai A ke C
3. Masukkan nilai B ke A
4. Masukkan nilai C ke B
5. Selesai, Variabel telah ditukar

Cukup rumit ya, tapi memang begitulah cari berpikir komputer. Harus langkah demi langkah, *step by step*. Maka dari itulah ilmu Algoritma sangat diperlukan dalam memprogram komputer.

## Bahasa Pemrograman

Dalam membuat sebuah program komputer yang kita lakukan sebenarnya adalah memerintah komputer untuk melakukan keinginan kita. Misal kita menyuruh komputer untuk menghitung luas persegi. Maka kita membuat algoritma atau langkah langkah untuk menghitung persegi dan komputer akan mengeksekusi langkah-langkah yang telah kita buat.

Namun sayangnya komputer tidak memahami bahasa kita. Bahasa yang kita gunakan terlalu rumit untuk ukuran alat yang terdiri dari sirkuit digital. Komputer hanya mengerti 2 buah angka yaitu 0 dan 1. Bagi kita sangat tidak mungkin menggunakan 2 angka itu untuk mengkomunikasikan apa yang kita inginkan terhadap komputer. Maka dari itu munculah

### Bahasa Pemrograman.

Bahasa Pemrograman adalah sebuah instruksi standar untuk memerintah komputer agar menjalankan fungsi tertentu. Bahasa pemrograman sudah sangat dapat dipahami oleh manusia sehingga lebih memudahkan dalam memerintah atau memprogram komputer.

Terdapat 2 jenis bahasa pemrograman menurut kemiripannya dengan bahasa manusia yaitu:

#### 1. Bahasa Tingkat Rendah

Bahasa pemrograman tingkat rendah merupakan bahasa pemrograman yang lebih mirip dengan bahasa mesin. Dengan kemiripan terhadap bahasa mesin tersebut maka jenis bahasa ini masih cukup sulit dibaca oleh manusia sehingga sudah jarang digunakan. Contoh bahasa pemrograman tingkat tinggi adalah Assembly.

#### 2. Bahasa Tingkat Tinggi

Bahasa Pemrograman Tingkat Tinggi adalah bahasa pemrograman yang sudah sangat mirip dengan bahasa manusia. Dengan kemiripan terhadap bahasa manusia ini maka bahasa pemrograman tingkat tinggi sangat mudah digunakan. Contoh bahasa pemrograman tingkat tinggi adalah C, C++, Pascal, Java, Python, dan lain lain.

**Untuk olimpiade yang sering digunakan pada tingkat SMA adalah Pascal dan C++.**

### Struktur Bahasa Pascal

Struktur dari suatu program Pascal terdiri dari sebuah judul program dan suatu blok program atau badan program. Secara ringkas, struktur suatu program Pascal dapat terdiri dari:

1. Judul Program
2. Blok Program

- a. Bagian deklarasi
  - deklarasi label
  - definisi konstanta
  - definisi tipe
  - deklarasi variabel
  - deklarasi prosedur
  - deklarasi fungsi
- b. Bagian pernyataan

Suatu program Pascal yang paling sederhana adalah program yang hanya terdiri dari sebuah bagian pernyataan saja. Bagian pernyataan merupakan bagian yang terakhir dari suatu blok. Bagian ini diawali dengan kata **Begin** dan diakhiri dengan kata **End**. Jadi suatu program Pascal yang paling sederhana dapat berbentuk:

```
Begin
    Pernyataan;
End.
```

Bagian pernyataan ini menunjukkan suatu tindakan yang akan dikerjakan oleh program. Tindakan yang dilakukan oleh program tergantung dari instruksi-instruksi yang diberikan. Pernyataan merupakan instruksi program. Pernyataan-pernyataan yang diberikan untuk dikerjakan ditulis diantara kata Begin dan End. Akhir penulisan End yang harus diakhiri tanda titik.

```
Begin
    Pernyataan;
    Pernyataan;
    Pernyataan;
End.
```

Misal:

```
Begin
    writeln('Hello, World');
End.
```

Program Pascal tidak mengenal aturan penulisan di kolom tertentu, jadi boleh ditulis di kolom mana saja. Penulisan pernyataan-pernyataan pada umumnya menjorok masuk beberapa kolom tidak berpengaruh pada proses, hanya dimaksudkan supaya mempermudah pembacaan program, sehingga akan lebih terlihat bagian-bagiannya. Setiap akhir pernyataan harus diakhiri dengan tanda titik koma.

Di Turbo Pascal, judul program sifatnya adalah opsional dan tidak terlalu berarti dalam program. Jika ditulis dapat digunakan untuk memberi nama program dan daftar dari parameter tentang komunikasi program dengan lingkungannya yang sifatnya sebagai dokumentasi saja. Judul program bila ditulis, harus terletak pada awal dari program dan diakhiri dengan titik koma. Judul program ditulis dalam satu kata, jika lebih dari dua kata harus disambung dengan tanda hubung bawah (tidak boleh ada spasi kosong). Judul program sifatnya sebagai dokumentasi saja, tidak berkaitan dengan proses program.

```
Program belajar_pascal;  
begin  
    writeln('Saya Pascal');  
    writeln('Saya baru kenal');  
end.
```

### Variabel dan Tipe Data

Variabel adalah 'penanda' identitas yang digunakan untuk menampung suatu nilai. Nilai tersebut dapat diubah sepanjang kode program. Secara teknis, variabel merujuk kepada suatu alamat di memory komputer. Setiap variabel memiliki nama yang sebagai identitas variabel tersebut.

Selain variabel juga terdapat konstanta yang dapat menampung data namun data yang ditampung konstanta tidak dapat diubah saat program dijalankan. Dalam bahasa pascal untuk menggunakan variabel dan konstanta harus dideklarasikan terlebih dahulu.

Bagian deklarasi sering juga kamus program. Bagian deklarasi digunakan bila di dalam program menggunakan pengenalan (identifier). Pengenal dapat berupa label, tipe, variabel, prosedur, atau fungsi. Kalau suatu program menggunakan pengenalan, Pascal menuntut supaya pengenalan tersebut dikenalkan dahulu sebelum digunakan, yaitu dideklarasikan atau dibuat kamusnya dulu pada bagian deklarasi.

#### a. Deklarasi konstanta

Bila ingin menggunakan pengenalan yang berisi nilai-nilai konstanta (tetap), maka harus didefinisikan terlebih dahulu pada bagian ini. Definisi konstanta diawali dengan kata cadangan Const diikuti oleh kumpulan pengenalan yang diberi suatu nilai konstanta.



```

Program belajar_pascal;
Const
    Potongan=0.2;
    Gaji=1000000;
    Nama_PT='PT Maju';
Begin
    Writeln('Gaji =',Gaji);
    Writeln('Potongan=',Potongan);
    Writeln('Nama =',Nama_PT);
End.

```

Bila program dijalankan akan menghasilkan:

Gaji = 1000000

Potongan = 2.000000000E-01

Nama = PT Maju

#### b. Deklarasi Variabel

Jika konstanta merupakan pengenal berisi data yang pasti, tetap, tidak berubah, maka variabel adalah pengenal yang berisi data yang dapat berubah-ubah nilai dalam program. Jadi dengan menggunakan konstanta, kita tidak dapat mengubah nilainya dalam program, tetapi dengan menggunakan variabel, nilai dapat berubah/diubah dalam program. Setiap variabel dalam program Pascal harus dideklarasikan sebelum digunakan. Kata cadangan Var digunakan sebagai judul di dalam bagian deklarasi variabel dan diikuti oleh satu atau lebih pengenal yang dipisahkan oleh koma, diikuti dengan titik dua dan tipe datanya serta diakhiri dengan titik koma.

```

Program luas_segi_empat;
Const
    P=5;
    L=4;
Var
    Luas:integer;
Begin
    Luas:=P*L;
    Writeln('Panjang=';p);
    Writeln('Lebar =' ;l);
    Writeln('Luas =' ;Luas);
End.

```

Bila program dijalankan akan menghasilkan:

Panjang=5

Lebar =4

Luas =20

Jika dilihat pada program diatas maka terdapat kata unik yaitu integer. Integer merupakan salah satu tipe data bawaan dari pascal. Apa itu tipe data? Tipe data data ini menunjukkan suatu nilai yang dapat digunakan oleh variabel bersangkutan. Terdapat beberapa tipe data yaitu:

1. Tipe data bilangan bulat

Data numerik integer merupakan nilai bilangan bulat baik dalam bentuk desimal maupun hexadesimal. Nilai integer hexadesimal diawali dengan tanda dollar (\$). Turbo Pascal menyediakan 5 macam tipe data integer yang masing-masing mempunyai jangkauan nilai yang berbeda seperti di bawah ini.

Tipe	Ukuran Memori	Jangkauan Nilai
Byte	1 byte	0..255
ShortInt	1 byte	-128..127
Integer	2 byte	-32768..32767
Word	2 byte	0..65535
LongInt	4 byte	-2147483648..2147483647

2. Tipe data bilangan pecahan atau desimal

Nilai konstanta numerik real berkisar dari 1E-38 sampai dengan 1E+38 dengan mantissa yang signifikan sampai dengan 11 digit. E menunjukkan nilai 10 pangkat. Nilai konstanta numerik real menempati memori sebesar 6

3. Tipe data boolean

Tipe data boolean mempunyai dua buah nilai, yaitu True dan False. Kita dapat mendeklarasikan suatu variabel dengan tipe boolean dan mengisi variabel tersebut dengan nilai data boolean True atau False. Tipe data ini tidak dapat dioperasikan ke dalam bentuk aritmatika.

4. Tipe data karakter

Nilai data karakter berupa karakter yang ditulis diantara tanda petik tunggal, seperti 'A', 'a', '!', '5' dan sebagainya. Penggunaan variabel untuk menyimpan data tipe karakter ini harus dideklarasikan dengan tipe Char.

5. Tipe data String

Nilai data string merupakan urutan dari karakter yang terletak di antara tanda petik tunggal. Bila karakter petik merupakan bagian konstanta string, maka



dapat ditulis dengan menggunakan dua buah petik tunggal berurutan. Nilai data string akan menempati memori sebesar banyaknya karakter stringnya ditambah dengan 1 byte. Bila panjang dari Mengenal Turbo Pascal–Awan Sundiawan 9 suatu string di dalam deklarasi variabel tidak disebutkan, maka dianggap panjangnya adalah 255 karakter.

## Percabangan

Hampir setiap program yang kompleks mengandung suatu penyeleksian kondisi. Dengan menyeleksi suatu kondisi, program dapat menentukan tindakan apa yang harus dikerjakan, tergantung dari hasil kondisi yang diseleksi tersebut. Untuk menyeleksi suatu kondisi, di dalam bahasa Pascal dapat dipergunakan pernyataan IF dan CASE.

### a. Percabangan IF

Struktur pernyataan IF dapat berupa If–Then atau If–Then–Else dengan bentuk umum:

If ungkapan Then pernyataan

Ungkapan adalah kondisi yang akan diseleksi oleh pernyataan If. Bila kondisi yang diseleksi terpenuhi, maka pernyataan yang mengikuti Then akan diproses. Sebaliknya, bila kondisi tidak terpenuhi, maka yang akan diproses adalah pernyataan berikutnya.

```
Var
  Nilai:real;
  Ket : string[11];
Begin
  Ket: 'Tidak Lulus';
  Write('Nilai Ujian = ');
  Readln(Nilai); {Penyeleksian kondisi}
  If Nilai>60 Then
    Ket:='Lulus';
  Writeln(Ket);
End.
```

Pada kondisi diatas jika variabel nilai memiliki nilai lebih dari 60 maka variabel ket akan diisi nilai 'Lulus' sehingga selanjutnya saat dicetak dengan writeln akan menghasilkan output 'Lulus'.

```
If kondisi Then
    pernyataan1;
Else
    pernyataan2;
```

Pernyataan1 akan diproses apabila kondisi benar(terpenuhi) sedang pernyataan2 akan diproses apabila kondisi tidak terpenuhi.

```
Var
    Nilai:Real;
Begin
    Writeln('Nilai Ujian = ');
    Readln(Nilai);
    If Nilai>60 Then
        Writeln('Lulus');
    Else
        Writeln('Tidak Lulus');
End.
```

Cara membaca percabangan tersebut adalah Jika nilai lebih dari 60 maka outputkan 'Lulus', Jika tidak maka outputnya 'Tidak Lulus'.

#### b. Percabangan CASE

Pernyataan Case dapat berbentuk struktur Case-Of atau Case-Of.... Else. Struktur Case-Of mempunyai suatu ungkapan logika yang disebut dengan selector dan sejumlah pernyataan yang diawali dengan suatu label permasalahan yang mempunyai tipe sama dengan selector. Pernyataan yang mempunyai label permasalahan yang bernilai sama dengan nilai selector akan diproses sedang pernyataan yang lainnya tidak. Perbedaan dengan struktur If adalah bila pernyataan If menyeleksi suatu kondisi yang terpenuhi, setelah memproses pernyataan dalam lingkungan yang terpenuhi tersebut, proses penyeleksian masih dilakukan terhadap pernyataan If berikutnya yang lain. Sedang pada struktur Case-Of bila salah satu kondisi terpenuhi dan pernyataan tersebut telah diproses, selanjutnya pernyataan yang lainnya dalam lingkungan Case tidak akan diseleksi lagi.

```
Var
    Nilai:Char;
Begin
    Writeln('Nilai huruf yang didapat = ');
    Readln(Nilai);
    Case Nilai of
        'A':Writeln('Sangat Baik');
        'B':writeln('Baik');
```

```
'C':Writeln('Cukup');
'D':Writeln('Kurang');
'E':Writeln('Gaga');
End;
End.
```

Nilai ujian yang diberikan dalam bentuk huruf A, B, C, D, E mempunyai arti sebagai berikut.

Nilai 'A' berarti sangat baik.

Nilai 'B' berarti baik.

Nilai 'C' berarti cukup.

Nilai 'D' berarti kurang.

Nilai 'E' berarti kurang sekali.

Nilai 'F' berarti gagal.

## Perulangan

Dalam membuat sebuah algoritma kadang kita memerlukan langkah langkah yang harus diulang. Misal saat membuat teh manis kita akan menambahkan gula, setelah menambah gula kamu akan mencicipinya, jika belum manis maka kamu akan melakukan langkah yang sama seperti sebelumnya kemudian akan terulang kembali.

Pengulangan (loop) merupakan bentuk yang sering ditemui di dalam suatu program aplikasi. Di dalam bahasa Pascal, dikenal tiga macam perulangan, yaitu dengan menggunakan pernyataan For, While-Do, dan Repeat...Until.

### a. Struktur Perulangan For

Perulangan dengan pernyataan For digunakan untuk mengulang pernyataan atau satu blok pernyataan berulang kali sejumlah yang ditentukan. Perulangan dengan pernyataan For dapat berbentuk perulangan positif, perulangan negatif dan perulangan bersarang.

```
Var
  I:Integer;
Begin
  For I:= 1 to 5 Do
    Writeln('Pascal');
End.
```

Output yang dihasilkan dari program tersebut adalah

Pascal

Pascal

Pascal

Pascal

Pascal

Program akan mengulangi pernyataan yang ada didalam for sesuai dengan jumlah yg ditulis disana.

### b. Struktur Perulangan While-Do

Perulangan ini menggunakan pernyataan While-Do. Pernyataan While-Do digunakan untuk melakukan proses perulangan suatu pernyataan atau blok pernyataan terus-menerus selama kondisi ungkapan logika pada While masih bernilai logika benar. Bentuk Umum

```
While kondisi ungkapan Do Pernyataan
```

Contoh :

```
Var
    I:Integer;
Begin
    I:=0;
    While I<5 Do
    Begin
        Writeln(I);
        I:=I+1;
    End;
End.
```

Cara membaca program tersebut adalah selama I memiliki nilai kurang dari lima maka pernyataan dalam perulangan tersebut akan di eksekusi sehingga menghasilkan:

- 1
- 2
- 3
- 4
- 5

### c. Struktur Repeat...Until

Struktur Repeat...Until digunakan untuk mengulang pernyataan-pernyataan atau blok pernyataan sampai kondisi yang diseleksi di Until terpenuhi. Bentuk umumnya adalah:

```
Repeat pernyataan Until kondisi
```

Contoh:

```
Var
  I: Integer
Begin
  I:=0;
  Repeat
    I:=I+1;
    Writeln(I);
  Until I=5;
End.
```

Cara membaca program tersebut adalah Ulangi perintah yang ada didalam perulangan hingga nilai I sama dengan 5 sehingga hasilnya adalah sebagai berikut:

```
1
2
3
4
5
```

### Larik atau Array

Sebuah variabel hanya dapat menyimpan sebuah nilai. Ia tidak dapat menyimpan beberapa data. Sedangkan dalam membuat program terkadang kita butuh penampungan data yang cukup banyak. Misalnya menampung data 100 nilai mahasiswa.

Sangat melelahkan jika harus membuat seperti ini :

```
nilai_mahasiswa_1 = 100
nilai_mahasiswa_2 = 90
nilai_mahasiswa_3 = 80
nilai_mahasiswa_4 = 40
```

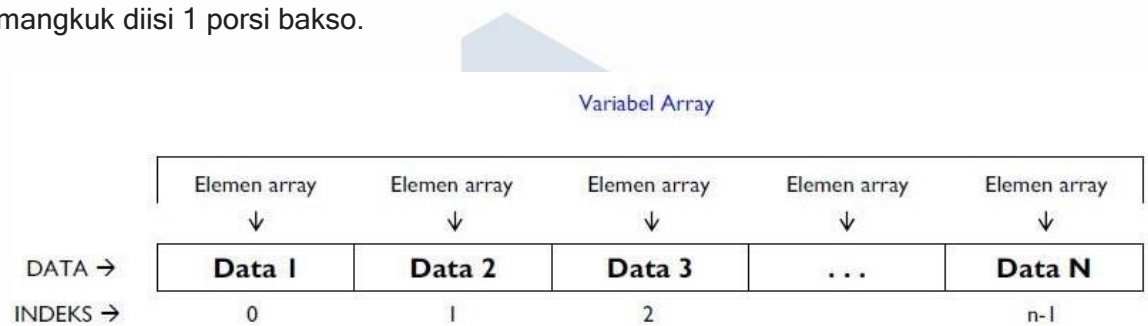
```
nilai_mahasiswa_5 = 80
```

```
•
•
•
```

```
nilai_mahasiswa_100 = 70
```

**Maka dari itu manusia menciptakan *list* atau *array*.**

Larik atau array atau list adalah suatu struktur data yang menyimpan sekumpulan data yang biasanya bertipe sama. Jika di ibaratkan, variabel merupakan sebuah mangkuk yang menyimpan 1 porsi bakso, maka array adalah sekumpulan mangkuk yang dijejer dan masing-masing mangkuk diisi 1 porsi bakso.



Saat kita ingin mengakses data dari suatu variabel maka kita hanya perlu memanggil nama variabelnya. Namun saat kita ingin mengakses data pada array tidak cukup dengan memanggil namanya. Mengapa demikian?

Seperti yang telah disebutkan di awal, array dapat menampung banyak data. Saat kita hanya memanggil nama arraynya saja, maka komputer akan bingung harus mengeluarkan data yang mana. Oleh sebab itu didalam array terdapat penanda atau alamat dari sebuah data yaitu index.

Suatu larik/matrik yang akan dipergunakan di dalam Pascal harus dideklarasikan terlebih dahulu. Deklarasi dari larik didahului dengan kata cadangan Array diikuti oleh tipe indeks yang diletakkan di antara tanda '[ ]', diikuti lagi oleh kata cadangan of dan tipe lariknya.

```
Var
    x:array[1..100] of integer;
```

Larik x telah dideklarasikan sebagai larik tipe integer dengan jumlah elemen maksimum sebanyak 100 elemen. Nilai-nilai elemen larik ini harus berisi nilai-nilai integer. Misalnya elemen-elemen (anggota) dari larik x adalah:

```
x[1]:=25;
```

```
x[2]:=55;
```



```
X[3]:=65;
X[4]:=75;
X[5]:=95;
.
.
.
X[100]:=95;
```

Apabila ingin menampilkan data ke-3 dari dari larik di atas maka dapat digunakan pernyataan:

```
writeln(x[3]);
```

hasil nya adalah  
65

### Subprogram

Subprogram merupakan blok kode yang akan dijalankan ketika dipanggil, dan dapat menerima input sesuai dengan parameter yang ditentukan ketika subprogram itu dibuat. Pada dasarnya, subprogram yang baik merupakan subprogram yang hanya fokus pada satu tujuan.

Subprogram banyak digunakan pada program yang terstruktur, karena:

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

Terdapat 2 jenis sub program yaitu procedure dan function

#### 1. Procedure

Prosecure adalah suatu program terpisah dalam blok sendiri yang berfungsi sebagai subprogram (program bagian). Prosedur diawali dengan kata cadangan Procedure di dalam bagian deklarasi prosedur. Prosedur dipanggil dan digunakan di dalam blok program yang lainnya dengan menyebutkan judul prosedurnya.

```

Procedure Hitung;
Var
    x,y:real;
Begin
    Writeln('Nilai x: ');
    Readln(x);
    Y:=x*x;
    Writeln('Nilai Y= ', y:6;2);
End;

Begin
    Hitung;
End.

```

#### Hasilnya

```

Nilai X : 4
Nilai Y = 16

```

## 2. Function

Fungsi hampir sama dengan prosedur, hanya fungsi harus dideklarasikan dengan tipenya. Tipe deklarasi ini menunjukkan tipe hasil dari fungsi. Tipe tersebut ditulis pada akhir deklarasi fungsi yang didahului dengan titik koma, sebagai berikut.

Function namafungsi(daftar\_parameter):type;

```

Function hitung(var x:integer):Integer;;
Var
    y:real;
Begin
    Y:=x*x;
    hitung:=Y;
End;

Begin
    Writeln('Nilai Y= ',hitung(4));

```

End.

Hasilnya

Nilai Y = 16

