

Algoritma Pemrograman Dalam Bahasa Pascal

Budi Permana, S.Kom

nobiasta@gmail.com

Lisensi Dokumen:

Copyright © 2003-2007 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Pendahuluan

Puji syukur penulis panjatkan kepada Allah SWT yang telah memberikan kesehatan dan kemudahan sehingga penulis dapat menyelesaikan modul ini. Tidak lupa penulis juga ucapkan terima kasih kepada almarhum ayah semoga amal ibadahnya di terima di sisi Allah SWT, Ibu dan Nenek.

Karena penulis juga manusia dan memiliki kekurangan semoga kritik dan saran membangun diperlukan untuk penulis guna menjadikan buku ini lebih baik lagi.

Kritik dan saran bisa disampaikan melalui :

Facebook : Budhi Nobi Permana

Email FB : nobinice@gmail.com

Email : nobiasta@gmail.com

YM : budinobipermana@yahoo.co.id

Akhir kata penulis ucapkan terima kasih semoga menjadi ilmu bermanfaat bagi mereka yang menggunakan buku ini amin.

BAB I

Dasar-dasar Algoritma

A. Sejarah Algoritma

Algoritma adalah suatu perintah yang berisi langkah-langkah untuk menyelesaikan masalah. Algoritma berasal dari nama tokoh ilmuwan islam pada masa itu yaitu Abu Ja'far Muhammad Ibu Musa Al Khawārizmi yang hidup sekitar abad ke-9. Dengan karya bukunya yang terkenal yaitu Al Jabar Wal Muqabala yang berarti “Buku Pemugaran dan Pengurangan”.

Pada awalnya kata algorisma adalah istilah yang merujuk kepada aturan-aturan aritmetis untuk menyelesaikan persoalan dengan menggunakan bilangan numerik arab (sebenarnya dari India, seperti tertulis pada judul di atas). Pada abad ke-18, istilah ini berkembang menjadi **algoritma**, yang mencakup semua prosedur atau urutan langkah yang jelas dan diperlukan untuk menyelesaikan suatu permasalahan.

Berikut adalah contoh bagaimana algoritma dapat menyelesaikan masalah :

Soal1 :

1. Terdapat dua buah gelas yang berisi air yang berwarna merah dan gelas yang berisi air berwarna biru. Kita asumsikan gelas yang berwarna merah adalah gelas A dan gelas yang berwarna biru adalah gelas B. Jika isi gelas A ingin kita pindahkan ke dalam gelas B dan isi gelas B berpindah ke gelas A agar tidak merubah warna masing-masing gelas tentukan cara bagaimana mana gelas tersebut dapat berpindah tempat tanpa merubah isi dalam gelas tersebut.

Penyelesaiannya :

1. Kita tambahkan gelas kosong yang kita asumsikan sebagai gelas C.
2. Pindahkan isi gelas A ke dalam gelas kosong C sehingga sekarang gelas A kosong dan isi gelas C adalah air berwarna merah.
3. Setelah itu pindahkan isi gelas B ke dalam gelas A sehingga isi gelas B kosong dan berpindah ke gelas A.
4. Pindahkan isi gelas C yang berisi air berwarna merah kedalam gelas B yang sudah kosong karena isinya sudah berpindah ke dalam gelas A.

5. Hasil akhirnya adalah A berisi air berwarna biru dan B berisi air berwarna merah.

Soal 2 :

1. Untuk menelepon menggunakan telpon umum maka kita di hadapkan pada langkah-langkah menelpon di telpon umum tersebut. Uraikan lah langkah-langkah tersebut.

Penyelesaian :

1. Pertama angka gagang telpon.
2. Kemudian masukan koin
3. Setelah itu tekan nomor telpon yang di tuju
4. Dan setelah telpon terhubung maka berbicara jika tidak tutup kembali telpon umum.

Perhatikan bagaimana suatu algoritma dapat menyelesaikan suatu masalah tanpa menimbulkan masalah baru. Algoritma adalah suatu hal yang penting yang harus di kuasai oleh para programmer. Karena setiap masalah yang dihadapi seorang programmer dapat diselesaikan dengan algoritma.

B. Dasar Algoritma

➤ Pernyataan dan aksi

Pernyataan adalah setiap langkah penyelesaian suatu algoritma. Sebuah pernyataan menggambarkan aksi (action) algoritmik yang dapat dieksekusi.

Berikut adalah contoh bagaimana perbedaan pernyataan dan aksi :

Tulis “**Tugas algoritma telah selesai**”

Disini tulisan yang berwarna merah adalah yang menggambarkan pernyataan sedangkan tulisan yang berwarna biru adalah yang menggambarkan aksi sehingga menghasilkan tulisan berwarna merah. Jika program diatas di jalankan di pascal maka akan menghasilkan sebagai berikut :

Tugas algoritma telah selesai

➤ Runtutan/Sequence

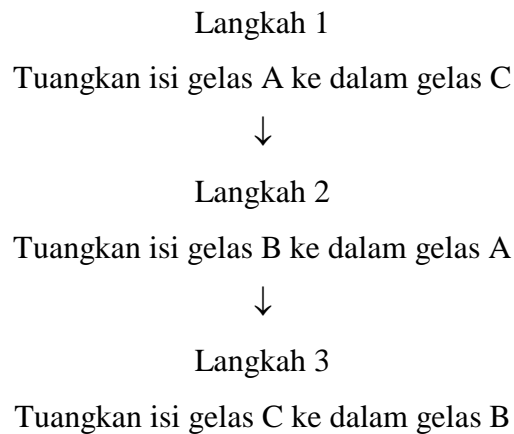
Algoritma merupakan runtutan satu atau lebih instruksi, yang berarti bahwa :

- a. Tiap instruksi di kerjakan satu persatu;
- b. Tiap instruksi dilaksanakan tepat sekali; tidak ada instruksi yang di ulang;
- c. Urutan instruksi yang dilaksanakan pemroses sama dengan urutan instruksi

sebagaimana yang tertulis didalam teks algoritmanya;

d. Akhir dari instruksi terakhir adalah akhir algoritma.

Contoh runtunan :



Contoh Runtunan di Pascal

```
Program Selamat_Pagi_Dunia;  
Uses wincrt;  
begin  
  Write('Selamat Pagi Dunia');  
End.
```

Penjelasan :

Runtunan pada algoritma di atas hanya menampilkan kata 'Selamat Pagi Dunia' ketika program di run.

➤ Pemilihan

Adakalanya sebuah program dihadapkan pada suatu kondisi dimana kondisi tersebut menentukan alur program yang akan kita buat.

Contoh 1

Jika keran air kita tutup maka
Air tidak akan keluar

Pernyataan diatas dapat kita tulis dalam pernyataan kondisional sebagai berikut :

If **kondisi** then Aksi

Jika kita tulis :

If keran air di tutup then Air tidak akan keluar

If keran air di buka then Air akan keluar

Contoh lainnya sebagai berikut :

If bensin habis then Motor mogok

If air mendidih then Matika kompor

If motor rusak then Jalan kaki saja

Dalam bahasa Indonesia If berarti jika dan then berarti maka dimana setiap kondisi bias menghasilkan nilai benar atau salah.

➤ Pengulangan

Jika di banding dengan manusia kelebihan computer adalah tidak mengenal kata lelah jika mengerjakan pekerjaan yang sama secara berulang kali. Tentunya kita pernah diberi tugas menulis kata yang sama sewaktu di bangku sekolah dasar oleh guru. Dimana kita disuruh menulis kata “Saya akan belajar yang rajin” sebagai bentuk latihan kita dalam membentuk huruf tulisan agar tulisan kita menjadi lebih baik bukan. Tetapi terkadang latihan yang kita kerjakan membuat kita jenuh dan bosan.

Contoh algoritma yang kurang cerdas untuk menuliskan kata sebanyak 100 x sebagai berikut :

```
Program menulis_kata;  
{Menuliskan kata sebanyak 100 kali}  
Deklarasi  
{Tidak ada}  
Algoritma  
  Tulis ‘Saya akan belajar yang rajin’;  
  Tulis ‘Saya akan belajar yang rajin’;  
  Tulis ‘Saya akan belajar yang rajin’;  
  Tulis ‘Saya akan belajar yang rajin’;  
  Tulis ‘Saya akan belajar yang rajin’;  
  .....
```

Alangkah baiknya jika pengulangan tersebut kita buat secara otomatis yaitu dengan menggunakan algoritma perulangan sebagai berikut :

```
Program menulis_kata;  
{Menuliskan kata sebanyak 100 kali}  
Deklarasi  
  
Algoritma  
For i dari 1 sampai 100  
  Tulis ‘Saya akan belajar yang rajin’;  
End for
```

Algoritma diatas akan mengulangi kata ‘Saya akan belajar yang rajin’ sampai 100 kali, apabila perulangan sudah terpenuhi maka perulangan pun akan berhenti.

BAB II

Tipe, Operator Dan Eskpresi

Variabel dan konstanta adalah objek data yang biasa dimanipulasi didalam sebuah program. Agar dapat di eksekusi oleh program terlebih dulu tipe data variable tersebut harus di deklarasikan sebelum program dapat di eksekusi. Hasil akhir dari eksekusi adalah menghasilkan sebuah result dimana program mengkombinasikan setiap variable sehingga menghasilkan suatu nilai baru. Tipe data dikelompokan menjadi 2 yaitu tipe dasar dan tipe bentukan.

1. Tipe Dasar

Tipe dasar adalah tipe yang sudah ada dan dapat langsung di pakai. Yang termasuk tipe dasar adalah bilangan logik, bilangan bulat, karakter, bilangan riil, dan string.

Contoh Tipe data dalam PASCAL:

- ✓ integer (contoh: -1, -2, 0, 83, 1283)
- ✓ real (contoh: -1.242, 39.0, 1.87)
- ✓ char (contoh: 'a', 'b', 'c')
- ✓ boolean (contoh: true, false)
- ✓ string (contoh: 'al-azhar', 'nama_saya')

Bilangan Logik

Nama tipe bilangan logik adalah *Boolean*. Kata booeelan ini diambil dari nama seorang tokoh matematikawan Inggris yang bernama George Boole. Bilangan logik ini hanya mengenal dua nilai yaitu *true* atau *false*.

Biasanya tipe Boolean ini sering di aplikasikan pada operasi logika dimana operasi logika ini menghasilkan nilai true atau false. Operator yang sering digunakan untuk operasi logika ini adalah not, and, or dan xor.

Bilangan Bulat

Bilangan bulat terdiri dari bilangan cacah (0, 1, 2, 3, ...) dan negatifnya (-1, -2, -3, ...; -0 adalah sama dengan 0 sehingga tidak lagi dimasukkan secara terpisah). Bilangan bulat dapat dituliskan tanpa mengandung komponen desimal atau pecahan. Nama tipe dari bilangan bulat ini adalah integer. Tipe bilangan bulat ini mempunyai ranah nilai yang tidak terbatas. Berikut tipe data *integer* yang memiliki rentang nilai tertentu, yaitu : *byte*, *shortint*, *word*, *integer* dan *longint*.

<i>Tipe</i>	<i>Rentang Nilai</i>
Byte	0 .. 255
Shortint	-128 .. 27
Word	0 .. 65535
Integer	-32768 .. 32767
Longint	-2147483648 .. -2147483648

Operasi bilangan bulat ada dua, yaitu

Operasi Aritmatika	Operasi Perbandingan
+	<
-	<=
*	>
Div	>=

Mod	=
	<>

Bilangan Riil

Dalam matematika, **bilangan riil** atau **bilangan real** menyatakan bilangan yang bisa dituliskan dalam bentuk desimal, seperti 2,4871773339... atau 3.25678. Nama tipe dari bilangan riil adalah *real*.

Tipe *real* dapat direpresentasikan ke dalam empat macam tipe, yaitu *real*, *single*, *double* dan *extended*.

Tipe	Rentang
Real	$2.9 \times 10^{-39} \dots 1.7 \times 10^{18}$
Single	$1.5 \times 10^{-45} \dots 3.4 \times 10^{38}$
Doble	$5.0 \times 10^{-324} \dots 1.7 \times 10^{308}$
Extended	$3.4 \times 10^{-4932} \dots 1.1 \times 10^{4932}$

Setiap konstanta yang bertipe bilangan riil harus ditulis dengan tanda titik desimal. Operasi yang dilakukan terhadap bilangan riil ada dua macam, yaitu :

Operasi Aritmatika	Operasi Perbandingan
+	<
-	<=
*	>
/	>=
	=
	<>

Karakter

Karakter adalah mencakup huruf abjad, tanda baca, angka yang diberi tanda petik ('0') dan karakter khusus seperti '&', '@'. Nama tipe dari karakter ini adalah *char*.

Operasi tipe karakter adalah operasi perbandingan

Operasi Perbandingan	
=	<
<>	<=
>=	>

String

Tipe data string adalah tipe data yang digunakan untuk menyimpan barisan karakter. Nama tipe dari string ini adalah *string*. Biasanya Konstanta *string* harus diapit oleh tanda petik tunggal, contoh : 'Presiden', 'Indonesia'.

Operasi

Operasi yaitu operator aritmatika (+) dan operasi perbandingan.

Operasi Aritmatika	Operasi Perbandingan
+	<
	<=
	>
	>=
	=
	<>

2. Tipe Bentukan

Tipe bentukan adalah tipe yang dibuat sendiri sesuai keinginan pemrogram.

Contoh :

Type Bilangan : real;

Tipe bilangan diatas adalah tipe bilangan real. Dalam implementasinya jika kita memiliki variable R yang bertipe bilangan maka tipe R tersebut sama dengan tipe bilangan real.

BAB III

Pemilihan Dalam Bahasa Pascal

Didalam membuat program tentunya ada masalah yang dapat di pecahkan secara langsung seperti menghitung sebuah persegi panjang yang hasilnya bias kita dapatkan dari perkalian panjang * lebar. Tentunya setiap masalah memiliki penanganan yang berbeda, seperti didalam sebuah perkuliahan bagaimana kita bias menentukan nilai grade A, grade B grade C dari hasil nilai angka yang kita dapatkan.

1. Statemen IF

Penggunaan statement If ini adalah bagaimana kita mengkondisikan sebuah expresi dimana kita menentukan sendiri mana expresi yang menghasilkan nilai true dan mana expresi yang menghasilkan nilai false. Untuk aturan penulisan statement IF ini adalah sebagai berikut :

```
If kondisi then  
    Statement1  
    else  
    statement2  
end;
```

Statement di atas dapat kita ilustrasikan sebagai berikut :

1. Jika suatu kondisi menghasilkan nilai true maka statement 1 akan dijalankan.
2. Sebaliknya jika suatu kondisi menghasilkan nilai false maka statement 2 lah yang akan di jalankan.

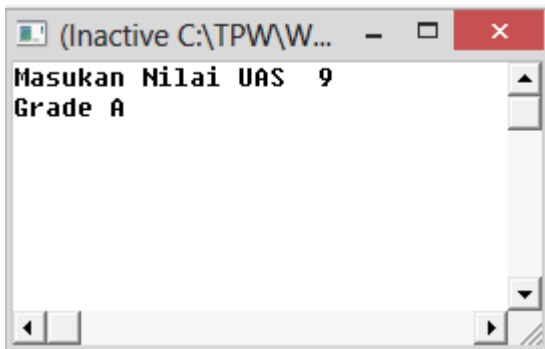
Berikut kita akan membuat program grade menggunakan statement If dengan bahasa pascal.

Contoh program 1 :

```

program Grade;
uses
    WinCrt;
    var n : integer;
begin
    Write('Masukan Nilai UAS ');readln(n);
    if n >= 8 then
        write('Grade A')
    else
        if (n >= 7) and (n <8) then
            write('Grade B')
        else
            write('C');
end.
    
```

Kode di atas jika di jalankan akan menghasilkan output sebagai berikut :



Penjelasa kode :

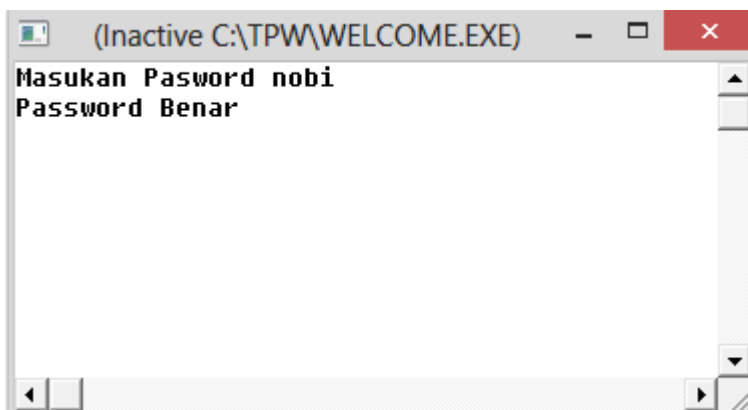
Kode	Keterangan
Write ('Masukan Nilai Uas ')	Berfungsi untuk menampilkan teks 'Masukan Nilai'
Readln(n)	Berfungsi sebagai inputan nilai Uas.
If n >= 8 then write ('Grade A')	Jika nilai inputan lebih besar dari n maka akan menampilkan Grade A
If (n >= 7) and (n <8) then write('Grade	Jika nilai inputan lebih besar sama dengan

B')	7 dan kurang dari 8 maka akan menampilkan Grade B.
Else write ('Grade C')	Jika inputan tidak memenuhi kriteria A dan tidak memenuhi kriteria B maka akan menampilkan Grade C.

Contoh program 2 :

```
program Password;  
uses  
  WinCrt;  
var n : string;  
begin  
  Write('Masukan Pasword ');readln(n);  
  If n = 'nobi' then  
    write('Password Benar')  
  else  
    write('Password Salah');  
end.
```

Jika kode yang dimasukan nobi maka akan menghasilkan output sebagai berikut :



Penjelasan kode :

Kode	Keterangan
Write ('Masukan ')	Berfungsi untuk menampilkan teks 'Masukan Password'
Readln(n)	Berfungsi sebagai inputan password
If n = 'nobi' then write ('Password Benar')	Jika nilai inputan benar = nobi maka akan menampilkan 'Password Benar'
Else write('Password Salah')	Jika nilai inputan salah tidak sama dengan nobi maka akan menampilkan 'Password Salah'

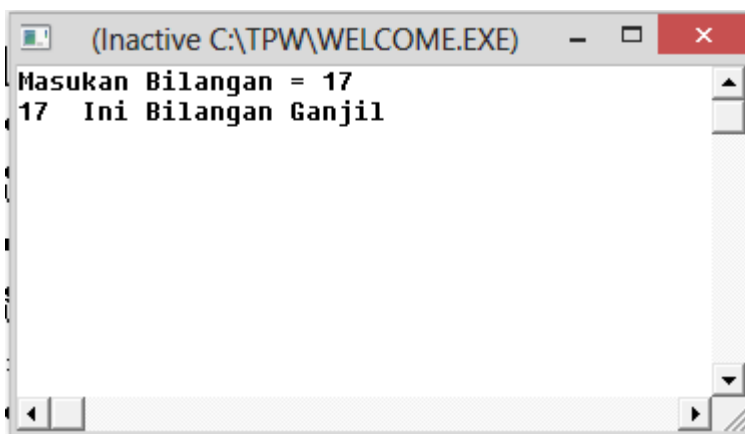
Contoh Program 3:

```

program Gan_jilGenap;
uses
  WinCrt;
var n : integer;
begin
  Write('Masukan Bilangan = '); readln(n);
  if n mod 2 = 0 then
    write('Ini Bilangan Genap')
  else
    write( n, ' Ini Bilangan Gan_jil');
end.

```

Jika di run maka akan menghasilkan output sebagai berikut :



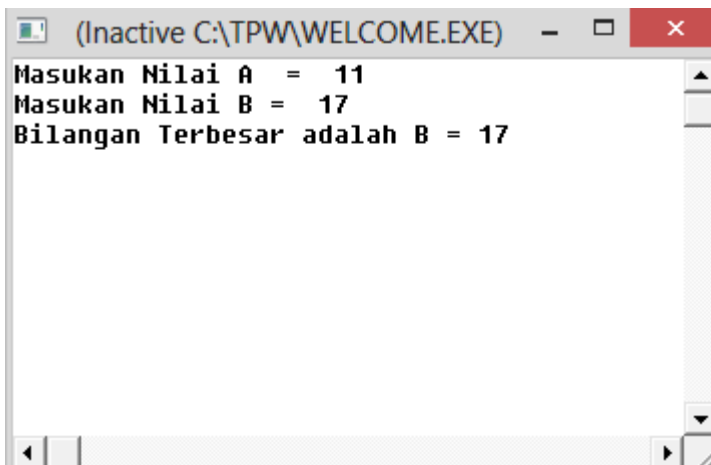
Penjelasan kode :

Kode	Keterangan
Write ('Masukan ')	Berfungsi untuk menampilkan teks 'Masukan Bilangan'
Readln(n)	Berfungsi sebagai inputan angka
If n mod 2 = 0 write ('ini bilangan Genap')	Jika kondisi n memenuhi kriteria maka n adalah bilangan genap.
Else ('ini bilangan Ganjil')	Jika tidak maka n adalah bilangan ganjil

Contoh Program 4 :

```
program BilanganTerbesar;  
uses  
  WinCrt;  
var  
  n : integer;  
  i : integer;  
begin  
  Write('Masukan Nilai A = '); readln(n);  
  write('Masukan Nilai B = '); readln(i);  
  
  if n > i then  
    write('Bilangan Terbesar adalah A = ', n)  
  else  
    write('Bilangan Terbesar adalah B = ', i );  
end.
```

Jika di run maka akan menghasilkan output sebagai berikut :



Penjelasan kode :

Kode	Keterangan
Write ('Masukan Nilai A ')	Berfungsi untuk menampilkan teks 'Masukan Nilai A dan B'
Write ('Masukan Nilai B ')	
Readln(n)	Berfungsi sebagai inputan angka
If n > I then write('Bilangan terbesar adalah A')	Jika nilai n lebih besar dari I maka akan menampilkan pesan Bilangan terbesar adalah A.
Else write('Bilangan terbesar adalah B')	Jika kondisi pertama tidak terpenuhi maka akan menampilkan pesan Bilangan terbesar adalah B

Contoh Program 5 :

```

Program Cuaca_sekarang;

uses
  wincrt;

var
  cuaca : string;

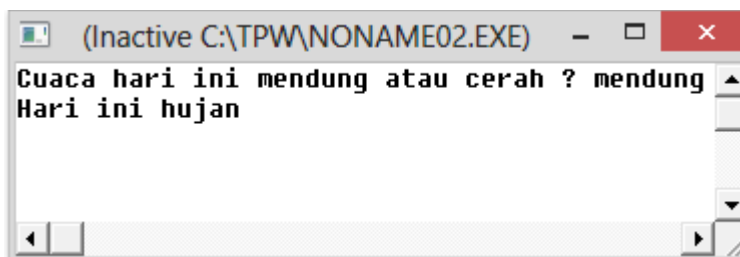
begin
  write('Cuaca hari ini mendung atau cerah ? '); readln(cuaca);

  if Cuaca = 'mendung' then
    writeln('Hari ini hujan')
  else
    writeln('Hari ini panas');

  end.

```

Jika di run maka akan menghasilkan output sebagai berikut :



Penjelasan kode :

Kode	Keterangan
Write ('Cuaca hari ini mendung atau cerah')	Berfungsi untuk menampilkan teks 'cuaca hari ini mendung atau cerah'
Readln(n)	Berfungsi sebagai inputan mendung atau cerah
If cuaca = 'mendung' then writeln('Hari ini Hujan')	Jika inputan cuaca sama dengan mendung maka akan menampilkan pesan hari ini hujan.
Else write('Hari ini panas')	Jika inputan cuaca sama dengan cerah maka akan menampilkan pesan hari ini panas

3. Statement Case Of

Sejauh ini, Anda telah belajar bagaimana menggunakan statemen If. Namun dalam beberapa kasus, statement Case Of lebih disukai untuk pernyataan If karena mengurangi beberapa kode yang tidak perlu, tetapi makna yang sama tetap dipertahankan. Case of ini sangat mirip dengan statement If dalam penggunaannya. Berikut adalah aturan penggunaan statement Case Of :

```
case (expression) of  
  L1 : S1;  
  L2: S2;  
  ...  
  ...  
  Ln: Sn;  
end;
```

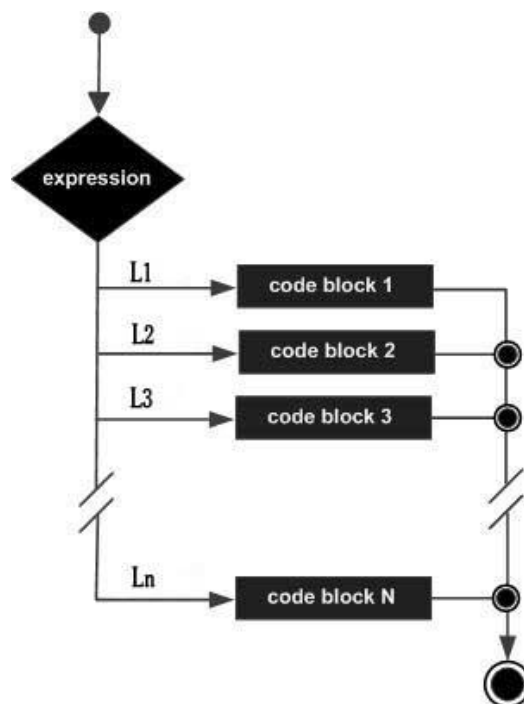
Dimana, L1, L2 ... adalah Case label, atau nilai-nilai masukan yang bisa integer, karakter, boolean atau disebutkan item data. S1, S2, ... adalah pernyataan, masing-masing pernyataan mungkin memiliki satu atau lebih dari satu label kasus yang terkait

dengannya. Ekspresi ini disebut Case pilihan atau Case indeks. Case Indeks mengasumsikan nilai-nilai yang sesuai dengan case label.

Aturan berikut berlaku untuk case statement :

- ✓ Ekspresi yang digunakan dalam sebuah case statment harus memiliki tipe integral atau enumerasi, atau menjadi tipe kelas di mana kelas memiliki fungsi konversi tunggal untuk tipe integral atau enumerasi.
- ✓ Setiap Case diikuti dengan nilai yang akan dibandingkan dengan titik dua.
- ✓ Case Label untuk case harus tipe data yang sama sebagai ekspresi dalam case statment, dan harus berupa sebuah konstanta atau literal.
- ✓ Compiler akan mengevaluasi ekspresi case. Jika salah satu dari nilai case label ini sesuai dengan nilai ekspresi, pernyataan yang mengikuti label ini dijalankan. Setelah itu, program akan berakhir.
- ✓ Jika tidak ada case label yang sesuai dengan nilai ekspresi aliran program akan berakhir.

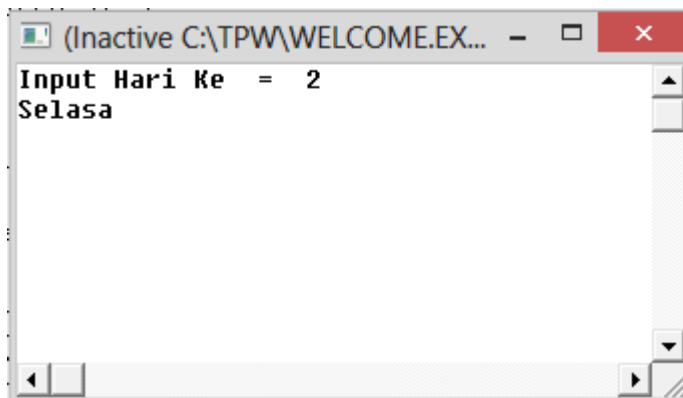
Perhatikan data flow diagram berikut :



Contoh Program 1 :

```
program Pilih_Hari;  
uses  
  WinCrt;  
var  
  pilih : integer;  
begin  
  write('Input Hari Ke = ');readln(pilih);  
  case pilih of  
    1 : write('Senin');  
    2 : write('Selasa');  
    3 : Write('Rabu');  
    4 : write('Kamis');  
    5 : write('Jumat');  
    6 : write('Sabtu');  
    7 : write('Minggu');  
  end;  
end.
```

Jika di run maka akan menghasilkan output sebagai berikut :



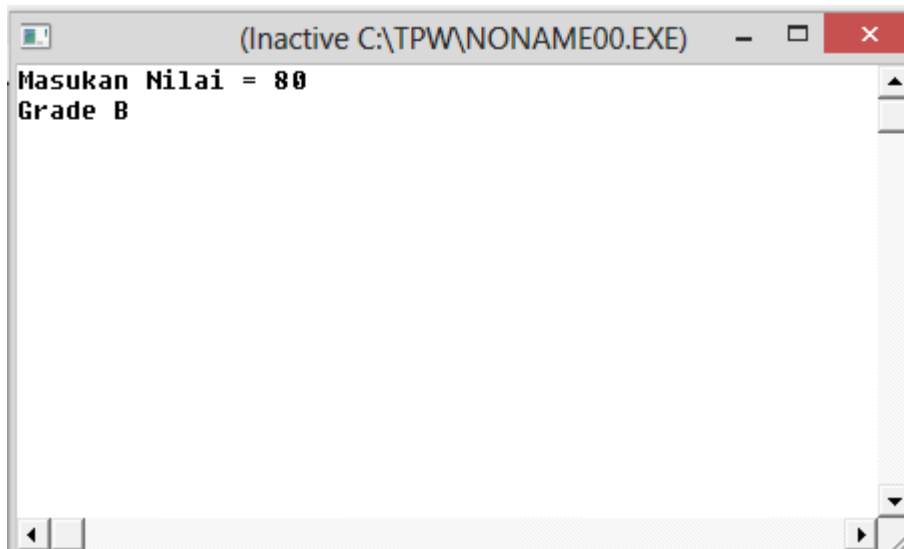
Penjelasan kode :

Kode	Keterangan
Write ('Masukan ')	Berfungsi untuk menampilkan teks 'Input hari ke'
Readln(n)	Berfungsi sebagai inputan pilihan
Case pilih of	Jika kita menginput angka no 1 maka akan tampil pesan senin.

Contoh Program 2 :

```
Program Grade;  
uses wincrt;  
var nilai : integer;  
  
Begin  
Write('Masukan Nilai = ');readln(nilai);  
  
Case nilai of  
85..100 : Write('Grade A');  
79..84 : write('Grade B');  
60..78 : write('Grade C');  
0..59 : write('Grade D');  
end;  
  
end.
```

Jika di run maka akan menghasilkan output sebagai berikut :



Penjelasan kode :

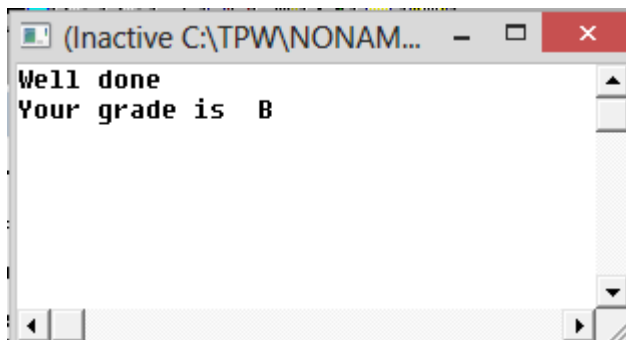
Kode	Keterangan
Write ('Masukan Nilai ')	Berfungsi untuk menampilkan teks 'Masukan Nilai'
Readln(n)	Berfungsi sebagai inputan nilai
Case nilai of	Jika kita menginputkan nilai 80 maka akan tampil pesan Grade B. case nilai of berisi list nilai grade.

Contoh program 3 :

```
program Grade;
uses wincrt;
var
  Nilai_Huruf: char;
begin
  Nilai_Huruf := 'B';

  case (Nilai_Huruf) of
    'A' : writeln('Excellent! ');
    'B', 'C': writeln('Well done' );
    'D' : writeln('You passed' );
    'F' : writeln('Better try again' );
  end;
  writeln('Your grade is ', Nilai_Huruf );
end.
```

Jika di run maka akan menghasilkan output sebagai berikut :



Penjelasan kode :

Kode	Keterangan
Nilai_Huruf = 'B'	Jika kita memasukan huruf B dan menjalankan program maka akan tampil pesan 'Well Done', 'Your Grade is B' kita bias mengganti B menjadi A atau C.
Case nilai_huruf of	Berisi nilai list grade.

Contoh Program 4 :

```

program Kode_Pelajaran;

uses
  WinCrt;

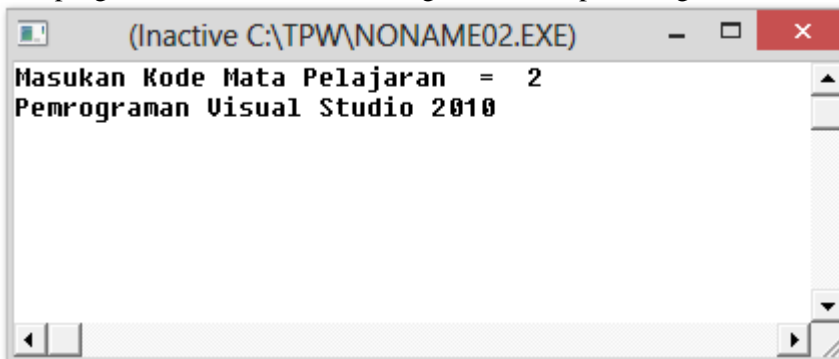
var
  pilih : integer;

begin
  write('Masukan Kode Mata Pelajaran = ');readln(pilih);

  Case pilih of
  1 : write('Pascal');
  2 : write('Pemrograman Visual Studio 2010');
  3 : Write('Pemrograman Java Script');
  4 : write('Pemrograman PHP');
  5 : write('Merakit Komputer');
  6 : write('Sistem OperasiJaringan');
  7 : write('Teknik Digital');
  end;
end.

```

Jika program di run maka akan menghasilkan output sebagai berikut :



Penjelasan kode:

Kode	Keterangan
Write ('Masukan mata pelajaran ')	Berfungsi untuk menampilkan teks 'Masukan kode mata pelajaran'
Readln(pilih)	Berfungsi sebagai inputan pilihan
Case pilih of	Jika kita menginput angka no 2 maka akan tampil pesan Pemrograman Visual Studio 2010. Case pilih of ini berisi list pilihan mata pelajaran.

BAB IV

Perulangan Dalam Bahasa Pascal

Salah satu kelebihan komputer apabila di bandingkan dengan manusia adalah dapat melakukan suatu perintah yang sama secara berulang-ulang tanpa mengenal lelah dan bosan. Kelebihan ini lah yang tidak di miliki oleh manusia. Sehingga manusia dapat memanfaatkan kelebihan ini untuk kepentingannya dalam menyelesaikan tugas-tugas keseharian.

Di dalam bahasa pascal terdapat 3 pernyataan perulangan yang lazim sering digunakan oleh para pemrogram. Ketiga pernyataan itu adalah sebagai berikut :

1. Pernyataan For,
2. Pernyataan While, dan
3. Pernyataan Repeat.

Ada baiknya setiap pernyataan tersebut kita pahami satu persatu. Karena dalam setiap masalah yang kita hadapi perlu penanganan yang berbeda pula. Oleh Karen itu setiap pernyataan tersebut memiliki fungsi yang berbeda-beda dalam setiap pengaplikasiannya.

1. Pernyataan For

Perulangan for digunakan untuk mengulang statmen sebanyak yang telah didefinisikan sebelum perulangan tersebut di eksekusi. Perulangan For..to.. do biasanya digunakan apabila pencacah sudah di ketahui.

```
For pencacah <- nilai awal to nilai akhir do  
    pernyataan  
end for;
```

Keterangan :

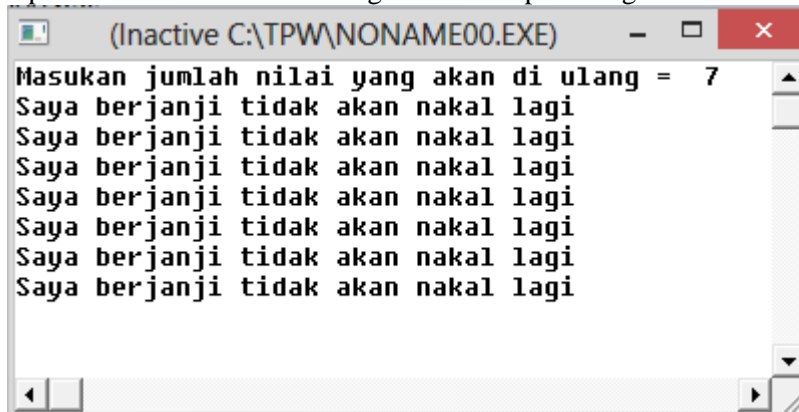
- a. Pencacah harus berupa tipe data integer atau karakter.
- b. Nilai awal harus lebih kecil atau sama dengan nilai akhir.
- c. Pernyataan adalah statmen yang di ulang sebanyak berulang kali.

Berikut adalah contoh-contoh pengaplikasian pernyataan for dalam bahasa Pascal.

Contoh Program 1 :

```
program Perulangan_Janji ;  
  
uses  
    WinCrt ;  
  
var  
n : integer ;  
i : integer ;  
  
begin  
    write('Masukan jumlah nilai yang akan di ulang = ');readln(n);  
    for i := 1 to n do  
        writeln('Saya berjanji tidak akan nakal lagi');  
  
end.
```

Apabila di run maka akan menghasilkan output sebagai berikut :



Penjelasan kode :

Kode	Keterangan
Write ('Masukan jumlah nilai yang akan di ulang = ')	Berfungsi untuk menampilkan teks 'Masukan jumlah nilai yang akan di ulang ='
Readln(n)	Berfungsi sebagai inputan jumlah nilai angka
For I := 1 to n Writeln('Saya berjanji tidak akan nakal lagi')	Berfungsi sebagai perulangan dimana pengulangan sesuai dengan nilai n dan akan menampilkan pesan saya berjanji tidak akan nakal lagi.

Contoh Program 2 :

```

program cetak_angka;

uses
    WinCrt;

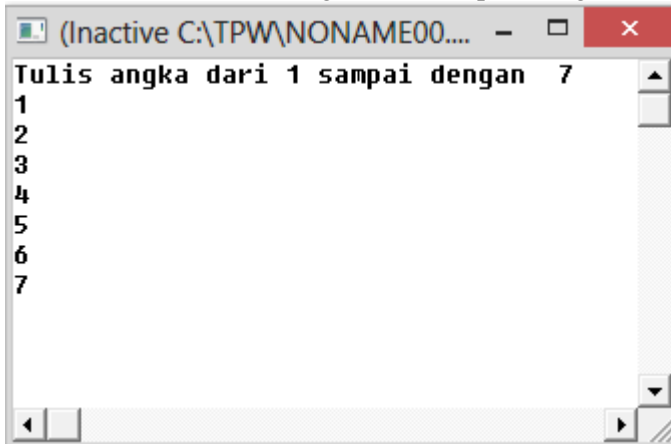
var
    n : integer;
    i : integer;

begin
    write('Tulis angka dari 1 sampai dengan ');readln(n);

    for i := 1 to n do
        writeln(i);

end.
    
```


Jika di run maka akan menghasilkan output sebagai berikut :



Penjelasan kode :

Kode	Keterangan
Write ('Tulis angka dari 1 sampai dengan ')	Berfungsi untuk menampilkan teks 'Tulis angka dari 1 sampai dengan'
Readln(n)	Berfungsi sebagai inputan jumlah nilai angka
For I := 1 to n Writeln(i)	Berfungsi sebagai perulangan dimana pengulangan sesuai dengan nilai n dan akan menampilkan angka sebanyak n.

Contoh Program 3:

```

program deret_angka_terbesar_ke_terkecil;

uses
    WinCrt;

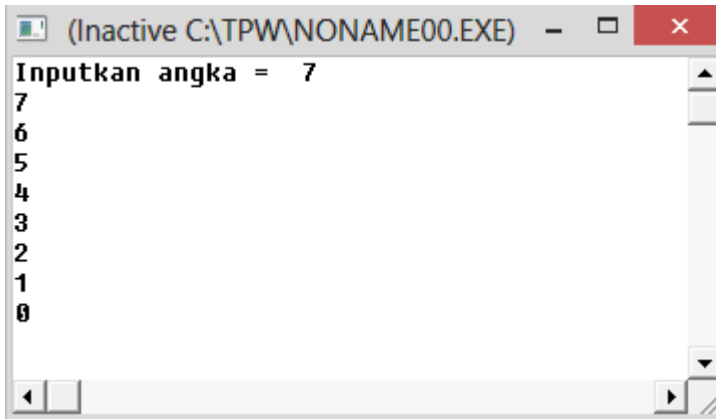
var
    i : integer;
    n : integer;

begin
    write('Inputkan angka = ');readln(n);

    for i := n downto 0 do
        writeln(i);

end.
    
```

Jika di run maka akan menghasilkan output sebagai berikut :



Penjelasan kode :

Kode	Keterangan
Write ('Input angka = 7')	Berfungsi untuk menampilkan teks 'input angka ='
Readln(n)	Berfungsi sebagai inputan jumlah nilai angka n
For I := n downto 0 Writeln(i)	Berfungsi sebagai perulangan dimana pengulangan sesuai dengan nilai n dan akan menampilkan angka terbesar hingga terkecil.

2. Pernyataan While

Pernyataan while ini akan melakukan perulangan selama kondisi bernilai benar/true. Jika perulang bernilai salah/false maka perulangan akan berhenti.

While kondisi do
 pernyataan
endwhile;

contoh program 1:

```

program pen_jumlahan_berderet;
uses
    WinCrt;

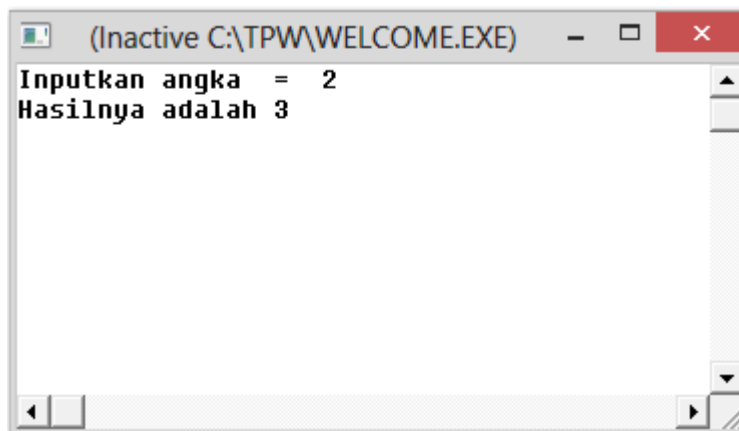
var
    n : integer;
    i : integer;
    j : integer;

begin
    write('Inputkan angka = ');readln(n);

        j := 0;
        i := 1;
        while i <= n do
            begin
                j := j + i;
                i := i + 1;
            end;
        writeln('Hasilnya adalah ', j);

end.
    
```

Jika di run maka akan menghasilkan output sebagai berikut :



Penjelasan kode :

Kode	Keterangan
Write ('Input angka ')	Berfungsi untuk menampilkan teks 'Input angka'
Readln(n)	Berfungsi sebagai inputan angka
J := 0 dan I := 1	Sebagai inisialisasi j sebagai 0 dan I sebagai 1
while i <= n do begin	Selama I kurang dari n maka proses akan terus di lanjutkan

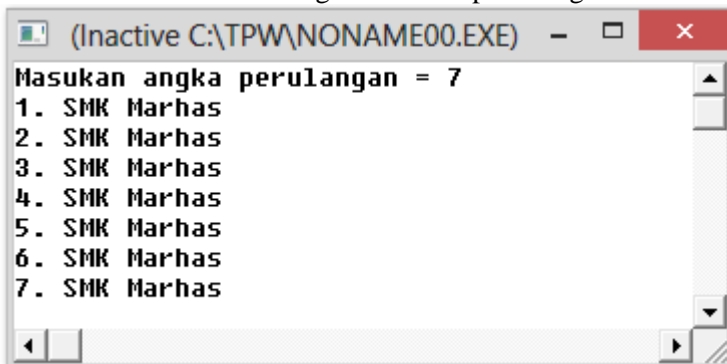
<pre>j := j + 1; i := i + 1; end;</pre>	<p>Variable j ditambah dengan 1 kemudian hasil penambahan akan di tampung di variable j</p> <p>Variable i ditambah dengan 1 kemudian hasil penambahan akan di tampung di variable I apabila data yang ditampung melebihi nilai n maka proses akan berhenti.</p>
---	---

3. Pernyataan repeat

Contoh program 1 :

```
program perulangan_kata;  
uses  
winCRT;  
  
var i : integer;  
n : integer;  
  
begin  
write('Masukan angka perulangan = ');readln(n);  
i := 1;  
repeat  
writeln(i, '. SMK Marhas');  
i := i + 1;  
until i > n ;  
end.
```

Jika di run maka akan menghasilkan output sebagai berikut :



Penjelasan kode :

Kode	Keterangan
Write ('Masukan angka perulangan= ')	Berfungsi untuk menampilkan teks 'Masukan angka perulangan= '
Readln(n)	Berfungsi sebagai inputan angka
I := 1;	Inisialisasi I sebagai 1
Repeat	Pernyataan perulangan
Writeln(I,'SMK Marhas')	Untuk menampilkan pesan SMK Marhas sedangkan 'I' akan menampilkan angka sebanyak di ulang.
i = I + 1; until I > n	Berfungsi apabila nilai I lebih besar dari n maka perulangan akan berhenti.

Contoh program 2 :

```

program password;
uses
  wincrt;

var i : string;
    n : string;

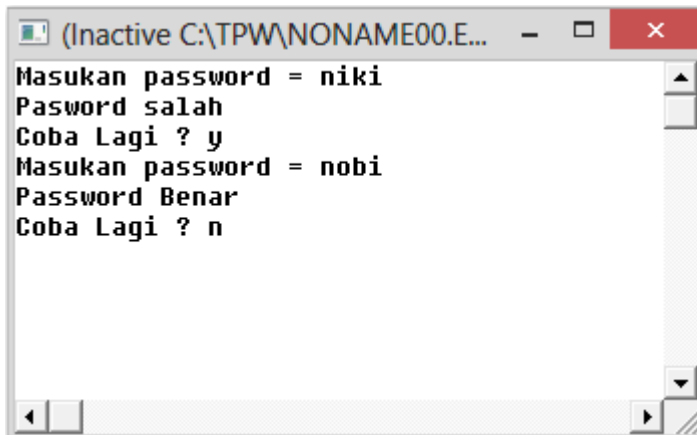
begin
  repeat
    write('Masukan password = ');readln(n);
    i := 'n';

  begin
    if n = 'nobi' then
      writeln('Password Benar')
    else
      writeln('Pasword salah');
    end;
    write('Coba Lagi ? ');readln(i);
  until i = 'n';

end.

```

Jika di run maka akan menghasilkan output sebagai berikut :



Penjelasan kode :

Kode	Keterangan
Write ('Masukan Password ')	Berfungsi untuk menampilkan teks 'Masukan Password'
Readln(n)	Berfungsi sebagai inputan password
I := 'n'	Inisialisasi I sebagai karakter n
If n = 'nobi' then writeln('Password Benar')	Jika inputan n sama dengan nobi maka akan menampilkan pesan password benar.
Else writeln('Password Salah')	Jika kondisi pertama tidak terpenuhi maka akan menampilkan pesan Password salah.
Write('Coba lagi ?') readln(i) Until I = 'n'	Akan menampilkan pesan pertanyaan Coba lagi. Jika ya maka program akan di ulang dan jika memasukan nilai n maka program akan berhenti.

Contoh program 3:

```

Program Cuaca_sekarang;

uses
  wincrt;

var
  cuaca : string;
  i : string;

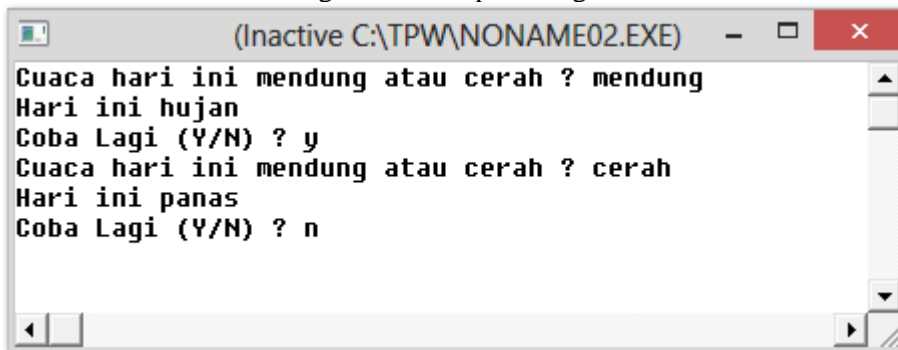
begin
repeat

  write('Cuaca hari ini mendung atau cerah ? '); readln(cuaca);
  i := 'n';

begin
  if cuaca = 'mendung' then
    writeln('Hari ini hujan')
  else
    writeln('Hari ini panas');
  end;
  write('Coba Lagi (Y/N) ? ');readln(i);
  until i = 'n';
end.

```

Jika di run maka akan menghasilkan output sebagai berikut :



Penjelasan kode :

Kode	Keterangan
Write ('Cuaca hari ini mendung atau cerah ? ')	Berfungsi untuk menampilkan teks 'Cuaca hari ini mendung atau cerah ? '
Readln(cuaca)	Berfungsi sebagai inputan nilai
I := 'n'	Inisialisasi I sebagai karakter n
If cuaca = 'mendung' then writeln('Hari ini hujan')	Jika inputan cuaca sama dengan mendung maka akan menampilkan pesan hari ini hujan.
Else writeln('Hari ini panas')	Jika kondisi pertama tidak terpenuhi maka akan menampilkan pesan Hari ini panas.

<p>Write('Cobal lagi ?') readln(i)</p> <p>Until I = 'n'</p>	<p>Akan menampilkan pesan pertanyaan Coba lagi. Jika ya maka program akan di ulang dan jika memasukan nilai n maka program akan berhenti.</p>
---	---

BAB V

Procedure Dalam Bahasa Pascal

Prosedur adalah sebuah sub program yang dibuat terpisah dari program utama yang dapat di panggil sewaktu-waktu oleh program utama. Procedure merupakan program yang tidak berdiri sendiri karena tidak dapat di eksekusi secara langsung. Procedure dapata memiliki parameter atau juga tidak. Procedure yang memiliki parameter biasanya variable-variabel/peubahnya biasa di deklarasikan pada bagian header procedure.

Penulisan Procedure

```
Procedure Nama_Procedure (param_formal1:tipe_param1, param_formal2:tipe_param2);  
{jika prosedur tidak memiliki parameter, tanda kurung tidak disertakan}  
var  
{bagian deklarasi prosedur, sifatnya lokal}  
begin  
    statement-1;  
    statement-2;  
  
    Statement-n;  
end;
```

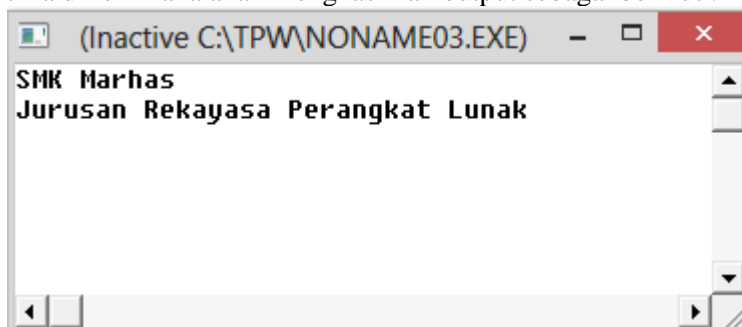
Berikut adalah contoh penulisan procedure tanpa parameter:

```
Procedure cetak_string;  
Begin  
writeln('SMK Marhas');  
writeln('Jurusan Rekayasa Perangkat Lunak');  
end;
```

Berikut adalah cara pemanggilan procedure tanpa parameter di dalam program:

```
Program presdure_sederhana;  
uses wincrt;  
  
{procedure}  
Procedure cetak_string;  
Begin  
writeln('SMK Marhas');  
writeln('Jurusan Rekayasa Perangkat Lunak');  
end;  
  
{program utama}  
begin  
cetak_string;  
end.
```

Jika di run maka akan menghasilkan output sebagai berikut :

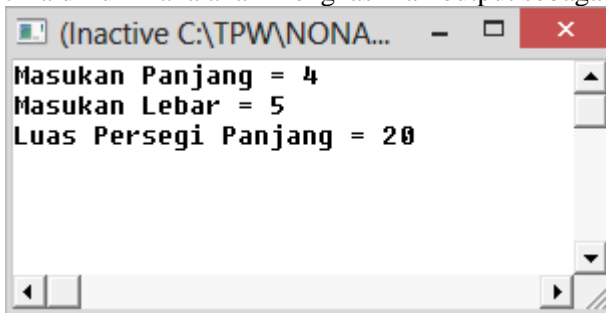


Pada program diatas kita dapat perhatikan bagaimana procedure dipanggil melalui program utama. Sehingga ketika program di jalankan program utama langsung memanggil procedure.

Contoh program procedure dengan parameter :

```
Program persegi_pan_jang;  
uses wincrt;  
var  
p : integer;  
l : integer;  
  
Procedure hitung_luas(panjang,lebar : integer);  
var  
luas : integer;  
  
Begin  
luas := panjang*lebar;  
writeln('Luas Persegi Panjang = ',luas);  
end;  
  
{program utama}  
begin  
write('Masukan Panjang = '); readln(p);  
write('Masukan Lebar = ');readln(l);  
hitung_luas(p,l);  
end.
```

Jika di run maka akan menghasilkan output sebagai berikut :



Agar dapat memahami langkah-langkah pembuatan program seperti di atas, sebagai latihan buatlah program menghitung luas segitiga menggunakan procedure.

BAB VI

Fungsi Dalam Bahasa Pascal

Fungsi adalah program kecil yang berada dalam program utama yang digunakan untuk menyelesaikan masalah khusus dengan paramater yang diberikan. Penggunaan fungsi hamper mirip dengan procedure.

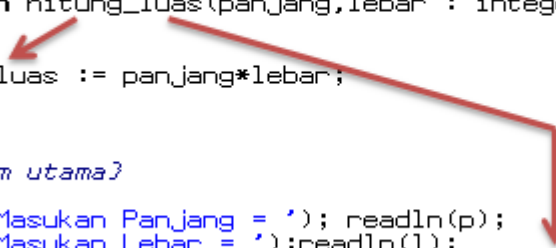
Penulisan Fungsi

```
Function Nama_Fungsi (param_formal:tipe_param,...):tipe_hasil;  
var  
{bagian deklarasi fungsi, sifatnya lokal}  
  
begin  
    statement-1;  
    statement-2;  
  
    Statement-n;  
    Nama_Fungsi:=hasil;  
end;
```

Konsep fungsi di dalam bahasa pascal mirip dengan konsep fungsi di dalam matematika dimana fungsi digunakan untuk menghitung nilai berdasarkan nilai masukan.

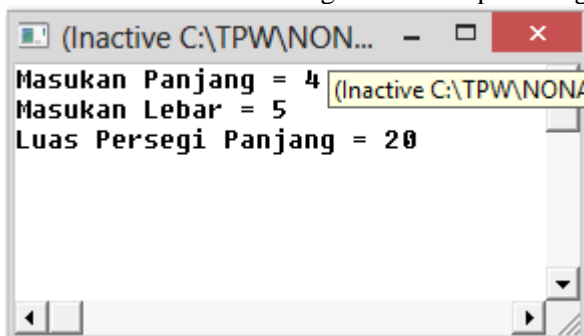
Berikut adalah contoh program menggunakan fungsi :

```
Program persegi_panjang;  
uses wincrt;  
var  
p : integer;  
l : integer;  
  
function hitung_luas (panjang, lebar : integer):integer;  
Begin  
hitung_luas := panjang*lebar;  
end;  
  
{program utama}  
begin  
write('Masukan Panjang = '); readln(p);  
write('Masukan Lebar = '); readln(l);  
writeln('Luas Persegi Panjang = ', hitung_luas(p,l));  
end.
```



Perhatikan bagaimana fungsi ini digunakan.

Jika di run maka akan menghasilkan output sebagai berikut :



Agar dapat memahami langkah-langkah pembuatan program seperti di atas, sebagai latihan buatlah program menghitung luas segitiga menggunakan fungsi.

Refrensi

1. Fathul Wahid.2004. *Dasar-dasar Algoritma dan Pemrograman*.Yogyakarta. Penerbit Andi
2. Rinaldi Munir. 2005. *Algoritma dan Pemrograman Dalam Bahasa Pascal dan C*. Bandung Edisi Ke 3 Buku 1.Bandung.Penerbit Informatika.
3. <http://id.wikipedia.org/wiki/Algoritma>.
4. <http://www.itechsoft.web.id/tipe-operator-dan-ekspresi-dalam-algoritma-pemogramaman.html>
5. <http://iamazzahra.blogspot.com/2009/06/algoritma-dan-struktur-data.html>
6. http://id.wikipedia.org/wiki/Bilangan_bulat
7. http://id.wikipedia.org/wiki/Bilangan_riil
8. <http://id.wikipedia.org/wiki/String>
9. <http://www.iriertools.com/iriepascal/progref357.html>
10. http://www.tutorialspoint.com/pascal/pascal_case_statement.htm
11. http://id.wikibooks.org/wiki/Pemrograman_C/Prosedur_dan_Fungsi

Biografi Penulis



Budi Permana, S.Kom adalah alumni Stmik Mardira Indonesia dari kota Bandung, setelah lulus melanjutkan kuliah selama 6 bulan untuk mengambil Akta IV di UNISBA. Penulis pernah bekerja sebagai teknisi jaringan dan guru di Smk Marhas dan pernah juga bekerja sebagai Konsultan IT.