
Pascal (1)

Bambang Heru S
Departemen Teknik Kimia
Universitas Indonesia

Intro Pascal

- Contoh Program Pascal

```
program Hallo;  
(* Program untuk menampilkan tulisan "Hallo,  
World" di layar *)  
var  
    i:integer;  
begin  
    writeln('Hallo, World');  
end.
```

Intro Pascal

- `program Hallo;`
 - program menandai awal dari program yang bernama "Hallo"

- `(* Program untuk menampilkan tulisan "Hallo, World" di layar *)`
 - Bagian yang berada dalam `(* dan *)` merupakan komentar dan tidak akan dieksekusi pada saat kompilasi maupun runtime. Contoh komentar yang lain : `{ Ini juga komentar}`

Intro Pascal

- `var`
 - Kata kunci `var` untuk penanda variabel apa saja yang akan digunakan dalam program

- `i : integer;`
 - Deklarasi variabel, sebelum dapat dipakai, variabel harus dideklarasikan terlebih dahulu. Cara pendeklarasian dijelaskan kemudian

- `begin`
 - Penanda bagian awal blok algoritma program

Intro Pascal

- `writeln('Hallo, World');`
 - Prosedur untuk menulis ke layar. Prosedur akan dijelaskan kemudian

- `end.`
 - Akhir Blok Algoritma Program

Identifier

- Identifier : nama pengenal variabel, konstanta, nama fungsi, atau prosedur.

- Penulisan identifier harus diawali A..Z atau a..z atau `_`

- Pascal tidak case sensitive

Data

- Nilai yang digunakan dalam program

- Jenis
 - Konstanta
Data yang nilainya didefinisikan sekali dan tidak dapat diubah – ubah
 - Variabel
Data yang nilainya dapat diubah - ubah

Tipe Data

- Tipe Data ialah representasi suatu data dalam komputer. Tiap tipe memiliki beberapa atribut :
 - Nama
 - Integer, float, character , string
 - Domain harga
 - [A..Z, a..z, 0..9]

Tipe Data

- Cara penulisan Literal
 - 233, 'a', "hallo"

- Operator
 - +, -, =, *, /

Data pada Pascal

- Deklarasi Data
 - Konstanta
`Const <nama> = <nilai>`

 - Variabel
`Var <nama> : <tipe>`

Tipe Data pada Pascal

- integer
- real
- boolean
- char
- string

Integer

■ Deklarasi

```
numOfPoint : integer;  
i, j : integer;
```

■ Operator

```
□ +, -, *, /, div, mod, =, <>, <, >
```

Real

- Deklarasi

```
kecepatan : real;  
Suhu : real;
```

- Operator

- +, -, *, /, =, <>, <, >

Boolean

- Deklarasi

```
found : boolean;
```

- Operator

- and, or, not, xor

char

- Deklarasi

```
menu : char;
```

- Operator

- =, <>

string

- Deklarasi

```
nama : string;
```

- Operator

- +, =, <>

Assignment

- Assignment ialah mengisi suatu data dengan nilai
 - Contoh $x = 5$; artinya ialah mengassign 5 sebagai nilai dari variabel x
 - Assignment di Pascal

```
x := 5;
```

Input / Output

- Input
 - Nilai yang dimasukkan user untuk kemudian digunakan di dalam program
 - Dalam Pascal
 - `read(var);`
 - `readln(var);`

Input / Output

- Output

- Nilai yang dihasilkan program dan ditampilkan ke layar

- Dalam Pascal

- write(var);
- writeln(var);

Input

- Read

```
read(x);
```

- Readln

```
readln(x);
```

Output

■ Write

```
write('Hallo');  
Write ('Hallo');  
write('Hallo',nama);
```

■ Writeln

```
writeln('Hallo');  
writeln(nama);  
writeln('Hallo', nama);
```

Input / Output

■ Contoh Program

```
program IO;  
(* program yang menerima masukan nama dan umur lalu menuliskannya ke layar  
*)  
var  
    nama : string;  
    umur : integer;  
begin  
    write('Masukkan nama Anda : ');  
    read(nama);  
    write('Masukkan usia Anda : ');  
    read(umur);  
    writeln ('----Masukan yang didapat----');  
    writeln ('Nama Anda : ', nama);  
    writeln ('Usia Anda : ', umur);  
    writeln ('Tulis dalam satu baris');  
    writeln ('Nama Anda : ', nama, ' - usia Anda : ', umur);  
end.
```

Pascal (2)

Bambang Heru S
Departemen Teknik Kimia
Universitas Indonesia

Percabangan

- Pemilihan blok algoritma yang akan dieksekusi pada saat program berjalan.
 - Karena program tidak mungkin berjalan lurus saja dari atas sampai ke bawah.
-

Percabangan

- Mengevaluasi sebuah ekspresi boolean
- Struktur percabangan
 - If – Then
 - If – Then – Else
 - Case

If – Then

- Pola : if x then y
- Arti : jika kondisi x terpenuhi maka lakukan aksi y

- Pascal

```
if x = 1 then writeln(x);
```

```
if not found then writeln('kosong');
```

If - Then

```
program ifThen;
var
  a : integer;
begin
  write('Masukkan integer : ');
  readln(a);
  if a >= 0 then
    writeln ('masukan positif');
end.
```

If - Then - Else

- Pola : if x then y else z
- Arti : jika kondisi x terpenuhi maka lakukan aksi y jika tidak maka lakukan aksi z

- Pascal

```
if x = 1 then writeln(x) else writeln('x
bukan angka 1');
if not found then writeln('kosong') else
writeln('ketemu');
```

If – Then - Else

```
program ifThenElse;  
var a : integer;  
begin  
  write('Masukkan integer : ');  
  readln(a);  
  if (a>=0) then writeln ('masukan positif')  
  else writeln ('Masukan negatif');  
end.
```

- Perhatikan sebelum else tidak ada titik koma

Case

- Pascal

```
case x of  
  0 : hari := 'Minggu';  
  1 : hari := 'Senin';  
  2 : hari := 'Selasa';  
  3 : hari := 'Rabu';  
  4 : hari := 'Kamis';  
  5 : hari := 'Jumat';  
  6 : hari := 'Sabtu';  
end;
```

Case

- Arti : jika x bernilai 0 maka assign variabel hari dengan nilai “Minggu” jika bernilai 1 maka assign variabel hari dengan nilai “Senin”, dst...
- Penggunaannya sama dengan depend on
- Buat program yang menerima masukan integer[1..7] yang mencetak nama hari sesuai masukan, jika masukan berbeda, maka tampilkan pesan kesalahan

Solusi

```
program conditional;
(* Contoh penggunaan conditional branch *)
var
    angkaHari : integer;

begin
    write ('Masukkan Angka 1..7 : ');
    read (angkaHari);

    case angkaHari of
        1 : writeln('Minggu');
        2 : writeln('Senin');
```


Solusi (Lanjutan)

```
3 : writeln( 'Selasa');
4 : writeln( 'Rabu');
5 : writeln( 'Kamis');
6 : writeln( 'Jumat');
7 : writeln( 'Sabtu');
else writeln('pesan kesalahan');
end;

end.
```

Tambahan...

- Untuk memngevaluasi banyak kasus dapat juga menggunakan if - then – else bersarang (nested)
 - Contoh
- ```
if (angkaHari = 1) then writeln('Minggu')
else if angkaHari = 2 then writeln ('Senin')
else if ...
```

## Blok Instruksi

- Penggunaan Blok instruksi untuk mengelompokkan > 1 aksi.

- Contoh

```
if a >= 0 then
 writeln('Masukan Positif');
 writeln('Karena Masukan lebih dari
0');
```

## Blok Program

- Seharusnya

```
if a >= 0 then
 begin
 writeln('Masukan Positif');
 writeln('Karena Masukan lebih dari
0');
 end;
```

## Perulangan

- Melakukan aksi yang sama berulang – ulang.
- Struktur
  - For – do
  - While
  - Repeat - until

## Perulangan

- Contoh kasus : Faktorial.
- Buat program yang menerima integer(N) dan menghasilkan N!.
- $N! = 1 \rightarrow (N = 1)$
- $N * (N - 1)! \rightarrow$  untuk n lainnya

## For - do

```
program faktorial;
var
 a, i, N : integer;
begin
 write ('Masukkan N : ');
 readln(N);
 a := 1;
 for i:=1 to N do
 a := a * i;
 writeln(N, '! = ', a);
end.
```

## While

```
program Faktorial2;
var
 i, a, N : integer;
begin
 i := 1;
 a := 1;
 write ('Masukkan N : ');
 readln(N);
 while i <= N do
 begin
 a := a * i;
 i := i + 1;
 end;
 writeln(N, '! = ', a);
end.
```

## Repeat - Until

```
program Faktorial3;
var
 i, a, N : integer;
begin
 i := 1;
 a := 1;
 write ('Masukkan N : ');
 readln(N);
 repeat
 a := a * i;
 i := i + 1;
 until i >= N;
 writeln(N, '! = ', a);
end.
```

## Penggunaan for

- Digunakan jika kita ingin melakukan aksi yang sama untuk setiap nilai pencacah. Misalnya untuk menginisialisasi atau untuk mengisi array. Array akan dijelaskan kemudian.

## Penggunaan While

- Struktur ini biasanya digunakan jika kita ingin mencari suatu nilai untuk kemudian digunakan, atau ingin melakukan aksi hanya sampai kondisi tertentu terpenuhi saja, sehingga tidak perlu melakukannya terhadap sisanya. Penggunaan **While – Do** paling sering untuk algoritma search, contohnya pada fungsi yang mengembalikan nilai boolean apakah ada suatu nilai x dalam array.

## Penggunaan Repeat - Until

- Struktur ini penggunaannya hampir sama dengan **While – Do**, tetapi untuk menggunakannya kita harus memastikan bahwa untuk lelaran(pass) pertama kondisi pasti belum dipenuhi. Salah satu penggunaan struktur ini ialah pada desain menu. Menu minimal ditampilkan satu kali pada eksekusi program.

## Pemilihan Struktur Loop

- Pada contoh sebelumnya struktur loop yang tepat ialah For – do. Karena operasi perkalian dilakukan untuk setiap nilai  $i$ , dari 1 sampai  $N$  (nilai masukan).

## Type Data (2)

- Array
  - Data yang mengacu kepada sekumpulan elemen bertipe sama yang dikenali indeksnya
- Record / Tipe Bentukan
  - Tipe bentukan ialah tipe data yang dibentuk dari satu atau lebih tipe data dasar atau tipe record lain yang sudah didefinisikan

## Array

- Deklarasi Array

```
arrInt : array [1..10] of integer;
```

```
arrInt2 : array ['A'..'J'] of integer;
```

## Array

- Cara pengaksesan elemen array

```
arrInt[1] := 2;
```

artinya mengassign 2 ke array arrInt yang indeksinya 1.

```
x := arrInt['C'];
```

artinya mengassign nilai array yang berindex 'c' ke variabel x.



## Contoh Penggunaan Array

```
program cobaarray;
var
 tab : array [1..5] of integer;
 i : integer;
begin
 tab[1] := 5;
 tab[2] := 4;
 tab[3] := 3;
 tab[4] := 2;
 tab[5] := 1;
 write('Masukkan integer[1..5] <- ');
 read (i);
 if (i > 0) and (i <= 5) then writeln ('Isi tab[' , i, ']'
 : ' , tab[i])
 else writeln ('masukan tidak valid');
end.
```

## Contoh Kasus

- Modifikasi program conditional. Gunakan array untuk menyimpan nama hari.
- Hint : Masukan yang diterima (misalnya variabel `indeks`) disimpan sebagai indeks dari array. Lalu cetak array yang berindeks masukan tersebut.

## Solusi

```
program conditional2;
var
 arrHari : array [1..7] of string;
 idx : integer;
begin
 arrHari[1] := 'Minggu';
 arrHari[2] := 'Senin';
 arrHari[3] := 'Selasa';
 arrHari[4] := 'Rabu';
 arrHari[5] := 'Kamis';
 arrHari[6] := 'Jumat';
 arrHari[7] := 'Sabtu';
```

## Solusi(Lanjutan)

```
writeln('Masukkan indeks 1..7 : ');
readln(idx);

writeln ('Hari ini hari ' + arrHari[idx]);

end.
```

## Tipe Bentukan

- Tipe Bentukan
  - Diturunkan dari tipe dasar.
  - Memudahkan pengelolaan data

- Deklarasi

```
type nama = string;
var
 namaDepan, namaBelakang : nama;
```

## Record

- Mahasiswa memiliki nama, NIM, dan nilai. dapat disimpan dalam satu variabel.
- Contoh

```
type mahasiswa = record
 nama : integer;
 NIM : string;
 nilai : char;
end;
```

## Record

### ■ Cara pengaksesan

- Jika `mhs1` adalah variabel yang bertipe `mahasiswa` maka elemennya diakses dengan :

`mhs1.nama`

`mhs1.NIM`

`mhs1.nilai`

## Contoh Program

```
program kuadranpoint;
type point = record
 x : integer; {absis}
 y : integer; {ordinat}
end;
var
 p : point;
 kuadran : integer;
begin
 writeln ('Masukkan absis p');
 readln (p.x);
 writeln ('Masukkan ordinat p');
 readln (p.y);
```

## Contoh Program(Lanjutan)

```
if (p.x > 0) and (p.y > 0) then
 kuadran := 1
else if (p.x < 0) and (p.y > 0) then
 kuadran := 2
else if (p.x < 0) and (p.y < 0) then
 kuadran := 3
else if (p.x > 0) and (p.y < 0) then
 kuadran := 4
else
 kuadran := 0;

writeln ('Point (' , p.x, ', ', p.y, ') berada di
 kuadran ', kuadran);
end.
```

## Pascal (3)

Bambang Heru S  
Departemen Teknik Kimia  
Universitas Indonesia

## Type Data (2)

- **Array**
  - Data yang mengacu kepada sekumpulan elemen bertipe sama yang dikenali indeksnya
- **Record / Tipe Bentuk**
  - Tipe bentuk ialah tipe data yang dibentuk dari satu atau lebih tipe data dasar atau tipe record lain yang sudah didefinisikan

## Array

- **Deklarasi Array**

```
arrInt : array [1..10] of integer;
```

```
arrInt2 : array ['A'..'J'] of integer;
```

## Array

- Cara pengaksesan elemen array

```
arrInt[1] := 2;
```

artinya mengassign 2 ke array arrInt yang indeksinya 1.

```
x := arrInt['C'];
```

artinya mengassign nilai array yang berindex 'c' ke variabel x.

## Contoh Penggunaan Array

```
program cobaarray;
var
 tab : array [1..5] of integer;
 i : integer;
Begin
 for i:=1 to 5 do
 tab[i] := 6-i;
 write('Masukkan integer[1..5] <- ');
 read (i);
 if (i > 0) and (i <= 5) then writeln ('Isi tab[',
 i, '] : ', tab[i])
 else writeln ('masukan tidak valid');
end.
```

## Tipe Bentukan

- Tipe Bentukan
  - Diturunkan dari tipe dasar.
  - Memudahkan pengelolaan data

- Deklarasi

```
type nama = string;
var
 namaDepan, namaBelakang : nama;
```

## Record

- Mahasiswa memiliki nama, NIM, dan nilai. dapat disimpan dalam satu variabel.
- Contoh

```
type mahasiswa = record
 nama : integer;
 NIM : string;
 nilai : char;
end;
```



## Record

### ■ Cara pengaksesan

- Jika `mhs1` adalah variabel yang bertipe `mahasiswa` maka elemennya diakses dengan :

`mhs1.nama`

`mhs1.NIM`

`mhs1.nilai`

## Contoh Program

```
program kuadranpoint;
type point = record
 x : integer; {absis}
 y : integer; {ordinat}
end;
var
 p : point;
 kuadran : integer;
begin
 writeln ('Masukkan absis p');
 readln (p.x);
 writeln ('Masukkan ordinat p');
 readln (p.y);
```

## Contoh Program(Lanjutan)

```
if (p.x > 0) and (p.y > 0) then
 kuadran := 1
else if (p.x < 0) and (p.y > 0) then
 kuadran := 2
else if (p.x < 0) and (p.y < 0) then
 kuadran := 3
else if (p.x > 0) and (p.y < 0) then
 kuadran := 4
else
 kuadran := 0;

writeln ('Point (' , p.x, ', ' , p.y, ') berada di
 kuadran ', kuadran);
end.
```

## SubProgram

- Untuk memecah-mecah program.
- Subprogram
  - Prosedur
  - Fungsi

## Prosedur

- Sederetan instruksi algoritmik yang memiliki nama, dan akan memberikan efek neto yang dapat dilihat dengan membandingkan keadaan awal dan keadaan akhir.

## Prosedur

- Intinya : prosedur ialah kumpulan beberapa aksi untuk mengubah status program menjadi status lainnya.
- Untuk menghemat space jika aksi-aksi itu sering dilakukan.
- Prosedur untuk menuliskan teks ke layar  
`writeln('Hello, World!');`
- Prosedur untuk membaca masukan  
`readln(x);`

## Prosedur

### ■ Contoh

```
procedure tulisIsiArray;
var
 i := integer;
begin
 for i:=1 to N do
 writeln(tab[i]);
end;
```

## Prosedur

### ■ Variabel : Global vs Lokal

- Nilai variabel ditentukan oleh lingkup dimana ia dideklarasikan dan diakses.

## Contoh Program

```
program mahasiswa;
type mhs = record
 nama : string;
 nilai : char;
end;

var
 mhs1 : mhs;
 Chr : char;
```

## Contoh Program

```
procedure inputMhs;
var
 Chr : char;
 Str : string;
begin
 write('Nama : ');
 readln(Str);
 mhs1.nama := Str;
 write('Nilai : ');
 readln(Chr);
 mhs1.nilai := Chr;
end;
```

## Contoh Program

```
procedure cetakMhs;
begin
 writeln ('Nama : ', mhs1.nama);
 writeln ('Nilai : ', mhs1.nilai);
end;

begin
 writeln('Meminta masukan mahasiswa');
 inputMhs;
 writeln('Data yang didapat');
 write('Nilai (sekali lagi) : ');
 readln(Chr);
 cetakMhs;
end.
```

## Prosedur

### ■ Parameter

#### □ Perhatikan kembali

```
Writeln('Hello, World!');
```

#### □ Parameter berguna untuk menghindari kesalahan pada penggunaan variabel global.

## Prosedur

### ■ Contoh

```
procedure cetakMhs(inputmhs : mhs);
begin
 writeln ('Nama : ', mhs.nama);
 writeln ('Nilai : ', mhs.nilai);
end;
```

### ■ Cara memanggil

```
cetakMhs (mhs1);
cetakMhs (mhs2);
```

## Contoh Program

- Modifikasi program mahasiswa. Gunakan prosedur dengan parameter. Lalu tambah beberapa variabel yang bertipe `mhs`

## Prosedur

- Parameter : Formal vs Aktual
  - Parameter formal yaitu parameter yang ditulis pada saat deklarasi.
  - Parameter aktual yaitu parameter yang ditulis ketika memanggil prosedur.

## Prosedur

- Parameter : input, output, input/output
  - Input : diperlukan oleh prosedur untuk berjalan.
  - Output : parameter yang nilainya dihasilkan oleh prosedur.
  - Input/output : diperlukan oleh prosedur kemudian nilainya diubah didalam prosedur lalu nilainya yang berubah itu dikembalikan oleh program.



## Prosedur

- Parameter : pass by value vs pass by reference
  - By value : yang diberikan ke parameter adalah nilai dari tipe parameter.
  - By reference : yang diberikan ke parameter ialah referensi dari tipe parameter

## Prosedur

- Mudahnya
  - By value : nilainya tidak akan berubah meskipun didalam prosedur diubah
  - By reference : nilainya dapat diubah dari dalam prosedur

## Prosedur

- Jadi...
  - Pass by value digunakan untuk parameter input
  - Pass by reference digunakan untuk parameter output dan input/output

## Prosedur

- Pass Parameter di Pascal

```
procedure coba(a : integer);
```

```
procedure coba(var a : integer);
```

## Contoh Program

- Program untuk menukar dua buah nilai integer

```
program proc2;
var
 a, b : integer;

procedure tukar(a, b : integer);
var
 temp : integer;
begin
 temp := a;
 a := b;
 b := temp;
end;
```

## Contoh Program

```
procedure swap(var a, b : integer);
var
 temp : integer;
begin
 temp := a;
 a := b;
 b := temp;
end;
```

## Contoh Program

```
begin
 write('Masukkan nilai a : ');
 readln(a);
 write('Masukkan nilai b : ');
 readln(b);

 writeln('a,b sekarang : ', a, ', ', b);

 tukar(a,b);
 writeln('a,b setelah tukar : ', a, ', ', b);

 swap(a,b);
 writeln('a,b setelah swap : ', a, ', ', b);
end.
```

## Pascal (4)

Bambang Heru S  
Departemen Teknik Kimia  
Universitas Indonesia

## Fungsi

- Sebuah transformasi akibat pemetaan suatu nilai ke nilai lain.
- Sejumlah ekspresi algoritmik yang mengolah masukan menjadi data keluaran yang diinginkan

## Fungsi

- Contoh fungsi
  - `pow(a,b)`
  - `abs(a)`
  - `floor(a)`

## Contoh Fungsi

```
function pow(a, b: integer) : longint;
var
 hasil : longint;
 i : integer
begin
 hasil := 1;
 for i := 1 to b do
 hasil := hasil * a;
 pow := hasil;
end;
```

## Fungsi

### ■ Perhatikan di fungsi

- hanya ada parameter input
- tipe hasil fungsi dituliskan pada saat deklarasi
- pengembalian nilai fungsi dengan operator assignment (:=)
- nilai kembalian harus bertipe dasar
- cara pemanggilan :

```
namaVar := namaFungsi(list_parameter_aktual);
x := pow(2,3);
x := pow(a,b);
```

## Contoh Program

```
program fungsiInt;
var
 x, y : integer;
 hslPangkat : longint;
function pow(a, b: integer) : longint;{tuliskan
 algoritma untuk menghitung perpangkatan disini }
begin
 writeln('2^10 = ', pow(2,10));
 writeln('masukkan integer : ');
 readln(x);
 writeln('masukkan pangkatnya');
 readln(y);
 hslPangkat := pow(x,y);
 writeln(x, '^', y, ' = ', hslPangkat);
end.
```

## Contoh Fungsi(2)

```
function f(x: integer) : integer;
{menghitung $x^2 - 9x + 10$ }
var
 hasil : integer;
begin
 hasil := pow(x,2) - (9*x) + 10;
 f := hasil;
end;
```

## Contoh Kasus

- Tambahkan ke program fungsint sebuah fungsi yang menerima dua buah integer dan mengembalikan nilai yang lebih besar

## Untitled

- Softcopy slide – slide dan modul dapat didownload di
  - <http://students.if.itb.ac..id/~if13052/etc/pascal/>



---

## Pascal (2)

---

Bambang Heru S  
Departemen Teknik Kimia  
Universitas Indonesia

---

## Percabangan

- Pemilihan blok algoritma yang akan dieksekusi pada saat program berjalan.
  - Karena program tidak mungkin berjalan lurus saja dari atas sampai ke bawah.
-

## Percabangan

- Mengevaluasi sebuah ekspresi boolean
- Struktur percabangan
  - If – Then
  - If – Then – Else
  - Case

## If – Then

- Pola : if x then y
- Arti : jika kondisi x terpenuhi maka lakukan aksi y

- Pascal

```
if x = 1 then writeln(x);
```

```
if not found then writeln('kosong');
```

## If - Then

```
program ifThen;
var
 a : integer;
begin
 write('Masukkan integer : ');
 readln(a);
 if a >= 0 then
 writeln ('masukan positif');
end.
```

## If - Then - Else

- Pola : if x then y else z
- Arti : jika kondisi x terpenuhi maka lakukan aksi y jika tidak maka lakukan aksi z

- Pascal

```
if x = 1 then writeln(x) else writeln('x
bukan angka 1');
if not found then writeln('kosong') else
writeln('ketemu');
```

## If – Then - Else

```
program ifThenElse;
var a : integer;
begin
 write('Masukkan integer : ');
 readln(a);
 if (a>=0) then writeln ('masukan positif')
 else writeln ('Masukan negatif');
end.
```

- Perhatikan sebelum else tidak ada titik koma

## Case

- Pascal

```
case x of
 0 : hari := 'Minggu';
 1 : hari := 'Senin';
 2 : hari := 'Selasa';
 3 : hari := 'Rabu';
 4 : hari := 'Kamis';
 5 : hari := 'Jumat';
 6 : hari := 'Sabtu';
end;
```

## Case

- Arti : jika x bernilai 0 maka assign variabel hari dengan nilai “Minggu” jika bernilai 1 maka assign variabel hari dengan nilai “Senin”, dst...
- Penggunaannya sama dengan depend on
- Buat program yang menerima masukan integer[1..7] yang mencetak nama hari sesuai masukan, jika masukan berbeda, maka tampilkan pesan kesalahan

## Solusi

```
program conditional;
(* Contoh penggunaan conditional branch *)
var
 angkaHari : integer;

begin
 write ('Masukkan Angka 1..7 : ');
 read (angkaHari);

 case angkaHari of
 1 : writeln('Minggu');
 2 : writeln('Senin');
```

## Solusi (Lanjutan)

```
3 : writeln('Selasa');
4 : writeln('Rabu');
5 : writeln('Kamis');
6 : writeln('Jumat');
7 : writeln('Sabtu');
else writeln('pesan kesalahan');
end;

end.
```

## Tambahan...

- Untuk memngevaluasi banyak kasus dapat juga menggunakan if - then – else bersarang (nested)
  - Contoh
- ```
if (angkaHari = 1) then writeln('Minggu')
else if angkaHari = 2 then writeln ('Senin')
else if ...
```

Blok Instruksi

- Penggunaan Blok instruksi untuk mengelompokkan > 1 aksi.

- Contoh

```
if a >= 0 then
  writeln('Masukan Positif');
  writeln('Karena Masukan lebih dari
0');
```

Blok Program

- Seharusnya

```
if a >= 0 then
  begin
    writeln('Masukan Positif');
    writeln('Karena Masukan lebih dari
0');
  end;
```

Perulangan

- Melakukan aksi yang sama berulang – ulang.
- Struktur
 - For – do
 - While
 - Repeat - until

Perulangan

- Contoh kasus : Faktorial.
- Buat program yang menerima integer(N) dan menghasilkan N!.
- $N! = 1 \rightarrow (N = 1)$
- $N * (N - 1)! \rightarrow$ untuk n lainnya

For - do

```
program faktorial;
var
  a, i, N : integer;
begin
  write ('Masukkan N : ');
  readln(N);
  a := 1;
  for i:=1 to N do
    a := a * i;
  writeln(N, '! = ', a);
end.
```

While

```
program Faktorial2;
var
  i, a, N : integer;
begin
  i := 1;
  a := 1;
  write ('Masukkan N : ');
  readln(N);
  while i <= N do
  begin
    a := a * i;
    i := i + 1;
  end;
  writeln(N, '! = ', a);
end.
```

Repeat - Until

```
program Faktorial3;
var
  i, a, N : integer;
begin
  i := 1;
  a := 1;
  write ('Masukkan N : ');
  readln(N);
  repeat
    a := a * i;
    i := i + 1;
  until i >= N;
  writeln(N, '! = ', a);
end.
```

Penggunaan for

- Digunakan jika kita ingin melakukan aksi yang sama untuk setiap nilai pencacah. Misalnya untuk menginisialisasi atau untuk mengisi array. Array akan dijelaskan kemudian.

Penggunaan While

- Struktur ini biasanya digunakan jika kita ingin mencari suatu nilai untuk kemudian digunakan, atau ingin melakukan aksi hanya sampai kondisi tertentu terpenuhi saja, sehingga tidak perlu melakukannya terhadap sisanya. Penggunaan **While – Do** paling sering untuk algoritma search, contohnya pada fungsi yang mengembalikan nilai boolean apakah ada suatu nilai x dalam array.

Penggunaan Repeat - Until

- Struktur ini penggunaannya hampir sama dengan **While – Do**, tetapi untuk menggunakannya kita harus memastikan bahwa untuk lelaran(pass) pertama kondisi pasti belum dipenuhi. Salah satu penggunaan struktur ini ialah pada desain menu. Menu minimal ditampilkan satu kali pada eksekusi program.

Pemilihan Struktur Loop

- Pada contoh sebelumnya struktur loop yang tepat ialah For – do. Karena operasi perkalian dilakukan untuk setiap nilai i , dari 1 sampai N (nilai masukan).

Type Data (2)

- Array
 - Data yang mengacu kepada sekumpulan elemen bertipe sama yang dikenali indeksinya
- Record / Tipe Bentukan
 - Tipe bentukan ialah tipe data yang dibentuk dari satu atau lebih tipe data dasar atau tipe record lain yang sudah didefinisikan

Array

- Deklarasi Array

```
arrInt : array [1..10] of integer;
```

```
arrInt2 : array ['A'..'J'] of integer;
```

Array

- Cara pengaksesan elemen array

```
arrInt[1] := 2;
```

artinya mengassign 2 ke array arrInt yang indeksinya 1.

```
x := arrInt['C'];
```

artinya mengassign nilai array yang berindex 'c' ke variabel x.

Contoh Penggunaan Array

```
program cobaarray;
var
  tab : array [1..5] of integer;
  i : integer;
begin
  tab[1] := 5;
  tab[2] := 4;
  tab[3] := 3;
  tab[4] := 2;
  tab[5] := 1;
  write('Masukkan integer[1..5] <- ');
  read (i);
  if (i > 0) and (i <= 5) then writeln ('Isi tab[' , i, ']'
    : ' , tab[i])
  else writeln ('masukan tidak valid');
end.
```

Contoh Kasus

- Modifikasi program conditional. Gunakan array untuk menyimpan nama hari.
- Hint : Masukan yang diterima (misalnya variabel `indeks`) disimpan sebagai indeks dari array. Lalu cetak array yang berindeks masukan tersebut.

Solusi

```
program conditional2;
var
  arrHari : array [1..7] of string;
  idx : integer;
begin
  arrHari[1] := 'Minggu';
  arrHari[2] := 'Senin';
  arrHari[3] := 'Selasa';
  arrHari[4] := 'Rabu';
  arrHari[5] := 'Kamis';
  arrHari[6] := 'Jumat';
  arrHari[7] := 'Sabtu';
```

Solusi(Lanjutan)

```
writeln('Masukkan indeks 1..7 : ');
readln(idx);

writeln ('Hari ini hari ' + arrHari[idx]);

end.
```

Tipe Bentukan

- Tipe Bentukan
 - Diturunkan dari tipe dasar.
 - Memudahkan pengelolaan data

- Deklarasi

```
type nama = string;  
var  
    namaDepan, namaBelakang : nama;
```

Record

- Mahasiswa memiliki nama, NIM, dan nilai. dapat disimpan dalam satu variabel.
- Contoh

```
type mahasiswa = record  
    nama : integer;  
    NIM : string;  
    nilai : char;  
end;
```


Record

■ Cara pengaksesan

- Jika `mhs1` adalah variabel yang bertipe `mahasiswa` maka elemennya diakses dengan :

`mhs1.nama`

`mhs1.NIM`

`mhs1.nilai`

Contoh Program

```
program kuadranpoint;
type point = record
  x : integer; {absis}
  y : integer; {ordinat}
end;
var
  p : point;
  kuadran : integer;
begin
  writeln ('Masukkan absis p');
  readln (p.x);
  writeln ('Masukkan ordinat p');
  readln (p.y);
```

Contoh Program(Lanjutan)

```
if (p.x > 0) and (p.y > 0) then
    kuadran := 1
else if (p.x < 0) and (p.y > 0) then
    kuadran := 2
else if (p.x < 0) and (p.y < 0) then
    kuadran := 3
else if (p.x > 0) and (p.y < 0) then
    kuadran := 4
else
    kuadran := 0;

writeln ('Point (' , p.x, ', ', p.y, ') berada di
    kuadran ', kuadran);
end.
```

Pascal (3)

Bambang Heru S
Departemen Teknik Kimia
Universitas Indonesia

Type Data (2)

- **Array**
 - Data yang mengacu kepada sekumpulan elemen bertipe sama yang dikenali indeksnya
- **Record / Tipe Bentukan**
 - Tipe bentukan ialah tipe data yang dibentuk dari satu atau lebih tipe data dasar atau tipe record lain yang sudah didefinisikan

Array

- **Deklarasi Array**

```
arrInt : array [1..10] of integer;
```

```
arrInt2 : array ['A'..'J'] of integer;
```

Array

- Cara pengaksesan elemen array

```
arrInt[1] := 2;
```

artinya mengassign 2 ke array arrInt yang indeksinya 1.

```
x := arrInt['C'];
```

artinya mengassign nilai array yang berindex 'c' ke variabel x.

Contoh Penggunaan Array

```
program cobaarray;
var
  tab : array [1..5] of integer;
  i : integer;
Begin
  for i:=1 to 5 do
    tab[i] := 6-i;
  write('Masukkan integer[1..5] <- ');
  read (i);
  if (i > 0) and (i <= 5) then writeln ('Isi tab[',
    i, '] : ', tab[i])
  else writeln ('masukan tidak valid');
end.
```

Tipe Bentukan

- Tipe Bentukan
 - Diturunkan dari tipe dasar.
 - Memudahkan pengelolaan data

- Deklarasi

```
type nama = string;  
var  
    namaDepan, namaBelakang : nama;
```

Record

- Mahasiswa memiliki nama, NIM, dan nilai. dapat disimpan dalam satu variabel.
- Contoh

```
type mahasiswa = record  
    nama : integer;  
    NIM : string;  
    nilai : char;  
end;
```

Record

■ Cara pengaksesan

- Jika `mhs1` adalah variabel yang bertipe `mahasiswa` maka elemennya diakses dengan :

`mhs1.nama`

`mhs1.NIM`

`mhs1.nilai`

Contoh Program

```
program kuadranpoint;
type point = record
  x : integer; {absis}
  y : integer; {ordinat}
end;
var
  p : point;
  kuadran : integer;
begin
  writeln ('Masukkan absis p');
  readln (p.x);
  writeln ('Masukkan ordinat p');
  readln (p.y);
```

Contoh Program(Lanjutan)

```
if (p.x > 0) and (p.y > 0) then
    kuadran := 1
else if (p.x < 0) and (p.y > 0) then
    kuadran := 2
else if (p.x < 0) and (p.y < 0) then
    kuadran := 3
else if (p.x > 0) and (p.y < 0) then
    kuadran := 4
else
    kuadran := 0;

writeln ('Point (' , p.x, ', ' , p.y, ') berada di
    kuadran ', kuadran);
end.
```

SubProgram

- Untuk memecah-mecah program.
- Subprogram
 - Prosedur
 - Fungsi

Prosedur

- Sederetan instruksi algoritmik yang memiliki nama, dan akan memberikan efek neto yang dapat dilihat dengan membandingkan keadaan awal dan keadaan akhir.

Prosedur

- Intinya : prosedur ialah kumpulan beberapa aksi untuk mengubah status program menjadi status lainnya.
- Untuk menghemat space jika aksi-aksi itu sering dilakukan.
- Prosedur untuk menuliskan teks ke layar
`writeln('Hello, World!');`
- Prosedur untuk membaca masukan
`readln(x);`

Prosedur

■ Contoh

```
procedure tulisIsiArray;  
var  
    i := integer;  
begin  
    for i:=1 to N do  
        writeln(tab[i]);  
end;
```

Prosedur

■ Variabel : Global vs Lokal

- Nilai variabel ditentukan oleh lingkup dimana ia dideklarasikan dan diakses.

Contoh Program

```
program mahasiswa;  
type mhs = record  
  nama : string;  
  nilai : char;  
end;  
  
var  
  mhs1 : mhs;  
  Chr : char;
```

Contoh Program

```
procedure inputMhs;  
var  
  Chr : char;  
  Str : string;  
begin  
  write('Nama : ');  
  readln(Str);  
  mhs1.nama := Str;  
  write('Nilai : ');  
  readln(Chr);  
  mhs1.nilai := Chr;  
end;
```

Contoh Program

```
procedure cetakMhs;
begin
  writeln ('Nama : ', mhs1.nama);
  writeln ('Nilai : ', mhs1.nilai);
end;

begin
  writeln('Meminta masukan mahasiswa');
  inputMhs;
  writeln('Data yang didapat');
  write('Nilai (sekali lagi) : ');
  readln(Chr);
  cetakMhs;
end.
```

Prosedur

■ Parameter

□ Perhatikan kembali

```
Writeln('Hello, World!');
```

□ Parameter berguna untuk menghindari kesalahan pada penggunaan variabel global.

Prosedur

■ Contoh

```
procedure cetakMhs(inputmhs : mhs);  
begin  
    writeln ('Nama : ', mhs.nama);  
    writeln ('Nilai : ', mhs.nilai);  
end;
```

■ Cara memanggil

```
cetakMhs (mhs1);  
cetakMhs (mhs2);
```

Contoh Program

- Modifikasi program mahasiswa. Gunakan prosedur dengan parameter. Lalu tambah beberapa variabel yang bertipe `mhs`

Prosedur

- Parameter : Formal vs Aktual
 - Parameter formal yaitu parameter yang ditulis pada saat deklarasi.
 - Parameter aktual yaitu parameter yang ditulis ketika memanggil prosedur.

Prosedur

- Parameter : input, output, input/output
 - Input : diperlukan oleh prosedur untuk berjalan.
 - Output : parameter yang nilainya dihasilkan oleh prosedur.
 - Input/output : diperlukan oleh prosedur kemudian nilainya diubah didalam prosedur lalu nilainya yang berubah itu dikembalikan oleh program.

Prosedur

- Parameter : pass by value vs pass by reference
 - By value : yang diberikan ke parameter adalah nilai dari tipe parameter.
 - By reference : yang diberikan ke parameter ialah referensi dari tipe parameter

Prosedur

- Mudahnya
 - By value : nilainya tidak akan berubah meskipun didalam prosedur diubah
 - By reference : nilainya dapat diubah dari dalam prosedur

Prosedur

- Jadi...
 - Pass by value digunakan untuk parameter input
 - Pass by reference digunakan untuk parameter output dan input/output

Prosedur

- Pass Parameter di Pascal

```
procedure coba(a : integer);
```

```
procedure coba(var a : integer);
```

Contoh Program

- Program untuk menukar dua buah nilai integer

```
program proc2;
var
  a, b : integer;

procedure tukar(a, b : integer);
var
  temp : integer;
begin
  temp := a;
  a := b;
  b := temp;
end;
```

Contoh Program

```
procedure swap(var a, b : integer);
var
  temp : integer;
begin
  temp := a;
  a := b;
  b := temp;
end;
```


Contoh Program

```
begin
  write('Masukkan nilai a : ');
  readln(a);
  write('Masukkan nilai b : ');
  readln(b);

  writeln('a,b sekarang : ', a, ', ', b);

  tukar(a,b);
  writeln('a,b setelah tukar : ', a, ', ', b);

  swap(a,b);
  writeln('a,b setelah swap : ', a, ', ', b);
end.
```

Pascal (4)

Bambang Heru S
Departemen Teknik Kimia
Universitas Indonesia

Fungsi

- Sebuah transformasi akibat pemetaan suatu nilai ke nilai lain.
- Sejumlah ekspresi algoritmik yang mengolah masukan menjadi data keluaran yang diinginkan

Fungsi

- Contoh fungsi
 - `pow(a,b)`
 - `abs(a)`
 - `floor(a)`

Contoh Fungsi

```
function pow(a, b: integer) : longint;
var
  hasil : longint;
  i : integer
begin
  hasil := 1;
  for i := 1 to b do
    hasil := hasil * a;
  pow := hasil;
end;
```

Fungsi

■ Perhatikan di fungsi

- hanya ada parameter input
- tipe hasil fungsi dituliskan pada saat deklarasi
- pengembalian nilai fungsi dengan operator assignment (:=)
- nilai kembalian harus bertipe dasar
- cara pemanggilan :

```
namaVar := namaFungsi(list_parameter_aktual);
x := pow(2,3);
x := pow(a,b);
```

Contoh Program

```
program fungsiInt;
var
    x, y : integer;
    hslPangkat : longint;
function pow(a, b: integer) : longint;{tuliskan
    algoritma untuk menghitung perpangkatan disini }
begin
    writeln('2^10 = ', pow(2,10));
    writeln('masukkan integer : ');
    readln(x);
    writeln('masukkan pangkatnya');
    readln(y);
    hslPangkat := pow(x,y);
    writeln(x, '^', y, ' = ', hslPangkat);
end.
```

Contoh Fungsi(2)

```
function f(x: integer) : integer;
{menghitung  $x^2 - 9x + 10$ }
var
    hasil : integer;
begin
    hasil := pow(x,2) - (9*x) + 10;
    f := hasil;
end;
```

Contoh Kasus

- Tambahkan ke program fungsint sebuah fungsi yang menerima dua buah integer dan mengembalikan nilai yang lebih besar

Untitled

- Softcopy slide – slide dan modul dapat didownload di
 - <http://students.if.itb.ac..id/~if13052/etc/pascal/>