

1. Mengetahui Pascal

1.1 Struktur Program 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

1.2 Program Pascal yang Paling Sederhana

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.

Bentuk umum dari bagian pernyataan ini adalah sebagai berikut.

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

Contoh

```
Begin
    Writeln('Saya pascal');
End.
```

1.3 Penulisan Program Pascal

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.

Contoh 1

```
Begin writeln('Saya Pascal');  
Writeln('Saya mulai mengenal Pascal');  
End.
```

Contoh 2

```
Begin  
    writeln('Saya Pascal');  
    Writeln('Saya mulai mengenal Pascal');  
End.
```

1.4 Judul Program

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).

Contoh

```
Program Contoh_awal;  
Begin  
    Writeln('Saya Pascal');  
    Writeln('Saya baru kenal');  
End.
```

Judul program sifatnya sebagai dokumentasi saja, tidak berkaitan dengan proses program.

1.5 Bagian Deklarasi

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.

Contoh

```
Program Contoh_Konstanta;  
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 = 2.00000000E-01  
Nama      = PT Maju
```

Turbo Pascal mempunyai beberapa konstanta yang tidak perlu didefinisikan dapat dipergunakan langsung, diantaranya yaitu:

False dengan tipe Boolean, bernilai logika salah
True dengan tipe Boolean, bernilai logika benar
MaxInt dengan tipe numerik integer, bernilai 32767
MaxLongInt dengan tipe numerik integer, bernilai 2147483647

Contoh

```
Program Contoh_Konstanta_Terdefinisi;  
Begin  
    Writeln('Nilai logika benar= ';true);  
    Writeln('Nilai logika salah= ';false);  
    Writeln('Nilai MaxInt      = ';MaxInt);  
    Writeln('Nilai MaxLongInt   = ';MaxLongInt);  
End.
```

Bila program dijalankan akan menghasilkan:

```
Nilai logika benar      = TRUE  
Nilai logika salah     = FALSE  
Nilai MaxInt           = 32767  
Nilai MaxLongInt      = 2147483647
```

Contoh deklarasi konstanta yang dapat dilakukan dengan benar:

```
Const a=8;
Const c='8';
Const s='konstanta';
Const Is=sizeof(integer);
Const b=length('konstanta');
Const x:integer = 10;
Const z=ord(#64);
const a= 10; s= chr (a);
```

Contoh deklarasi konstanta yang **salah**:

```
const a=integer:=10;
const a(integer):=10;
const integer(a):=10;
const integer:a= 10;
const x.1=2.7182818;
const a= 10'; s= length (a);
const a= 10'; s= pos ('1' , a);
const a= 10; s= high (a);
```

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.

Contoh

```
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
```

Pada contoh di atas terdapat 2 konstanta dan 1 variabel, yaitu **Luas** dengan tipenya datanya integer.

c. Deklarasi Prosedur

Prosedur merupakan bagaian yang terpisah dari program dan dapat diaktifkan di manapun di dalam program. Prosedur dapat berupa prosedur standar (sudah disediakan oleh Pascal) maupun prosedur yang akan dibuat sendiri. Prosedur dibuat sendiri bilamana program akan dibagi-bagi menjadi beberapa modul-modul. Prosedur dibuat di dalam program dengan cara mendeklarasikannya di bagian deklarasi prosedur. Kata cangan **Procedure** digunakan sebagai judul dari bagian deklarasi prosedur, diikuti oleh pengenalan yang merupakan nama prosedurnya dan secara opsional dapat diikuti lagi oleh kumpulan parameter yang diakhiri dengan titik koma.

Contoh

```
Program prosedur;
PROCEDURE HITUNG (X, Y: INTEGER;      VAR HASIL: INTEGER);
BEGIN
    HASIL := X + Y;
END;
VAR
    Z: INTEGER;

{program utama}
BEGIN
    HITUNG (2, 3, Z);
    WRITELN ('2 + 3 = ', Z);
END.
```

Bila program dijalankan akan menghasilkan:

2 + 3 = 5

Lebih lanjut akan dibahas khusus pada materi prosedur.

d. Deklarasi Fungsi

Fungsi juga merupakan bagian program yang terpisah mirip dengan prosedur, tetapi ada beberapa perbedaannya. Fungsi dapat berupa fungsi standar (sudah tersedia dalam Pascal) atau fungsi dibuat sendiri. Bila pemakai membuat sendiri, maka harus dideklarasikan terlebih dahulu. Kata cadangan **Function** mengawali bagian deklarasi fungsi diikuti oleh pengenalan yang merupakan nama fungsinya dan secara opsional dapat diikuti oleh kumpulan parameter, tipe dari fungsinya dan diakhiri dengan titik koma.

Contoh

```
Program Fungsi;
FUNCTION TAMBAH (X, Y: INTEGER) : INTEGER;
BEGIN
    TAMBAH := X + Y;
END;

{program utama}
BEGIN
    WRITELN ('2 + 3 = ', TAMBAH (2, 3));
END.
```

Jika program dijalankan akan mengasilkan:

2 + 3 = 5

Lebih lanjut akan dibahas khusus pada materi fungsi.

2. Elemen-Elemen Program Pascal

Untuk mempelajari suatu bahasa komputer guna dapat membuat program dengan benar dan dapat mengembangkannya, langkah pertama yang harus kita ketahui terlebih dahulu adalah mengenal struktur dari program yang akan dibuat dengan bahasa tersebut. Selanjutnya mengetahui elemen-elemen yang membentuk program tersebut.

Elemen-elemen program Pascal sebagai berikut.

1. Simbol-simbol dasar
2. Kata-kata cadangan
3. Pengenal didefinisikan oleh pemakai
4. Data
5. Karakter kontrol
6. Tanda Operasi
7. Komentar Program
8. Pernyataan
9. Prosedur
10. Fungsi

2.1 Simbol-Simbol Dasar

Program Pascal dapat dibentuk dari simbol-simbol yang terdiri dari huruf-huruf, angka-angka dan simbol-simbol khusus.

1. Huruf

Huruf yang digunakan adalah huruf A sampai dengan Z, a sampai dengan z dan garis bawah _ (garis bawah). Huruf besar dan huruf kecil dianggap sama, tidak dibedakan.

2. Angka

Angka yang digunakan adalah angka 0 1 2 3 4 5 6 7 8 9

3. Simbol khusus

Simbol khusus yang dapat digunbakan adalah: + - * / = ^ () [] {} . , ; ' # \$ < = > = := .. (* *) (.)

2.2. Kata Cadangan

Kata cadangan adalah kata-kata yang sudah didefinisikan oleh Pascal yang mempunyai maksud tertentu. Kata-kata cadangan tidak boleh didefinisikan ulang oleh pemakai, sehingga tidak dapat dipergunakan sebagai pengenal.

Contoh kata cadangan:

And	Array	Begin	Case	Cosnt
Div	Do	Downto	Else	End
File	For	Forward	Function	Goto
If	In	Label	Mod	Nil
Not	Of	Or	Procedure	Program
Record	Repeat	Set	Shl	Shr
String	Then	To	Type	Until
Uses	Var	While	With	Xor

2.3 Pengenal (*Identifier*) Didefinisikan Pemakai

Nama yang dipergunakan di dalam program Pascal disebut dengan pengenal. Pemakai dapat mendefinisikan sendiri suatu nama sebagai pengenal untuk tipe, pengenal konstanta, pengenal variabel, pengenal prosedur, pengenal fungsi, pengenal unit, pengenal program dan pengenal record.

Pengenal yang didefinisikan sendiri bebas, tetapi harus mengikuti ketentuan sebagai berikut.

1. Terdiri dari gabungan huruf dan angka dengan karakter pertama harus berupa huruf. Huruf besar dan huruf kecil dianggap sama.
2. Tidak boleh mengandung spasi kosong.
3. Tidak boleh mengandung simbol-simbol khusus, kecuali garis bawah.
4. Panjangnya bebas, tetapi hanya 63 karakter pertama yang dianggap signifikan.

Contoh

Pengenal	Keterangan
NamaSiswa	Benar
No_Induk	Benar
P3K	Benar
1X	Salah, karakter pertama harus huruf
A&B	Salah, tidak boleh mengandung simbol khusus
A B	Salah, tidak boleh ada spasi kosong

2.4 Tipe Data

Kalau kita menggunakan pengenal variabel di dalam program, maka variabel tersebut harus kita deklarasikan terlebih dahulu. Pada waktu kita mendeklarasikan suatu variabel, maka kita harus menentukan tipe dari datanya. Tipe data data ini menunjukkan suatu nilai yang dapat digunakan oleh variabel bersangkutan.

1. Tipe data numerik integer

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

Contoh:

Misalnya kita akan menggunakan suatu pengenal variabel yang bernama Jumlah dan nilai akan dikandungnya tidak lebih dari 255, maka dapat digunakan tipe data byte. Oleh karena itu digunakan suatu variabel, maka pengenal variabel ini harus dideklarasikan terlebih dahulu, sebagai berikut.

```
Var
    Jumlah:byte;
Begin
    Jumlah:=200;
    Writeln('Nilai Jumlah=',Jumlah);
End.
```

2. Tipe data numerik real

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

Contoh

Milsanya kita akan membuat pengenalan variabel yang bernama Luas dan akan bernilai desimal atau ada pecahannya. Seperti Luas segi tiga atau ada proses matematika yang bernilai bilangan real.

Contoh

```
Program Luas_Segi_Tiga;
Var
  Luas:real;
  Alas,tinggi:byte;
Begin
  Alas:=5;
  Tinggi:=10;
  Luas:=0.5*Alas*Tinggi;
  Writeln('Alas  =',Alas);
  Writeln('Tinggi=',Tinggi);
  Writeln('Luas  =,'Luas);
End.
```

Bila program dijalankan akan menghasilkan

```
Alas  =5
Tinggi=10
Luas  =2.5E+1
```

3. 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**.

Contoh:

```
Var
  Karakter:char;
Begin
  Karakter:='A';
  Writeln('Karakter tersebut=',karakter);
End.
```

Bila program dijalankan

```
Karakter tersebut=A
```

4. Tipe data string

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

suatu string di dalam deklarasi variabel tidak disebutkan, maka dianggap panjangnya adalah 255 karakter.

Contoh:

```
Var
  Tanggal:string[10];
  Hari   :string[6];
  Pasaran:string[6];
Begin
  Tanggal:='26-12-2005';
  Hari   :='Senin';
  Pasaran:='Pahing';
  Writeln(Tanggal,' ',hari,' ',Pasaran);
End.
```

Bila program dijalankan

26-12-2005 Senin Pahing

5. 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 kedalam bentuk aritmatika.**

Contoh

```
Var
  Lagi:boolean;
Begin
  Lagi:=True;
  Writeln('Nilai lagi=',lagi);
End.
```

Bila program dijalankan

Nilai lagi=TRUE

6. Tipe data pointer

Suatu pointer adalah suatu variabel yang berisi alamat di memori. Dengan kata lain, pointer akan menunjukkan letak dari data di memori.

Contoh

```
Type
  Tipestring=string[40];
  Pointerstring:^Tipestring;
Var
  Letaknama:pinterstring;
Begin
  Letaknama^:='Pascal';
  Writeln(Letaknama^);
End.
```

Pada deklarasi variabel, dideklarasikan sebuah pengenalan variabel, yaitu `LetakNama`, Variabel ini adalah variabel pointer yang menunjukkan letak data di memori sebanyak 40 karakter. Jadi variabel `LetakNama` ini tidak berisi dengan datanya, tetapi berisi dengan pointer letak dari data di memori. Untuk meletakkan nilai ke memori yang letaknya ditunjukkan oleh variabel `LetakNama` dapat dilakukan sebagai berikut.

```
LetakNama^:='Pascal';
```

7. Tipe data ordinal

Tipe data ordinal adalah tipe data yang merupakan subset dari tipe data sederhana. Yang dimaksud tipe data sederhana sebenarnya adalah semua tipe data sederhana kecuali tipe data numerik real. Jadi yang termasuk tipe data ordinal adalah:

- a. Tipe data standar (`Inetgerm`, `ShortInt`, `LongInt`, `Byte`, `Word`, `Boolean`, `Char`)
- b. Tipe data yang didefinisikan sendiri oleh pemakai.

Tipe data ordinal mempunyai 4 karakteristik sebagai berikut.

1. Nilai dari tipe data ordinal mempunyai nilai yang berurutan.
2. Fungsi standar ***Ord*** dapat digunakan untuk menghasilkan nilai urutannya.
3. Fungsi standar ***Pred*** dapat digunakan untuk menghasilkan nilai urutan sebelumnya.
4. Fungsi standar ***Succ*** dapat digunakan untuk menghasilkan nilai urutan sesudahnya.

Contoh

```
Begin
  Writeln(Ord('A'));
  Writeln(Pred('B'));
  Writeln(Succ('C'));
End.
```

Bila program dijalankan

```
65
A
D
```

2.5 Karakter Kontrol

Turbo Pascal memungkinkan karakter-karakter kontrol untuk dilekatkan di dalam suatu string. Dua buah notasi digunakan untuk karakter kontrol, yaitu:

1. Simbol `#` yang diikuti oleh suatu nilai integer diantara 0 sampai dengan 255, baik berupa nilai desimal maupun hexadesimal untuk menunjukkan suatu karakter yang dihubungkan dengan kode ASCII.

Contoh:

`#65` (ASCII 65 adalah karakter A)

2. Simbol `^` diikuti oleh suatu karakter, menunjukkan hubungannya dengan karakter kontrol.

Contoh

`^G` (Control-G berarti bel, sama dengan ASCII 7)

2.6 Tanda Operasi

Tanda operasi di dalam bahasa Pascal dikelompokkan ke dalam 9 kategori, yaitu:

1. Assignment operator
2. Binary operator
3. Unary operator
4. Bitwise operator
5. Relational operator
6. Logical operator
7. Address operator
8. Set operator
9. String operator

2.6.1 Assignment operator

Assignment operator adalah operator pengerjaan, operator ini menggunakan simbol titik dua diikuti oleh tanda sama dengan (:=).

Contoh

```
Luas:=panjang*lebar;  
Kota:='Kuningan';
```

2.6.2 Binary operator

Binary operator adalah tanda operasi biner, karena operator ini digunakan untuk mengoperasikan dua buah operand. Operand dapat berbentuk konstanta ataupun variabel. Operator ini digunakan untuk operasi aritmatika yang berhubungan dengan nilai tipe data *integer* dan *real*.

Tabel Tanda Operasi Biner

Operator	Operasi	Tipe Operand	Tipe Hasil
*	Perkalian	real, real integer, integer real, integer	real integer real
DIV	Pembagian bulat	integer, integer	Integer
/	Pembagian real	real, real integer, integer real, integer	real real real
MOD	Sisa pembagian	integer, integer	integer
+	Penambahan	real, real integer, integer real, integer	real integer real
-	Pengurangan	real, real integer, integer real, integer	real integer real

Contoh

```
Begin  
  Writeln(15*5);  
  Writeln(23/3);  
  Writeln(20 Div 3);  
  Writeln(20 Mod 3);  
End.
```

Bila program dijalankan

```
75  
6.6666666667E+00  
6  
2
```

Operator **DIV** digunakan untuk pembagian dengan hasil yang dibulatkan ke bawah atau menghilangkan angka pecahannya.

2.6.3 Unary operator

Unary operator adalah operator yang hanya menggunakan sebuah operand saja, yaitu tanda plus atau minus. Tanda ini digunakan diawal angka.

2.6.4 Bitwise operator

Bitwise operator digunakan untuk operasi bit per bit pada nilai binari (biner).

Acuan bilangan desimal ke bilangan biner sebagai berikut.

desimal	biner
0=0	0
1=2 ⁰	1
2=2 ¹	10
4=2 ²	100
8=2 ³	1000
16=2 ⁴	10000
32=2 ⁵	100000
dst...	dst...

Pangkat pada desimal merupakan jumlah nol pada biner.

Contoh

desimal	biner
25= 16	10000
8	1000
1	1
	10001

1. Operator NOT

Operator **Not** digunakan untuk pembalikan bitwise, yaitu nilai bit 1 menjadi bit 0 dan sebaliknya nilai bit 0 menjadi bit 1.

Contoh

```
Begin
  Writeln(not 0);
  Writeln(not 5);
  Writeln(not 18);
  Writeln(not -17);
End.
```

Bila program dijalankan

```
-1
-6
-19
16
```

2. Operator And

Operator **And** digunakan untuk membandingkan dua buah elemen, hasilnya akan benar bila keduanya benar. Bila logika benar diberi simbol 1 dan logika salah diberi simbol 0, maka hubungan operator And tampak pada tabel berikut.

Tabel Hubungan operator And

A	B	A And B
1	1	1
1	0	0
0	1	0
0	0	0

Contoh

```
Begin
  Writeln(12 And 23);
End.
```

Bila program dijalankan akan menghasilkan

4

Operator **And** bekerja dengan membandingkan bit demi bit dari elemen-elemen yang dibandingkan, sebagai berikut.

12 nilai binarinya adalah **01100**
23 nilai binarinya adalah **10111**
12 AND 23 bernilai **00100** adalah **4** (2^2)

3. Operator Or

Operator **Or** digunakan untuk membandingkan dua buah elemen, hasilnya akan benar bila salah satu atau keduanya benar. Bila logika benar diberi simbol 1 dan logika salah diberi simbol 0, maka hubungan operator Or tampak pada tabel berikut.

Tabel Hubungan operator Or

A	B	A Or B
1	1	1
1	0	1
0	1	1
0	0	0

Contoh

```
Begin
  Writeln(12 Or 23);
End.
```

Bila program dijalankan akan menghasilkan

31

Ungkapan 12 Or 23 didapat dari:

12 nilai binarinya 1100
23 nilai binarinya 10111
12 Or 23 bernilai **11111** adalah **31** ($2^4+2^3+2^2+2^1+1$)= $(16+8+4+2+1)$

4. Operator Xor

Operator **Xor** digunakan untuk membandingkan dua buah elemen, hasilnya akan benar bila salah satunya saja yang benar. Bila logika benar diberi simbol 1 dan logika salah diberi simbol 0, maka hubungan operator Xor tampak pada tabel berikut.

Tabel Hubungan operator Xor

A	B	A Or B
1	1	0
1	0	1
0	1	1
0	0	0

Contoh

```
Begin
  Writeln(12 XOR 23);
End.
```

Bila program dijalankan akan menghasilkan

27

Ungkapan 12 XOR 23 bernilai 27 didapat dari:

12 nilai binarinya 1100

23 nilai binarinya 10111

12 Or 23 bernilai **11011** adalah **27** ($2^4+2^3+2^1+1$)=(16+8+2+1)

5. Operator Shl

Operator Shl digunakan untuk menggeser sejumlah bit ke kiri dengan nilai bit 0 atau menambah jumlah nol (0) ke kanan.

Contoh

```
Begin
  Writeln(5 shl 2);
End.
```

Bila program dijalankan akan menghasilkan

20

Nilai integer digeser sebanyak 2 bit ke kiri dengan nilai bit 0 didapatkan hasil 20 sebagai berikut.

5 nilai binarinya adalah 101

Digeser 2 bit ke kiri (ditambah nolai 0 sebanyak 2 buah) menjadi 101**00**=20

6. Operator Shr

Operator Shr digunakan untuk menggeser sejumlah bit ke kanan dengan nilai bit 0 atau membuang angka paling kanan.

Contoh

```
Begin
  Writeln(20 shr 2);
End.
```

Bila program dijalankan akan menghasilkan

5

20 nilai binarinya adalah 10100

digeser 2 bit ke kanan (dibuang 2 bit dari kanan) menjadi 101=5

7. Operator Hubungan

Operator hubungan digunakan untuk membandingkan hubungan antara dua buah operand dan akan didapatkan hasil tipe boolean yaitu *true* atau *false*.

Operator	Operasi
=	sama dengan
<>	tidak sama dengan
>	lebih dari
>=	lebih dari sama dengan
<	kurang dari
<=	kurang dari sama dengan
IN	seleksi dari anggota himpunan

3. Input dan Output

Setiap program aplikasi pasti membutuhkan operasi *input* dan *output*. Memasukkan data input secara interaktif merupakan ciri dari program-program aplikasi yang banyak ditemukan. Suatu program yang tidak dapat menghasilkan output, maka proses program tersebut akan sia-sia, karena hasilnya tidak akan dapat dinikmati.

3.1 Memasukkan Data

Cara yang paling banyak digunakan untuk memasukkan data adalah dengan cara mengetikannya lewat papan ketik. Turbo Pascal menyediakan prosedur untuk maksud tersebut, yaitu prosedur standar **Read** dan **Readln**. Prosedur standar **Read** dan **Readln** mempunyai aturan tertentu untuk beberapa tipe pengenalan variabel. Tiap-tiap data yang dimasukkan harus sesuai dengan tipe variabelnya.

Tipe variabel dan data yang diizinkan.

Tipe Variabel	Keterangan
Char	Memasukkan sebuah karakter, bila lebih akan terpotong, yang dianggap hanya yang pertama.
String	Memasukkan string maksimum sepanjang yang didefinisikan.
Integer	Memasukkan data numerik bulat mulai -32768 sampai dengan 32767
Word	Memasukkan data numerik bulat mulai 0 sampai dengan 65535
Byte	Memasukkan data numerik bulat mulai 0 sampai dengan 255
ShortInt	Memasukkan data numerik bulat mulai -128 sampai dengan 127
LongInt	Memasukkan data numerik bulat mulai -2147483648 sampai dengan 2147483647
Real	Memasukkan data numerik real maksimum 30 digit dapat tanpa titik desimal.
Boolean	Data tipe ini tidak diijinkan

Contoh

```
Program Masuk;
Var
  A,B : Integer;
  C    : Real;
  D    : String[10];
  E    : Char;
Begin
  {pernyataan memasukan data}
  Write('Data A=');Readln(A);
  Write('Data B=');Readln(B);
  Write('Data C=');Readln(C);
  Write('Data D=');Readln(D);
  Write('Data E=');Readln(E);
```

```

    {Pernyataan menampilkan data}
    Writeln('Data A=',A);
    Writeln('Data B=',B);
    Writeln('Data C=',C);
    Writeln('Data D=',D);
    Writeln('Data E=',E);
End.

```

Bila program dijalankan, pertama kali menunggu memasukkan data mulai dari data A sampai dengan E. Setelah semua diisi, maka akan menampilkan data yang dimasukkan tadi. Prosedur **Readln** untuk memasukkan data perbaris, artinya setelah mengetikkan data harus diakhiri dengan menekan **ENTER**, maka akan ganti baris.

3.2 Menampilkan Hasil

Untuk menampilkan hasil dengan bahasa Pascal digunakan prosedur standar **Write** dan **Writeln**. Perbedaannya adalah, prosedur **Write** menampilkan hasil tanpa ganti baris dan tampilan berikutnya akan disambung dalam baris yang sama. Sedangkan prosedur standar **Writeln** digunakan untuk menampilkan tampilan perbaris, akan ganti baris untuk tampilan berikutnya.

Contoh:

```

Program Tampil;
Var
    Nama:string[10];
Begin
    Nama:='Nina';
    Write('Nama : ');
    Write(Nama);
    Writeln;
    Writeln(Nama: ');
    Writeln(Nama);
End.

```

Bila program dijalankan akan menghasilkan

Nama:Nina -----> efek dari write

Nama:
Nina -----> efek dari writeln

3.3 Tampilan Terformat

Walaupun bentuk standar dianggap cukup untuk menampilkan bentuk pada program yang sederhana, tetapi untuk program-program aplikasi dibutuhkan bentuk format tertentu yang harus diatur kembali. Untuk mengatur format tampilan dengan prosedur standar **write** atau **writeln** dipergunakan parameter-parameter sebagai berikut.

1. Parameter CH:N

Digunakan untuk membentuk ornat tampilan **Char** seleber n karakter dengan spasi kosong dimuka sebanyak n-1.

Contoh

```
Program tampil_fomat_1;
Var
  Ket1, Ket2: Char;
Begin
  Ket1:= 'A';
  Ket2:= 'B';
  Writeln(Ket1:5, Ket2:3);
End.
```

Bila Program dijalankan akan menghasilkan:

```
  A  B
----5--3  karakter
```

2. Parameter S:N

Parameter ini digunakan untuk membentuk format tampilan string dengan lebar n akarakter.

Contoh:

```
Program Tampil_format_2;
Var
  Tulisan:string[6];
Begin
  Tulisan:='Pascal';
  Writeln('Bahasa:', Karakter:10);
End.
```

Bila Program dijalankan akan menghasilkan:

```
Bahasa:      Pascal
----- 10 karakter, 4 spasi kosong di muka
```

3. Parameter I:N

Parameter ini digunakan untuk membentuk format tampilan *integer* dengan lebar n digit.

Contoh:

```
Program tampil_format_3;
Var
  N: Integer;
Begin
  N:=275;
  Writeln('Jumlah Data=', N:5);
End.
```

Bila program dijalankan akan menghasilkan:

```
Jumlah Data=  275
-----5 digit, 2 digit spasi kosong di muka
```

4. Tampilan R:N:M

Parameter ini digunakan untuk membentuk format tampilan nilai numerik *Real* dengan lebar n digit rata sebelah kanan dan m digit angka di belakang koma tidak dalam bentuk eksponen.

Contoh:

```
Program Tampil_format_4;  
Var  
  Nilai:Real;  
Begin  
  Nilai:=123.456;  
  Writeln('Nilai Real=',Nilai:0:0);  
  Writeln('Nilai Real=',Nilai:3:0);  
  Writeln('Nilai Real=',Nilai:3:2);  
  Writeln('Nilai Real=',Nilai:7:1);  
End.
```

Bila program dijalankan akan menghasilkan:

```
123  
123  
123.46  
 123.5
```

4. Perulangan

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**.

4.1 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 tersarang.

1. Perulangan positif.

Perulangan positif adalah perulangan dengan penghitung dari kecil ke besar atau penambahan positif. Perulangan positif dapat dibentuk dengan menggunakan pernyataan **For-To-Do**, dengan bentuk umum:

For variabel-kontrol:=nilai awal **To** nilai akhir **Do** pernyataan

Variabel kontrol, nilai awal, nilai akhir harus betipe integer

Contoh

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

Bila program dijalankan akan menghasilkan:

```
Pascal
Pascal
Pascal
Pascal
Pascal
```

Penjelasan:

Pernyataan **Writeln('Pascal')** akan diulang sebanyak 5 kali, yaitu dengan penghitung dari nilai awal 1 sampai dengan nilai akhir 5.

Apabila pernyataan diulang lebih dari satu pernyataan maka setelah **DO** harus memakai **Begin** kemudian beberapa pernyataan yang akan diulang dan diakhiri dengan **End;**.

Contoh:

```
Var
  I:Integer;
Begin
  For I:= 1 to 2 Do
  Begin
    Write('Pascal');
    Writeln('Turbo');
  End;
End.
```

Bila program dijalankan akan menghasilkan:

```
PascalTurbo  
PascalTurbo
```

2. Perulangan negatif

Perulangan negatif adalah perulangan dengan penghitung dari besar ke kecil atau penambahan negatif. Perulangan negatif dapat dibentuk dengan menggunakan pernyataan **For-DownTo-Do**, dengan bentuk umum:

For variabel-kontrol:=nilai awal **DownTo** nilai akhir **Do** pernyataan

Contoh:

```
Var  
  I:integer;  
Begin  
  For I:=5 downto 1 do  
    Begin  
      Writeln(I);  
    End;  
  End.
```

Bila program dijalankan maka akan menghasilkan:

```
5  
4  
3  
2  
1
```

3. Perulangan tersarang

Perulangan tersarang adalah perulangan yang berbeda di dalam perulangan yang lainnya. Perulangan yang lebih dalam akan diproses terlebih dahulu sampai habis, kemudian perulangan yang lebih luar baru akan bertambah, mengerjakan perulangan yang lebih dalam lagi mulai dari nilai awalnya dan seterusnya.

Contoh:

```
Var  
  I, J: Integer;  
Begin  
  For I:=1 to 5 Do  
    Begin  
      For j:=1 to 3 Do  
        Begin  
          Write(I:8, J:3);  
        End;  
      Writeln;  
    End;  
  End.
```

Bila program dijalankan akan menghasilkan:

```
1 1      1 2      1 3
2 1      2 2      2 3
3 1      3 2      3 3
4 1      4 2      4 3
5 1      5 2      5 3
```

Perulangan tersarang banyak digunakan dalam aplikasi matrik dengan menggunakan variabel tipe larik. (Lebih lanjut akan di bahas pada materi matrik)

4.2 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.
```

Bila program dijalankan akan menghasilkan:

```
0
1
2
3
4
```

Penjelasan:

Perulangan dari while akan terus menerus dikerjakan bila kondisinya masih benar. Dalam hal ini kondisinya adalah I dan bila nilai I kurang dari 5, berarti kondisi di dalam While masih terpenuhi dan perulangan akan selesai setelah nilai I lebih besar atau sama dengan 5.

4.3 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**

Contoh:

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

Bila program dijalankan akan menghasilkan:

```
1
2
3
4
5
```

Penjelasan:

Pencetakan variabel I akan diulang-ulang sampai harga I bernilai 5.

5. Penyeleksian Kondisi

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

5.1 Pernyataan 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.

Contoh:

```
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.
```

Bila program dijalankan akan menghasilkan:

```
Nilai Ujian = 70-----→ diisi melalui papan ketik
Lulus
```

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

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

Contoh:

```
Var
    Nilai:Real;
Begin
    Writeln('Nilai Ujian = ');
    Readln(Nilai);

    If Nilai>60 Then
        Writeln('Lulus');
    Else
        Writeln('Tidak Lulus');
End.
```

5.2 Pernyataan 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-pernyataan yang lainnya dalam lingkungan **Case** tidak akan diseleksi lagi.

Bentuk dari struktur **Case—Of** adalah:

```
Case ungkapan Of
    daftar case-label1:pernyataan1;
    daftar case-label2:pernyataan2;
    daftar case-label3:pernyataan3;
    daftar case-label4:pernyataan4;
    .
    .
    daftar case-labeln:pernyataann;
End;
```

Daftar *case label* (label permasalahan) dapat berupa konstanta, atau range (himpunan) dari konstanta yang bukan bertipe real.

Contoh:

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.

```

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.
  
```

Struktur **Case—Of...Else** merupakan pengembangan dari struktur **Case—Of** dan juga merupakan pengembangan dari standar Pascal. Pada struktur **Case—Of** bila tidak ada kondisi yang terpenuhi berarti tidak ada pernyataan di lingkungan **Case—Of** yang diproses. Dengan struktur **Case—Of...Else**, bila tidak ada kondisi yang terpenuhi, maka pernyataan yang akan diproses di dalam lingkungan **Case—Of** adalah pernyataan yang ada di **Else**.

Contoh:

Pembelian yang mencapai nilai tertentu akan mendapatkan potongan pembelian yang besarnya disesuaikan dengan tabel berikut ini.

Total Pembelian (Rp)	Potongan Rp
0-1000	100
1001-5000	200
>5000	500

Akan dihitung besarnya potongan dan nilai yang harus dibayar untuk masing-masing penjualan. Programnya adalah:

```

Var
    Bersih,Beli,potong:Integer;
Begin
    Writeln('Total Pembelian = '); Readln(beli);

    Case Beli of
        0..1000:potong:=100;
        1001..5000:potong:=200;
    Else
        potong:=500;
    End;

    Bersih:=beli-ptong;
    Writeln('Potongan      = ', potongan);
    Writeln('Jumlah Bersih = ', Bersih);
End.
  
```

6. Operasi String

Suatu string di dalam Pascal dapat dioperasikan dengan berbagai macam tujuan. Beberapa string dapat dirangkai menjadi satu, dapat digunakan sebagai kondisi yang diseleksi dan lain sebagainya. Pascal menyediakan beberapa prosedur standar dan fungsi standar untuk operasi string.

6.1 Merangkai String

Pengerjaan suatu string hanya mempunyai sebuah operator, yaitu operator '+'. Bila operator ini digunakan untuk penambahan nilai numerik, maka akan berfungsi menjumlahkan dua buah nilai elemen numerik. Sedang pada string, operator ini digunakan untuk merangkai dua buah elemen string menjadi sebuah string. Panjang maksimum suatu string yang diijinkan oleh Pascal adalah 255 karakter.

Contoh:

```
Const
    Tulisan1='Bahasa ' ;
    Tulisan2=' Turbo ' ;
    Tulisan3=' Pascal ' ;
Var
    Sifat:string[10];
    Kalimat:[100];
Begin
    Sifat:=' Terstruktur' ;
    Kalimat:=tulisan1+tulisan2+tulisan3+sifat;
    Writeln(kalimat);
End.
```

Bila program dijalankan akan menghasilkan:

Bahasa Turbo Pascal Terstruktur

6.2 Prosedur Standar untuk Operasi String

Pascal menyediakan beberapa prosedur standar yang berguna untuk memanipulasi string, yaitu prosedur standar **Delete**, **Insert**, **Str**, **Val**.

6.2.1 Prosedur standar Delete

Porsedur uni digunakan untuk membuang sejumlah karakter (jumlah yang ditunjukkan oleh nilai **jumlah**, mulai dari posisi tertentu (dintunjukkan oleh nilai **posisi**) dari suatu tulisan (ditunjukkan oleh **pengenal variabel string** atau **tulisan**). Jika **posisi** lebih besar dari panjang tulisan, maka tidak ada karakter yang akan terhapus.

Bentuk umum dari prosedur ini adalah:

Delete(tulisan, posisi, jumlah)

Contoh:

```
Begin
  Writeln(Delete('Pascal',4,3));
End.
```

Hasilnya

Pas

```
program abac;
uses wincrt;
Var
  Tulisan:string[10];
  I,j,posisi:integer;
Begin
  Tulisan:='Pascal';
  Writeln(tulisan);
  J:=length(tulisan);
  Posisi:=j;
  For I:=1 to j do
  Begin
    delete(Tulisan,posisi,1);
    writeln(tulisan);
    posisi:=j-i;
  End;
End.
```

Hasilnya

Pascal
Pasca
Pasc
Pas
Pa
P

6.2.2 Prosedur standar Insert

Prosedur ini digunakan untuk menyisipi suatu string (dintunjukkan oleh tulisan1) ke nilai string yang lainnya (dintunjukkan oleh tulisan2) mulai posisi yang ditentukan oleh nilai posisi. Bila hasil dari penyisipan menjadi string yang panjangnya lebih dari 255 karakter, maka akan menjadi sampai dengan 255 karakter saja.

Betuk umumnya:

Insert(tulisan1,tulisan2)

Contoh:

```
Var
    Nama:[string];
Begin
    Nama:='Pascal';
    Insert('Bahasa ',Nama);
    Writeln(Nama);
End.
```

Hasilnya

Bahasa Pascal

6.2.3 Prosedur standar Str

Prosedur standar ini digunakan untuk mengubah nilai numerik (ditunjukkan oleh nilai N) menjadi nilai string (ditunjukkan oleh nilai S). Nilai N dapat berupa nilai numerik **integer** maupun nilai numerik **real**. Hasil dari perubahan ke nilai string dapat terformat maupun tidak terformat. Bila **lebar** atau **desimal** disebutkan, maka akan dihasilkan nilai string dengan format panjang tertentu. Nilai lebar menunjukkan format panjang dari nilai utuh dan nilai desimal menunjukkan format dari panjang nilai di belakang koma.

Bentuk Umumnya:

Str(N:lebar:desimal,S)

Contoh:

```
Var
    N1, N2:integer;
    S1, S2:string[5];
Begin
    N1:=1234;
    N2:=567;
    Writeln(N1+N2);
    Str(N1:4,S1);
    Str(N2:4,S2);
    Writeln(S1+S2);
End.
```

Hasilnya:

1801
1234 567

6.2.4 Prosedur standar Val

Prosedur standar ini digunakan untuk mengonversi nilai string (ditunjukkan oleh nilai S) menjadi nilai numerik (ditunjukkan oleh nilai N). Walaupun S merupakan nilai string, tetapi harus berisi angka atau tanda plus atau minus, bila tidak berarti salah dan letak kesalahannya ditunjukkan oleh nilai variabel kode. Nilai variabel akan nol bila tidak mengandung kesalahan.

Bentuk umumnya:

Val(S, N, Kode);

Contoh:

```
Var
    S1:string[6];
    N:real;
    Kode:integer;
Begin
    S:='123.45';
    Val(S,N,kode);
    Writeln(S);
    Writeln(N);
    Writeln(kode);
End.
```

Hasilnya

```
123.45
 1.2345000000E+2
0
```

6.3 Fungsi Standar untuk Operasi String

Di samping prosedur standar untuk memanipulasi string, Pascal juga menyediakan beberapa fungsi standar untuk operasi string, yaitu **Copy**, **Concat**, **Pos** dan **Length**.

6.3.1 Fungsi standar Copy

Fungsi standar ini digunakan untuk menyalin (**copy**) sejumlah karakter (jumlah karakter yang disalin ditunjukkan oleh nilai **jumlah**) mulai dari posisi yang ditunjukkan oleh nilai **posisi** dari nilai string yang ditunjukkan oleh nilai S.

Bentuk Umumnya:

Copy(S, posisi, jumlah)

Contoh:

```
Var
    Tulisan:string[15];
Begin
    Tulisan:'Pascal';
    Writeln(Tulisan,3,3);
    Writeln(Tulisan,3,1);
End.
```

Hasilnya:

```
sca
s
```

6.3.2 Fungsi standar ConCat

Fungsi standar ini mempunyai operasi yang sama dengan operator '+' yaitu merangkai beberapa nilai string yang ditunjukkan oleh nilai S1, S2, ... Sn.

Bentuk Umumnya:

ConCat(S1,S2,..Sn)

Contoh:

```
Const
    S1=' Pascal' ;
    S2=' Turbo ' ;
Begin
    Writeln(Concat (S2, S1)) ;
End.
```

Hasilnya

Turbo Pascal

6.3.3 Fungsi standar Pos

Fungsi standar ini digunakan untuk mencari posisi letak dari suatu nilai **string** (ditunjukkan oleh subS) yang ada di dalam nilai string yang lain (ditunjukkan oleh S). Nilai yang dihasilkan adalah berupa nilai byte yang menunjukkan letaknya. Bila bernilai nol berarti nilai string yang dicari tidak ada.

Bentuk Umumnya:

Pos(subS, S)

Contoh:

```
Const
    Tulisan=' Turbo' ;
Begin
    Writeln('T',tulisan);
    Writeln('o',tulisan);
    Writeln('e',tulisan);
End.
```

Hasilnya

1
5
0

6.3.4 Fungsi standar Length

Fungsi standar ini digunakan untuk menghitung panjang atau jumlah karakter yang ada di nilai string yang ditunjukkan oleh S. Hasil dari fungsi ini adalah nilai numerik integer positif.

Bentuk umum

Length(S)

Contoh:

```
Var
    Nama:string[50];
    Jumlah:integer;
Begin
    Writeln('Nama Anda : ');
    Readln(Nama);
    Jumlah:length(nama);
    Writeln('Panjang Nama Anda adalah = ',jumlah);
End.
```

7. Fungsi Standar Aritmatika

7.1 ABS

Bentuk umum: **Abs(x)**;

Digunakan untuk memutlakkan suatu nilai yang ditunjukkan oleh argumen x atau memosotikan suatu bilangan.

Contoh

```
Writeln(abs(-2));
```

Hasilnya

2

7.2 EXP

Bentuk umum: **Exp(x:real):real**;

Digunakan untuk menghitung nilai pangkat dari bilangan e (bilangan alam=2.7182818285, yaitu sebesar e^x).

Contoh

```
Writeln(exp(2):9:2);
```

Hasilnya

7.39

7.3 LN

Bentuk umum: **Ln(x:real):real**;

Digunakan untuk menghitung nilai logaritma alam dari nilai x.

Contoh

```
Writeln(LN(100):5:2);
```

Hasilnya

2.00

7.4 INT

Bentuk umum: **Int(real):real**;

Digunakan untuk menghasilkan nilai integer dari x atau membulatkan ke bawah dengan membuang pecahannya.

Contoh

```
Writeln((9.99):5:2);
```

Hasilnya

9.00

7.5 **FRAC**

Bentuk umum: **Frac(x:real):real;**

Digunakan untuk mendapatkan nilai pecahan dari argumen x.

Contoh

```
Writeln(Frac(9.99):5:3);
```

Hasilnya

0.990

7.6 **SQR**

Bentuk Umum **Sqr(x);**

Digunakan untuk menghitung nilai pangkat kuadrat dari x.

Contoh

```
Writeln(Sqrt(5));
```

Hasilnya

25

7.7 **SQRT**

Bentuk umum **Sqrt(x);**

Digunakan untuk menghitung nilai akar dari x.

Contoh

```
Writeln(Sqrt(36));
```

Hasilnya

6

8. Fungsi Standar Transfer

Fungsi standar transfer digunakan untuk mengubah suatu nilai ke bentuk nilai yang lain. Fungsi-fungsi yang tersedia yaitu fungsi standar **Char**, **Ord**, **Round**, dan **Trunc**.

8.1 CHR

Bentuk umum: **Chr(x)**;

Digunakan untuk mengubah nilai x ke bentuk karakter ASCII.

Contoh

```
Writeln(Chr(65));
```

Hasilnya

A

8.2 ORD

Bentuk umum: **Ord(x)**;

Digunakan untuk mengubah nilai x ke bentuk nilai *longint* yang sesuai dengan kode ASCIInya (menentukan urutan karakter pada kode ASCII). Fungsi *Ord* kebalikan dari *Chr*.

Contoh

```
Writeln(Ord(B));
```

Hasilnya

66

8.3 ROUND

Bentuk umum: **Round(x:real):longint**;

Digunakan untuk membulatkan nilai x ke longint yang terdekat. Bila nilai pecahan sama dengan atau lebih dari 0.5 akan dibulatkan ke atas, sedang bila bila nilai pecahan lebih kecil dari 0.5 akan dibulatkan ke bawah.

Contoh

```
Writeln(Round(10/3));  
Writeln(Round(20/3));
```

Hasilnya

3

7

8.4 TRUNC

Bentuk umum: **Trunc(X)**

Digunakan untuk membulatkan nilai X ke nilai longint terkecil. Atau membulatkan ke bawah dengan membuang pecahannya.

Contoh

```
Writeln(Trunc(10/3));  
Writeln(Trunc(20/3));
```

Hasilnya

3

6

9. Matrik/Larik

9.1. Definisi Matrik

Matrik adalah sekumpulan informasi yang setiap datanya mengacu pada dua buah indeks. Matrik disebut juga tabel lajur yang memiliki indeks baris dan indeks kolom. Karena adanya dua buah indeks tersebut, matrik juga disebut larik dua dimensi. Istilah dua dimensi karena data tersebut ada pada pertemuan baris dan kolom.

Contoh:

	Fisika	Kimia	Biologi	Matematika
Ahmad	6	7	7	8
Joni	7	6	6	5
Toni	8	4	5	6
Dede	6	7	8	6

Tabel di samping disebut matrik 4x4, karena memiliki 4 baris dan memiliki 4 kolom atau larik 4x4.

Cara menentukan nilai elemen data tabel dengan matrik sebagai berikut:

- Nilai Fisika punya Ahmad ada pada baris 1 kolom 1, maka matriksnya [1,1]
- Nilai Biologi punya Joni ada pada baris 2 kolom 3, maka matriksnya [2,3]
- Nilai Kimia punya Dede ada pada baris 4 kolom 2, maka matriksnya [4,2]
- dst.

9.2 Definsi Larik

Suatu larik (*array*) adalah tipe terstruktur yang terdiri dari sejumlah komponen-komponen yang mempunyai tipe sama. Suatu larik mempunyai jumlah komponen yang banyaknya tetap. Banyaknya komponen dalam suatu larik ditunjukkan oleh suatu indkes yang disebut tipe indeks. Tipe indeks ini berbentuk ungkapan tipe ordinal (tipe data yang berurutan). Tiap-iap komponen pada larik dapat diakses dengan menunjukkan nilai indeksnya.

1. Dimensi Larik

a. Larik dimensi satu

Larik ini memiliki data yang berurutan dengan satu jenis yang sama dan tidak berbentuk tabel.

Contoh: larik **bunga** datanya *mawar, melati, angrek*

larik **nama hari** datanya *Minggu, Senin, Selasa, Rabu, Kamis, Jumat, Sabtu*

Deklarasi larik satu dimensi

`bunga : array[1..3] of string=('mawar, melati, angrek')`

nama larik kata cadangan untuk larik jumlah elemen data tipe data larik data larik

elemen-elemen data larik bunga adalah:

`bunga[1]: 'mawar'`
`bunga[2]: 'melati'`
`bunga[3]: 'angrek'` } nilai data larik

nama larik indeks/urutan dalam larik

b. *Larik dua dimensi*

Larik dua dimensi disebut juga matriks. Larik dua dimensi memiliki data yang tersusun berdasarkan urutan baris dan kolom pada tabel.

Contoh:

Larik nilai memiliki data urutan baris yaitu: Ahmad, Joni, Toni, Dede dan data urutan kolom yaitu fisika, kimia, biologi, matematika. Data di atas dapat digambarkan sebagai berikut:

		Fisika	Kimia	Biologi	Matematika
		1	2	3	4
Aahmad	1	6	7	7	8
Joni	2	7	6	6	5
Toni	3	8	4	5	6
Dede	4	6	7	8	6

Dari data tabel di atas dapat dibuat indeksinya yaitu:

- Nilai[Ahmad,Fisika] atau Nilai[1,1] = 6
- Nilai[Ahmad,Kimia] atau Nilai[1,2] = 7
- Nilai[Ahmad,Biologi] atau Nilai[1,3] = 7
- Nilai[Ahmad,Matematika] atau Nilai[1,4]=8
- Nilai[Joni,Fisika] atau Nilai[2,1] = 7
- Nilai[Joni,Kimia] atau Nilai[2,2] = 6
- dst.

Deklarasi Larik dua dimensi

*nama larik:array[*minbaris..maxbaris*,*minkolom..maxkolom*] of tipe larik*

Contoh1:

Nilai:array[1..4,1..4] of integer;

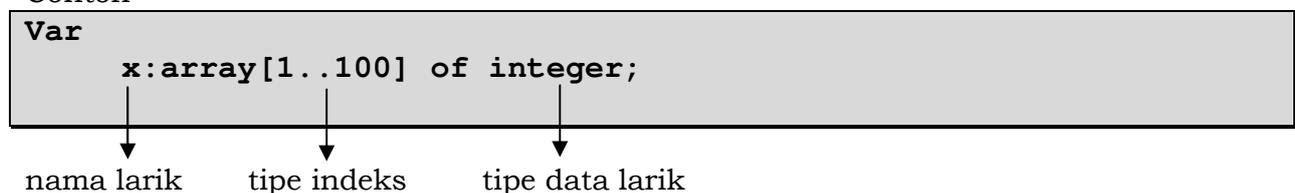
atau

Contoh2:

Nilai:array[1..4,1..4] of integer=((6,7,7,8),(7,6,6,5),(8,4,5,6),(6,7,8,6))

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 diantara tanda '[]', diikuti lagi oleh kata cadangan **of** dan tipe lariknya.

Contoh



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:

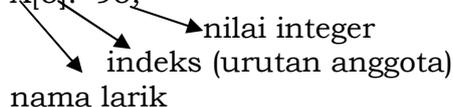
X[1]:=25;

X[2]:=55;

X[3]:=65;

X[4]:=75;

X[5]:=95;



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

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

hasilnya

65

Bila nilai-nilai elemen larik dibatasi dengan nilai-nilai tertentu yang tidak dapat ditulis dalam suatu range, tetapi dapat ditentukan secara beurutun, maka dapat dipergunakan larik tipe skalar.

Contoh

```
Type
  Hari=(Minggu, Senin, Selasa, Rabu, Kamis, Jumat, Sabtu);
Var
  Jamkerja:array[1..25] of hari;
```

Dari deklarasi ini, larik Jamkerja mempunyai maksimum 25 elemen dan masing-masing nilai elemennya hanya dapat mempunyai nilai sebanyak 7 macam nilai saja.

9.3 Cara Membuat Matrik

- a. *Membuat satu buah matriks, elemennya berdasarkan nilai perulangan*
 - Tentukan jumlah baris
 - Tentukan jumlah kolom
 - Buat pengulangan untuk baris, dan pengulangan kolom.

Contoh ke-1 listing program mambuat matrik:

```
Program Matrik1;
Uses WinCrt;
Var
  I, J      :Integer;
Begin
  ClrScr;
  {pengulangan jumlah baris}
  FOR I:= 1 TO 5 DO
  BEGIN
    {pengulangan jumlah kolom}
    FOR J:=1 TO 4 DO
    BEGIN
      {pencetakan matriks}
      WRITE(I:6, j:3);
    END;
    {ganti baris pada matriks}
    WRITELN;
  END;
END.
```

Hasilnya

```
1 1      1 2      1 3      1 4      1 5
2 1      2 2      2 3      2 4      2 5
3 1      3 2      3 3      3 4      3 5
4 1      4 2      4 3      4 4      4 5
5 1      5 2      5 3      5 4      5 5
```

- b. *Membuat satu buah matrik, elemennya ditetapkan di awal.*
- Tentukan jumlah baris
 - Tentukan jumlah kolom
 - Buat pengulangan untuk baris, dan pengulangan kolom.
 - Cetak matrik yang berindeks

Contoh ke-2 listing program mambuat matrik:

- c. *Membuat satu buah elemennya berdasarkan nilai perulangan*
- Tentukan jumlah baris
 - Tentukan jumlah kolom
 - Buat pengulangan untuk baris, dan pengulangan kolom.

Contoh ke-2 listing program mambuat matrik:

```
PROGRAM M2;
uses wincrt;
CONST matrik:array[1..3,1..3]of
INTEGER=((1,2,3),(4,5,6),(7,8,9));
VAR b,k:integer;
begin
    CLRSCR;
    FOR B:=1 TO 3 DO
    BEGIN
        FOR K:=1 TO 3 DO
        BEGIN
            WRITE (MATRIK [B, K] :3);
        END;
        WRITELN;
    END;
END.
```

Hasilnya

```
1 2 3
4 5 6
7 8 9
```

10. Prosedur (*Procedure*) dan Fungsi (*Fuction*)

Prosedur 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.

Prosedur 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 diruliskan sekali saja dalam prosedur dan dapat dipanggil atau dipergunakan sewaktu-waktu bila diperlukan.

Contoh1 prosedur tanpa acuan

```
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  
16
```

Contoh2 prosedur secara acuan

```
Procedure Hitung(var a,b,c);  
Begin  
    C:=a+b;  
    Writeln(c);  
End;  
  
Var  
    X,y,z:integer;  
Begin  
    X:=2;Y:=3;  
    Hitung(x,y,z);  
End.
```

Hasilnya

```
5
```

Prosedur Memanggil Dirinya Sendiri (*Rekursi*)

Prosedur memanggil dirinya sendiri merupakan suatu prosedur yang memanggil atau menggunakan prosedur itu juga. Proses dari suatu program bagian yang memanggil dirinya sendiri dikenal dengan istilah rekursi.

Contoh

```
Var
    I:integer;
Procedure Cetak;
Begin
    Writeln('Pascal');
    I:=i+1;
    If i<10 then cetak;
End;

Begin
    I:=1;
    Cetak;
End.
```

Bila program dijalankan, maka prosedur cetak akan diproses sebanyak 9 kali, yaitu dengan cara memanggil dirinya sendiri, hasilnya sebagai berikut.

Pascal
Pascal
Pascal
Pascal
Pascal
Pascal
Pascal
Pascal
Pascal
Pascal

Fungsi

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;

Contoh

```
Function hitung(var nilai:integer):Integer;
Function pangkat (x,y:real):real
```

Soal Latihan Pemahaman Konsep #1

SOAL-SOAL DEKLARASI

1. Manakah yang mendeklarasikan tipe enumerasi dengan tepat?
 - a. Type a=integer;
 - b. Type a=1..300;
 - c. Type a=(baik, jelek, buruk);
 - d. Type a=[baik, jelek, buruk];
 - e. Type a=baik, jelek, buruk;
2. Tipe di bawah ini mana yang tidak dapat melakukan operasi aritmatika?
 - a. integer
 - b. byte
 - c. real
 - d. boolean
 - e. word
3. Deklarasi prosedur manakah yang dibenarkan?
 - a. procedure hapus;
 - b. procedure hapus(s:string);
 - c. procedure hapus(var s:string);
 - d. procedure hapus(s:string):boolean;
 - e. procedure hapus(var data);
4. Deklarasi function manakah yang tidak diizinkan?
 - a. Function density(x:real):real;
 - b. Function density(b:byte):byte;
 - c. Function density(var s:string):real;
 - d. Function density(var data):byte;
 - e. Function density;
5. Tipe variabel ekspresi manakah yang tidak dapat ditampilkan dengan procedure Writeln?
 - a. Type T=Integer;
 - b. Type T=String;
 - c. Type C=Char;
 - d. Type T=(Small, Medium, Large)
 - e. Semua valid
6. Dengan deklarasi berikut:
Type warna=(merah, kuning, hijau, biru, hitam, putih, jingga);
Var w:warna;
Perintah mana yang salah?
 - a. If w in [warna] then writeln('ada');
 - b. w:=merah;
w:=w + kuning;
 - c. w:=[merah];
 - d. w:=hijau;
dec(w);
 - e. w:='Merah';

7. Pada deklarasi di atas, jika variabel W1 berisi [merah,kuning,hijau] dan variabel W2 berisi [merah,kuning,hitam] maka, jika diberikan statemen $W3 := W1 + W2$, W3 akan berisi:
- [merah,kuning,hijau,hitam]
 - [merah,kuning,hijau,merah,kuning,hitam]
 - [hijau,hitam]
 - [merah,kuning,merah,kuning,hijau, hitam]
 - [merah,kuning]
8. Jika diberikan statemen $W3 := W1 - W2$, W3 akan berisi:
- [merah,kuning,hijau,hitam]
 - [merah,kuning,hijau,merah,kuning,hitam]
 - [hijau]
 - [merah,kuning,merah,kuning,hijau, hitam]
 - [merah,kuning]
9. Jika diberikan statemen $W3 := W1 * W2$, W3 akan berisi:
- [merah,kuning,hijau,hitam]
 - [merah,kuning,hijau,merah,kuning,hitam]
 - [hijau,hitam]
 - [merah,kuning,merah,kuning,hijau, hitam]
 - [merah,kuning]

SOAL-SOAL INPUT / OUTPUT

10. Perhatikan potongan program berikut ini :

```
Begin
    Writeln((10 shr 1) shl 2);
end.
```

Apa yang dihasilkan oleh program diatas...

- 18
- 19
- 20
- 21
- 22

SOAL-SOAL STRUKTUR KONTROL

11. Bagaimana keluaran program di bawah ini?

```
Var
    I: integer;
Begin
    I:=2;
    Case I of
        1, 3, 5, 7, 9: writeln('Ganjil');
        2: writeln('Prima genap');
        0..10: writeln('Normal');
    else writeln('Tidak normal');
    end;
end;
```

- a. Prima genap
- b. Normal
- c. Prima genap
Normal
- d. Normal
Prima genap
- e. Prima genap
Tidak normal

Perhatikan program di bawah ini:

```

var I, j, k: integer;
    L: byte;
begin
    i:=3;
    j:=4;
    k:=32;
    L:=0;

    {If - 1 }
    if i + j and k =0 then
        writeln('Betul')
    else
        writeln('Salah');

    {If - 2 }
    if (i = 2) and (j < i) or (k > i) then
        writeln('Betul')
    else
        writeln('Salah');

    {If - 3}
    if not L in [1..120] then
        writeln('Betul')
    else
        writeln('Salah');
end.

```

Program diatas berisi tiga perintah if then else yang saling tidak berkaitan, masing-masing IF diberi nama IF – 1, IF – 2, IF – 3.

12. Perintah if manakah yang tidak dibenarkan:

- a. If - 1
- b. If - 2
- c. If - 3
- d. If – 1 dan if - 2
- e. Tidak ada if yang salah

13. Pada program di atas, if mana yang menghasilkan output “Betul”?

- a. If - 1
- b. If - 2
- c. If - 3
- d. If – 1 dan if - 2
- e. Tidak ada if yang menghasilkan “Betul”

SOAL-SOAL PERULANGAN

14. Perhatikan penggalan program berikut ni :

```
const
    Data: array [1..3,1..3] of char=
        (('1','1','2'),('2','2','4'),('4','4','8'));
var i, j : byte;
begin
    for i:= 1 to 3 do
        begin
            for j:=3 downto 1 do
                write(Data[i,j]):
            writeln;
        end;
    end.
```

Apa keluaran program di atas ?

- a. 112
224
448
- b. '1''1''2'
'2''2''4'
'4''4''8'
- c. 211
422
844
- d. '2''1''1'
'4''2''2'
'8''4''4'
- e. 124
124
248

15. Perhatikan program dibawah ini :

```
type data=set of char;
var setchar:data;
    s:string;
    i:integer;
begin
    setchar:=[];
    readln(s);
    for i:=1 to length(s) do
        begin
            if not(s[i] in setchar) then
                begin
                    setchar:=setchar+[s[i]];
                    write(s[i]);
                end;
        end;
    writeln;
end.
```

Output dari program di atas jika input 'To be or Not To be that is the question' adalah

- a. 'To berNthaisqun.'
- b. 'To berNhaisqu`
- c. 'to@bernhaisquN'
- d. 'T N.'
- e. 'OBERTHAISQUN'

16. Gunakan program berikut untuk menjawab pertanyaan :

```
type data=set of byte;
var setint:data;
    i:integer;
begin
    setint:=[1];
    setint:=setint+[3];
    setint:=[5];
    for i:=1 to 5 do
    begin
        if (i in setint) then continue else setint:=[i];
    end;
end.
```

Output dari program di atas adalah:

- a. [1, 2, 3, 4, 5]
- b. [1, 3, 5]
- c. [5]
- d. [1, 3]
- e. []

SOAL-SOAL PROSEDUR DAN FUNGSI

17. Perhatikan program berikut :

```
var s:string;
begin
    s:='TOKI GO GET GOLD!';
    delete(s,1,length(s)-12);
    writeln(s);
end.
```

Apa keluaran program di atas ?

- a. GO GET GOLD!
- b. GO GET GOLD!
- c. GET GOLD!
- d. TOKI GO GET
- e. TOKI GO GE

18. Perhatikan penggalan program berikut :

```
var i,k: integer;
begin
i:=5; k:=0;
k:=trunc(sqrt(i))+1;
writeln(k);
end.
```

Apa keluaran program di atas?

- a. 3
- b. 2.24
- c. 2
- d. 0
- e. program tidak dapat dijalankan

19. Mengacu pada program berikut :

```
var
A,B:string;
C:string[10];
begin
A:='TOKI MEMANG';
B:='HEBAT';
C:=A+B;
if (Pos(B)>0) then
Begin
Writeln('A');
end else
Writeln('B');
end.
```

Apa yang terjadi jika program di atas di jalankan...

- a. Huruf 'A' tercetak
- b. Huruf 'B' tercetak
- c. Tidak dapat dipastikan
- d. Terjadi error
- e. Tidak bisa di compile

20. Perhatikan potongan program berikut :

```
begin
writeln(round(frac(3.7)));
end.
```

Apa keluaran program di atas ?

- a. 0
- b. 1
- c. 2
- d. 3
- e. 4

21. Diketahui deklarasi fungsi dan variabel sebagai berikut:

```
var St: String;
procedure Sulap(var S: String);
begin
    if S = 'Kecil' then S := 'kecil' else
    if S = 'Besar' then S := 'BESAR';
end;
```

Di antara potongan program berikut, manakah yang salah?

- a. St := Chr(60);
Sulap(St);
- b. St := 'KECIL';
Sulap(St);
- c. St := Chr(45) + Chr(65);
Sulap(St);
- d. Sulap('Besar');
- e. Semua ekspresi di atas benar

SOAL-SOAL OPERASI FILE

22. Perintah mana yang tidak boleh digunakan untuk file bertipe text?

- a. Assign
- b. Reset
- c. EOF
- d. FilePos
- e. Semua boleh digunakan untuk Text

Gunakan program berikut ini untuk menjawab soal di bawah ini:

```
program Uji;
var T:Text;
    i, j, k:integer;
begin
    Assign(T, 'INPUT.TXT');
    Reset(T);
    Readln(T, i, j, k);
    Writeln(i, ' ', j, ' ', k);
    Readln(T, i);
    Readln(T, j);
    Writeln(i, ' ', j);
    Close(T);
End.
```

23. Misalkan file INPUT.TXT berisi baris-baris sebagai berikut:

```
3 1 4 9
5 2 6
8 7
0
```

Bagaimanakah output dari program tersebut?

- a. 3 1 4 9
5 2 6
8 7
- b. 3 1 4
9 5
- c. 3 1 4
5 2
- d. 3 1 4
5 8
- e. Terjadi runtime error karena isi file INPUT.TXT tidak sesuai untuk program ini.

SOAL-SOAL KASUS/MEMBACA PROGRAM

Program berikut ini dipakai untuk menjawab dua soal di bawah ini

```
var Bil:Integer;
procedure Find(B:Integer;I:Integer);
var J,R:Integer;
begin
    R:=Round(sqrt(B));
    J:=2;
    while (J<=R) and (B Mod J<>0) do
        inc(J);
        if J<=R then
            begin
                Write(J,'*');
                Find(B div J, I+1);
            end
        else if I>0 then
            Writeln(B,'=',Bil)
        else
            Writeln('Bilangan Prima!');
    end;

begin
    Write('Masukkan bilangan : ');
    Readln(Bil);
    Find(Bil,0);
end.
```

24. Bagaimana output program di atas bila inputnya 42?

- a. $7 * 3 * 2 = 42$
- b. Bilangan prima
- c. =42
- d. $2 * 3 * 7 = 42$
- e. Salah semua

25. Bagaimana output program di atas bila, inputnya 23?

- = 23
- Bilangan prima
- $23 * 1 = 23$
- = 23 Bilangan prima!
- Salah semua

Joni, petugas statistik yang baru saja belajar Pascal. Mencoba membuat program perata-rata sebagai berikut.

```
var Amatan:array[5] of integer;
    Jumlah:Integer;
    RataRata:Integer;
    I:Integer;
begin
  for I:=1 to 5 do
  begin
    Write('Amatan ke-',I, ' : ');
    Readln(Amatan[I]);
  end;
  Jumlah:=0;
  For I:=1 to 5 do
  begin
    Jumlah:=Jumlah+Amatan[I];
    RataRata:=Jumlah/5;
  end;
  Writeln('Jumlah = ',Jumlah);
  Writeln('Rata-rata = ',RataRata);
  Readln;
end.
```

Gunakan program yang dibuat oleh Joni ini untuk menjawab soal-soal berikut.

26. Ketika si Joni mencoba menjalankan program tersebut, ternyata, compiler menunjukkan sebuah pesan kesalahan yang membuat ia kebingungan. Tahukah Anda kesalahan pertama yang dibuat Joni?

- Judul program (program Statistik) terlalu panjang, maksimum 8 karakter (misalnya: program Stat)
- Procedure Readln (pada baris terakhir program sebelum end.) tidak boleh dipanggil tanpa parameter. Jadi seharusnya: Readln(I);
- Statement for dengan variabel sama tidak boleh diulangi dua kali. Seharusnya dideklarasikan variabel lain, misalnya var I: Integer untuk for yang kedua
- Deklarasi array salah, semestinya: var Amatan: array[1..5] of Integer;
- Nama variabel seperti RataRata tidak valid, seharusnya Ratarata

27. Setelah Anda memberi saran demikian, ternyata Joni masih belum bisa meng-compile programnya. Apa sebabnya?

a. setiap variabel harus dideklarasikan dengan keyword var sendiri-sendiri.

Misalnya:

```
var Jumlah: Integer;  
var RataRata: Integer;  
Var I: Integer;
```

b. Variabel RataRata tidak harus bertipe Real

c. Semua variabel, kecuali I seharusnya adalah Real, tidak boleh Integer

d. Pemisah antara parameter dalam Write dan Writeln harus titik koma, bukan koma, Misalnya

```
Writeln('Jumlah = ';Jumlah);
```

e. Semua alasan di atas salah

28. Joni mengganti operator "/" dengan "div" pada baris ke-15 program tersebut. Apa akibatnya?

a. program tidak mau di-compile karena. operator div tidak dapat digunakan di situ

b. nilai rata-ratanya menjadi 5

c. nilai rata-ratanya menjadi 6

d. nilai rata-ratanya menjadi 0

e. nilai rata-ratanya menjadi 2

Gunakan program berikut ini untuk menjawab beberapa soal selanjutnya:

```
uses wincrt;  
var j:array['A'..'Z'] of Byte;  
c:char;  
Kal:string;  
procedure HH(S:String);  
var i:integer; {baris-6}  
    m:char;  
begin  
    for i:= 1 to length(S) do  
        begin  
            m:=S[i]; {baris-11}  
            if m in ['A'..'Z'] then {baris-12}  
                inc(J[i]);  
        end;  
end;  
  
begin  
for c:='A' to 'Z' do J[c]:=0;  
Kal:='PASAR';  
HH(Kal);  
for c:='A' to 'Z' do  
if J[c]>0 then write(c,J[c], ' ');  
writeln;  
Kal:='RAYA';  
HH(Kal);  
for c:='Z' downto 'A' do  
if j[c]>0 then write(c,J[c], ' ');  
writeln;  
end.
```

29. Bila terdapat kesalahan yang menyebabkan program sama sekali tidak dapat dijalankan sebutkan pada baris berapa, dan bagaimana perbaikannya?
- Kesalahan semacam ini tidak ada
 - Baris 12, seharusnya ditulis
`If [m] in ['A'..'Z'] then`
 - Baris 6 seharusnya ditulis
`var i: Char;`
 - Baris 13, seharusnya ditulis
`inc(J[m]);`
 - Index array hanya boleh berupa angka. Jadi deklarasi variabel seharusnya ditulis:
`const A = 1; Z = 26;`
`var J: array[A..Z] of Byte;`
`c: Byte;`
`Kal: String;`
dan semua konstanta karakter dalam perintah for harus diganti, misalnya: `for c:=A to Z do` dan seterusnya
30. Dengan perbaikan seperti nomor sebelumnya (kalau ada), maka program bisa dijalankan. Apakah hasil dari program tersebut?
- A2 PI RI SI
A4 PI R2 SI YI
 - A2 P1 RI SI
YI RI A2
 - A2 PI RI SI
YI R2 A4
 - PI A2 SI RI
YI A4 R2
 - A2 P1 RI SI
YI SI R2 PI A4
31. Tindakan apakah yang dilakukan oleh subrutin HH ketika dipanggil oleh baris 19 program di atas, dengan string S berisi kata "PASAR"?
- Menghitung frekuensi kemunculan huruf-huruf alfabet dan menyimpannya dalam array J
 - Mengumpulkan huruf-huruf alfabet yang muncul lebih dari satu kali ke dalam array J
 - Mencatat letak setiap huruf alfabet ke dalam array J
 - Menentukan huruf yang paling sering dan paling jarang muncul dalam array J
 - Mengurutkan huruf-huruf menurut urutan alfabet dari yang terkecil sampai yang terbesar.

Soal Latihan Pemahaman Konsep #2

Soal 1.

Deklarasi konstanta manakah yang salah?

- a. `const rata-rata=10;`
- b. `const tinggibadan=150;`
- c. `const pi=22/4;`
- d. `const suhu=20000;`
- e. `const duaxtiga=8;`

Soal 2.

Manakah yang benar dari deklarasi di bawah ini?

- a. Program `abc;`
`var x,y:real;`
- b. Program `pqrstu2343;`
`var x,y=real;`
- c. Program `4343;`
`var x:=boolean;b:=integer;`
- d. Program `ku;`
`var z,y:char='a';`
- e. Program `gampang;`
`var a:byte, c:real;`

Soal 3.

Manakah yang mendeklarasikan tipe enumerasi dengan tepat?

- a. `Type a=integer;`
- b. `Type a=1..300;`
- c. `Type a=(baik, jelek, buruk);`
- d. `Type a=[baik, jelek, buruk];`
- e. `Type a=baik, jelek, buruk;`

Soal 4.

Bagaimana mendeklarasikan konstanta bertipe array 3 x 3?

- a. `Const a:array[3,3]=((1,2,3), (2,3,4), (3,4,5));`
- b. `Const a[3,3]=(1,2,3, 2,3,4, 3,4,5);`
- c. `Const a:array[1..3,1..3] of byte =((1,2,3), (2,3,4), (3,4,5));`
- d. `Const a:array[1..3,1..3] of byte=(1,2,3,2,3,4,3,4,5);`
- e. `Const a:array[1..3] of array[1..3] of byte = ((1,2,3), (2,3,4), (3,4,5));`

Soal 5.

Apakah output program berikut ini?

```
var U: Integer,  
begin  
U:=3;  
writeln('Umur saya', U:3,'tahun'),  
End.
```

- a. Umur saya 3 tahun
- b. Umur saya 1 tahun
- c. Umur saya 3 tahun
- d. Umur saya 003 tahun
- e. Umur saya 3.00 tahun

Soal 6.

Bagaimana cara anda mengetikkan input untuk program berikut ini?

```
var Umur, Tinggi, Berat: Real,  
begin  
write('Masukkan umur, tinggi 'dan berat badan Anda: ');  
Readln(Umur, Tinggi, Berat);  
end.
```

- a. 16,170, 61.5
- b. 16; 170; 61.5
- c. 16 170 61.5
- d. 16 170 61.5
- e. 16;170;161,5

Soal 7.

```
var r : real;  
begin  
r:=147.0;  
writein(r:0:5),  
end.
```

Apa tampilan program di atas ?

- a. 00147
- b. 147.0000
- c. 0147.0
- d. 147.00000
- e. 147.000

Soal 8.

```
var i, j:byte;  
begin  
i:=100; j:=200;  
writeln(i*j)  
end.
```

Apa tampilan program di atas?

- a. 12
- b. 22
- c. 32
- d. 20000
- e. 400000

Soal 9.

Perhatikan source code di bawah inii

```
var  
Mail: word;  
begin  
Mail:='TOKI';  
writein ('TOKI');  
writeln (Mail);  
end.
```

Output yang tercetak di layar setelah eksekusi adalah

- a. TOKI
- b. TOKITOKI
- c. TOKITOKITOKI
- d. terjadi compiler error
- e. tidak ada jawaban yang benar

Soal 10.

Bagaimana keluaran program di bawah ini?

```
Var
I:integer;
Begin
I:=2;
Case I of
1,3,5,7,9:writeln('Ganjil');
2:writeln('Prima genap');
0..10:writeln('Normal');
else writeln('Tidak normal');
end;
end;
```

- a. Prima genap
- b. Normal
- c. Prima genap
Normal
- d. Normal
Prima genap
- e. Prima genap
Tidak normal

Perhatikan program di bawah ini:

```
var I,j,k:integer;
L:byte;
begin
i:=3;
j:=4;
k:=32;
L:=0;
{If - 1 }
if i + j and k =0 then
writeln('Betul')
else
writeln('Salah');
{If - 2 }
if (i = 2) and (j < i) or (k > i) then
writeln('Betul')
else
writeln('Salah');
{If - 3}
if not L in [1..120] then
writeln('Betul')
else
writeln('Salah');
end.
```

Program ini berisi tiga perintah if then else yang saling tidak berkaitan, masing-masing IF diberi nama IF – 1, IF – 2, IF – 3.

Soal 11.

Perintah if manakah yang tidak dibenarkan:

- a. If - 1
- b. If - 2
- c. If - 3
- d. If – 1 dan if - 2
- e. Tidak ada if yang salah

Soal 12.

Pada program di atas, if mana yang menghasilkan output “Betul”?

- a. If - 1
- b. If - 2
- c. If - 3
- d. If – 1 dan if - 2
- e. Tidak ada if yang menghasilkan “Betul”

Soal 13.

Pada program di atas, manakah pernyataan yang benar?

- a. Ekspresi not L in [1..120] ekuivalen dengan L in [0,121..255]
- b. Ekspresi $I + J$ and $K = 0$ dalam if – 1 ekuivalen dengan $(I + J = 0)$ and $(K = 0)$
- c. Kondisi pada if – 1 akan bernilai lain jika kondisi diubah menjadi $(I + J)$ AND $K = 0$
- d. Kondisi pada if – 2 akan bernilai lain jika kondisi diubah menjadi $((I=2)$ and $(J<1))$ or $(k>1)$
- e. Kondisi pada if – 3 akan bernilai lain jika kondisi diubah menjadi $\{NOT L in [1..120])$ OR $(L in [121,122])$

Soal 14.

Berapa keluaran program bila pemakai memberi nilai $M=9$?

```
Var C, M, J:integer;  
Begin  
J:=1;  
write('M = '); readln(M);  
For c:=5 to M do  
J := j * (M-4);  
writeln('J = ',J);  
nd.
```

- a. J = 120
- b. J = -120
- c. J = 1
- d. J = 24
- e. Salah semua

Soal 15.

```
Var  
i,j: shortint;  
Begin  
for i:=1 to 200 do  
Inc(j);  
end.
```

Program diatas akan mengbasilkan:

- a. Nilai j = 200;
- b. Nilai j = 127;
- c. Tidak dapat dipastikan
- d. Terjadi error (code 76).
- e. Program tidak dapat berhenti

Soal 16.

```
var
h,i,j:integer;
begin
h:=0;
for j:=1 to 10 do
inc(h);i:=1;
j:=0;
repeat
inc(i);
inc(j);
until i=10;
if h=j then writeln('GOLD')
else writeln('SILVER');
end.
```

Apa keluaran program di atas

- a. GOLD
- b. SILVER
- c. GOLD
SILVER
- d. SILVER
GOLD
- e. tidak ada keluaran

Soal 17.

```
var i,j : integer;
begin
j := 1;
for i := 1 to 5 do
begin
writeln(i,' ',j);
j := i-1;
end;
end.
```

Output dari program di atas adalah

- | | | | | |
|--------|--------|--------|--------|--------|
| a. 1 1 | b. 1 1 | c. 1 0 | d. 1 5 | e. 1 1 |
| 2 1 | 2 2 | 2 1 | 2 4 | 2 0 |
| 1 3 | 3 3 | 3 2 | 3 3 | 3 1 |
| 4 1 | 4 4 | 4 3 | 4 2 | 4 2 |
| 1 5 | 5 5 | 5 4 | 5 1 | 5 3 |

Program untuk 2 soal berikutnya

```
1. var s : string;
2. i: integer;
3. begin
4. for i := 1 to length(s) do
5. begin
6. s[i]:= s[length(s)-i+1];
7. end;
8. end.
```

Soal 18.

Output dari program di atas jika s='SC!P!O' adalah

- a. s='SC!!CS'
- b. s='SSSSSS'
- c. s='O!PP!O'
- d. s='CCCCCC'
- e. s='!O!O!O'

Soal 19.

Output dari program di atas jika s='HaShMaT' dan baris ke-4 diubah menjadi "for i:=length(s) downto 1" adalah

- a. s='TaMhMaT'
- b. s='HaShSaH'
- c. s='HHHHHHH'
- d. s='aaaaaaa'
- e. s='aHaHaHa'

Soal 21.

```
Function Sum(const A,B:Integer): Integer;
Begin
Sum:=A+B;
end;
Begin
writeln(Sum(5,10));
end.
```

Apa yang dihasilkan oleh program diatas.

- a. 5
- b. 10
- c. 15
- d. 20
- e. Tidak bisa di compile.

Soal 22.

```
Begin
Assign(Output, 'ABC.TXT');
Rewrite(Output);
Write('GO GET GOLD');
Close(Output);
end.
```

Program diatas akan menghasilkan:...

- a. File'ABC' dengan isi 'GO GET GOD'
- b. Run-time error.
- c. Tidak dapat dicompile karena variable 'Ouiput' tidak ada
- d. Tidak menghasilkan apa-apa
- e. Tidak ada yang benar

Soal 23.

Perintah mana yang tidak boleh digunakan untuk file bertipe text?

- a. Assign
- b. Reset
- c. EOF
- d. FilePos
- e. Semua boleh digunakan untuk Text

Gunakan program berikut ini untuk menjawab dua soal di bawah ini:

```
program Uji;  
var T:Text;  
i,j,k:integer;  
begin  
    Assign(T, 'INPUT.TXT');  
    Reset(T);  
    Readln(T, i, j, k);  
    writeln(i, ' ', j, ' ', k);  
    Readln(T, i);  
    Readln(T, j);  
    writeln(i, ' ', j);  
    Close(T);  
End.
```

Soal 24.

Misalkan file INPUT.TXT berisi baris-baris sebagai berikut:

```
3 1 4 9  
5 2 6  
8 7  
0
```

Bagaimanakah output dari program tersebut?

- a. 3 1 4 9
5 2 6
8 7
- b. 3 1 4
9 5
- c. 3 1 4
5 2
- d. 3 1 4
5 8
- e. Terjadi runtime error karena isi file INPUT.TXT tidak sesuai untuk program ini.

Soal 25.

Bila isi file INPUT.TXT adalah sebagai berikut:

1
3
2
4
5
6

bagaimana output program ini?

a. 2 3 1

4 5

b. 1 0 0

3 2

c. 1 3 2

4 5

d. 1 3 2

5 6

e. Terjadi runtime error karena isi file INPUT.TXT tidak sesuai untuk program ini.

Soal Latihan Membuat Program #1

TIPE DATA SEDERHANA

A. Tujuan

Siswa dapat mengenal dan menggunakan tipe data sederhana dengan benar.

B. Dasar Teori

Suatu variabel dapat digunakan hanya dapat menyimpan sebuah nilai dari sebuah tipe data sederhana. Untuk memanipulasi kelompok tipe data sebuah nilai pada suatu variabel maka harus menentukan tipe data yang tepat untuk sebuah variabel tersebut.

Berikut contoh program penggunaan tipe data sederhana:

```
Program Luas;
Uses WinCrt;
Var
  Lebar, panjang : byte;
  Luas           : integer;
Begin
  CLRSCR;
  {input data}
  WRITE('MASUKAN NILAI PANJANG : '); READLN(PANJANG);
  WRITE('MASUKAN NILAI LEBAR   : '); READLN(LEBAR);

  {prpses data}
  LUAS:=PANJANG*LEBAR;

  {output}
  WRITELN('LUAS PERSEGI PANJANG : ', LUAS);
  READLN;
END.
```

C. Praktik

1. Tulislah program di atas, jalankan program tersebut. Kemudian perhatikan, isilah nilai panjang dengan nilai 5 dan lebar 4, dan catat hasilnya.
2. Jalankan lagi program di atas, kemudian isi nilai panjang dengan 700 dan lebar dengan 500, bagaimanakah hasilnya, mengapa terjadi kesalahan? Cobalah ganti tipe datanya. Jalankan kembali program tadi dan isilah panjang dengan 700 dan lebar dengan 500. Bagaimana hasilnya sekarang? Jika masih salah cobalah perbaiki sampai benar!
3. Jika sudah benar rekam dengan nama file MODUL1A.
4. a. Buatlah program untuk mencari luas segitiga! Rekam dengan nama file MODUL1B
b. Buatlah listing program untuk masalah berikut ini:
Input : nama, nama barang, jumlah barang, harga
*Proses: jumlah harga:=jumlah barang*harga*
*potongan:=10/100*jumlah harga;*
jumlah bayar:=jumlah harga-potongan;
Output : jumlah harga, potongan dan jumlah bayar
Rekam dengan nama file MODUL1C

D. Pertanyaan

Semua pertanyaan berikut harus dijawab dalam laporan resmi!

1. Tulislah perbedaan antara tipe data **byte**, **word**, **integer**, **longint**, **real**, **char**, **string**.
2. Kapankah digunakan tipe data **real**?
3. Mengapa soal praktik ke-2 mengalami kesalahan? Jelaskan!

Soal Latihan Membuat Program #2

PERULANGAN FOR TO DO

A. Tujuan

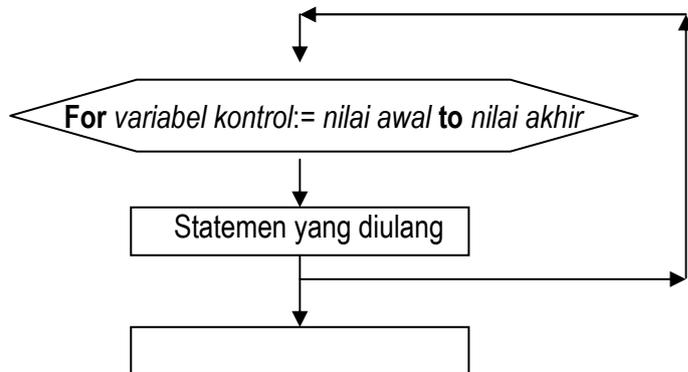
Siswa dapat membuat program untuk mengulangi sebuah statemen berulang kali dengan benar.

B. Dasar Teori

Perulangan (*loop*) merupakan bentuk yang sering ditemui di dalam suatu program aplikasi. Di dalam bahasa Pascal, dikenal tiga macam perulangan, salah satunya statemen **FOR**.

1. Struktur Perulangan For

Perulangan dengan statemen *For* digunakan untuk mengulang statemen atau satu blok statemen berulang kali sejumlah yang ditentukan. Perulangan dengan statemen *For* dapat berbentuk perulangan positif, perulangan negatif, dan perulangan tersarang. Sintak dari statemen *For* dalam bentuk diagram adalah sebagai berikut:



2. Perulangan Positif

Perulangan positif adalah perulangan dengan penghitung (*counter*) dari kecil ke besar atau dengan kata lain pertambahannya positif. Perulangan positif dapat dibentuk dengan menggunakan statemen **For To Do**, dengan bentuk umum:

For variabel kontrol:=nilai awal **To** nilai akhir **Do** statemen

Variabel kontrol, nilai awal, nilai akhir harus mempunyai tipe yang sama, yaitu bertipe integer.

Contoh ke-1:

```
Program pengulangan_positif;
Uses Wincrt;
Var
  I:Real;
Begin
  Clrscr;
  For I:=1 to 5 Do
  Begin
    Write(I);
    Writeln(' Pascal');
  End;
End.
```

3. Perulangan Negatif

Perulangan negatif adalah perulangan dengan penghitung (*counter*) dari besar ke kecil atau dengan kata lain pertambahannya negatif. Perulangan negatif dapat dibentuk dengan menggunakan statemen **For DownTo Do**.

For variabel kontrol:=nilai awal **DownTo** nilai akhir **Do** statemen

Variabel kontrol, nilai awal, nilai akhir harus mempunyai tipe yang sama, yaitu bertipe integer.

Contoh ke-2:

```
Program pengulangan_negatip;
Var
  I, X2:Real;

Begin
  Clrscr;
  For I:=15 DownTo 1 Do
  Begin
    X2:=I*I;
    Write(I:5); Write(X2:5);
  End;
End.
```

4. Perulangan Tersarang

Perulangan tersarang (*nested loop*) adalah yang berada di dalam perulangan yang lainnya. Perulangan yang lebih dalam akan diproses terlebih dahulu sampai habis, kemudian perulangan yang lebih luar baru akan bertambah.

Contoh ke-3:

```
Program pengulangan_tersarang;
Uses Wincrt;
Var
  I, J:Integer;

Begin
  Clrscr;
  For I:=1 To 10 Do
  Begin
    For J:=1 To 5 Do
    Begin
      Write(I:3);
      Write(J:8);
    End;
    Writeln;
  End;
End.
```

C. Praktik

1. Tulislah program contoh ke-1, jalankan program tersebut. Kemudian perhatikan, dan catat hasilnya. Mengapa terjadi kesalahan? Cobalah perbaiki sampai benar. Bagaimana hasilnya sekarang? Jika masih salah cobalah perbaiki sampai benar! Jika sudah benar rekam dengan nama file MODUL2A.
2. Tulislah program contoh ke-2, jalankan program tersebut. Kemudian perhatikan, dan catat hasilnya. Mengapa terjadi kesalahan? Cobalah perbaiki sampai benar. Bagaimana hasilnya sekarang? Jika masih salah cobalah perbaiki sampai benar! Jika sudah benar rekam dengan nama file MODUL2B.
3. Tulislah program contoh ke-3, jalankan program tersebut. Kemudian perhatikan, dan catat hasilnya. Jika masih salah cobalah perbaiki sampai benar! Jika sudah benar rekam dengan nama file MODUL2C.

D. Pertanyaan

1. Tulislah hasil praktikum contoh ke-1, contoh ke-2, dan ke-3.
2. Tipe apakah yang disarankan untuk variabel dalam statemen **For**?
3. Mengapa soal praktik ke-1 dan ke-2 mengalami kesalahan? Jelaskan!

Soal Latihan Membuat Program #3

PENYELEKSIAN KONDISI

A. Tujuan

Siswa dapat menggunakan bentuk pemilihan kondisi dengan benar.

B. Dasar Teori

Suatu keadaan kadang-kadang harus diputuskan oleh suatu pemilihan kondisi yang benar. Untuk memilih suatu perintah pemilihan kondisi tersebut harus cermat karena berkaitan dengan tipe data variabel yang digunakan. IF THEN ELSE berlaku untuk semua tipe data sedangkan CASE OF hanya untuk tipe data integer atau character.

Berikut contoh program penggunaan pemilihan kondisi:

Contoh ke-1:

```
Program Pilihan1;
Uses WinCrt;
Var
    pilih      : string;
    ket        : string;
Begin
    CLRSCR;
    {tampilan}
    WRITELN('PILIH LAH SALAH SATU PILIHAN YANG DI SUKAI');
    WRITELN('A. WARNA PUTIH');
    WRITELN('B. WARNA HIJAU');
    WRITELN('C. WARNA KUNING');
    WRITELN('D. WARNA MERAH');

    {input data}
    WRITELN('PILIHAN ANDA [A-D] '); READLN(PILIH);

    {proses data}
    CASE PILIH OF
        'A','a':ket:='bersih, suci';
        'B','b':ket:='teduh, damai';
        'C','c':ket:='jaya, optimis';
        'D','d':ket:='berani, gagah';
    END;

    {output}
    WRITELN;
    WRITELN('TERNYATA ANDA MEMILIKI SIFAT : ',KET);
    READLN;
END.
```

Contoh ke-2:

```
Program Pilihan2;
Uses WinCrt;
Var
    pilih      : string;
    ket        : string;
Begin
    CLRSCR;
    {tampilan}
    WRITELN('PILIH LAH SALAH SATU PILIHAN YANG DI SUKAI');
    WRITELN('A. HARI SENIN');
    WRITELN('B. HARI JUMAT');
    WRITELN('C. HARI MINGGU');
    WRITELN('D. HARI RABU');
    {input data}
    WRITELN('PILIHAN ANDA [A-D] '); READLN(PILIH);

    {proses data}
    IF PILIH='A' THEN ket:='pendiam, pemberani';
    IF PILIH='B' THEN ket:='lembut hatinya';
    IF PILIH='C' THEN ket:='suka panas hati';
    IF PILIH='D' THEN ket:='pendiam, ulet';

    {output}
    WRITELN;
    WRITELN('TERNYATA ANDA MEMILIKI SIFAT : ',KET);
    READLN;
END.
```

C. Praktik

1. Tulislah program contoh ke-1, jalankan program tersebut. Kemudian perhatikan, dan catat hasilnya. Mengapa terjadi kesalahan? Cobalah perbaiki sampai benar. Bagaimana hasilnya sekarang? Jika masih salah cobalah perbaiki sampai benar! Jika sudah benar rekam dengan nama file MODUL2A.
2. Tulislah program contoh ke-2, jalankan program tersebut. Kemudian perhatikan, dan catat hasilnya. Mengapa terjadi kesalahan? Cobalah perbaiki sampai benar. Bagaimana hasilnya sekarang? Jika masih salah cobalah perbaiki sampai benar! Jika sudah benar rekam dengan nama file MODUL2B.

a. Buatlah listing program untuk masalah berikut ini:

Input : nama, kode barang, jumlah barang

Proses: Jika kode barang='m', nama barang='sabun mandi', harga:=950

Jika kode barang='c', nama barang='sabun cuci', harga:=2500;

Jika kode barang='p', nama barang='pasta gigi', harga 2730;

*jumlah harga:=jumlah barang*harga*

*Jika jumlah harga>10000 maka potongan:=10/100*jumlah harga;*

*Jika jumlah harga>5000 maka potongan:=5/100*jumlah harga;*

Jika jumlah harga<=5000 maka potongan:=0;

jumlah bayar:=jumlah harga-potongan;

Output : jumlah harga, potongan dan jumlah bayar

Rekam dengan nama file MODUL2C

D. Pertanyaan

Semua pertanyaan berikut harus dijawab dalam laporan resmi!

1. Mengapa soal praktik ke-1 mengalami kesalahan? Jelaskan!

Soal Latihan Membuat Program #4

MATRIK

A. Tujuan

Siswa dapat membuat matrik dengan berbagai variasi indeks dengan benar.

B. Dasar Teori

2. Definisi Matrik

Matrik adalah sekumpulan informasi yang setiap datanya mengacu pada duah buah indeks. Matrik disebut juga tabel lajur yang memiliki indeks baris dan indeks kolom. Karena adanya dua buah indeks tersebut, matrik juga disebut larik dua dimensi. Istilah dua dimensi karena data tersebut ada pada pertemuan baris dan kolom.

Contoh:

	Fisika	Kimia	Biologi	Matematika
Ahmad	6	7	7	8
Joni	7	6	6	5
Toni	8	4	5	6
Dede	6	7	8	6

Tabel di samping disebut matrik 4x4, karena memiliki 4 baris dan memiliki 4 kolom atau larik 4x4.

Cara menentukan nilai elemen data tabel dengan matrik sebagai berikut:

- Nilai Fisika punya Ahmad ada pada baris 1 kolom 1, maka matriksnya [1,1]
- Nilai Biologi punya Joni ada pada baris 2 kolom 3, maka matriksnya [2,3]
- Nilai Kimia punya Dede ada pada baris 4 kolom 2, maka matriksnya [4,2]
- dst.

3. Definsi Larik

Suatu larik (*array*) adalah tipe terstruktur yang terdiri dari sejumlah komponen-komponen yang mempunyai tipe sama. Suatu larik mempunyai jumlah komponen yang banyaknya tetap. Banyaknya komponen dalam suatu larik ditunjukkan oleh suatu indkes yang disebut tipe indeks. Tipe indeks ini berbentuk ungkapan tipe ordinal (tipe data yang berurutan). Tiap-iap komponen pada larik dapat diakses dengan menunjukkan nilai indeksnya.

4. Dimensi Larik

a. Larik dimensi satu

Larik ini memiliki data yang berurutan dengan satu jenis yang sama dan tidak berbentuk tabel.

Contoh: larik **bunga** datanya *mawar, melati, angrek*

larik **nama hari** datanya *Minggu, Senin, Selasa, Rabu, Kamis, Jumat, Sabtu*

Deklarasi larik satu dimensi

```
bunga : array[1..3] of string=('mawar,melati,angrek')
```

nama larik kata cadangan untuk larik **jumlah elemen data** tipe data larik **data larik**
elemen-elemen data larik bunga adalah:

```
bunga[1]: 'mawar'  
bunga[2]: 'melati'  
bunga[3]: 'angrek'
```

} nilai data larik

nama larik indeks/urutan dalam larik

b. *Larik dua dimensi*

Larik dua dimensi disebut juga matriks. Larik dua dimensi memiliki data yang tersusun berdasarkan urutan baris dan kolom pada tabel.

Contoh:

Larik nilai memiliki data urutan baris yaitu: Ahmad, Joni, Toni, Dede dan data urutan kolom yaitu fisika, kimia, biologi, matematika. Data di atas dapat digambarkan sebagai berikut:

		Fisika	Kimia	Biologi	Matematika
		1	2	3	4
Aahmad	1	6	7	7	8
Joni	2	7	6	6	5
Toni	3	8	4	5	6
Dede	4	6	7	8	6

Dari data tabel di atas dapat dibuat indeksnya yaitu:

- Nilai[Ahmad,Fisika] atau Nilai[1,1] = 6
- Nilai[Ahmad,Kimia] atau Nilai[1,2] = 7
- Nilai[Ahmad,Biologi] atau Nilai[1,3] = 7
- Nilai[Ahmad,Matematika] atau Nilai[1,4]=8
- Nilai[Joni,Fisika] atau Nilai[2,1] = 7
- Nilai[Joni,Kimia] atau Nilai[2,2] = 6
- dst.

C. Deklarasi Larik dua dimensi

nama larik:array[minbaris..maxbaris,minkolom..maxkolom] of tipe larik

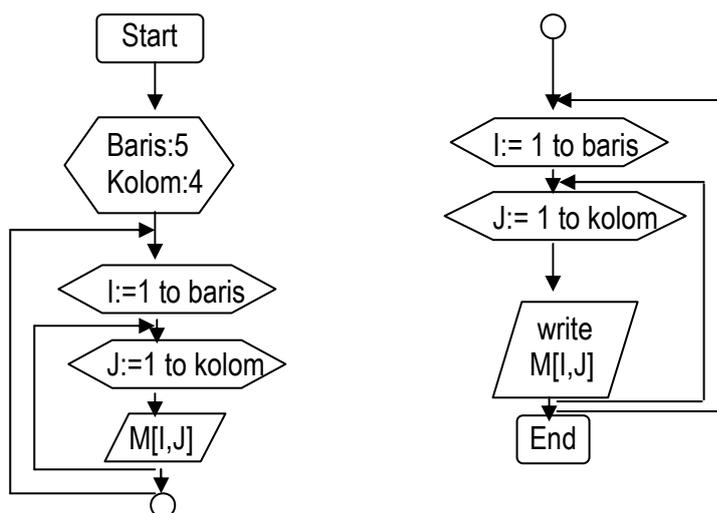
Contoh:

Nilai:array[1..4,1..4] of integer;

Cara Membuat Matrik

[] *Membuat satu buah matrik*

- Tentukan jumlah baris
- Tentukan jumlah kolom
- Isi data mulai dari baris pertama kolom pertama s.d. baris pertama kolom terakhir dan sampai dengan baris terakhir kolom pertama s.d. baris terakhir kolom terakhir.
- Jika pengisian sudah selesai tampilkan matriks yang sudah diisi.



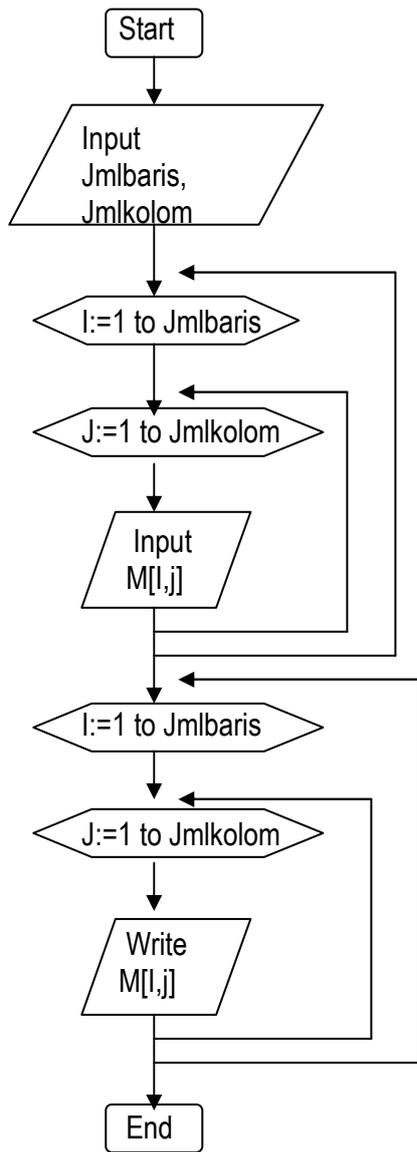
Contoh ke-1 listing program mambuat matrik:

```
Program Matrik1;
Uses WinCrt;
Var
    I,J      :Integer;
    Matrik   :array[1..5,1..4] of integer;
Begin
    ClrScr;
    {Bagian pemasukan data matriks}
    {pengulangan jumlah baris}
    FOR I:= 1 TO 5 DO
    BEGIN
        {pengulangan jumlah kolom}
        FOR J:=1 TO 4 DO
        BEGIN
            {pemasukan data matriks}
            WRITE('Data matriks ',I,j,' : ',);
            Readln(Matrik[I,j]);
        END;
    END;
    {Bagian pencetakan matriks}
    {pengulangan jumlah baris}
    FOR I:= 1 TO 5 DO
    BEGIN
        {pengulangan jumlah kolom}
        FOR J:=1 TO 4 DO
        BEGIN
            {pencetakan matriks}
            WRITE(Matrik[I,j]:3);
        END;
        {ganti baris pada matriks}
        WRITELN;
    END;
END.
```

[] Membuat matriks dengan jumlah baris dan jumlah kolom ditentukan ketika program dijalankan (menunggu pemasukan jumlah baris dan jumlah kolom dari operator komputer)

- Tanyakan jumlah baris dan jumlah kolom.
- Siapkan pengulangan sesuai jumlah baris dan jumlah kolom
- Isi data sesuai matriks yang dibuat
- Tampilkan elemen data matriks sesuai matriks yang dibuat.

Flowchart membuat matriks



Contoh program ke-2:

```

Program Matrik2;
Uses WinCrt;
Var
  Jmlbar, Jmlkol,
  I, J      :Integer;
  Matriks   :array[1..10,1..10]of integer
Begin
  ClrScr;

  {BAGIAN PEMASUKAN DATA MATRIKS}

  {Menentukan jumlah baris}
  Write('Jumlah Baris: ');Readln(jmlbar);
  {Menentukan jumlah kolom}
  Write('Jumlah Kolom: ');Readln(jmlkol);

  {Pengulangan baris}
  For I:=1 to Jmlbar do
  Begin
    {Pengulangan kolom}
    For J:=1 to Jmlkol do
    Begin
      {Pemasukan data matriks}
      Write('Data ke-',I,',',J,': ');
      Readln(Matriks[I, j]);
    End;
  End;

  {BAGIAN PENCETAKAN DATA MATRIKS}

  {Pengulangan baris}
  For I:=1 to Jmlbar do
  Begin
    {Pengulangan kolom}
    For J:=1 to Jmlkol do
    Begin
      {Pencetakan data matriks}
      Write(Matriks[I, j]);
    End;
    {ganti baris pada matriks}
    Writeln;
  End;
End;
End.
  
```

E. Praktik

1. Salinlah contoh program ke-1 dan ke-2 ke dalam bahasa pemrograman Turbo Pascal, amati hasilnya, jika terdapat kesalahan betulkan kesalahan tersebut. Jika sudah betul cetak hasilnya ke kertas!
2. Buatlah flowchart dan listing program untuk membuat matriks berikut ini:

a. 1 2 3
4 5 6
7 8 9

Rekam dengan nama file **MATRIKS3.PAS**

- b. Dua buah matriks sekaligus yang tersusun ke bawah hasilnya sebagai berikut:

matriks pertama

3 2 1 5
6 5 4 6
7 8 9 7

matriks kedua

5 4 9
3 6 7
8 2 3

Rekam dengan nama file **MATRIKS4.PAS**

- c. Dua buah matriks sekaligus yang tersusun ke kanan, hasilnya sebagai berikut:

matriks pertama

1 9 3
4 6 8
5 7 2
2 3 4

matriks kedua

4 5 7
9 1 6
2 3 8
7 4 5

Rekam dengan nama file **MATRIKS5.PAS**

Catatan:

Untuk membuat dua buah matriks sekaligus dengan cara: masukan data matriks pertama sampai selesai, masukan data matriks kedua sampai selesai, cetak data matriks pertama kemudian cetak data matriks kedua (untuk dua matriks yang tersusun ke bawah), cetak matriks pertama dan matriks kedua secara bersamaan (untuk dua matriks yang tersusun ke kanan).

F. Pertanyaan dan Tugas

1. Dapatkah elemen data matriks bernilai negatif? Jelaskan!
2. Apakah perbedaan lariks berdimensi satu dengan berdimensi 2?
3. Buatlah 3 contoh matriks lengkap dengan elemen datanya yang Anda ketahui dalam kehidupan sehari-hari!