

MODUL PRAKTEK
ALGORITMA DAN PEMROGRAMAN



PROGRAM STUDI D3 INSTRUMENTASI DAN ELEKTRONIKA
JURUSAN FISIKA
FAKULTAS SAINS DAN MATEMATIKA
UNIVERSITAS DIPONEGORO
SEMARANG

Peraturan Praktikum :

1. Praktikan harus menghadiri setiap sesi praktikum, tidak hadir tanpa keterangan akan menyebabkan hak menjadi praktikan gugur (nilai E).
2. Sebelum dan setelah praktikum dimungkinkan adanya pretset dan post test.
3. Laporan praktikum dianggap sah apabila praktikan telah melakukan asistensi sesuai dengan materi praktikum.
4. Laporan resmi adalah gabungan dari semua laporan praktikum tiap sesi.

Grade Penilaian Praktikum :

1. Kehadiran = 15% → asisten
 2. Pretest & post test = 20% → asisten
 3. Asistensi dan laporan praktikum = 30% → asisten
 4. Laporan resmi = 35% → dosen
-

DAFTAR ISI

MODUL 1: STRUKTUR PROGRAM PASCAL
MODUL 2: PENYELEKSIAN KONDISI
MODUL 3: LOOPING (PERULANGAN)
MODUL 4: TIPE ARRAY BERDIMENSI SATU
MODUL 5: TIPE ARRAY BERDIMENSI DUA
MODUL 6: PROCEDURE DAN FUNGSI

MODUL I

STRUKTUR PROGRAM PASCAL

I. TUJUAN

Mahasiswa mengenal struktur program bahasa PASCAL dan program sederhana dengan bahasa PASCAL.

II. TEORI SINGKAT

a. Editor Pascal

Editor TURBO PASCAL merupakan pelayanan terpadu antara editor dengan compiler. Untuk mengaktifkan editor TURBO PASCAL dapat dilakukan dengan :

- Aktifkan komputer dalam keadaan prompt C:\> (DOS)
- Masuk ke direktori TURBO PASCAL
- Masuk ke direktori BIN
- Ketikkan TURBO
C:\> CD TP
C:\TP\CD BIN
C:\TP\BIN\ TURBO

Sehingga akan muncul jendela editor TURBO PASCAL deengna menu antara lain : File, Edit, Search, Run, compile, Debug, Tools. Options, Window, Help.

Operasi-operasi pada editor PASCAL :

1. Untuk menciptakan lembar kerja baru → File, New
2. Untuk menyimpan hasil pekerjaan → File, Save/Save as, ketikkan nama filenya (Tekan tombol F2)
3. Untuk membuka file data → File, Open, pilih file data, Tekan tombol Open (Tekan tombol F3)
4. Untuk berpidah ke direktori kerja → File, Change dir..., Tentukan direktori kerja, Tekan tombol OK
5. Untuk keluar ke prompt DOS sementara → File, DOS shell
6. Untuk kembali ke editor PASCAL → ketikkan Exit
7. Untuk menjalankan program PASCAL → Run, Run (Ctrl+F9)
8. Untuk melihat hasil RUN → Alt+F

b. Struktur Program PASCAL :

PROGRAM <i>NamaKepalaProgram</i> ;	→ Optional
LABEL	
< <i>Deklarasi Label</i> >	→ Optional
CONST	
< <i>Deklarasi Konstantal</i> >	→ Optional
TYPE	
< <i>Deklarasi Tipe</i> >	→ Optional
VAR	
< <i>Deklarasi Variabell</i> >	→ Optional
< <i>Deklarasi Procedure dan Fuction</i> >	→ Optional
Begin	
< <i>Pernyataan</i> >	
End.	

NamaKepalaProgram

Dipakai untuk keperluan dokumentasi. Menunjukkan dari identifikasi program yang akan dibuat. Nama kepala program harus dimulai dengan huruf, dapat terdiri dari angka dan karakter _ (garis bawah)dan tidak boleh ada spasi. *NamaKepalaProgram* mempunyai aturan sama dengan *Namavariabel*. *Namakonstanta*, *NamaTypeData*, *NamaProcedure*, *NamaFuction*.

c. Deklarasi Variabel

Sebelum suatu variabel digunakan maka terlebih dahulu harus dipesan dalam blok deklarasi variabel dengan bentuk :

```
Var  NamaVar : Typevariabelnya;  
     NamaVar, NamaVar, ... : Typevariabelnya;
```

Jika menggunakan variabel yang belum di deklarasikan maka saat program di RUN (dikompilasi) akan menyebabkan kesalahan dengan muncul pesan “Unknown identifier”.
Selanjutnya variabel bisa diisi data dengan statemen assingment.

```
NamaVariabel := data;
```

:= adalah operator untuk pengisian variabel dengan nilai data.

Dalam PASCAL dikenal banyak type yang bisa digunakan untuk mendeklarasikan type suatu data.

Type	Kisaran nilai
<u>Type sederhana :</u>	
Byte	0 .. 255
Word	0 .. 65535
ShortInt	-128 .. 127
Integer	-32768 .. 32767
Longint	-2147483648 .. 2147483648
Real	1 E-38 .. 1 E+28
Single	1.5 E-45 .. 3.4 E+38
Double	5.0 E-324 .. 1.7E+308
Extended	1.9 E-4951 .. 1.1 E+4932
Boolean	Kondisi True atau False
Char	‘a’, ‘b’, .., ‘z’, ‘0’ .. ‘9’ → semua
String	karakter
<u>Type terstruktur</u>	‘rangkaian karakter’
:	
Array	larik karakter, numerik
Record	numerik, string, char
File	numerik, string, larik, record
Set	data ordinal
Pointer	alamat memori

Type diatas adalah type standard bawaan PASCAL (sudah didefinisikan oleh PASCAL). Kita dapat membuat/mendefinisikan sendiri type sesuai dengan keinginan kita dengan cara mendeklarasikan pada bagian type.

```
Bentuk :  
      Type  
      NamaTypeData = typedata;
```

selanjutnya NamaTypeData yang sudah kita definisikan bisa kita pakai sebagai type data pada variabel yang kita gunakan.

d. Operator

Dalam PASCAL dikenal beberapa operator antara lain : Operator aritmetika, Operator string, Operator logika.

Operator aritmetika

/ pembagian
 * perkalian
 div pembagian integer
 mod sisa hasil pembagian integer
 + - tambah dan kurang

di dalam PASCAL tidak dikenal operator pangkat. apabila dibutuhkan dapat dibuat untuk operasi pangkat.

operator string

Hanya ada satu operator string yaitu + yang artinya rangkai. Misalnya 'Sistem' + 'Informasi' → 'Sistem Informasi'.

operator logika

operator logika yang terpenting adalah NOT, AND dan OR

e. Deklarasi Konstanta

Konstanta adalah suatu nilai yang tidak dapat diubah dalam tubuh program dan harus dideklarasikan di bagian deklarasi konstanta sebagai berikut :

```
Const
    NamaKonstanta1 = NilaiKonstanta;
```

f. Komentar Program

Komentar adalah sesuatu yang tidak akan diproses pada saat program dijalankan. Pemberian komentar ditandai dengan :

```
{komentar ... komentar}
```

g. Deklarasi Unit

Unit adalah suatu kumpulan prosedur dan fungsi yang disediakan oleh PASCAL yang disesuaikan dengan tujuannya. Dalam PASCAL dikenal beberapa Unit standard antara lain Crt, Dos, Graph, Printer, Dos.

Contoh Unit yang tersedia dalam Unit Crt :

Clrscr : Untuk membersihkan layar

GotoXY : Untuk memposisikan kursor pada kolom X dan baris Y yang di layar

Cara mendeklarasikan Unit adalah :

```
Uses NamaUnit, NamaUnit, ...;
```

h. Prosedur Read dan Readln

Prosedur ini digunakan untuk membaca data dari keyboard dengan memakai variabel yang sudah dideklarasikan dalam deklarasi variabel. Sintaknya :

```
Read(NamaVariabel, NamaVariabel, ...);
Readln(NamaVariabel, NamaVariabel, ...);
```

Read : Tidak akan menurunkan kursor setelah proses pembacaan

Readln : menurunkan kursor setelah proses pembacaan

i. Prosedur Write dan Writeln

Write : Mencetak dan kursor disebelahnya

Writeln : Mencetak dan kursor satu baris ke baris awal

III. PETUNJUK PRAKTIKUM

a. Membuat program untuk input dan output

```
{
  Contoh program menginputkan data
  dengan perintah read atau readln
  kemudian menampilkannya dengan
  perintah write atau writeln
}
Program InputData;
uses Crt;
Var
  no_mhs : integer;
  nama, alamat : string;
Begin
  clrscr;
  write('Nomor mahasiswa : ');readln(no_mhs);
  write('Nama Mahasiswa : ');readln(nama);
  write('Alamat : ');readln(alamat);
  clrscr;
  write('Dia adalah ',nama);
  writeln(' yang mempunyai nomor mahasiswa
  ',no_mhs);
  writeln('beralamat di ',alamat,' Semarang');
  readln;
end.
```

b. Program untuk menghitung luas lingkaran :

```
Program lingkaran;
const
  phi = 3.14;
var
  jari, luas : real;

Begin
  Jari := 10.0;
  luas := phi * jari * jari;
  writeln('H A S I L');
  writeln('-----');
  writeln('Jari-jari lingkaran :
  ',jari:6:2);
  writeln('Luas lingkaran :
  ',luas:6:2);
end.
```

IV. TUGAS

1. Buatlah program untuk menghitung keliling lingkaran

MODUL II PENYELEKSIAN KONDISI

I. TUJUAN

Mempelajari struktur kondisi dalam PASCAL yaitu Struktur If ... Then ..., If ... Then ... Else ... dan Case ... Of ...

II. TEORI SINGKAT

a. If ... Then ... :

Statemen If ... Then ... mempunyai syntax :

If <syarat_logika> Then Statemen

Atau

```

If <syarat_logika> Then
  Begin
    Statemen1;
    Statemen2;
    ..... ;
  End;
```

} disebut blok statemen

<syarat_logika> adalah operasi relasi (seleksi kondisi) yang apabila bernilai TRUE (benar) maka statemen atau blok statemen setelah Then akan dilaksanakan. Apabila statemen yang berkaitan dengan nilai TRUE lebih dari satu statemen maka blok statemen dipergunakan, jika hanya terdiri dari satu statemen maka blok statemen tidak perlu dipergunakan. Blok statemen ditandai dengan *Begin ...End;*

b. If ... Then ...Else ... :

Mirip dengan syntax If ... Then, tata aturan if then else adalah :

If <syarat_logika> Then Statemen1 Else Statemen2;

Atau

```

If <syarat_logika> Then
  Begin
    Statemen1;
    Statemen2;
    ... ;
  End
Else
  Begin
    Statemen1;
    Statemen2;
    ... ;
  End;
```

} blok jika syarat True

} blok jika syarat False

Jika <syarat_logika> bernilai TRUE maka statemen atau blok statemen setelah Then akan dikerjakan, jika <syarat_logika> bernilai FALSE maka statemen atau blok statemen setelah Else akan dikerjakan.

c. Statemen Case NamaVar Of :

Statemen Case NamaVar Of dipakai untuk penyeleksian berganda. NamaVar adalah nama variabel yang harus bertipe ordinal seperti Char, byte atau integer.

Padanan dengan If ... Then , misalnya kita memiliki variabel *Pil* yang bertipe Char, dan dihadapi seleksi pilihan :

```

If Pil = 'a' Then Statemen2;
If Pil = 'b' Then Statemen2;
Dst
```

Dapat ditulis dengan statemen Case ... Of dengan syntax sbb :


```

Case Pil Of
  'a' : Statemen1;
  'b' : Statemen2;
  .... : ...;
End;

```

V. PETUNJUK PRAKTIKUM

a. Membuat program menggunakan if

```

Program TentukanLulus;
Uses Crt;
Label
  Mulai;
Var
  NamaSis   : String[35];
  Nilai     : Real;
  Ket,Pesan : String[35];
  Jawab     : Char;
Begin
  Mulai :
    ClrScr;
    Write('Nama Siswa   : ');Readln(NamaSis);
    Write('Nilai Angka  : ');Readln(Nilai);
    If (Nilai >= 60) Then
      Begin
        Ket   := 'Anda Lulus';
        Pesan := ' SELAMAT, PERTAHANKAN TERUS !';
      End
    Else
      Begin
        Ket   := 'Anda Tidak Lulus';
        Pesan := ' TINGKATKAN BELAJARNYA!';
      End;
    Writeln;
    Writeln('Hasil : ');
    Writeln(Ket,Pesan);
    Writeln;
    Write('Mau mencoba lagi [Y/T] ? ');Readln(jawab);
    If (Jawab = 'Y') or (Jawab = 'y') Then
      Goto mulai
End.

```

b. Membuat program menggunakan case

```

Program TekanTombol;
Uses Crt;
Label
  Mulai;
Var
  Tombol,
  Jawab : Char;
Begin
  Mulai:
    ClrScr;

```

```

Writeln('PROGRAM TENTUKAN TOMBOL');
Writeln('-----');
Write('Tekan Sembarang TOMBOL : ');
Readln(Tombol);
Case Tombol Of
  'a'..'z','A'..'Z': Begin
    Writeln('Anda menekan tombol ',Tombol);
    Writeln('Tombol ',Tombol,' Termasuk HURUF');
    End;
  '0'..'9'      : Begin
    Writeln('Anda menekan tombol ',Tombol);
    Writeln('Tombol ',Tombol,' Termasuk ANGKA');
    End;
Else
    Begin
    Writeln('Anda menekan tombol ',Tombol);
    Writeln('Tombol ',Tombol,' Bukan HURUF/ANGKA');
    End;
End;
Write('Mau mencoba lagi [Y/T] ? ');Readln(jawab);
  If (Jawab = 'Y') or (Jawab = 'y') Then
    Goto mulai
End.

```

IV. TUGAS

1. Buatlah program menggunakan perintah if.. Then... untuk konversi angka ke hari, jika
1→senin, 2→selasa, 3→rabu, 4→kamis, 5→jumat, 6→sabtu. 7→minggu
2. Buatlah soal 1 menggunakan perintah case

MODUL III LOOPING (PERULANGAN)

I. TUJUAN

Mempelajari struktur dari statemen kontrol PASCAL untuk menangani proses berulang yaitu : For .. Do, While.. Do, dan Repeat .. Until.

II. TEORI SINGKAT

a. For .. Do ..

Syntax for .. Do

```
For NamaVarKendali := Awal to Akhir Do
    Pernyataan;
```

Atau

```
For NamaVarKendali := Awal to Akhir Do
    Begin
        Pernyataan1;
        Pernyataan2;
        ...
    End;
```

} blok pernyataan

Atau

```
For NamaVarKendali := Awal Downto Akhir Do
    Pernyataan;
```

Atau

```
For NamaVarKendali := Awal Downto Akhir Do
    Begin
        Pernyataan1;
        Pernyataan2;
        ...
    End;
```

} blok pernyataan

Statemen/pernyataan atau blok pernyataan akan dikerjakan selama nilai dari *NamaVarKendali* dari *Awal* ke *Akhir* dengan kenaikan/penurunan satu langkah. Jika *Awal* lebih kecil *Akhir* kita pakai *to*, sedangkan jika *Awal* lebih besar *Akhir* kita pakai *downto*.

NamaVarKendali adalah *idetifier* yang bertype ordinal seperti byte, integer atau char dan tidak dapat bertype string atau real. Berikut adalah program untuk mencetak karakter mulai dari awal sampa akhir dan dari akhir sampai awal, diman awal dan akhir diinputkan dari keyboard. Awal, akhir bertype char.

b. While .. Do ..

Kontrol For .. Do .. kita pakai untuk memproses dengan jumlah ulangan diketahui sebelumnya. Apabila proses yang diulang tidak kita ketahui berapa putaran harus dilakukan, yaitu diputar selama kondisi tertentu dipenuhi, maka statemen yang kita pakai adalah While .. Do ..

Syntaxnya :

```
While <syarat_logika> Do
    <pernyataan>;
```

atau

```
While <syarat_logika> Do
    Begin
        Pernyataan1;
        Pernyataan2; } blok pernyataan
        ...
    End;
```

<syarat_logika> adalah suatu relasi yang selama relasi tersebut bernilai benar (TRUE) maka pernyataan/blok pernyataan akan terus menerus dikerjakan. Pernyataan/blok pernyataan akan berhenti dikerjakan jika <syarat_logika> tidak lagi bernilai benar.

c. Repeat .. Until ..

Statemen Repeat .. Until .. kita pakai apabila kita akan mengulang-ulang suatu proses terus menerus sampai kondisi tertentu dipenuhi.

Syntax :

```
Repeat
    Pernyataan1;
    Pernyataan2;
    ...
Until <kondisi>;
```

Pada contoh program diatas dapat kita modifikasi dengan memakai Repeat .. Until. Konstanta boolean *keyPressed* adalah konstanta yang bernilai FALSE selama kita tidak menekan sesuatu dan jika kita menekan sesuatu maka *KeyPressed* akan bernilai TRUE.

VI. PETUNJUK PRAKTIKUM

a. Membuat program menggunakan for

```
Program CetakKarakter;
Uses Crt;
Var
    Awal, Akhir, Karak : Char;
Begin
    Clrscr;
    Write('Karakter awal  : ');Readln(Awal);
    Write('Karakter akhir : ');Readln(Akhir);
    writeln;
    Write('N A I K      : ');
    For Karak := Awal to Akhir Do
        Write(karak);
    Writeln;
    Write('T U R U N   : ');
    For Karak := Akhir downto Awal Do
        Write(karak);
```

```

    Readln
End.

```

b. Membuat program menggunakan while

Contoh berikut akan mencetak karakter '*' terus menerus pada posisi baris kolom yang acak, setiap kita menekan suatu sembarang tombol. Proses berulang selam yang dibaca <> #27 (Esc).

Untuk meletakkan pada baris klo yang acak, setiap dipakai fungsi Random(x). Fuction standar *Random(x)* adalah fungsi untuk membangkitkan bilangan acak antara nol sampai x.

Selanjutnya satu karakter dibaca dengan Procedure standar *ReadKey*, dengan syntax :

NamaVarKar := ReadKey;

Pembacaan data dari keyboard dengan *ReadKey* adalah untuk variabel bertipe Char dan bersifat tidak meng-echo-kan (menampilkan ke layar).

```

Program Acak_dg_While_Do;
Uses Crt;
Const
    Escape = #27;
Var
    Baca          : Char;
    Baris, Kolom  : Byte;

Begin
    Clrscr;
    Baca := ' ';
    While Baca <> Escape Do
        Begin
            GotoXY(34,12);Write('Tekan Escape Selesai ...');
            Baris := Random(23);
            Kolom := Random(80);
            GotoXY(Kolom,Baris);Write('*', ' Kolom ',Kolom, '
Baris ',Baris);
            Baca := ReadKey;Clrscr;
        End;
    End.

```

#27 adalah konstanta karakter. Dalam PASCAL ada 2 macam cara untuk memperoleh konstanta karakter, yaitu dengan #n atau Chr(n) dimana n = kode ASCII karakter yang bersangkutan. Misalnya karakter CR (ENTER) mempunyai kode ASCII 13 dapat ditulis dengan #13 atau Chr(13).

c. Membuat program menggunakan repeat

```

Program Acak_dg_Repeat_Until;
Uses Crt;
Var
    Baris, Kolom  : Byte;

Begin
    Clrscr;
    Repeat
        GotoXY(34,12);Write('Tekan Escape Selesai ...');

```

```
Baris := Random(23);  
Kolom := Random(80);  
GotoXY(Kolom,Baris);Write('*', ' Kolom ',Kolom, '  
Baris ',Baris);  
Delay(10);  
Until KeyPressed;  
End.
```

IV. TUGAS

1. Buatlah program untuk menambah angka ganjil dari 1 hingga 1001.

MODUL IV TIPE ARRAY BERDIMENSI SATU

I. TUJUAN

Mempelajari tipe terstruktur Array dimensi satu.

II. TEORI SINGKAT

Array merupakan tipe data terstruktur yang berguna untuk menyimpan sejumlah data yang bertipe sama, dimana masing-masing elemen array dapat diakses melalui indek array.

Array dimensi satu adalah array yang hanya memiliki satu tipe indek.

Deklarasi variabel bertipe array dimensi :

```
Var
    NamaArray : Array[tipeIndek] of TArray;
```

NamaArray adalah nama variabel bertipe array dan *TArray* adalah tipenya. *TArray* dapat berupa Integer, real, char, string atau record. Sedangkan *tipeIndek* dari array merupakan tipe ordinal seperti char, byte, integer, atau enumerasi.

Contoh deklarasi tipe array :

```
Type
    NanaArray = (senin,selasa,rabu,kamis,jumat,sabtu,minggu);
Var
    Hari : DafHari;
    Incom : Array [DafHari] of integer;
    Masuk : Array [senin .. kamis] of integer;
```

DafHari adalah NamaType enumerasi yang kita jadikan indeks dari array *Incom* dan *Masuk*. *Incom* dan *Masuk* adalah array integer dengan indek enumerasi. Selanjutnya kita dapat mengisi elemen array dengan *Incom[senin]* atau *Masuk[rabu]* dan seterusnya.

Data bertipe string sesungguhnya adalah suatu larik data bertipe char.

Misalnya :

```
Var
    Pesan : string[20];
```

Variabel *Pesan* merupakan string. Kita juga bisa memandangnya sebagai larik char. *Pesan[1]* adalah karakter pertama, *Pesan[2]* adalah karakter kedua, dst. Sedangkan *Pesan[0]* adalah karakter yang kode ASCII-nya adalah cacah karakter dari isi pesan. Dengan demikian kita dapat mengetahui ada berapa cacah karakter isi dari *Pesan* dengan *Ord(Pesan[0])*, disamping dapat juga dipakai fungsi *Length(Pesan)*.

III. PETUNJUK PRAKTIKUM

Contoh berikut adalah membaca variabel pesan yang bertipe string yang secara otomatis dapat dipandang sebagai larik karakter dengan *Ord(Pesan[0])* dapat diperoleh cacah karakter yang disimpan dalam *CcHuruf*.

Contoh Program 1 :

```
Uses Crt;
Const
    Vokal =
    ['A','I','U','E','O','a','i','u','e','o'];
Var
    Huruf : char;
    Pesan : string;
    i,CcHuruf,CcVokal : integer;
```

```

Begin
  clrscr;
  Write('Ketikkan Kalimat : ');
  Readln(Pesan);
  CcHuruf := Ord(Pesan[0]);
  CcVokal := 0;
  For i := 1 to CcHuruf Do
    Begin
      Huruf := Pesan[i];
      If Huruf in Vokal then
        CcVokal := CcVokal + 1;
      End;
    End;
  Writeln('DIBALIK');
  For i := CcHuruf downto 1 do
    write(Pesan[i]);
  Writeln;
  Write('VOKAL : ',CcVokal,' BUAH');
  Readln
End.

```

Contoh Program 2 :

```

program Array_max_min;
var
  nilai:array[1..5] of INTEGER;
  maks,mins,total,i,data : integer;
  rata : real;
begin
  write('Jumlah Data = ');readln(data);
  write('Data ke-1 = ');readln(nilai[1]);
  maks :=nilai[1];
  mins :=nilai[1];
  total :=nilai[1];
  for i := 2 to data do
    begin
      write('Data ke-',i,' = ');readln(nilai[i]);
      if nilai[i]>maks then
        maks:=nilai[i];
      if nilai[i]<mins then
        mins:=nilai[i];
      total:=total+nilai[i];
    end;
  rata:=total/data;
  writeln('Nilai terbesar adalah = ',maks);
  writeln('Nilai terkecil adalah = ',mins);
  writeln('Nilai total adalah = ',total);
  writeln('Nilai rata-rata adalah = ',rata:0:2);
  readln;
end.

```

IV. TUGAS

1. Buatlah program untuk mengurutkan suatu data angka yang dimasukkan dari keyboard, secara urut menaik dan menurun. Gunakan variabel bertipe array.

MODUL V

TIPE ARRAY BERDIMENSI DUA

I. TUJUAN

Mempelajari tipe terstruktur Array dimensi dua.

II. TEORI SINGKAT

Array merupakan tipe data terstruktur yang berguna untuk menyimpan sejumlah data yang bertipe sama, dimana masing-masing elemen array dapat diakses melalui indek array.

Array dimensi dua adalah array yang memiliki dua tipe indek.

Deklarasi variabel bertipe array dimensi dua :

```
Var
    NamaArray : Array{tipeIndek1,tipeIndek2} of TArray;
```

III. PETUNJUK PRAKTIKUM

Penggunaan secara umum untuk matrik dua dimensi adalah untuk memasukkan data matrik. Berikut penggalan program untuk memasukkan data matrik dengan ordo 2 x 2

```
Prog Matrik2Dimensi;
Use Crt;
var
    Matrik      : Array[1..10,1..10] Of integer;
    Baris,Kolom : Integer;
Begin
    ClsRcr;
    {-Memasukkan data matrik-}
    For Baris := 1 to 2 Do
        Begin
            For kolom := 1 to 2 Do
                Begin
                    Write('Baris ',Baris,' Kolom ',Kolom,' : ');
                    Readln(Matrik(Baris,Kolom));
                End;
            End
        End
    End;
    {--Menampilkan Hasil--}
    For Baris := 1 to 2 Do
        Begin
            For kolom := 1 to 2 Do
                Begin
                    Write(Matrik(Baris,Kolom):5);
                End;
            End;
        End;
    Readln
End.
```

IV. TUGAS:

1. Buatlah program untuk menghitung penjumlahan dua buah matrik. Syarat penjumlahan dua buah matrik adalah kedua matrik tersebut mempunyai jumlah baris dan kolom yang sama.

Misalnya :

$$\begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix} + \begin{bmatrix} 2 & 2 \\ 3 & 2 \end{bmatrix} = \begin{bmatrix} 3 & 4 \\ 5 & 5 \end{bmatrix}$$

MODUL VI PROCEDURE DAN FUNGSI

V. Tujuan

Mempelajari pemecahan program ke dalam modul-modul program dalam bentuk prosedur dan fungsi

VI. Teori Singkat

a. PROCEDURE

Procedure merupakan sub program yang berdiri sendiri yang dapat dipanggil dari program utama.

Deklarasi prosedur :

```
PROCEDURE <Nama_prosedur> (<Daftar_parameter_formal>);  
<Bagian_deklarasi>  
Begin  
    <Bagian _pernyataan>  
End;
```

Dalam prosedur atau fungsi dikenal adanya parameter. Parameter adalah suatu nilai atau variabel yang akan dilewatkan dalam prosedur atau fungsi. Pengiriman paramter dalam prosedur atau fungsi dibedakan menjadi 2 yaitu parameter nilai dan parameter referensi. Pelewatan parameter secara nilai dimaksudkan jika parameter yang dileatkan dimasukkan tidak untuk diubah oleh prosedur atau fungsi sekalipun perubahan telah terjadi di dalam prosedur atau fungsi tersebut. Sedangkan parameter referensi dimasukkan untuk mengubah nilai atau variabel seluruhnya dari prosedur atau fungsi. Perbedaan cara pengiriman parameter didalam prosedur atau fungsi terletak pada deklarasi prosedur atau fungsinya. Contoh deklarasi prosedur dengan pengiriman parameter secara refensi :

```
Procedure Tukar(var x,y : real);
```

Untuk mengubah sifat pengiriman parameter secara nilai, tinggal diubah deklarasi prosedur dengan menghilangkan var dalam kepala prosedur sehingga menjadi :

```
Procedure Tukar(x,y : real);
```

b. FUNCTION

Fungsi memiliki struktur yang hampir sama dengan procedure, bedanya fungsi memiliki tipe hasil sedangkan procedure tidak.

Deklarasi fungsi :

```
FUNCTION <Nama_fungsi> (<Daftar_parameter>): tipe;  
<Bagian_deklarasi>  
Begin  
    <Bagian _pernyataan>  
End;
```

VII. PETUNJUK PRAKTIKUM

a. Berikut contoh program pemakaian untuk menukarkan dua nilai memakai prosedur :

```
program tukar_data;  
var  
    a, b : real;           {deklarasi variabel global}  
  
procedure info;  
begin
```

```
writeln('*****');
writeln('**    Program Tukar Data    **');
writeln('**    Dengan Procedure    **');
writeln('**          Oleh      :      **');
writeln('**    Ki Joko Bodho      **');
writeln('*****');
end;
{Prosedur penukaran data}
procedure tukar(var x,y : real);
var
    tamp : real;
begin
    tamp := x;
    x     := y;
    y     := tamp;
end;
{Program utama}
Begin
    info;
    a := 12.33;
    b := 67.98;
    writeln('Sebelum memanggil prosedur Tukar');
    writeln('a = ',a:7:2,'    b = ',b:7:2);
    tukar(a,b);
    writeln('Setelah memanggil prosedur Tukar');
    writeln('a = ',a:7:2,'    b = ',b:7:2);
    readln;
End.
```

b. Berikut contoh program memakai fungsi:

```
program kali_data;
var
    a, b : integer;           {deklarasi variabel global}

procedure info;
begin
    writeln('*****');
    writeln('**    Program Tukar Data    **');
    writeln('**    Dengan Procedure    **');
    writeln('**          Oleh      :      **');
    writeln('**    Ki Joko Bodho      **');
    writeln('*****');
end;
{Prosedur perkalian data}
function kali(x,y : integer):integer;
begin
    kali := x * y;
end;
{Program utama}
Begin
    info;
    a := 12;
    b := 67;
    writeln(a, ' x ',b, ' = ',kali(a,b));
```

```
writeln('12 x 13 = ',kali(12,13));  
readln;  
End.
```

VIII. TUGAS

1. Buatlah program dengan menggunakan prosedur untuk membalik suatu angka
Misalnya :
Masukkan angka : 12309
Hasil pembalikan : 90321
 2. Buatlah program dengan menggunakan fungsi untuk mencari jumlah dan rata-rata dari sejumlah data yang dimasukkan.
-