

## ***MEMULAI TURBO PASCAL***

*Dengan Hard Disk*

1. C:\Turbo\turbo
2. Tunggu Layar kerja.

## ***PERINTAH-PERINTAH TOMBOL***

## ***OPERASI BLOK***

secara cepat, pengcopyan, pemindahan teks dan lain- lain.

1. Letakkan kursor pada awal teks yang akan diblok
2. Tekan tombol SHIFT (jangan dilepas)
3. Sambil menekan tombol SHIFT, ikutilah dengan menekan tombol-tombol berikut:
  - a. Tekan anak panah kanan untuk memblok per karakter ke kanan.
  - b. Tekan anak panah kiri untuk memblok per karakter ke kiri.
  - c. Tekan tombol End untuk memblok hinga ke akhir baris.
  - d. Tekan tombol Home untuk memblok hinga ke awal baris.
  - e. Tekan anak panah atas untuk memblok ke baris atas.
  - f. Tekan anak panah bawah untuk memblok ke baris bawah.
4. Selanjutnya blok teks tersebut sudah dapat dicopy, dihapus, dipindahkan atau yang lain sesuai dengan keperluan anda.

Jika ingin membatalkan blok, tekan CTRL+KH, maka tanda blok akan hilang. Sebaliknya, tekan kembali CTRL+KH, maka tanda blok akan mencul kembali.

Mencopy blok adalah mengcopy teks yang diblok ke daerah lain, sehingga teks (dalam hal ini listing program) tidak diketik ulang.

1. Blok terlebih dahulu teks yang akan dicopy dengan acra yang telah diterangkan sebelumnya.
  2. Tekan Ctrl + Insert untuk mengcopy. {Maka blok teks akan dikirim ke Clipboard}
  3. Pindahkan kursor ke mana teks akan di copy
  4. Tekan Shift+ Insert.
  5. Maka teks akan dicopy dimana kursor berada.
  6. Selanjutnya tekan Ctrl+KH untuk menghilangkan tanda blok.
- 
1. Blok teks yang akan dihapus.
  2. Tekan Ctrl+KY.
  3. Maka seleuruh teks yang diblok terhapus.

- a. Mahasiswa mengerti tentang Tipe Data, pendeklarasian variabel, dan operasi aritmatika serta statement input dan output pada Turbo Pascal 7.0
- b. Mahasiswa dapat membuat program perhitungan sederhana menggunakan Turbo Pascal 7.0

- a. Tipe Data
- b. Identifier/Pengenal
- c. Operasi Aritmatika
- d. Operasi Input dan Output

a. Tipe Data

Dalam turbo pascal tipe data dibagi menjadi beberapa tipe yaitu :

1. Tipe Bilangan Bulat

Pada tipe bilangan bulat ini ada 5 (lima) macam tipe bawaan (defined) yang berkaitan dengan bilangan bulat yaitu : shortint, integer, longint, byte dan word. Adapun untuk range dan memory size dari masing masing tipe adalah sebagai berikut :

Tipe	Range	Memory Size
Shortint	-128 .. 127	1 byte
Integer	-32768 .. 32767	2 byte
Longint	-2147483648 .. 2147483647	4 byte
Byte	0 .. 255	1 byte
Word	0 .. 65535	2 byte

Untuk range bilangan heksadesimal dimulai dari \$00000000 \$FFFFFFFF. Sedangkan untuk penulisannya harus diawali dengan tanda \$.

2. Tipe Bilangan Real (Bilangan Pecahan)

Tipe real adalah tipe bilangan pecahan. Pada tipe real ini dapat dibedakan menjadi 5 (lima) macam tipe juga yaitu : real, single, double, extended, dan comp. Adapun untuk penjelasan masing masing tipe dapat dilihat pada tabel berikut ini :

Tipe	Range	Ketelitian	Memory size
Real	$\pm 2.9 \times 10^{-39} \dots 1.7 \times 10^{38}$	11 .. 12 digit	6 byte
Single	$\pm 1.5 \times 10^{-45} \dots 3.4 \times 10^{38}$	7 .. 8 digit	4 byte
Double	$\pm 5.0 \times 10^{-324} \dots 1.7 \times 10^{308}$	15 .. 16 digit	8 byte
Extended	$\pm 5.0 \times 10^{-4951} \dots 1.1 \times 10^{4932}$	19 .. 20 digit	10 byte
Comp	$-2^{63} \dots 2^{63} - 1$ ( $-9.2 \times 10^{18} \dots 1.1 \times 10^{18}$ )	19 .. 20 digit	8 byte

Untuk konstanta bilangan pecahan ini dapat ditulis dengan menggunakan tanda exponential E atau e. format penulisannya adalah sebagai berikut :

d.dddddddddE dddd

dimana : d menyatakan digit. Contoh :

1.6E5            1.6E05  
1.6E+5        1.6e+05

Keempat nilai diatas menyatakan nilai yang sama yaitu bilangan :  
1.6 x 10 4

### 3. Tipe Data Char

Tipe char ini merupakan tipe data yang hanya terdiri dari sebuah karakter (Huruf). Karakter ini dapat berupa sembarang karakter ASCII, dan untuk menyatakan tipe data ini harus diawali dan diakhiri dengan tanda petik tunggal/apostrof ('). Contoh :

'A' → Menyatakan karakter A

'1' → Menyatakan karakter angka 1

'>' → Menyatakan karakter simbol >

'"' → Menyatakan karakter apostrof

Beberapa karakter ASCII ada yang tidak bisa dinyatakan dengan tanda petik tunggal. Karakter yang demikian tersebut yaitu karakter yang memiliki kode ASCII 0 - 31, atau biasa sebagai karakter kontrol. Untuk penulisannya menggunakan notasi : ^<karakter>. Contoh :

^G → Bell, digunakan untuk membunyikan speaker komputer

^[ → Karakter Escape

^L → Karakter form feed

Karakter ASCII ini juga bisa ditulis dengan notasi #<nilai\_ascii>. Dengan

### 4. Tipe Data String

Tipe data string ini dapat berupa nol atau sejumlah karakter. Suatu konstanta string ditulis dengan diapit oleh tanda petik tunggal (').

Contoh :

Apabila didalam suatu konstanta string mengandung tanda petik tunggal maka penulisannya dapat dilakukan dengan menambah satu tanda petik tunggal lagi. Contoh :

→

untuk string yang mengandung karakter ASCII dengan kode 0 - 31 maka penulisannya menjadi sebagai berikut :

→

→

b. Identifier/Pengenal

Identifier/pengenal banyak bermanfaat dalam program, diantaranya untuk memberikan nama pada konstanta, variabel, procedure, function, unit dan program.

Aturan aturan penamaan identifier adalah sebagai berikut :

1. Pengenal/identifier harus diawali dengan huruf
2. Tidak mengandung spasi
3. Karakter kedua dan seterusnya dapat berupa huruf, angka atau garis bawah ( )
4. Panjang identifier yang dinyatakan berarti hanya 63 karakter pertama.
5. Tidak membedakan huruf kapital atau huruf kecil

Contoh pemberian identifier yang benar adalah :

HASIL

Hasil → Identik dengan HASIL

Matrix1

Kode\_Barang

Contoh pemberian identifier yang salah :

3Dimensi → Karakter pertama berupa angka

Jumlah Barang → Mengandung Spasi

Matrik-B → Mengandung Simbo Khusus

Da am pemberian *identifier* tidak boleh memakai *Reserved Word* (Kata Tercadang) atau nama *Defined Identifier* (Pengenal Terdefinisi) atau simbol khusus dari Turbo Pascal.

c. Operasi Aritmatika

Da am operasi aritmatika, operator aritmatik terbagi menjadi 2 (dua) yaitu operator **binary** dan **unary**. Operator **binary** adalah jika operator tersebut melibatkan 2 (dua) buah operand. Contoh :

→ penjumlahan A dengan B

Sedangkan operator **unary** adalah jika operator tersebut hanya melibatkan 1 (satu) buah operand saja. Contoh :

→ nilai negatif dari A

Adapun untuk daftar operator, jenis operasi dan sifatnya tercantum pada tabel dibawah ini :

Operator	Operasi	Sifat
+	Penjumlahan	Binary
-	Pengurangan	Binary
*	Perkalian	Binary
/	Pembagian Bilangan Bulat	Binary
div	Pembagian Real	Binary
mod	Sisa Pembagian Bulat	Binary
+	Tanda Plus (Positif)	Unary
-	Tanda Minus (Negatif)	Unary

**Contoh ungkapan aritmatik :**

$$2 + 3 = 5$$

$$7 / 2 = 3.5$$

$$7 \text{ div } 2 = 3$$

$$7 \text{ mod } 2 = 1$$

Dalam operasi aritmatik dikenal adanya prioritas operator. Adapun prioritas operator adalah sebagai berikut :

Operator **unary** memiliki prioritas tertinggi

Operator **\***, **/**, **div** dan **mod** memiliki prioritas menengah

Operator **+** dan **-** memiliki prioritas terendah

Jika dalam suatu operasi terdapat prioritas yang sama, maka perhitungan dimulai dari kiri ke kanan. Contoh :

$$\begin{aligned} 7 * 10 \ 5 \text{ mod } 3 * 4 + 9 &\rightarrow 70 \ 5 \text{ mod } 3 * 4 + 9 \\ &\rightarrow 70 \ 2 * 4 + 9 \\ &\rightarrow 70 \ 8 + 9 \\ &\rightarrow 62 + 9 \\ &\rightarrow 71 \end{aligned}$$

Untuk memberi memberikan prioritas yang lebih tinggi terhadap prioritas yang lebih rendah, maka dapat kita tambahkan tanda kurung. Contoh :

$$\begin{aligned} (7 * (10 \ 5) \text{ mod } 3 * 4 + 9 &\rightarrow (7 * 5 \text{ mod } 3) * 4 + 9 \\ &\rightarrow (35 \text{ mod } 3) * 4 + 9 \\ &\rightarrow 2 * 4 + 9 \\ &\rightarrow 8 + 9 \\ &\rightarrow 17 \end{aligned}$$

d. Statement Input dan Output

Pernyataan **Input - output** dalam Pascal menggunakan prosedur standard yaitu **Write**, **Writeln**, **Read**, **Readln**. Untuk operasi **input** (masukan) dapat digunakan **Read** atau **Readln**. Sedangkan untuk operasi **output** dapat digunakan **Write** atau **Writeln**. Bentuk pernyataan untuk prosedur standard **Read** dan **Readln** adalah :

dalam pernyataan diatas adalah variable yang digunakan untuk menyimpan data yang dimasukkan melalui **keyboard**.

Bentuk pernyataan untuk prosedur standar **Write** dan **Writeln** adalah :

pada pernyataan diatas bisa berupa nilai dari suatu variabel, suatu konstanta, suatu fungsi atau hasil dari suatu ungkapan yang melibatkan operator dan operand.

a. Percobaan I

Program sederhana ini akan menampilkan hasil dari suatu perhitungan tanpa meminta masukan dari **keyboard**.

```

Program Percobaan_I;
{-----}
{ Program untuk menampilkan hasil dari suatu ungkapan }
{-----}
Var
  A, B, C : Integer;
Begin
  A := 10; B := 2; C := 5;
  Write(A*B/C);
End.

```

b. Percobaan II

Pada Percobaan II ini akan diperlihatkan efek perbedaan dari **write** dan **writeln**.

```

Program Percobaan_II;
{-----}
{ Program ini akan menampilkan efek dari write dan writeln }
{-----}
Begin
  Write( Hello );
  Write( The World );
  Readln;
  Writeln( Hello );
  Writeln( The World );
End.

```

c. Percobaan III

Program pada Percobaan III ini akan menampilkan data yang disimpan pada suatu variabel dengan sebelumnya meminta masukan dari **keyboard**.

```

Program Percobaan_III
{-----}
{ Program ini akan meminta masukan dari keyboard }
{ selanjutnya akan ditampilkan pada layar monitor anda }
{-----}
Var
  A, B, C : Integer;
Begin
  Write( Masukkan Nilai A : ); readln(A);
  Write( Masukkan Nilai B : ); readln(B);
  Write( Masukkan Nilai C : );readln(C);
  Readln;
  Writeln( Nilai yang anda masukkan adalah : );
  Writeln( A = ,A); Writeln( B = ,B); Writeln( C = ,C);
  Readln;
End.

```

Buatlah suatu program yang bias menghitung luas dan keliling persegi panjang dengan ukuran panjang dan lebarnya tergantung pada masukan dari **keyboard**.

Rumus:

$$\text{Luas} = \text{Panjang} * \text{Lebar}$$
$$\text{Keliling} = 2 * (\text{Panjang} + \text{Lebar})$$



- a. Mahasiswa mengerti tentang Tipe Data Boolean dan proses seleksi (percabangan) serta proses pengulangan (looping)
- b. Mahasiswa dapat membuat program seleksi dari suatu syarat tertentu dengan menggunakan Turbo Pascal 7.0
- c. Mahasiswa dapat membuat program pengulangan proses dengan menggunakan Turbo Pascal 7.0

- a. Tipe Data Boolean
- b. Statement IF... THEN... ELSE
- c. Statement CASE... OF...
- d. Statement WHILE... DO...
- e. Statement REPEAT... UNTIL...
- f. Statement FOR... TO... dan FOR... DOWNTO...

a. Tipe Data Boolean

Tipe Data Boolean ini hanya memiliki 2 (dua) nilai saja, yaitu `false` dan `true`. Dalam pengoperasian tipe data ini, bisa digunakan operator `and`, `or`, `xor`, or atau `not`. Nilai dari variabel yang menggunakan tipe data ini sangat penting untuk pengambilan suatu keputusan, terutama jika dalam program yang membutuhkan proses seleksi. Pada tabel berikut ini akan dijelaskan tentang berbagai nilai operand dari operator operator boolean.

Operand 1	Operand 2	Hasil Operasi		
		AND	OR	XOR
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

Angka 0 = `false` dan 1 = `true`.

Sedangkan prioritas untuk operator penghubung adalah sebagai berikut :

**Prioritas Tertinggi** NOT

**Prioritas Terendah** = < > <= >= <>

b. Statement ...

Statement ini merupakan statement bersyarat yang akan mengeksekusi pernyataan dibawahnya apabila syarat logika yang diberikan memenuhi. Bentuk statement ini adalah :

atau

c. Statement

Statement ini digunakan untuk melakukan proses seleksi pada suatu kondisi yang memiliki kondisi lebih dari satu. Bentuk statement dari CASE... OF... ini adalah :

Pada statement ini, < > akan dieksekusi hanya jika nilai dari < **pemilih** > tercantum pada < **kondisi i** > . Namun jika nilai tersebut tidak ada maka pernyataan pernyataan yang mengikuti **ELSE** yang akan dieksekusi.

d. Statement

Statement ini digunakan untuk melakukan proses pengulangan sampai pada batas yang telah ditentukan. Bentuk pernyataannya adalah :

e. Statement

REPEAT memiliki kegunaan yang hampir sama dengan **WHILE**, namun perbedaannya adalah kondisi pengulangan pada **REPEAT** akan diperiksa setelah pernyataan yang berada dalam Blok **REPEAT... UNTIL...** dieksekusi. Sehingga pernyataan yang berada dalam **REPEAT** akan dieksekusi paling sedikit satu kali. Bentuk pernyataan dari **REPEAT** adalah :

**REPEAT**

**UNTIL**

Pengeksekusian terhadap pernyataan ini dilakukan terus selama **<syarat>** masih bernilai **False**

f. Statement

Statement **FOR** ini digunakan untuk melakukan pengulangan eksekusi terhadap suatu pernyataan. Pengendalian pengulangan ini dilakukan oleh **<variabel>** dengan jumlah pengulangan ditentukan oleh **<nilai awal>** dan **<nilai akhir>**. Bentuk pernyataan dari **FOR... DOWNTO...** adalah :

**<pernyataan>**

**<pernyataan>**

Pernyataan **FOR... DOWNTO...** digunakan untuk pengulangan secara menaik (**ascending**). Sedangkan **FOR... DOWNTO...** digunakan untuk pengulangan secara menurun (**descending**).

a. Percobaan I

Pada Percobaan I ini program akan melakukan seleksi terhadap suatu pilihan dengan menggunakan statement **IF... THEN... ELSE...**

```

Program Percobaan_2_1;
{-----}
{Penggunaan IF THEN ELSE}
{-----}

Uses Crt;
Var
  Kode : Byte;

Begin
Repeat
  ClrScr;
  Write('Masukkan Kode Hari [1 .. 7] : ');
  Readln(Kode);
Until (Kode in [1..7]);

Writeln;

if Kode = 1 Then
  Writeln('Hari Senin')
else
  if Kode = 2 Then
    Writeln('Hari Selasa')
  else
    if Kode = 3 Then
      Writeln('Hari Rabu')
    else
      if Kode = 4 Then
        Writeln('Hari Kamis')
      else
        if Kode = 5 Then
          Writeln('Hari Jum"at')
        else
          if Kode = 6 Then
            Writeln('Hari Sabtu')
          else
            if Kode = 7 Then
              Writeln('Hari Minggu');

Readln;
End.

```

b. Percobaan II

Pada Percobaan II ini program akan melakukan proses looping (*perulangan*) untuk menggambar bentuk segitiga dengan menggunakan karakter \* .

```
Program Percobaan_II;
{-----}
{Program Contoh Proses Looping}
{-----}
Uses Crt;

Var
  i, j,
  Bar : Byte;
Begin
  ClrScr;
  Write('Jumlah Baris : ');Readln(Bar);
  ClrScr;
  i := 0;
  Repeat
    For j := 1 To i Do
      Write('*':3);
      Writeln;
      inc(i);
    Until i > Bar;
  For i := Bar - 1 Downto 0 Do

    Begin
      J := 1;
      While j <= i Do
        Begin
          Write('*':3);
          j := j + 1;
        end;
        writeln;
      end;
  Readln;
End.
```

c. Percobaan III

Pada percobaan III ini program akan menghitung jumlah huruf kapital yang ada pada kata yang diinputkan.

```
Program Percobaan_III;
{-----}
{Contoh Program Seleksi dalam Looping}
{-----}
Uses Crt;
Var
  Kata : String[40];
  Kap, I : Byte;
Begin
  Repeat
  ClrScr;
    Writeln( Masukkan Kata : );
    Readln(Kata);
  Until Length(Kata) <= 40;
  For I := 1 to Length(Kata) Do
  Begin
    If Kata[I] in [ A .. Z ] Then
      Kap := Kap + 1;
  End;
  Writeln( Jumlah Huruf Besar : ,Kap);
  Readln;
End.
```

Buatlah program untuk memisahkan Huruf Vokal dan Konsonan serta Angka dan karakter lainnya dengan meminta masukan dari keyboard. Gunakan Statement **Case... Of...** untuk proses seleksinya.

Contoh :

Catatan : Kode ASCII untuk Spasi adalah 32

- a. Mahasiswa mengerti tentang Array 1 Dimensi maupun Array 2 Dimensi
  - b. Mahasiswa dapat membuat program dengan memanfaatkan Array
- a. Array 1 Dimensi
  - b. Array 2 Dimensi

Array adalah sekumpulan tipe data terstruktur yang berisi sekumpulan komponen/elemen dengan tipe yang sama.

a. Array 1 Dimensi

Array berdimensi 1 merupakan array yang memiliki satu indeks saja dengan tipe data yang sama. Bentuk pernyataan untuk mendeklarasikan array berdimensi 1 adalah :

Tipe data merupakan tipe data yang dikenal dalam pascal yang telah dijelaskan pada BAB 1. contoh pendklarasian array 1 dimensi adalah sebagai berikut :

Sedangkan untuk mengisikan nilai pada array adalah : Dalam hal ini nilai 32 akan diisikan pada array Daftar pada elemen yang mempunyai indeks array 6 atau elemen ke 7 dari array (elemen pertama adalah indeks 0). Adapun untuk menampilkan elemen tersebut adalah : **writeln(Daftar[6]);**

b. Array 2 Dimensi

Jika pada array 1 dimensi hanya memiliki 1 indeks array, maka pada array 2 dimensi ini dimungkinkan array memiliki 2 buah indeks array. Antara indeks array yang satu dengan yang lainnya dipisahkan oleh tanda koma. Bentuk dari pernyataan pendeklarasiannya adalah :

Contoh pendeklarasian array 2 dimensi ini adalah sebagai berikut :

Untuk memberikan nilai ada array 2 dimensi ini adalah : Pada contoh tersebut nilai 56 diisikan pada baris ke - 2 dan kolom ke 1 pada elemen array. Sedangkan untuk menampilkan nilai tersebut adalah :

a. Percobaan I

Pada percobaan I ini akan disajikan contoh program pengurutan data secara menaik (**ascending**) dengan memanfaatkan array 1 dimensi.

Program Percobaan\_3\_1;

{-----}

```

{Pengurutan Data Dengan Menggunakan Array}
{-----}
USES CRT, DOS;
VAR
  DATA : ARRAY[1..10] OF INTEGER;
  TMP,N,I,J: INTEGER;
BEGIN
  CLRSCR;
  N := 0;

  REPEAT
  CLRSCR;
  WRITE('MASUKKAN JUMLAH DATA (MAX 10) : ');
  READLN(N);
  UNTIL (N <= 10) AND (N <> 0);

  CLRSCR;
  FOR I := 1 TO N DO
  BEGIN
    WRITE('DATA KE - ',I, ' : ');
    READLN(DATA[I]);
  END;
  CLRSCR;
{THIS PART TO WRITE THE ORIGINAL DATA}
  GOTOXY(5,1);
  WRITELN('YOUR DATA IS : ');
  FOR J := 1 TO N DO
  BEGIN
    GOTOXY(4+J,2);
  WRITELN(DATA[J]);
  END;
{THIS PART TO SORTING THE DATA}
  FOR I := 1 TO N DO
  BEGIN
    FOR J := I+1 TO N DO
    BEGIN
      IF DATA[I] > DATA[J] THEN
      BEGIN
        TMP := DATA[J];
        DATA[J] := DATA[I];
        DATA[I] := TMP;
      END;
    END;
  END;
{THIS PART TO WRITE DATA AFTER SORTING}
  GOTOXY(5,3);
  WRITELN('DATA AFTER SORTING : ');
  FOR I:= 1 TO N DO
  BEGIN
    GOTOXY(4+I,4);
    WRITELN(DATA[I]);
  END;

```



```
    READLN;
```

```
END.
```

b. Percobaan II

Pada percobaan II ini program akan membaca dan menampilkan sebuah array berdimensi 2. Program ini bisa dimanfaatkan untuk operasi matrik.

Program Percobaan\_3\_2;

```
{-----}
```

```
{Program Membuat Matrik dengan array 2 Dimensi}
```

```
{-----}
```

```
uses crt;
```

```
const
```

```
    MaxData = 5;
```

```
var
```

```
    matrik : array [1..MaxData,1..MaxData] of integer;
```

```
    I,J,X,Y : integer;
```

```
Begin
```

```
    {Membaca Jumlah Baris Dan Kolom untuk elemen matrik}
```

```
    clrscr;
```

```
    repeat
```

```
write( Masukkan Jumlah Baris : ); Readln(X);
```

```
    until X <= MaxData;
```

```
    repeat
```

```
write( Masukkan Jumlah Kolom : ); Readln(Y);
```

```
    until Y <= MaxData;
```

```
    {Membaca Data Untuk Tiap Elemen Matirk}
```

```
    for I := 1 to X do
```

```
    begin
```

```
        for J := 1 to Y do
```

```
        begin
```

```
            write( Elemen ( , I , , J , ) );
```

```
            readln(matrik[I,J]);
```

```
        end;
```

```
    writeln;
```

```
    end;
```

```
    {Menampilkan Nilai Dari Matrik}
```

```
    writeln( MATRIKS : );
```

```
    writeln;
```

```
    for I := 1 to X do
```

```
    begin
```

```
        for J := 1 to Y do
```

```
            write(matrik[I,J],8);
```

```
        writeln;
```

```
    end;
```

```
    readln;
```

```
End.
```

Buatlah program Transpose Matriks. Program ini meminta masukan berupa jumlah baris dan jumlah kolom. Tetapkan jumlah baris maupun jumlah kolom maximal 10. Contoh dari Transpose Matriks ini adalah :

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \text{ maka } A^T = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{bmatrix}$$

Jadi pada *Transpose Matriks* ini adalah penukaran baris menjadi kolom.

- a. Mahasiswa mengerti tentang Tipe Himpunan, Tipe Record dan Operasi File.
- b. Mahasiswa dapat membuat program dengan memanfaatkan Tipe Himpunan dan Tipe Record serta penyimpanan data dengan menggunakan File.

- a. Tipe Himpunan
- b. Tipe Record
- c. File :
  1. File Acak (Random File)/File Bertipe
  2. File Teks
  3. File Tak Bertipe

Himpunan (set) adalah tipe data terstruktur yang berisi sekumpulan dari object dengan tipe yang sama, dengan masing masing object (yang biasanya disebut sebagai anggota atau elemen himpunan) tidak bisa diakses secara langsung.

Contoh :

1. Himpunan bilangan dari 1 sampai 100  $\rightarrow [1..100]$
2. Himpunan huruf besar dari C sampai X  $\rightarrow [C .. X]$
3. Himpunan huruf vokal  $\rightarrow [A, I, U, E, O]$

Pendefinisian himpunan ini adalah :

Jadi pendefinisian himpunan secara umum adalah :

***Variabel\_Himpunan = SET OF tipe\_elemen\_himpunan;***

Seperti halnya variabel lainnya, variabel bertipe himpunan ini bisa dioperasikan secara matematis. Adapun operasi dari himpunan ini meliputi penambahan, pengurangan, perkalian, maupun perbandingan.

1. Penambahan

Pada operasi penambahan atau sering disebut juga sebagai operasi union, akan menghasilkan gabungan dari 2 elemen himpunan yang ditambahkan. Contoh :

$$[1,2,3] + [4,5,6,7] \rightarrow [1,2,3,4,5,6,7]$$

2. Pengurangan

Pada operasi pengurangan ini akan menghasilkan elemen elemen himpunan yang ada pad

Contoh pendefinisian dari record adalah sebagai berikut :

Elemen dari record ini dapat diakses dengan menggunakan bentuk :

Digunakan untuk menyimpan data pada media penyimpanan data (Harddisk atau disket).

1. File Acak/File Bertipe

File ini bisa diakses secara acak, sehingga data dapat diakses secara langsung pada posisi yang diinginkan dan tidak harus dimulai dari awal file.

Pendefinisian dari file acak ini adalah :

Pada pendefinisian diatas semua komponen file bertipe skalar. Komponen tersebut bisa berupa record yang mengandung beberapa field. Contoh :

2. File Teks

Berbeda dengan file acak, file teks ini tidak memberikan keleluasaan pengaksesan komponen komponen file secara acak. Pada file ini semua

komponen harus dibaca mulai dari depan atau dibaca secara berurutan (*sequential*). Elemen dasar pembentuk file teks ini adalah karakter. Dalam sebuah file teks komponen file merupakan sederetan karakter yang membentuk sebuah baris dan masing masing baris diakhiri dengan sebuah tanda akhir baris (eoln/end of line). Tanda eoln ini terdiri dari karakter CR dan LF secara berderet. Pendefinisian untuk file teks ini adalah :

### 3. File Tak Bertipe

Pada file tak bertipe ini dimungkinkan untuk membaca sembarang file yang akan dibuka. Keistimewaan dari file ini adalah dapat melakukan pembacaan maupun penulisan secara serentak dalam sekali waktu saja. Pendefinisian file tak bertipe ini adalah :

#### a. Percobaan I

Pada Percobaan 4.1 ini akan disajikan tentang pemanfaatan Tipe Himpunan untuk mencari semua karakter yang ada pada kalimat ke 1 dan tidak ada pada kalimat ke 2 kecuali karakter spasi.

```
program percobaan_4_1;
uses crt;
const spasi = #32;
type
  himpkarakter = set of char;
var
  i,j   : integer;
  karakter : char;
  kalimat : array [1..2] of string;
  himpunan : array [1..2] of himpkarakter;
  himpselisih : himpkarakter;
begin

  clrscr;
  {----- input kalimat -----}
  writeln('kalimat pertama : '); readln(kalimat[1]);
  writeln('kalimat kedua : '); readln(kalimat[2]);

  {----- proses elemen -----}
  for i := 1 to 2 do
  begin
    himpunan[i] := []; {definisikan himpunan kosong}
    for j := 1 to length(kalimat[i]) do
      if not (kalimat[i][j] in himpunan[i]) then
        himpunan[i] := himpunan[i] + [kalimat[i][j]];
    end;
  end;
  himpselisih := himpunan[1] - himpunan[2];
```

```

{----- tampilkan elemen himpunan hasil selisih -----}
writeln;
writeln('daftar karakter yang ada pada kalimat pertama');
writeln('dan tidak ada pada kalimat kedua : ');
writeln;
karakter := succ(spasi); {untuk memperoleh karakter setelah spasi}
while (himpiselisih <> []) do
begin
  if karakter in himpselisih then
  begin
    write(karakter:2);
    himpselisih := himpselisih - [karakter];
    end;
    karakter := succ(karakter); {untuk memperoleh karakter berikutnya}
end;
end;

readln;
end.

```

b. Percobaan II

Pada Percobaan II ini program akan merekam data pada record, kemudian akan menampilkannya kembali.

```
Program Percobaan_4_2;
Uses CRT;
Const
  JNilai = 5;
  MaxSiswa = 20;
Type
  TSiswa = RECORD
    NIM : String[10];
    Nama : String[20];
    Nilai : Array [1..JNilai] of Real;
    Rata : Real;
  End;
  NSiswa = Array [1..MaxSiswa] of TSiswa;
Var
  Jumlah : 1..MaxSiswa;
  i,j : Integer;
  TotalNilai : Real;
  Siswa : NSiswa;
  Temporary : TSiswa;
Begin
  clrscr;
  {bagian memasukkan data}
  repeat
    write('Masukkan Jumlah Data (Max 20) : ');Readln(Jumlah);
  until Jumlah <= MaxSiswa;
  writeln;
  for i := 1 to Jumlah do
    begin
      write('N.I.M : ');
      readln(Siswa[i].Nim);
      write('Nama : ');
      readln(Siswa[i].Nama);
      writeln('Nilai : ');
      for j := 1 to JNilai do
        begin
          write('Nilai ',j,' : ');
          readln(Siswa[i].Nilai[j]);
        end;
      writeln;
    end;
  {bagian menghitung nilai rata - rata}
  for i := 1 to MaxSiswa do
    begin
      TotalNilai := 0;
      for j := 1 to JNilai do
        TotalNilai := TotalNilai + Siswa[i].Nilai[j];
      Siswa[i].Rata := TotalNilai / JNilai;
    end;
end;
```



```

{bagian mengurutkan data berdasarkan nilai rata
 rata tertinggi}
for i := 1 to Jumlah -1 do
begin
for j := i + 1 to Jumlah do
if Siswa[i].Rata < Siswa[j].Rata then
begin
{menukarkan isi record}
Temporary := Siswa[i];
Siswa[i] := Siswa[j];
Siswa[j] := Temporary;
end;
end;
{Tampilkan data yang sudah diurut}
writeln;
{012345678901234567890123456789012345678901234567890123}

writeln('=====');
writeln('|N.I.M   |NAMA           |Rata - Rata   |');

writeln('=====');
for i := 1 to Jumlah do
writeln('|',Siswa[i].Nim:10,'|',Siswa[i].Nama:20,'|',
Siswa[i].Rata:20:1,'|');

writeln('=====');
readln;
end.

```

c. Percobaan III

Program pada Percobaan III ini sebagian data yang ada pada Percobaan II yaitu NIM, Nama, Nilai Rata Rata akan disimpan pada File Acak dengan nama Nilai.Dat.

1. Program untuk membuat file baru dan menyimpan datanya ke dalam file tersebut Program Percobaan\_4\_3\_1;

```
uses crt;
type
  TSiswa = Record
    NIM   : String[10];
    Nama  : String[20];
    Nilai : Real;
end;
var
  FileSiswa : FILE OF TSiswa;
  Siswa     : TSiswa;
  Kar       : char;
  i         : integer;
begin
  clrscr;
  {tentukan nama dan directory dari file}
  assign(FileSiswa,'D:\Nilai.dat');
  {buat file baru dengan isi kosong}
  rewrite(FileSiswa);
  i := 0;
  {baca masukan dari user}
  repeat
    with Siswa do
      begin
        write('NIM   : ');readln(nim);
        write('Nama  : ');readln(nama);
        write('Nilai : ');readln(nilai);
      end;
    write('Tambah Data Lagi (Y/T)?');readln(kar);
    if kar in ['Y','y'] then
      begin
        write(FileSiswa,Siswa);{tuliskan data pada file}
        inc(i);
      end;
  until Kar in ['T','t'];
  close(FileSiswa);{menutup file}
end.
```

2. Program untuk membaca data di dalam file yang telah dibuat melalui program ke 1.

Program Percobaan\_4\_3\_2;

```
uses crt;
type
  TSiswa = Record
    NIM   : String[10];
    Nama  : String[20];
```

```

        Nilai : Real;
    end;
var
    FileSiswa : FILE OF TSiswa;
    Siswa     : TSiswa;
    Kar       : char;
    i         : integer;

begin
    clrscr;
    {tentukan nama dan directory dari file}
    assign(FileSiswa,'D:\Nilai.dat');
    {buka file tanpa menghapus}
    reset(FileSiswa);
    i := 0;
    {baca masukan dari user}
    while not eof(FileSiswa) do
    begin
        read(FileSiswa,Siswa);
        write('NIM   : ');writeln(Siswa.nim);
        write('Nama   : ');writeln(Siswa.nama);
        write('Nilai  : ');writeln(Siswa.nilai:0:1);
        write('=====');
        writeln('=====');
    end;
    close(FileSiswa);{menutup file}
    readln;
end.

```

3. Program untuk menambahkan data di akhir file.

```
Program Percobaan_4_3_3;
uses crt;
type
  TSiswa = Record
    NIM   : String[10];
    Nama  : String[20];
    Nilai : Real;
  end;
var
  FileSiswa : FILE OF TSiswa;
  Siswa     : TSiswa;
  Kar       : char;
  i         : integer;
begin
  clrscr;
  {tentukan nama dan directory dari file}
  assign(FileSiswa,'D:\Nilai.dat');
  {buka file tanpa menghapus}
  reset(FileSiswa);
  seek(FileSiswa,filesize(filesiswa)); {menempatkan pointer file pada posisi
  terakhir}
  i := 0;
  {baca masukan dari user}
  repeat
    with Siswa do
      begin
        write('NIM   : ');readln(nim);
        write('Nama  : ');readln(nama);
        write('Nilai : ');readln(nilai);
      end;
    write('Tambah Data Lagi (Y/T)?');readln(kar);
    if kar in ['Y','y'] then
      begin
        write(FileSiswa,Siswa);{tuliskan data pada file}
        inc(i);
      end;
  until Kar in ['T','t'];
  close(FileSiswa);{menutup file}
end.
```

Buatlah program yang bias menyimpan data karyawan. Data karyawan tersebut berisi NIP, Nama, Golongan, Gaji/Hari, Gaji Total. Gaji/Hari ditentukan melalui golongan karyawan tersebut, ketentuannya adalah :

Golongan I Rp. 3.000/hari  
Golongan II Rp. 4.500/hari  
Golongan III Rp. 6.000/hari  
Golongn IV Rp. 7.500/hari

Program tersebut bisa digunakan untuk menambah data, dan melihat data.

- a. Mahasiswa mengerti tentang Modular Programing
- b. Mahasiswa mengerti tentang perbedaan Prosedur dan Fungsi
- c. Mahasiswa bisa membuat program yang lebih terstruktur

- a. Parameter, Variabel Global dan Variabel Lokal
- b. Prosedur
- c. Fungsi
- d. Rekursif

- a. Parameter, Variabel Global dan Variabel Lokal

Parameter adalah suatu mekanisme untuk melewatkan nilai atau variabel ke dalam subprogram. Dengan parameter ini, pemberian nilai awal atau nama variabel yang berbeda terhadap subprogram dapat dilakukan. Parameter ini dibedakan menjadi 2 yaitu Parameter Formal dan Parameter Aktual. **Parameter Formal adalah parameter yang terdapat dalam kepala subprogram** **Parameter Aktual adalah parameter yang diberikan apabila prosedur atau fungsi** dipanggil. Parameter Aktual ini dibagi menjadi 2 yaitu Parameter Nilai dan *Parameter Variabel*. *merupakan parameter yang nilainya tidak akan berubah* pada saat subprogram selesai diproses meskipun ada pernyataan dalam subprogram yang mengubahnya. *Parameter Variabel adalah parameter yang nilainya bisa berubah ketika* subprogram selesai diproses.

variabel tersebut tidak dideklarasikan didalam subprogram itu sendiri, namun dideklarasikan pada subprogram yang memiliki level lebih tinggi atau pada program utama, dan pendeklarasiannya dilakukan sebelum subprogram.

dilakukan didalam subprogram itu sendiri. Variabel ini tidak bisa digunakan diluar blok subprogram yang mendeklarasikannya. Bentuk pendeklarasian prosedur adalah :

#### b. Prosedur dan Fungsi

Prosedur merupakan kumpulan dari statemen statemen yang dalam satu subprogram dan dapat dipanggil berulang kali baik oleh subprogram lain maupun oleh program utama. Prosedur berdiri sendiri sebagai sebuah pernyataan. Fungsi hampir sama dengan prosedur namun fungsi ini tidak dapat berdiri sendiri sebagai sebuah pernyataan. Fungsi digunakan sebagai elemen dalam pernyataan penugasan atau suatu ungkapan. Fungsi ini juga digunakan untuk memperoleh sebuah nilai, karena fungsi bisa memberikan suatu nilai balik.

Bentuk pendeklarasian fungsi adalah :

```
FUNCTION <nama_fungsi> (<daftar_parameter_formal>) : <tipe_hasil>;
```

#### c. Rekursif

Rekursif merupakan suatu subprogram yang bisa memanggil dirinya sendiri. Suatu subprogram selain bisa dipanggil oleh subprogram lain bisa juga dipanggil oleh subprogram itu sendiri. Cara rekursif ini banyak dipakai pada persoalan yang dapat dipecahkan secara induktif.

#### a. Percobaan I

Program berikut ini akan melakukan penukaran nilai variabel melalui prosedur.

```
Program Percobaan_5_1;
Uses crt;
Var
  X,Y : real;
Procedure Swap(Var X,Y : real);
Var
  Temp : real;
Begin
  Temp := X;
  X := y;
  Y := Temp;
End;

Begin {Program Utama}
  Clrscr;
  X := 12;
  Y := 20;

  Writeln( Nilai Variabel Sebelum Prosedur Swap Dipanggil : );
  Writeln( X = ,X:5:2, Y = ,Y:5:2);
```

```
Swap(X,Y);
Writeln( Nilai Variabel Setelah Prosedur Swap Dipanggil : );
Writeln( X = ,X:5:2, Y = ,Y:5:2)
Readln;
End.
```

b. Percobaan II

Pada percobaan ini program akan melakukan penjumlahan pada 2 variabel dengan memanfaatkan fungsi.

Program Percobaan\_4\_2;

```
Uses crt;
Var
  A,B : integer;

Function Add(A, B : Integer) : Integer;
Begin
  Add := a + b;
End;

Begin
  Clrscr;
  A := 5;
  B := 10;
  Writeln( A + B = , Add(A,B) : 5 : 2);
  Writeln( 15 + 20 = , Add(15,20) : 5 : 2);
  Readln;
End.
```



c. Percobaan III

Pada Percobaan III ini akan ditampilkan fungsi rekursif untuk menghitung nilai dari suatu deret fibbonacci.

```
Program Percobaan_4_3;
Uses crt;
Var
  I, Nilai : Integer;

Function Fibo(N : Integer) : Integer;
Begin
  If (N = 1) or (N = 2) then
    Fibo := 1;
  Else
    Fibo := Fibo(N - 1) + Fibo(N - 2);
  End;

Begin {Pogram Utama}
  Clrscr;
  Write( 'Masukkan Batas Deret : ');
  Readln(Nilai);
  For I := 1 to Nilai do
    Writeln(I : 3, Fibo(I) : 8);
  Readln;
End.
```

Buatlah suatu fungsi yang dapat menghitung nilai faktorial dengan memanfaatkan fungsi rekursif. Kemudian manfaatkan fungsi tersebut untuk menghitung nilai suatu kombinasi dengan rumus :

$n!$

$C =$

$(n)!$

$n! m m n -$



Untuk membuat variabel dinamis, prosedur yang dipakai yaitu New. Prosedur ini memiliki parameter yang berupa variabel pointer. Contoh penggunaan prosedur

**New ini :**

NEW(<variabel\_pointer>);

Selain New untuk membuat variabel dinamis ini diperlukan satu kata tercadang yaitu NIL. Contoh pembuatan pointer baru ini :

b. Membaca Data Variabel Dinamis (Pointer)

Seperti halnya variabel statis, variabel dinamis ini juga bisa dibaca dan ditampilkan. Cara membaca variabel dinamis ini dengan menggeser pointer yang ada sampai pointer tersebut sama dengan NIL. Contoh :

c. Menghapus data dari variabel dinamis

Variabel dinamis ini sebenarnya akan disimpan dalam memory. Jadi pada variabel ini bisa dilakukan operasi penghapusan. Baik seluruh data maupun sebagian data. Contoh penghapusan data pada pointer adalah :

Pada Percobaan VI ini akan ditampilkan program untuk operasi pointer. Operasi yang dilakukan adalah penambahan data, penghapusan data, dan menampilkan data.

```
uses crt;
type linked = ^list;
list = record
    nim : string[10];
    nama : string[20];
    alamat : string[25];
```

```

        next : linked;
        end;
var
    awal, akhir : linked;
    p : char;
function newlist : linked;
var
    first : linked;
begin
    new(first);
    with first^ do
        begin
            write('nim  : ');readln(nim);
            write('nama  : ');readln(nama);
        write('alamat  : ');readln(alamat);
            next := nil;
        end;
        newlist := first;
    end;

procedure tambah(jml : integer);
var
    baru : linked;
    j : char;
begin
    repeat
        clrscr;
        if jml <> 0 then
            begin
                writeln('menambah data baru');
                writeln('-----');
            end;
        writeln;
        baru := newlist;
        if awal = nil then
            akhir := baru
        else
            baru^.next := awal;
            awal := baru;
        write('tambah data lagi <y/n> ?');readln(j);
        until upcase(j) = 'n';
        clrscr;
    end;

procedure delete;
begin
    if awal <> nil then
        begin
            awal := awal^.next;
            writeln('simpul pertama telah dihapus');
        end
    else
        writeln('link list masih kosong');
end;

```

```

        writeln;
        writeln('tekan enter untuk kembali ke menu utama');
repeat until readkey = #13;
    clrscr;
end;

procedure baca;
var
    temp : linked;
    i,j,y : integer;
begin
    i := 1;
    writeln('membaca isi link list');
    temp := awal;
    if temp = nil then
        writeln('link list masih kosong')
    else
        begin
            gotoxy(1,2);writeln('no');
            gotoxy(19,2);writeln('nim');
            gotoxy(39,2);writeln('nama');
            gotoxy(59,2);writeln('alamat');
        for j := 1 to 79 do
            begin
                gotoxy(j,3);write('-');
            end;
        y := 4;
        while temp <> nil do
            begin
                gotoxy(1,y);writeln(i);
                gotoxy(19,y);writeln(temp^.nim);
                gotoxy(39,y);writeln(temp^.nama);
                gotoxy(59,y);writeln(temp^.alamat);
                temp := temp^.next;
                inc(y);inc(i);
            end;
        for j := 1 to 79 do
            begin
                gotoxy(j,y);writeln('-');
            end;
        writeln;
        writeln('tekan sembarang enter untuk kembali ke menu utama');
        repeat until readkey = #13;
        clrscr;
    end;
end;
{----- menu utama -----}
begin
    awal := nil; akhir := nil;
    clrscr;
    repeat

```

```

gotoxy(30,3);writeln('daftar pilihan menu');
gotoxy(20,8);writeln('<1> menambah list');
gotoxy(20,9);writeln('<2> membaca list');
gotoxy(20,10);writeln('<3> menghapus list');
gotoxy(20,11);writeln('<4> keluar');
gotoxy(20,14);write('pilihan : ');readln(p);
case p of
  '1' : begin
    clrscr;
    tambah(1);
    end;
  '2' : begin
    clrscr;
    baca;
end;
  '3' : begin
    clrscr;
    delete;
    end;
  end;
until p = '4';
end.

```

Lengkapilah percobaan diatas dengan procedure prosedur sebagai berikut :

1. Tambah Data Ditengah (Inserting)
2. Tambah Data Diakhir Pointer
3. Menghapus Data Ditengah Pointer
4. Menghapus Data Diakhir Pointer