MODUL I
PEMROGRAMAN DASAR

LABORATORIUM DASAR KOMPUTER
JURUSAN TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI NASIONAL
BANDUNG
2016

PENGANTAR

Laboraturium Dasar Komputer setiap semester selalu melaksanakan kegiatan praktikum dan pada semester genap dilakukan kegiatan praktikum Pemrograman Dasar. Kegiatan praktikum ini dilaksanakan sebagai pendukung matakuliah yang sama yaitu Pemrograman Dasar. Pada tahun ajaran 2015/2016 materi praktikum Pemrograman Dasar adalah pemrograman pascal dengan menggunakan Aplikasi Lazarus IDE v1.4.0.

Pada praktikum Pemrograman Dasar ajaran 2015/2016, pembuatan modul dan *jobsheet* dibuat sepenuhnya oleh **Tim Asisten Laboratorium Dasar Komputer**.

Koordinator : Marisa Premitasari.,S.Kom.,MT

Tim Asisten Laboratorium

1.	15-2013-073	Mochamad Angga Anggriawan	**(Koordinator Asisten)**
2.	15-2013-052	Derangga Galuh Raharja	(Koordinator Kelas)
3.	15-2013-054	Gian Permana	(Koordinator Kelas)
4.	15-2013-066	Bagus Rachman Pribadi	
5.	15-2013-078	Firma Firmansyah Adi	
6.	15-2014-005	Rian Herdiansyah	
7.	15-2014-014	Hanif Al Kamal	
8.	15-2014-025	Ihsan Haryadi K	
9.	15-2014-048	Novi Nur’aini	
10.	15-2014-056	Renita Dewi	(Koordinator Kelas)
11.	15-2014-069	Bintang Candra A.S	
12.	15-2014-074	Adi Nugraha	

Dlakukan perbaikan oleh **Koordinator Laboratorium** dan **Tim Asisten Laboratorium Dasar Komputer tahun ajaran 2018/2019.**

Koordinator : Uung Ungkawa, Ir., M.T.

Tim Asisten Laboratorium

1.	15-2014-056	Renita Dewi	**(Koordinator Asisten)**
2.	15-2015-077	M. Alif Abhiesa Al Kautsar	
3.	15-2015-084	Fahmi Ahmad Fauzi	
4.	15-2015-097	Reza Fadilah Dwiputra	
5.	15-2015-098	Anisa Dwiyanti	
6.	15-2015-109	Ari Yudianto	(Koordinator Kelas B)
7.	15-2016-009	Maruli Ibrahim Batara D.	
8.	15-2016-012	Muhammad Fikrian Nahl	
9.	15-2016-015	Febby Fitriani Karina	
10.	15-2016-016	Fabio Fahri Pratama	
11.	15-2016-086	Lantip Wakito Aji	
12.	15-2016-088	Agistya Anugrah Dwiutama	
13.	15-2016-093	Futy Alpaz	
14.	15-2016-097	Ismail	(Koordinator Kelas D)
15.	15-2016-099	Dwi Adi Lenggana Putra	
16.	15-2016-100	Hasna Septi Dewi	
17.	15-2016-102	James Palmer P.	(Koordinator Kelas C)
18.	15-2016-118	Rifki Mirfiza	(Koordinator Kelas A)
19.	15-2016-125	Mochamad Fahrizky R. P.	
20.	15-2016-126	Yulianto Ardi Nugroho	

Perbaikan yang dilakukan adalah dengan menambahkan materi baru pada modul diantaranya :

1. Penggunaan Delphi7 dalam penerapan bahasa Pascal
2. Pemberian contoh *source code* dalam materi
3. Kumpulan kasus pemrograman yang harus diselesaikan.

DAFTAR ISI

MATERI 1- PENGENALAN PASCAL

1.1. Struktur Program

Program Pascal terdiri dari:

1. Kepala Program
   Missal: Program *nama*, Program *nama (input)*, Program *nama (input,output)*
2. Bagian Deklarasi
   - Bagian deklarasi label
   - Bagian deklarasi konstanta
   - Bagian deklarasi tipe
   - Bagian deklarasi variable
   - Bagian deklarasi subprogram

   Bagian-bagian di atas tidak harus dimiliki semua di dalam program.
3. Bagian Pernyataan

```
Begin
End
```

Sebagai bahasa pemrograman yang terstruktur, bahasa ini menyediakan sejumlah control (berupa REPEAT, WHILE dan FOR) serta mendukung pembuatan program yang tersusun atas sejumlah blok-blok kecil atau yang dikenal dengan sebutan rutin pada berbagai bahasa pemrograman.

Dengan menggunakan compiler, program tersebut harus bebas dari kesalahan kaidah agar program dapat dieksekusi (eksekusi adalah istilah dalam pemograman yang menyatakan bahwa program sedang dijalankan oleh komputer).Dengan menggunakan interpreter, eksekusi program dihentikan pada saat suatu perintah dinyatakan salah secara kaidah. Perintah-perintah yang sudah betul dan terletak sebelum yang salah akan tetap dijalankan. Biasanya setelah seorang pemograman telah mahir memprogram, ia akan lebih suka menggunakan compiler, karena kecepatan eksekusi program yang telah di kompilasi jauh lebih cepat daripada interpreter.

Pada model compiler, anda memberikan semua perintah ke penterjemah, tanpa harus menunggu sebuah perintah dikerjakan terlebih dulu. Proses untuk menterjemahkan semua perintah sekaligus pada compiler dikenal dengan sebutan kompilasi.

Beberapa istilah pada pemograman yang kiranya perlu diperkenalkan adalah *bug* dan *debugging*. *Bug* (baca : bag) menyatakan suatu kesalahan dalam program. Adapun *debugging* (baca : dibaging) adalah upaya untuk mencari kesalahan pada program. Kesalahan pada tahap ini umumnya berupa kesalahan logika. Contoh kesalahan logika :proses menampilkan data semua pegawai wanita, tetapi yang dituliskan di dalam program ternyata semua pegawai (baik pria maupun wanita).

1.2. Karakter

Elemen terkecil pada Pascal adalah karakter. Karakter dapat berupa :

- Karakter control (karakter dengan nilai ASCII di bawah 32, misalnya Tab dan ***Backspace)***
- karakter ASCII tidak standar pada PC (karakter dengan nilai ASCII antara 128 hingga 255)

Simbol khusus pada Pascal berupa :

+ - * / = <> [ ] , ( ) : ; ^ . @ { } $ #

Selain simbol khusus yang berupa sebuah karakter tunggal, ada sejumlah simbol khusus yang tersusun atas dua karakter.Kedua karakter pada simbol-simbol seperti ini tidak boleh di tulis dengan diantarai oleh spasi.

SIMBOL	KETERANGAN
<=	Operator Relasi
>=	Operator Relasi
:=	Operator Penugasan
<>	Operator Relasi
(*	Tanda awal komentar
*)	Tanda akhir komentar
(.	Identik dengan [
.)	Identik dengan]
..	Tanda subjangkauan

1.3. Jenis Data

Jenis data sederhana terdiri dari:

Standard:

- Integer: bilangan bulat
- Real: bilangan real
- Char: karakter
- Boolean: nilai boolean (benar/salah)

user-defined (bentukan)

- enumerated (terbilang)
- subrange

Jenis data yang lain adalah yang terstruktur yang terdiri dari:

- Array
- Record
- Set
- File
- Pointer

Operasi Integer

Operator	Arti	Operan	Hasil
+	penjumlahan	integer	integer
-	Pengurangan	integer	integer
*	Perkalian	integer	integer
/	Pembagian	integer	real
DIV	Pembagian terpotong	integer	integer
Mod	Sisa pembagian	integer	integer

Real

Operator	Arti	Operan	Hasil
+	penjumlahan	real	real
-	Pengurangan	real	real
*	Perkalian	real	real
/	Pembagian	real	real

Boolean

Operator	Arti
=	Sama dengan?
<>	Tidak sama dengan?
<	Lebih kecil?
>	Lebih besar?
<=	Lebih kecil atau sama dengan?
>=	Lebih besar atau sama dengan?
and	Benar jika kedua operan benar
or	Benar jika salah satu operan benar
not	Membalik nilai benar/salah

Jenis Enumerated

type hari = (Ahad, Senin, Selasa, Rabu, Kamis, Jum'at, Sabtu)

pred(Senin) = Ahad

ord(Ahad)=0

ord(Selasa)=2

Ahad < Senin = true

write(ord(Senin)) tertulis 1

Jenis Subrange:

Mengambil sebagian dari rentang data:

type nama = data_pertama .. data_terakhir

Contoh:

type hari = (Ahad, Senin, Selasa, Rabu, Kamis, Jum'at, Sabtu)

harikerja = Senin .. Jum'at;

bulan = 1..12;

tanggal = 1..31;

1.4. Input dan Output Data

Data input dan output:

a. Dapat disimpan di dalam file yang akan dibaca dan ditulis pada saat program berjalan

b. Diberikan atau disajikan melalui layar dan keyboard

Contoh:

PROGRAM penggajian(input, output);

Statemen-statemen yang digunakan untuk input/output.

1.4.1 Read dan Readln

Perintah ini digunakan untuk memasukkan [input] data lewat keyboard ke dalam suatu variabel.
Sintaks: Read/Readln(x); (ingat, selalu diakhiri dengan titik koma [;])
Keterangan : x = variabel.

Read = pada statemen ini posisi kursor tidak pindah ke baris selanjutnya.
Readln = pada statemen ini posisi kursor akan pindah ke baris selanjutnya setelah di input..

1.4.2. Write dan Writeln

Digunakan untuk menampilkan isi dari suatu nilai variable di layar.

Sintaks: Write/Writeln(x);

Keterangan : x = variabel.

Write/Writeln= statement ini digunakan untuk mencetak variable ke dalam monitor

Perbedaan keduanya:

writeln menyebabkan write atau writeln berikutnya menuliskan outpun di baris baru, sedangkan write tidak demikian

Contoh program menggunakan bahasa pemrograman Pascal

```
//Menuliskan Pernyataan didalam writeln

program Output;   //header atau kepala
uses crt; //mengambil unit library dalam hal ini fungsi pada BIOS

begin  //bagian pernyataan program
  clrscr; //membersihkan layar
  writeln('Selamat Datang Mahasiswa Baru');
  //menampilkan pernyataan di layar
  readkey; //untuk membaca program
end. //bagian akhir program
```

Program di atas hanya sekedar menuliskan kalimat `'Selamat Datang Mahasiswa Baru'` . Ini berarti parameter yang ada di dalam kurung writeln akan dituliskan di layar. Apa yang tertulis di antara tanda petik tunggal akan dituliskan apa adanya.

Program Input dan Output :

```
program InputOutput;
uses crt;
var
   nama : String; //variabel nama sebagai wadah penyimpanan
begin
   clrscr;
   write('Masukkan Nama Anda : ');
   readln(nama); //menampung nama pada variabel nama
   //menampilkan hasil tampungan
   writeln('Selamat Datang Di Kampus ',nama);
   readkey;
end.
```

MATERI 2- STATEMENT PENUGASAN DAN PENGULANGAN

2.1. Statement IF

Statement `If` akan diikuti oleh ekpresi (sebagai kondisi yang yang akan di periksa) dan selalu berpasangan dengan kata kunci`then`. Apabila statement yang akan dilakukan hanya satu maka kita tidak perlu menuliskan blok `begin…end`. Namun apabila statementlebih dari satu maka `blok begin…end` harus dituliskan. Untuk pempermudah pembahasan, maka statement `if` akan kita klasifikasi ke dalam tiga bagian, yaitu statement `if` untuk satu kasus, dua kasus dan tiga kasus.

2.1.1 Satu Kasus

Statement if dengan satu kasus merupakan bentuk yang paling sederhana karena hanya melibatkan satu kondisi yang akan diperiksa. Apabila kondisi yang diperiksa bernilai benar, maka proses akan mengeksekusi bagian yang berada dalam blok. Bila sebaliknya, program akan mengakibatkan statement di dalam blok dan akan langsung melanjutkan eksekusi ke statement-statement berikutnya yang berada di bawah blok pemilihan.

Adapun bentuk umum untuk mendefinisikan blok pemilihan yang memiliki satu kasus adalah sebagai berikut :

```
program IFSatuKasus;
uses crt;
var
  bil : Integer; //variabel bil sebagai wadah penyimpanan
begin
  clrscr;
  bil := 2016; //deklarasi nilai pada variabel bil
  if bil > 2015 then //statement if
  writeln('Sekarang adalah tahun ',bil);
  readkey;
end.
```

2.1.2 Dua Kasus

Statement `if` yang melibatkan dua kasus merupakan perluasan dari bentuk pertama yang telah anda pelajari di atas. Konsepnya sederhana, pada bentuk pemilihan ini terdapat penambahan statement (bisa berbentuk gabungan statement dalam satu blok) yang digunakan untuk mengatasi kejadian pada saat kondisi yang didefinisikan bernilai salah atau tidak terpenuhi. Statement alternatif yang akan dieksekusi apabila kondisi yang didefinisikan bernilai salah. Untuk mendefinisikan blok pemilihan jenis ini, kita harus menggunakan kata kunci `else`.

Apabila kondisi bernilai benar (*true*), maka akan dilakukan adalah statement pertama. Sebaliknya, apabila salah (*false*) maka yang akan dilakukan adalah statement kedua, yaitu statement yang terdapat pada bagian `else`.Bentuk umum tersebut berlaku apabila kita hanya memiliki satu baris statement yang akan dilakukan. Satu hal lagi yang perlu di perhatikan adalah statement pertama (sebelum kata kunci `else`) tidak boleh diakhiri oleh titik koma (;). Apabila ingin menempatkan titik koma, maka harus menempatkan statement tersebut ke dalam sebuah blok (`begin... end`). Hal tersebut juga berlaku apabila statement lebih dari satu baris, seperti yang ditujukan di bawah ini :

```
program IFDuaKasus;
uses crt;

var
bil : Integer;

begin
clrscr;
write('Masukkan Angka/Bilangan : ');
readln(bil);

if bil mod 2 = 0 then
writeln('Genap')
else
writeln('Ganjil');
readkey;

end.
```

2.1.3 Tiga Kasus atau Lebih

Bentuk ini merupakan bentuk pilihan yang sedikit kompleks dibandingkan dua bentuk di atas. Pasalnya, bentuk ini memiliki tiga buah atau lebih kasus sehingga akan terdapat statement `if` di dalam `if` lainnya, yang sering dikenal dengan sebutan `if` bersarang (*nested*-`if`). Apabila kondisi pertama tidak terpenuhi, maka program akan mengecek kondisi kedua. Apabila ternyata kondisi kedua juga belum terpenuhi maka program akan mengecek kondisi berikutnya, begitu seterusnya sampai ditemukan kondisi yang sesuai. Namun apabila ternyata semua kondisi yang didefinisikan tidak terpenuhi maka program akan mengeksekusi statement alternatif yang berada pada bagian akhir (di dalam bagian `else`yang akhir). Berikut ini bentuk umum untuk pendefinisian statement `if` yang memiliki tiga kasus atau lebih.

```
program IFTigaKasusAtauLebih;
uses crt;

var
bil : Integer;

begin
clrscr;
write('Masukkan Angka[1-10] : ');
readln(bil);

if bil > 5 and bil <= 10 then
writeln('Baik!')
else
if bil > 0 and bil <= 5 then
writeln('Kurang!')
else
writeln('Tidak Terdaftar!');
readkey;

end.
```

2.2 Statement Case

Apabila kondisi yang didefinisikan di dalam statement `if` semakin banyak.maka hal tersebut tentu akan menjadi hal yang cukup kompleks. Oleh karena itu, Pascal menyediakan alternatif lain untuk melakukan pemilihan yang didasarkan pada nilai-nilai constant tertentu, yaitu dengan menggunakan statement `case`. Dalam statement `case`, nilai-nilai konstan yang didefinisikan sebagai nilai pilihan harus bersifat unik dan berasal dari tipe ordinal (misal `char`, `integer`, `byte`, `Boolean`).Nilai tersebut harus berupa nilai konstan (tidak boleh berupa variabel maupun ekspresi). Perhatikan bentuk umum pendefinisian statement `case` di bawah ini

```
programCaseOf;
uses crt;

var
bil : Integer;

begin
clrscr;
write('Masukkan Angka[1-10] : ');
readln(bil);

case bil of
 0..2 : begin
   writeln('Bayi');
 end;
 3..5 : begin
   writeln('Balita');
 end;
 6..10 : begin
   writeln('Anak-Anak');
 end
 else
   writeln('Tidak Terdaftar');
end;
readkey;

end.
```

Statement default yang berada pada bagian else di atas bersifat opsional, artinya bisa dituliskan bisa juga tidak, hal ini tentu tergantung dari kasus program yang akan kita tulis. Selain itu kita dapat mendefinisikan beberapa nilai untuk menyatakan suatu satu pilihan (sama seperti menggunakan operator `or` pada statement `if`), yaitu dengan cara memisahkan nilai-nilai tersebut dengan tanda koma. Selain itu, nilai juga dapat didefinisikan dalam bentuk subrange (rentang).

TUGAS DIBERIKAN OLEH ASISTEN LABORATOTIUM

DIKERJAKAN DIKELAS!

2.3 Statement For

Konstruksi pengulangan `for` pada umumnya digunakan untuk melakukan pengulangan yang banyaknya sudah diketahui secara pasti (tanpa adanya kondisi yang harus diperiksa).Dalam pengulangan jenis ini kita selalu membutuhkan sebuah variabel sebagai indeks pengulangan yang dapat bertipe bilangan bulat, karakter maupun enumerasi.Berikut ini bentuk umum pendefinisiannya.

```
program ForToDo;
uses crt;

var
n,i : Integer;

begin
clrscr;
write('Masukkan Angka : ');
readln(n);

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

readkey;
end.
```

Banyaknya pengulangan yang akan dilakukan dihitung dari mulai nilai batas awal sampai nilai batas akhir. Perlu diperhatikan bahwa nilai dari batas awal harus lebih kecil dari pada nilai batas akhir. Apabila nilai batas awal sama dengan nilai batas akhir, maka statement hanya akan dilakukan satu kali. Nilai awal dari indeks dapat dimulai dari berapa saja, asal nilai akhirnya disesuaikan dengan banyaknya pengulangan yang akan dilakukan. Adapun nilai dari variabel indeks tersebut akan dinaikkan sebesar 1 secara otomatis setiap akhir pengeksekusian statement.

2.4 Statement While

Berbeda dengan bentuk pengulangan `for`, pada konstruksi pengulangan `while` ini terdapat suatu kondisi yang harus diperiksa terlebih dahulu. Apabila kondisi yang didefinisikan bernilai benar (*true*) maka statement yang terdapat dalam blok pengulangan pun akan dieksekusi. Sebaliknya, apabila kondisi salah (*false*), maka program tidak akan pernah memasuki blok pengulangan. Dengan kata lain, statement dalam blok pengulangan akan diabaikan. Untuk itu, dalam pengulangan jenis ini, biasanya digunakan sebuah variabel sebagai indeks pengulangan sekaligus sebagai kondisi yang akan diperiksa. Nilai dari variabel tersebut juga tentu harus diinisialisasi terlebih dahulu sehingga variabel tersebut memiliki nilai awal pada saat pengecekan kondisi. Berikut ini bentuk umum dari pendefinisian pengulangan dengan menggunakan statement `while`.

```
program Whiledo;
uses crt;

var
n,i : Integer;

begin
clrscr;
write('Masukkan Angka : ');
readln(n);

i:=1;
while i<=n do
begin
  writeln(i);
  i := i+1;
end;

readkey;
end.
```

2.5 Statement Repeat

Blok pengulangan jenis ini sebenarnya mirip dengan blok pengulangan `while`, perbedaannya hanya pada jenis ini penanggulangannya akan terus dilakukan apabila kondisi yang didefinisikan masih bernilai *false*. Jadi, dengan kata lain pengulangan hanya akan dihentikan apabila kondisi bernilai *true*. Dalam pengulangan jenis ini kondisi akan dituliskan di akhir blok. Hal ini tentu menyebabkan pada blok pengulangan jenis ini statement minimal akan di eksekusi satu kali, walaupun ternyata kondisi yang didefinisikan terpenuhi atau bernilai *true*. Untuk itu, pengulangan jenis ini pada umumnya digunakan untuk kasus-kasus pengulangan yang tidak tergantung pada kondisi awal. Adapun bentuk umum pendefinisiannya adalah sebagai berikut :

```
program RepeatUntil;
uses crt;

var
n,i : Integer;

begin
clrscr;
write('Masukkan Angka : ');
readln(n);

i:=1;
repeat
begin
  writeln(i);
end;
i:=i+1;
until i > n ;

readkey;
end.
```

TUGAS DIBERIKAN OLEH ASISTEN LABORATOTIUM

DIKERJAKAN DIKELAS!

MATERI 3- ARRAY

3.1 Definisi Array

Array (larik) adalah sebuah variabel yang dapat menyimpan lebih dari satu nilai sejenis (memiliki tipe data sama). Hal ini tentu berbeda dengan variabel biasa yang hanya mampu menampung satu buah nilai.Setiap nilai yang di simpan di dalam array disebut dengan elemen array, sedangkan nilai urut yang digunakan untuk mengakses elemennya disebut dengan indeks array.

3.2 Deklarasi Array

Sama seperti variabel lainnya, array memiliki deklarasi oleh variabel. Bila akan didefinisikan sebagai tipe bentukan, maka array juga dideklarasikan di bagian definisi tipe. Berikut ini bentuk pendeklarasiannya.

```
NamaArray :array [IndeksAwal..IndeksAkhir] of tipe_data ;
```

Di bawah ini contoh kode yang dapat digunakan untuk mendeklarasikan 10 buah elemen array bertipe integer sebagai penggantian kode di atas.

```
Var
A1 : array[0..9] of integer ;
A2 : array[5..15] of integer ;
A3 : array['a'..'j'] of integer ;
A4 : array['A'..'J'] of integer ;
```

Dalam bahasa pascal, tersedia dua buah fungsi yang dapat digunakan untuk mengambil indeks terendah dan tertinggi dari sebuah array, yaitu fungsi Low dan High. Adapun parameter dari kedua fungsi tersebut adalah nama array yang akan dicari indeksnya.

Perhatikan contoh di bawah ini :

```
program DeklarasiArray;
uses crt;

var
aa: array[0..9] of Integer;
t,r : integer;
begin
clrscr;
  r := Low(aa);
  t := High(aa);
  writeln('Angka Terkecil : ',r,', Angka Terbesar : ',t);
readkey;
end.
```

3.3 Elemen Array

Setelah mengetahuicara pendeklarasian array, maka kini saatnya membahas bagaimana cara untuk memanipulasi array tersebut. Langkah pertama yang harus dilakukan adalah mengisikan nilai ke dalam elemen-elemen array bersangkutan. Adapun bentuk umum untuk pengisian elemen array adalah sebagai berikut :

```
NamaArray [indeks] : = nilai ;
```

Untuk lebih memahaminya, perhatikan contoh berikut :

```
program ArrayElemen;
uses crt;

var
aa: array[0..5] of Integer;
i : integer;
begin
clrscr;
aa[1]:= 2;
aa[2]:= 4;
aa[3]:= 6;
aa[4]:= 8;
aa[5]:= 10;

for i:= 1 to 5 do
begin
  writeln('Angka ',aa[i]);
end;

readkey;
end.
```

3.4 Array Berisi Record

Konsep dasar array berisi record hampir sama seperti pada saat mendefinisikan array untuk tipe dasar, seperti integer, char, real dan juga lainnya. Langkah awal yang harus dilakukan adalah mendefinisikan record terlebih dahulu yang selanjutnya akan digunakan sebagai tipe data pada saat pendeklarasian array. Untuk lebih jelasnya dapat melihat contoh di bawah ini :

```
program ArrayRecord;
uses crt;
Type
  Mahasiswa = record
    NRP : String[11];
    Nama : String[30];
    Umur : Integer;
  end;
  Person = Array[1..100] of Mahasiswa;
var
  A:Person;
  n,i : integer;
```

{Lengkapi Souce Code diatas dengan menampilkan 2 Mahasiswa}

Hasil :

```
Masukkan Jumlah Mahasiswa : 2
Mahasiswa ke-1
Masukkan NRP : 15-2013-073
Masukkan Nama : Mochamad Angga
Masukkan Umur : 20
-----------------------------------
Mahasiswa ke-2
Masukkan NRP : 15-2013-054
Masukkan Nama : Gian Permana
Masukkan Umur : 20
-----------------------------------
NRP :15-2013-073, Nama : Mochamad Angga, Umur : 20
NRP :15-2013-054, Nama : Gian Permana, Umur : 20
```

3.5 Mengurutkan Elemen Array

Di dalam pemrograman, terdapat beberapa metode untuk melakukan pengurutan data.Namun terdapat delapan yang umumnya banyak digunakan. Adapun metode-metode tersebut antara lain :

1. Bubble Sort
2. Maximum/Minimum Sort
3. Selection Sort
4. Insertion Sort
5. Heap Sort
6. Quick Sort
7. Merge Sort
8. Shell Sort

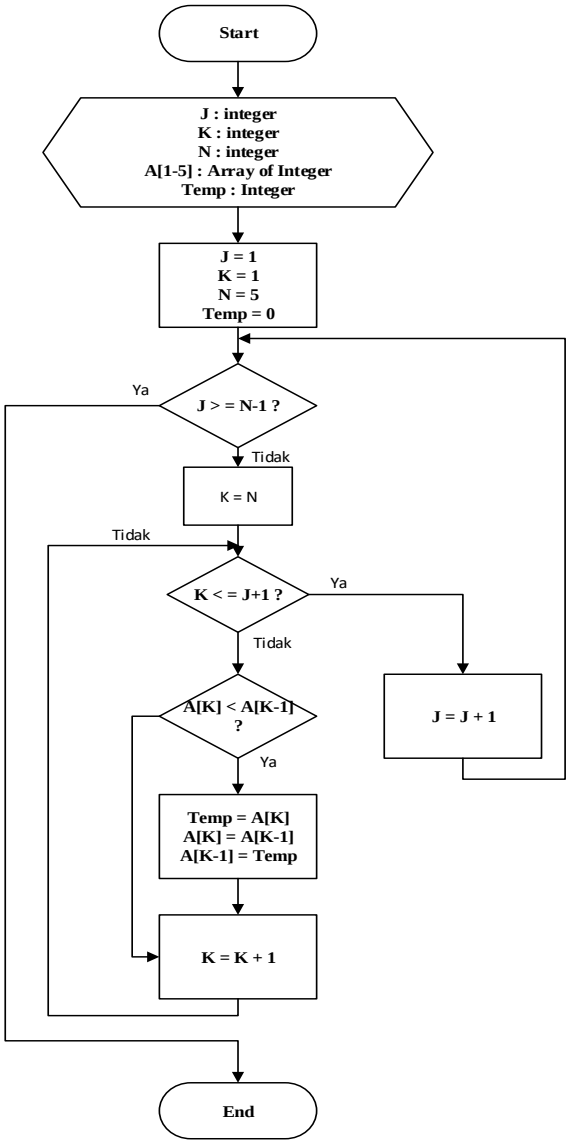
Berikut ini adalah contoh listing program mengurutkan bilangan

```
program pengurutan;
uses crt;
const
  N = 5;
  A : array [1..N] of Integer = (27,18,23,25,15);
var
  j,k,temp : Integer;
begin
clrscr;
//menampilkan data sebelum proses pengurutan
writeln('Data Sebelum Diurutkan');
for j := 1 to N do
begin
  writeln('A[',j,'] = ',A[j]);
end;

//melakukan proses pengurutan data
for j := 1 to N-1 do begin
  for k := N downto j+1 do
  begin
    if A[k] < A[K-1] then
  begin
  temp:= A[k];
  A[k] := A[k-1];
  A[k-1] := temp;
  end;
  end;
end;

//menampilkan proses pengurutan data
WriteLn();
writeln('Data Setelah Diurutkan');
for j := 1 to N do
begin
  writeln('A[',j,'] = ',A[j]);
end;
readkey;
end.
```

Flowchart :



Hasil :

```
Data Sebelum Diurutkan
A[1] = 27
A[2] = 18
A[3] = 23
A[4] = 25
A[5] = 15

Data Setelah Diurutkan
A[1] = 15
A[2] = 18
A[3] = 23
A[4] = 25
A[5] = 27
```

TUGAS DIBERIKAN OLEH ASISTEN LABORATOTIUM

DIKERJAKAN DIKELAS!

MATERI 4 - PROSEDUR DAN FUNGSI

4.1 Prosedur

Prosedur merupakan suatu rutin yang melakukan proses tertentu tanpa adanya pengambilan nilai. Procedure menyediakan suatu metode pengisolasian bagian program secara terpisah yang dapat dipanggil atau diaktivasi dari bagian manapun di dalam program.Definisi prosedur dalam bahasa pascal adalah `procedure`.Adapun bentuk umum definisi dari prosedur sebagai berikut :

```
Procedure namaprosedur(parameter1: tipe_data, parameter2: tipe_data,...);
     Parameter2: tipe_data,... );
     Const
         {daftar konstanta lokal}
     Var
         {Daftar pendeklarasian variabel lokal}
     Begin
         {Kode program yang akan ditulis}
     ...
```

Berikut contoh penggunaan procedure untuk menghitung penjumlahan dua bilangan :

```
program procedureJumlah;
uses crt;
var
  a,b : integer;

procedure jumlah (angk1,angk2: integer);
var
   jml : integer;
begin
  jml := angk1 + angk2 ;
  writeln('Hasil Penjumlahan');
  writeln(angk1,' + ',angk2,' = ',jml);
end;

begin
clrscr;
 write('Masukkan Angka ke-1 : ');
 readln(a);
 write('Masukkan Angka ke-2 : ');
 readln(b);
 jumlah(a,b);

readkey;
end.
```

Sourcecode diatas menunjukkan bahwa pada program tersebut ‘procedure jumlah’ digunakan untuk memanggil proses penjumlahan.

4.2 Fungsi

Definisi fungsi sebenarnya sama dengan sebuah prosedur. Perbedaannya, pada fungsi terdapat pengambilan nilai, sehingga pada saat pemanggilan, fungsi dapat langsung digunakan untuk mengisikan sebuah ekspresi.Berbeda dengan prosedur yang didefinisikan dengan kata kunci `procedure`, fungsi didefinisikan dengan kata kunci `function`.Berikut ini bentuk umum dari definisi fungsi.

```
function NamaFungsi
(parameter1 : tipe_data,
parameter 2 : tipe_data, …) : tipe_data;

cont   {daftar konstanta lokal}
var{daftar pendeklarasian variabel lokal}
begin  {kode program yang akan ditulis}
…
NamaFungsi := nilai_kembalian ;     {ingat baris ini}
end;
```

Berikut contoh penggunaan function untuk menghitung penjumlahan dua bilangan :

```
program FunctionPenjumlahan;
uses crt;
var
  a,b : integer;

function jumlah (angk1,angk2: integer):integer;
begin
 jumlah := angk1 + angk2 ;
 writeln('Hasil Penjumlahan');
 writeln(angk1,' + ',angk2,' = ',jumlah);
end;

begin
clrscr;
 write('Masukkan Angka ke-1 : ');
 readln(a);
 write('Masukkan Angka ke-2 : ');
 readln(b);
 jumlah(a,b);

readkey;
end.
```

Sourcecode diatas menunjukkan bahwa pada program tersebut ‘function penjumlah’ digunakan untuk memanggil hasil penjumlahan.

TUGAS DIBERIKAN OLEH ASISTEN LABORATOTIUM

DIKERJAKAN DIKELAS!