

MODUL PRAKTIKUM

ALGORITMA DAN PEMROGRAMAN I



Disusun Oleh :
Dosen Teknik Informatika dan Asisten Praktikum

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA
2020

**TATA TERTIB DAN TATA LAKSANA PRAKTIKUM
JURUSAN/PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK UPR**

TATA TERTIB

1. Praktikan **WAJIB** mengikuti semua modul praktikum.
2. Praktikan hanya boleh tidak mengikuti praktikum **1 (satu) kali DENGAN ATAU TANPA surat izin** dari jumlah pertemuan praktikum.
3. Praktikan yang berhalangan mengikuti praktikum, diwajibkan melaporkan ke asisten praktikum untuk menentukan jadwal praktikum sebagai pengganti jadwal yang berhalangan.
4. Praktikan yang lebih dari 1 (satu) kali tidak mengikuti praktikum, tidak diperbolehkan untuk mengikuti praktikum untuk modul-modul praktikum selanjutnya dan **NILAI AKHIR PRAKTIKUM** adalah **NOL**.
5. Praktikum dilaksanakan menggunakan media Zoom Meeting.
6. Semua peserta praktikum wajib menyalakan kamera saat praktikum berlangsung.
7. Praktikan wajib hadir 15 menit sebelum praktikum dimulai.
8. Tidak diperbolehkan saling bekerjasama.
9. Dilarang menggunakan kaos oblong dan sandal jepit selama praktikum. Bagi yang melanggar poin ini, tidak diperbolehkan mengikuti praktikum.
10. Segala bentuk kecurangan dan plagiarisme, baik pada laporan maupun test praktikum, akan berakibat pada nilai E sebagai nilai akhir praktikum.

TATA LAKSANA

1. Sebelum praktikum dimulai praktikan wajib mengumpulkan 2 buah laporan, yaitu: Laporan Rencana Praktikum, dan Laporan Hasil Praktikum modul sebelumnya. Tanpa mengumpulkan kedua laporan ini praktikan tidak diperkenankan mengikuti praktikum.
2. Setiap modul praktikum, akan dilakukan ***Pre-test***.
3. Format laporan meliputi:
Laporan Rencana Praktikum :
 - Format A4
 - Tulis Tangan.
 - Tujuan Praktikum dari modul yang akan dilaksanakan
 - Hal-hal yang akan dilakukan selama praktikum.
Laporan Hasil Praktikum :
 - Halaman Depan
 - Bab I. Tujuandan Landasan Teori
 - Bab II. Pembahasan
 - Bab III. Kesimpulan
 - Daftar Pustaka
 - Lampiran (disertai laporan rencana praktikum)

Format Penulisan Laporan Hasil Praktikum

- Spasi : 1,5
- Font : Times New Roman
- Font Size : 12
- Margins : Top 3, Left 4, Right 3, Bottom 4
- Kertas : A4

Penilaian Laporan Hasil Praktikum :

| | |
|------------------------------------|------|
| ▪ Sampul Depan | 5% |
| ▪ Bab I. Tujuan dan Landasan Teori | 30% |
| ▪ Bab II. Pembahasan | 30% |
| ▪ Bab III. Kesimpulan | 25% |
| ▪ <u>Daftar Pustaka</u> | 10% |
| Total | 100% |

4. Praktikan yang mengabaikan format penulisan poin 5, nilai akan dikurangi 5 setiap kesalahan.

5. Penilaian Akhir Praktikum :

| | |
|---------------------|-------|
| ▪ <i>Pre-Test</i> | : 5% |
| ▪ Praktikum | : 30% |
| ▪ Laporan Praktikum | : 15% |
| ▪ <u>Responsi</u> | : 50% |
| ▪ Total | 100% |

6. Penilaian Akhir Mata Kuliah :

| | |
|------------------------------------|-------------|
| ▪ Tugas di kelas + UTS + Praktikum | 50% |
| ▪ <u>UAS</u> | 50% |
| ▪ Total | 100% |

MODUL 1

PENGENALAN PASCAL DAN TIPE DATA

1. Tujuan

- 1.1. Mahasiswa mengenal bahasa Pascal
- 1.2. Mahasiswa mengetahui jenis tipe data dan variabel dalam pascal
- 1.3. Mahasiswa dapat menentukan variable dan tipe data yang sesuai .
- 1.4. Mahasiswa mengetahui operasi terhadap variable dengan jenis tipe data yang telah ditentukan

2. Pembahasan

Pascal adalah bahasa pemrograman yang pertama kali di buat oleh Profesor Niklaus Wirth, seorang anggota Internasional Federation Processing (IFIP) pada tahun 1971. Dengan mengambil nama dari matematikawan Perancis, Blaise Pascal, yang pertama kali menciptakan mesin penghitung, Profesor Niklaus Wirth membuat bahasa pascal ini sebagai alat bantu untuk mengajarkan konsep pemrograman komputer kepada mahasiswanya. Selain itu, Profesor Niklaus Wirth membuat pascal juga untuk melengkapi kekurangan-kekurangan bahasa pemrograman yang ada pada saat itu.

Struktur Pascal

1. Bagian Judul Program

2. Blok Program

Bagian Deklarasi

a. Deklarasi tipe data (TYPE)

Dalam bahasa Pascal terdapat beberapa jenis tipe data yang bisa digunakan untuk sebuah variabel atau konstanta pada program antara lain adalah:

| Tipe Data | Deskripsi (range variabel) |
|-----------------------|---------------------------------------|
| <u>Byte</u> | angka dari 0 - 255 |
| <u>Integer</u> | angka dari -32768 - 2767 |
| <u>Real</u> | semua nilai pecahan dari 1E-38 - E+38 |
| <u>Boolean</u> | nilai TRUE atau FALSE |
| <u>Char</u> | semua karakter dari tabel ASCII |
| <u>String</u> | semua huruf, spasi, frasa |

- b. Deklarasi variabel (VAR)
- c. Deklarasi konstanta (CONST)
- d. Deklarasi label (LABEL)
- e. Deklarasi sub-program (PROCEDURE dan FUNCTION)

3. Bagian Program Utama Perintah-perintah atau Statemen

Statemen diawali oleh kata BEGIN dan diakhiri dengan kata END.

Akhir dari setiap statemen diakhiri dengan titik koma [;].

a. Read atau readln

Perintah read maupun readln ini digunakan untuk memasukan atau menginputkan data ke dalam suatu variabel.

b. Write atau writeln

Perintah write maupun writeln ini digunakan untuk menampilkan atau mencetak isi dari suatu nilai variabel.

c. Clrscr

Clrscr (clear screen) pada bahasa pemrograman pascal ini bertujuan membersihkan tampilan pada layar.

Penulisan Variabel dalam Bahasa Pascal.

Variabel adalah nama yang mewakili suatu elemen data. Ada aturan tertentu yang wajib diikuti dalam pemberian nama variable, antara lain :

- a. Harus dimulai dengan abjad, tidak boleh dengan angka atau symbol.
- b. Tidak boleh ada spasi diantaranya
- c. Jangan menggunakan simbol.

3. Tugas

1. Tulislah 3 program dibawah ini dengan memberikan nama program

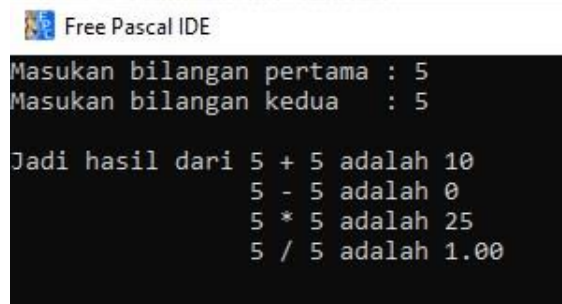
Alpro1_1, Alpro1_2, dan Alpro1_3 :

```
Program Alpro_1;  
Uses crt;  
Begin  
Writeln('Saya Belajar Bahasa Pascal');  
Readln;  
End.
```

```
Program Alpro1_2;  
Uses crt;  
Var kalimat : string;  
Begin  
Kalimat := 'Saya Belajar Bahasa Pascal';  
Writeln(kalimat);  
Readln;  
End.
```

```
Program Alpro1_3  
Uses crt;  
Const kalimat = 'Saya Belajar Bahasa Pascal';  
Begin  
Writeln(kalimat);  
End.
```

2. Buatlah program untuk melakukan operasi aritmatika dimana bilangan pertama dan kedua merupakan inputan dari piranti masukan.



```
Free Pascal IDE
Masukan bilangan pertama : 5
Masukan bilangan kedua   : 5

Jadi hasil dari 5 + 5 adalah 10
                  5 - 5 adalah 0
                  5 * 5 adalah 25
                  5 / 5 adalah 1.00
```

3. Buatlah program untuk menghitung luas lingkaran . Luas lingkaran didefinisikan sbb : $L = \text{Pi} * r * r$, dimana pi adalah konstanta yang bernilai 3,1415 sedangkan r adalah jari-jari lingkaran.
4. Buatlah program untuk mengkonversi suhu dalam derajat celcius ke derajat fahrenheit dan reamour dimana :
Fahrenheit = $9/5 * \text{celcius} + 32$
Reamour = $4/5 * \text{celcius}$

MODUL II

RUNTUTAN (SEQUENTIAL)

1. Tujuan

- 1.1. Dapat menggolongkan sebuah masalah ke dalam kategori runtunan (*sequential*).
- 1.2. Dapat memecahkan permasalahan yang sifatnya runtunan ke dalam sebuah program.

2. Landasan Teori

Runtunan adalah struktur algoritma paling dasar yang berisi rangkaian instruksi yang diproses secara sekuensial, artinya dikerjakan satu per satu, mulai dari instruksi pertama sampai instruksi akhir.

Sebuah instruksi dieksekusi setelah instruksi sebelumnya selesai dieksekusi. Urutan instruksi menentukan keadaan hasil dari algoritma. Bila urutannya diubah, maka hasil akhirnya mungkin juga berubah. Instruksi algoritma pada struktur runtunan ini akan dikerjakan semua dan tidak ada yang diabaikan. Pada dasarnya, semua program dibuat berdasarkan struktur algoritma runtunan.

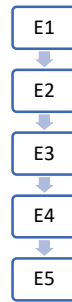
Menurut Goldshlager dan Lister (1998) struktur berurutan atau runtunan mengikuti ketentuan-ketentuan sebagai berikut.:

- a) Tiap instruksi dikerjakan satu persatu.
- b) Tiap instruksi dilaksanakan tepat sekali, tidak ada yang diulang.
- c) Urutan instruksi yang dilaksanakan pemroses sama dengan urutan aksi sebagaimana yang tertulis di dalam algoritmanya.
- d) Akhir dari instruksi terakhir merupakan akhir algoritma.

Setiap instruksi di dalam runtunan ditulis dalam satu baris, atau beberapa instruksi dalam baris yang sama tetapi antara setiap instruksi dipisahkan dengan tanda titik koma (;).

Contoh struktur runtunan adalah sebagai berikut.

Bila suatu instruksi runtunan dalam algoritma ditulis berturut-turut, kita lambangkan dengan variabel E1, E2, E3, E4, E5, maka urutan pelaksanaan instruksi tersebut adalah seperti gambar 1 dibawah ini:



Gambar 1 Runtunan yang Terdiri dari 5 Buah Instruksi

Keterangan :

Runtunan diatas, dimulai dari intruksi E1, kemudian E1 harus dikerjakan sampai selesai terlebih dahulu sebelum instruksi E2 dikerjakan, setelah instruksi E2 selesai dikerjakan juga, maka instruksi E3 dikerjakan lagi dan seterusnya sampai instruksi ke E5 setelah itu proses selesai dikerjakan dan baru berhenti dari proses pembacaan runtunan tersebut.

3. Tugas

1. Buatlah program untuk melakukan operasi aritmatika dimana bilangan pertama dan kedua merupakan inputan dari piranti masukan.
2. Buatlah program untuk mencari nilai x_1 dan x_2 dari persamaan linear $\mathbf{ax^2 + bx + c = 0}$ dengan rumus :

$$x_{12} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Dimana nilai a, b dan c merupakan inputan dari piranti masukan.

3. Algoritma Hitung Buku

Deklarasi

Const Bsoni = 15

var

Badi, Banis, Bluki, Bnisa: integer

Deskripsi

Badi \leftarrow Bsoni-10

Banis \leftarrow 2*(Bsoni+Badi)

Bluki \leftarrow Bsoni+Badi+Banis-5

Banis \leftarrow Bluki*4

Write(Badi, Banis, Bluki, Banis)

Buatlah program pascal untuk menghitung buku Adi, Anis, Luki, dan Nisa berdasarkan algoritma diatas.

4. Algoritma KonversiDetik

Deklarasi

var

Detik, KonvJam, KonvMnt, KonvDtk, Sisa : integer

Deskripsi

read(Detik)

KonvJam \leftarrow Detik div 3600

Sisa \leftarrow Detik mod 3600

KonvMnt \leftarrow Sisa div 60

KonvDtk \leftarrow Sisa mod 60

write(KonvJam, KonvMnt, KonvDtk)

Buatlah program pascal untuk mengkonversi waktu dari satuan detik ke satuan jam, menit, dan detik.

MODUL III

PERCABANGAN (DECISION)

1. Tujuan

- 1.1. Dapat mengetahui perbedaan penggunaan if-then-else dan case..of dalam setiap permasalahan.
- 1.2. Dapat menggunakan if-then-else dan case of dalam menyelesaikan sebuah permasalahan yang dituangkan dalam sebuah program.

2. Landasan Teori

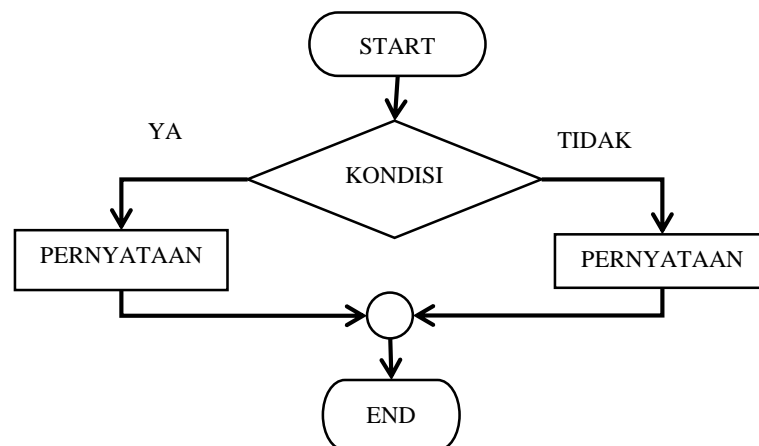
Penyeleksi kondisi atau pernyataan kondisi (conditional statement) merupakan suatu pernyataan yang menganalisa suatu keadaan dan mengambil keputusan berdasarkan pada hasil analisa itu. Hasil dari penyeleksian jika bernilai benar maka akan dikerjakan instruksi tertentu. Jika salah maka akan dikerjakan instruksi yang lain.

Pada struktur percabangan, program akan berpindah urutan pelaksanaan jika suatu kondisi yang disyaratkan terpenuhi. Pada Flowchart, simbol flowchart Decision yang digunakan pada proses ini. Simbol decision akan berisi pernyataan yang akan diuji kebenarannya. Nilai hasil pengujian akan menentukan cabang mana yang akan ditempuh.

Bentuk strukturnya adalah:

If *Kondisi* then
 Statement 1
Else
 Statement 2

Dalam bentuk Flowchart akan menjadi sebagai berikut:



Fungsi/kegunaan dari struktur pemilihan adalah untuk membuat struktur menu, memvalidasi inputan, dan mencegah error.

Konsep dasar dan bentuk struktur dari percabangan dalam bahasa pemrograman Pascal adalah sebagai berikut:

a. If – Then

```
If Kondisi Then  
begin  
  Kode Program  
end;
```

b. If – Then – Else

```
If Kondisi Then  
begin  
  Kode Program 1  
end  
Else  
begin  
  Kode Program 2  
End;
```

c. Case – Of

1) Bentuk Case - Of

```
Case kondisi of  
  nilai1      :  
  statemen1;  
  nilai2      :  
  statemen2;  
  nilai3      :  
  statemen3;  
  nilai4      :  
  statemen4;  
  .....  
End;
```

2) Bentuk Case-Of-Else

```
Case kondisi of  
  nilai1      :  
  statemen1;
```

```

.....
    nilaiN      :
    statemenN;
Else
    Statemen Lain;
End;

```

3. Tugas

1. Buatlah program untuk menentukan bilangan bulat terbesar dan yang terkecil dari tiga buah bilangan bulat!
2. Buatlah program untuk membaca nomor bulan (integer), lalu menuliskan nama bulan sesuai dengan angka bulannya. Misalnya jika dibaca angka 9, maka tercetak agustus.
3. Simulasikan sebuah kalkulator sederhana untuk melakukan operasi aritmatika sebagai berikut : dibaca operand 1, operator, dan operand 2. Tentukan hasil operasi aritmatika :

```

Operand 1      Operator      Operand 2
      7          "+"          3
Maka hasilnya adalah 10

```

4. Buatlah program untuk menentukan Harga Mutu serta keterangan Lulus/Tidak Lulus dari nilai akhir sebuah mata kuliah. Dengan ketentuan sebagai berikut.

| Nilai | Harga Mutu | Keterangan |
|-------|------------|-------------|
| > 80 | A | Lulus |
| 75-79 | B+ | Lulus |
| 70-74 | B | Lulus |
| 65-69 | C+ | Lulus |
| 60-64 | C | Lulus |
| 50-59 | D | Tidak Lulus |
| < 50 | E | Tidak Lulus |

Misalkan diberi masukan nilai 73 maka akan muncul output:

Harga mutu = B

Keterangan = Lulus

5. Buatlah program kalkulator sederhana menggunakan percabangan case..of seperti dibawah ini

```
===== My Calculator
```

```
=====
```

1. Penjumlahan

2. Pengurangan

3. Perkalian

4. Pembagian

pilih menu : 1

masukkan bilangan 1 = 3

masukkan bilangan 2 = 4

hasil perhitungan = $3 + 4 = 7$

Terima Kasih

MODUL IV

PENGULANGAN (LOOPING)

1. Tujuan

Setelah menyelesaikan praktikum ini praktikan diharapkan dapat :

- 1.1 Memahami dan menguasai metode pengulangan pada Pascal.
- 1.2 Memahami suatu metode pengulangan dengan **WHILE..DO**, **REPEAT..UNTIL**, dan **FOR**.
- 1.3 Membuat program dengan menggunakan statement **WHILE..DO**, **REPEAT..UNTIL**, dan **FOR**.

2. Landasan Teori

Terdapat dua model struktur pengulangan, yaitu:

- 1) Struktur Pengulangan Tanpa Kondisi (unconditional looping). Di dalam struktur ini, instruksi-instruksi di dalam badan pengulangan diulangi sejumlah kali yang dispesifikasikan (jumlah pengulangan sudah diketahui sebelum eksekusi). Contoh: Struktur FOR.
- 2) Struktur Pengulangan Dengan Kondisi (conditional looping). Di dalam struktur ini, jumlah pengulangan tidak diketahui sebelum eksekusi program. Yang dapat ditentukan hanya kondisi berhenti pengulangan, artinya instruksi-instruksi di dalam badan pengulangan diulangi sampai kondisi berhenti terpenuhi. Contoh: Struktur WHILE dan Struktur REPEAT.

Macam-macam struktur pengulangan:

a. Struktur FOR

Struktur pengulangan FOR digunakan untuk mengulang statemen atau satu blok statemen berulang kali. Jumlah pengulangan diketahui atau dapat ditentukan sebelum eksekusi. Untuk mencacah sudah jumlah pengulangan diperlukan sebuah variabel pencacah (*counter*). Variabel ini nilainya selalu bertambah satu setiap kali pengulangan dilakukan. Jika cacah pengulangan sudah mencapai jumlah yang dispesifikasikan, maka proses pengulangan berhenti. Pada struktur FOR, *pencacah* haruslah dari tipe data yang memiliki predecessor dan successor, yaitu

integer atau karakter. Tipe riil tidak dapat digunakan sebagai *pencacah*. Aksi adalah satu atau lebih instruksi yang diulang.

Bentuk struktur FOR ada dua macam:

1) Menaik (*Ascending*)

Pada struktur FOR menaik, nilai_awal harus lebih kecil atau sama dengan nilai_akhir. Jika nilai_awal lebih besar dari nilai_akhir, maka badan pengulangan tidak dimasuki. Pada awalnya, pencacah diinisialisasikan dengan nilai_awal. Nilai pencacah secara otomatis bertambah satu setiap kali aksi pengulangan dimasuki, sampai akhirnya nilai pencacah sama dengan nilai_akhir. Jumlah pengulangan yang terjadi = $\text{nilai_akhir} - \text{nilai_awal} + 1$. Struktur Bahasa Pascal untuk pengulangan FOR menaik ditunjukkan pada Gambar 4.1.

```
for pencacah:=nilai_awal to nilai_akhir do  
    aksi
```

Gambar 4.1 Struktur Bahasa Pascal Untuk Pengulangan FOR
Ascending

2) Menurun (*Descending*)

Pada struktur FOR menurun, nilai_akhir harus lebih besar atau sama dengan nilai_awal. Jika nilai_akhir lebih kecil dari nilai_awal, maka badan pengulangan tidak dimasuki. Pada awalnya, pencacah diinisialisasikan dengan nilai_akhir. Nilai pencacah secara otomatis berkurang satu setiap kali aksi diulangi, sampai akhirnya nilai pencacah sama dengan nilai_awal. Jumlah pengulangan yang terjadi = $\text{nilai_akhir} - \text{nilai_awal} + 1$. Struktur Bahasa Pascal untuk pengulangan FOR menurun ditunjukkan pada Gambar 4.2

```
for pencacah:=nilai_akhir downto nilai_awal do  
    aksi
```

Gambar 4.2 Struktur Bahasa Pascal Untuk Pengulangan FOR
Descending

b. Struktur WHILE

Pada struktur WHILE, *aksi* (atau runtunan *aksi*) akan dilaksanakan berulang kali selama *kondisi* bernilai *true*. Jika kondisi bernilai *false*, badan pengulangan tidak akan dilaksanakan, yang berarti pengulangan selesai. Yang harus diperhatikan adalah pengulangan harus berhenti. Pengulangan yang tidak pernah berhenti menandakan bahwa logika pemrograman tersebut salah. Pengulangan berhenti apabila *kondisi* bernilai *false*. Agar *kondisi* suatu saat bernilai *false*, maka di dalam badan pengulangan harus ada instruksi yang mengubah nilai variabel *kondisi*.

```
while kondisi do
    aksi
```

Gambar 4.3. Struktur Bahasa Pascal Untuk Pengulangan WHILE

c. Struktur REPEAT

Struktur ini mendasarkan pengulangan pada kondisi *boolean*. *Aksi* di dalam badan pengulangan diulang sampai *kondisi boolean* bernilai *true*. Dengan kata lain, jika kondisi *boolean* masih *false*, pengulangan masih terus dilakukan. Karena proses pengulangan suatu saat harus berhenti, maka di dalam badan pengulangan harus ada *aksi* yang mengubah nilai variabel *kondisi*. Struktur REPEAT mempunyai makna yang sama dengan WHILE, dan dalam beberapa masalah kedua struktur tersebut komplemen satu sama lain.

```
repeat
    aksi
until kondisi
```

Gambar 4.4. Struktur Bahasa Pascal Untuk Pengulangan REPEAT

3. Langkah Kerja

- 1) Buatlah program untuk mencetak tulisan “Fakultas Teknik” sebanyak N dimana N adalah nilai yang dimasukkan dari piranti masukan. Jumlah pengulangan : 3

Output =

Fakultas Teknik

Fakultas Teknik

Fakultas Teknik

Silahkan masukan kode program dibawah ini:

```
program modul4langkah1;
uses crt;
    var i,a : integer;

begin
    clrscr;
    write('Jumlah Pengulangan: '); readln(a);
    writeln;
    for i:=1 to a do
        writeln('Fakultas Teknik');
    readkey;
end.
```

- 2) Ubahlah program praktikum langkah 1 dengan menggunakan pengulangan while..do
- 3) Buatlah program untuk mencetak bintang berikut ini jika diberikan tinggi segitiga adalah N. Sebagai contoh, jika $N = 4$ maka outputnya adalah :

```
$$$$
$$$
$$
$
$
$
$$$
$$$$
```

Silahkan masukan kode program dibawah ini:

```
program modul4langkah3;
uses crt;
    var i,j,a : integer;

begin
    clrscr;
    write('Jumlah Pengulangan: '); readln(a);
    writeln
    for i:= a downto 1 do
        begin
            for j:=1 to i do
                write('$');
            writeln;
        end;
    readkey;
end.
```

- 4) Buatlah program kalkulator untuk melakukan berbagai operasi aritmatika

Output dari program adalah seperti pada tampilan di bawah ini :

```
===== MENU UTAMA =====  
PROGRAM KALKULATOR  
-----  
1. Penjumlahan  
2. Perkalian  
3. Pengurangan  
4. Pembagian  
5. Pangkat Dua ( N^2 )  
6. Keluar  
-----  
Pilihan Menu : 1  
2 + 5 = 7  
Anda ingin mengulang lagi? ( Y/T ) :
```

Aturannya adalah : Jika jawaban pilihan jawaban di atas : Y (Ya) maka masih melakukan proses menu penjumlahan (masih menu yang dipilih saat itu), jika T (Tidak), maka akan kembali ke Menu Utama.

MODUL V

PROCEDURE & FUNCTION

1. Tujuan

- 1.1. Dapat memahami konsep prosedur dan fungsi
- 1.2. Mampu membuat prosedur dan fungsi baik dengan parameter maupun tanpa parameter
- 1.3. Mampu membedakan kapan menggunakan prosedur maupun fungsi

2. Pembahasan

Procedure/prosedur merupakan suatu program yang terpisah dalam blok sendiri yang berfungsi sebagai sub-program (modul program) yang merupakan sebuah program kecil untuk memproses sebagian dari pekerjaan program utama. Procedure berguna untuk mengumpulkan statement-statement (Pernyataan) yang dapat dijalankan menjadi satu dalam blok dan untuk menjalankan kembali hanya dengan menuliskan nama procedure yang menampungnya, yaitu pemanggilan dilakukan pada program utama.

Sedangkan Fungsi hampir sama dengan Prosedur. Perbedaannya adalah bahwa fungsi memiliki tipe data sedangkan prosedut tidak memiliki tipe data. Artinya bahwa sebuah fungsi dapat mengembalikan nilai sesuai dengan tipenya ketika menerima panggilan.

Bentuk umum procedure & function

```
Procedure nama_procedure  
(parameter_input:tipe ; var parameter output:tipe)  
Begin  
  (kode program)  
End.
```

```
Function Nama_Fungsi  
(parameter_input:tipe; Var parameter_output : tipe);  
Begin  
  <pernyataan dalam Fungsi>;  
End;
```

Parameter

Parameter adalah nama variabel yang dideklarasikan pada bagian header Fungsi.

1. Parameter aktual (argumen) adalah parameter yang disertakan pada waktu pemanggilan Fungsi.
2. Parameter formal adalah parameter yang dideklarasikan di dalam bagian header Fungsi itu sendiri.

Aturan Parameter

Aturan yang harus diperhatikan antara parameter aktual dan parameter formal:

1. Jumlah parameter aktual pada pemanggilan Fungsi harus sama dengan jumlah parameter formal pada header Fungsinya.
2. Tiap parameter aktual harus bertipe sama dengan tipe parameter formal yang bersesuaian

- Procedure tanpa parameter

```
program luassegitiga;
uses crt;
var a,t,h:real;
procedure luasseg3;
begin clrscr;
write('masukan nilai alas:'); readln(a);
write('masukan nilai tinggi:'); readln(t);
h:=a*t/2;
writeln('Hasilnya adalah:',h:0:0);
end;
begin
luasseg3;
readln;
end.
```

- Procedure dengan parameter

```

program luassegitiga;
uses crt;
var a,t:real;
procedure luas segi3(a,t:real);
var h:real;
begin
h:=a*t/2;
writeln('Hasilnya adalah:',h:0:0);
end;
begin
clrscr;
write('masukan nilai alas:'); readln(a);
write('masukan nilai tinggi:'); readln(t);
luas segi3(a,t);
readln;
end.

```

- Function tanpa parameter

```

program luaspersegi_panjang;
uses crt;
function lpersegi panjang:string;
var p,l,h:real;
begin
write('masukan nilai panjang:'); readln(p);
write('masukan nilai lebar:'); readln(l);
h:=p*l;
writeln('Hasilnya adalah:',h:0:0);
end;
begin
clrscr;
lpersegi panjang;
readln;
end.

```

- function dengan parameter

```

program luaspersegi panjang;
uses crt;
function lpersegi panjang(p,l:integer):real;
begin lpersegi panjang:=p*l;
end;
var p,l:integer;
begin
clrscr;
write('masukan nilai panjang:'); readln(p);

```

```
write('masukan nilai lebar:');readln(l);  
writeln('Hasilnya adalah:',lpersegipanjang(p,l):0:0);  
readln;  
end.
```

3. TUGAS

1. Buatlah Kalkulator pengurangan, penjumlahan menggunakan Procedure tanpa parameter.
2. Buatlah Kalkulator perkalian dan pembagian menggunakan Procedure dengan parameter.
3. Buatlah procedure untuk menghitung volume balok dan volume kubus.
4. Buatlah Kalkulator (penjumlahan, pengurangan, perkalian, pembagian) menggunakan function
5. Buatlah program menghitung keliling dan luas segitiga sama kaki.
6. Buatlah program menghitung keliling dan luas lingkaran.