

MODUL PRAKTIKUM

PRAK. ALGORITMA & PEMROGRAMAN 2



Versi	3.1
Tahun Penyusunan	2012
Tim Penyusun	1. <i>Dina Anggraini</i>
	2. <i>Benny Irawan</i>
	3. <i>Kuwat Setiyanto</i>
	4. <i>Hela Yudha</i>
	5. <i>Rahma Aditia Donny Irianto</i>

Laboratorium Sistem Informasi

Sistem Informasi

Fakultas Ilmu Komputer

UNIVERSITAS GUNADARMA

Daftar Isi

Daftar Isi	2
Pertemuan 1	5
Konsep Dasar Pascal 7.0	5
P1.1 Teori	6
<i>Sejarah Singkat Bahasa Pascal</i>	6
Program dapat dibagi menjadi beberapa bagian/subprogram, yang terdiri dari satu program utama dan satu / lebih program bagian (bisa berupa prosedur / fungsi). Deklarasi prosedur/ fungsi terletak pada subprogram yang menggunakannya	14
<i>Operator-Operator dalam Pascal</i>	22
<i>Statement Input - Output</i>	23
P1.2 Contoh Kasus	26
P1.3 Latihan	27
P1.4 Daftar Pustaka	27
Pertemuan 2	28
Penyeleksian Kondisi dan Perulangan.....	28
P2.1 Teori	29
Pada umumnya satu permasalahan yang kompleks memerlukan suatu penyelesaian kondisi. Dengan menyeleksi suatu kondisi, maka selanjutnya dapat ditentukan tindakan apa yang harus dilakukan, tergantung pada hasil kondisi penyeleksian. Jadi suatu aksi hanya dikerjakan apabila persyaratan atau kondisi tertentu terpenuhi	29
Dalam Pascal dikenal adanya 2 tipe perulangan yaitu:.....	29
• Iteration statement digunakan untuk melakukan perulangan sekumpulan statement (compound statement).	29
• Loop (iteration) structure menggambarkan perulangan dari satu atau lebih instruksi	29
<i>Statement IF</i>	29
Struktur Penulisan IF-THEN :	30
aksi	30
Struktur penulisan IF-THEN-ELSE :	30
aksi2	30
Struktur penulisan IF-THEN-ELSE tersarang :	31
aksi3	31
<i>Statement Case</i>	31
<i>Kondisi1, kondisi2, ... kondisiN</i> dapat bernilai benar atau salah. Tiap <i>kondisi</i> diperiksa nilai kebenarannya mulai dari <i>kondisi</i> pertama sampai ditemukan <i>kondisi</i> yang benar. Jika <i>kondisi</i>	

ke- <i>k</i> benar, maka <i>aksi ke-k</i> dilaksanakan, selanjutnya keluar dari struktur CASE. <i>Aksi</i> yang dipasangkan dengan <i>kondisi ke-k</i> dapat lebih dari satu, karena itu ia berupa runtunan. Jika tidak ada satupun <i>kondisi</i> yang benar, maka <i>aksi</i> sesudah <i>otherwise</i> (optional) dikerjakan.	32
<i>Perulangan For</i>	32
<i>Perulangan While Do</i>	34
<i>Perulangan Repeat Until</i>	36
P2.2 Contoh Kasus	37
-----	39
P2.3 Latihan	40
P2.4 Daftar Pustaka	41
2. Munir R, Algoritma dan Pemrograman dengan Pascal dan C edisi Kedua, Bandung: Informatika, 2003.	41
3. Jogiyanto H, Turbo Pascal, Yogyakarta Andi, 2006	41
.	41
Pertemuan 3	42
Prosedur dan Fungsi.....	42
P3.1 Teori	43
Prosedur	43
Prosedur adalah suatu program terpisah dalam blok sendiri yang berfungsi sebagai subprogram (program bagian). Procedure diawali dengan kata cadangan <i>Procedure</i> di dalam bagian deklarasi prosedur. Prosedur banyak digunakan pada program yang terstruktur, karena:.....	43
1. Merupakan penerapan konsep program modular, yaitu memecah – memecah program yang rumit menjadi program – program bagian yang lebih sederhana dalam bentuk prosedur – prosedur.	43
2. Untuk hal – hal yang sering dilakukan berulang – ulang, cukup dituliskan sekali saja dalam prosedur dan dapat dipanggil atau dipergunakan sewaktu – waktu bila diperlukan.	43
Struktur prosedur terdiri dari 3 bagian yaitu :	43
end;	43
Function	46
Blok fungsi hamper sama dengan blok prosedur, hanya fungsi harus dideklarasikan dengan tipenya. Tipe deklarasi ini menunjukkan tipe hasil dari fungsi. Tipe tersebut ditulis pada akhir deklarasi fungsi yang didahului dengan titik koma, sebagai berikut :	46
<i>Function identifier(daftar-parameter): type;</i>	46
Contoh :	46
Function factorial (var fak,hasil: integer) : integer;	46

P3.2 Contoh Kasus	47
Penggunaan Prosedur	47
Penggunaan Prosedur dengan parameter masukan / nilai	47
Penggunaan Prosedur dengan parameter acuan	47
Prosedur memanggil prosedur yang lain	47
Prosedur tersarang	49
Prosedur rekursif	49
Function dengan parameter nilai	50
Function dengan parameter acuan	50
Function tanpa parameter	51
Function Rekursif	51
P3.3 Latihan	52
P3.4 Daftar Pustaka	52
2. Munir R, Algoritma dan Pemrograman dengan Pascal dan C edisi Kedua, Bandung: Informatika, 2003.	52
3. Jogiyanto H, Turbo Pascal, Yogyakarta Andi, 2006.	52
Pertemuan 4	53
Array 53	
P4.1 Teori	54
<i>Deklarasi Array</i>	54
<i>Array Multi Dimensi</i>	57
P4.2 Contoh Kasus	58
P4.3 Latihan	60
P4.4 Daftar Pustaka	61

Pertemuan 1

Konsep Dasar Pascal 7.0

Objektif:

1. Memahami Sejarah singkat bahasa pemrograman Pascal 7.0
2. Mengerti dan dapat menggunakan tipe-tipe data yang ada
3. Dapat menggunakan operator-operator dalam Pascal
4. Dapat menggunakan statement input dan output

P1.1 Teori

Sejarah Singkat Bahasa Pascal

Apa itu Turbo Pascal ?

Pascal merupakan suatu bahasa komputer tingkat tinggi yang dibuat sekitar tahun 1970 oleh Niklaus Wirth dan digunakan untuk pendidikan komputer. Nama Pascal diambil dari seorang ahli matematika yang bernama Blaise Pascal yang menemukan mesin hitung pertama. Bahasa Pascal dirancang untuk menyelesaikan masalah dari berbagai kalangan pemakai, mulai dari para mahasiswa, pendidik, dan ilmuwan. Salah satu kompiler pascal yang terkenal dan tercepat adalah Turbo PASCAL yang dibuat oleh perusahaan Borland (<http://www.borland.com>).

Adakah versi Turbo Pascal untuk pemrograman Visual ?

Pada trend pemrograman Visual, perusahaan Borland mengeluarkan Borland Delphi yang menggunakan dasar dialek Pascal pada lingkungan pemrograman Visual.

Apa artinya Pascal adalah bahasa pemrogram terstruktur ?

Pascal adalah suatu bahasa pemrograman terstruktur. Hal tersebut berarti semua program yang anda buat harus terstruktur dan teratur, dalam hal ini ada pemakaian goto dan jump.

Bagaimana Struktur program pascal yang paling sederhana ?

Struktur program pascal yang paling sederhana adalah :

```
uses ...;
var ...;
begin
    .... { Your program is here }
end.
```

Dapatkah anda memberi contoh sebuah program pascal yang sederhana ?

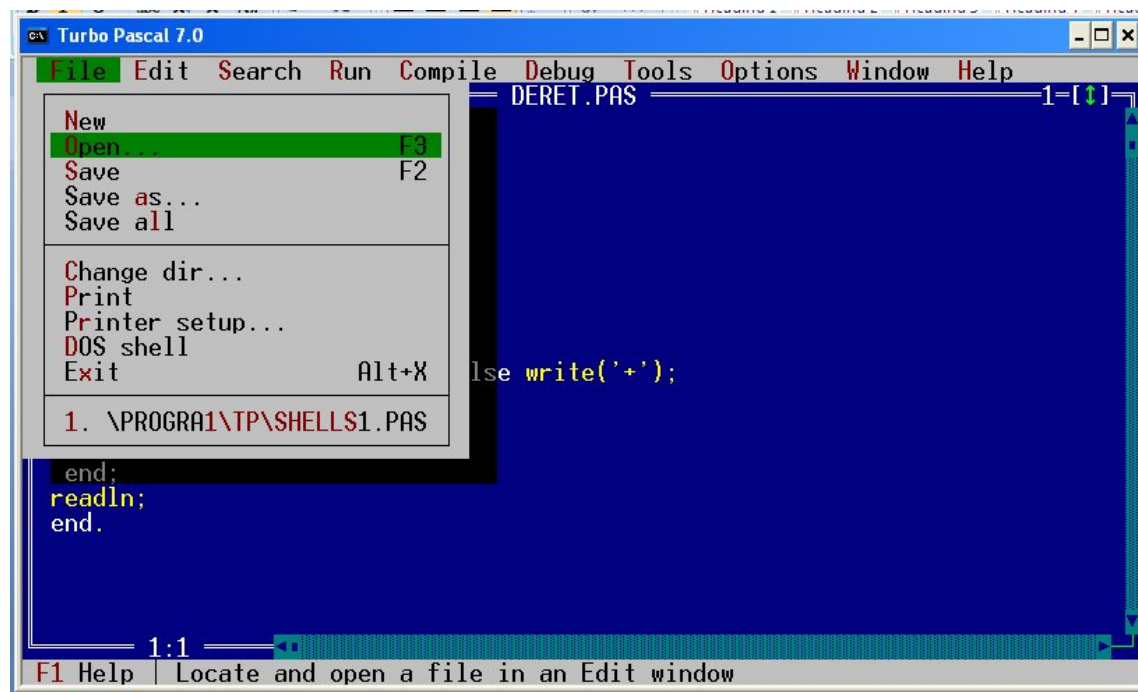
```
uses crt;  
begin  
  Clrscr;  
  Writeln('Saya sedang belajar Pascal !');  
  Writeln('Di PentaCOM');  
end.
```

Penggunaan Turbo Pascal

Dari modus prompt DOS, hal yang dilakukan sbb:

A:\> turbo

Menu dalam Pascal :



a. File (Alt-F)

- | | |
|-------------------|---|
| <i>New</i> | : untuk membuat program baru. |
| <i>Open</i> | : untuk mengambil program yang sudah ada di disk kerja. |
| <i>Save</i> | : untuk menyimpan program. |
| <i>Save as</i> | : untuk menyimpan file dengan nama yg beda |
| <i>Save all</i> | : untuk menyimpan semua program yg di edit |
| <i>Change dir</i> | : untuk mengganti directory yang aktif. |
| <i>Print</i> | : untuk mencetak program yg sedang dibuka |

Printer setup : untuk mensetting printer
DOS Shell : untuk menjalankan perintah-perintah DOS
Exit : mengakhiri turbo Pascal dan kembali keprompt DOS

b. Edit (Alt-E)

Digunakan untuk keperluan memperbaiki program.

c. Search (Alt-S)

Digunakan untuk mencari kata atau baris program

c. Run (Alt-R)

Digunakan untuk menjalankan program yang ada di jendela edit.

d. Compile (Alt-C)

Digunakan untuk mengkompilasi program.

Destination Memory (disimpan di memory).

Destination Disk (disimpan di disk dengan ext .EXE).

e. Debug dan Break/Watch (Alt-D & Alt-B)

Digunakan untuk melacak program.mengaktifkan Debug & Break/Watch.

f. Tools (Alt-T)

Digunakan untuk menampilkan pesan/ *output* dari program

g. Options (Alt-O)

Digunakan untuk mengatur/menentukan kembali bagaimana F1-help, F2-Save F3-new file, F4-import data, F9-expand, F10-contract dan Esc-exit integrated environment bekerja.

h. Window (Alt-W)

Untuk mengatur tampilan dari program-program yang dipanggil, bentuk *Tile* atau *cascade*.

****Cat :** tekan *Esc* untuk meninggalkan menu.

Struktur Program Pascal

Secara ringkas, struktur suatu program Pascal dapat terdiri dari :

1. Judul Program
2. Tubuh Program

Tubuh program dibagi menjadi dua bagian utama :

- a. Bagian deklarasi
 - deklarasi label
 - deklarasi konstanta
 - deklarasi tipe
 - deklarasi variabel/perubah
 - deklarasi prosedur
 - deklarasi fungsi
- b. Bagian Pernyataan/Terproses

Cat : baris-baris komentar untuk memperjelas program diletakkan diantara tanda (dan *) atau { dan } .*

1. Judul program

Judul program ini digunakan untuk memberi nama program dan sifatnya optional. Jika ditulis harus terletak pada awal dari program dan diakhiri dengan titik koma (;).

Contoh penulisan judul program :

```
PROGRAM latihan;  
PROGRAM latihan(input,output);  
PROGRAM lat_1;  
PROGRAM lat_satu(output);
```

2. Bagian Pernyataan/Terproses

Bagian ini adalah bagian yang akan terproses dan terdapat dalam suatu blok yang diawali dengan BEGIN dan diakhiri dengan END (penulisan END diikuti dengan tanda titik).

Bagian ini berisi pernyataan / statemen yang merupakan instruksi program. Setiap statemen diakhiri dengan tanda titik koma (;).

Bentuk umumnya adalah sbb :

```
BEGIN  
  
    ...  
    statemen;  
    statemen;  
    ...  
END.
```

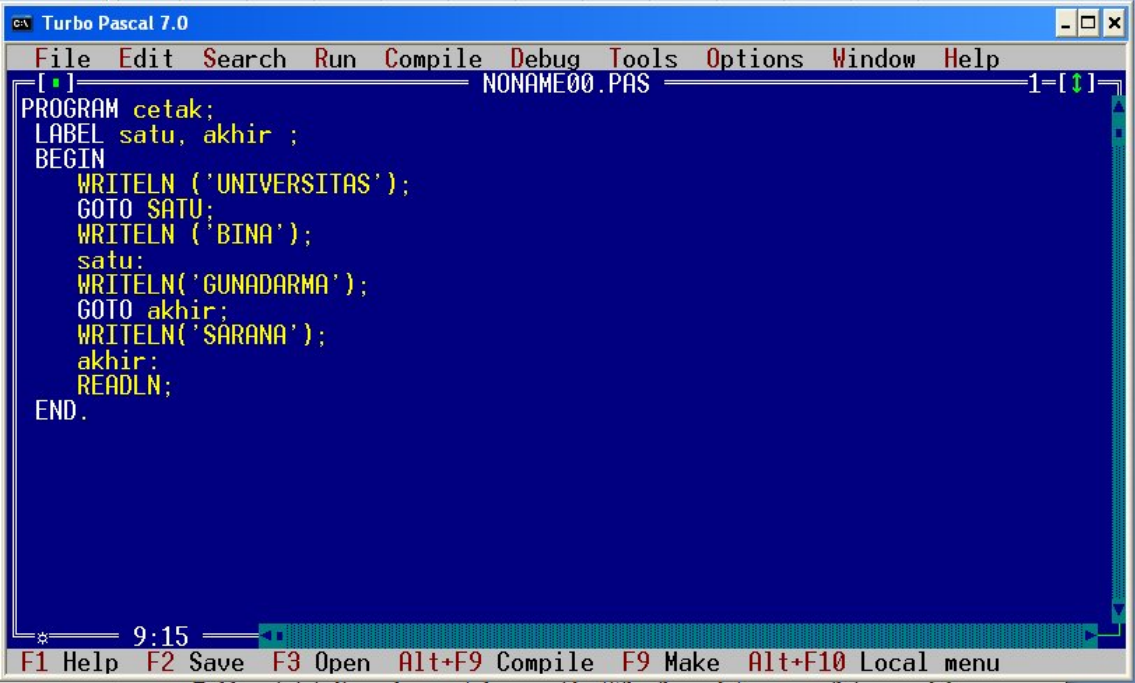
3. Bagian deklarasi

Bagian ini menjelaskan / memperkenalkan secara rinci semua data yang akan digunakan pada suatu program. Dalam penulisannya tidak boleh sama dengan kata-kata cadangan (reserved words) dan selalu diakhiri dengan titik koma (;).

Deklarasi label

Deklarasi label digunakan jika pada penulisan program akan menggunakan statemen GOTO (untuk meloncat ke suatu statement tertentu).

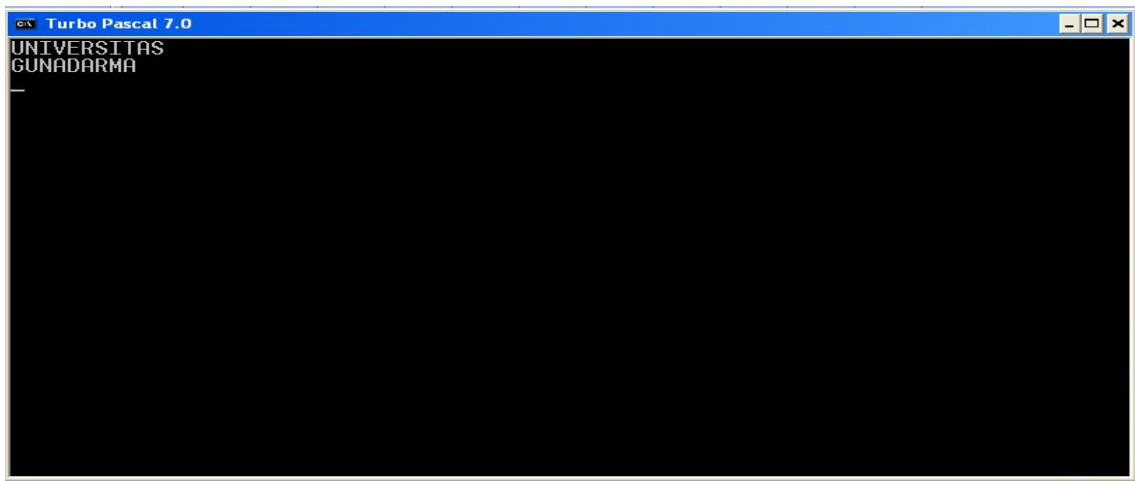
Contoh :



```
Turbo Pascal 7.0  
File Edit Search Run Compile Debug Tools Options Window Help  
[.] NONAME00.PAS 1-[1]  
PROGRAM cetak;  
LABEL satu, akhir ;  
BEGIN  
    WRITELN ('UNIVERSITAS');  
    GOTO SATU;  
    WRITELN ('BINA');  
    satu:  
    WRITELN('GUNADARMA');  
    GOTO akhir;  
    WRITELN('SARANA');  
    akhir:  
    READLN;  
END.  
9:15  
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu
```

Catt : Statement *Readln* digunakan untuk menghentikan tampilan. Karena jika tidak menggunakan *Readln*, ketika dijalankan, akan kembali ke Editor program

Bila program di atas dijalankan, output sbb :



Deklarasi konstanta

Deklarasi ini digunakan untuk mengidentifikasi data yang nilainya sudah ditentukan dan pasti, tidak dapat dirubah dalam program.

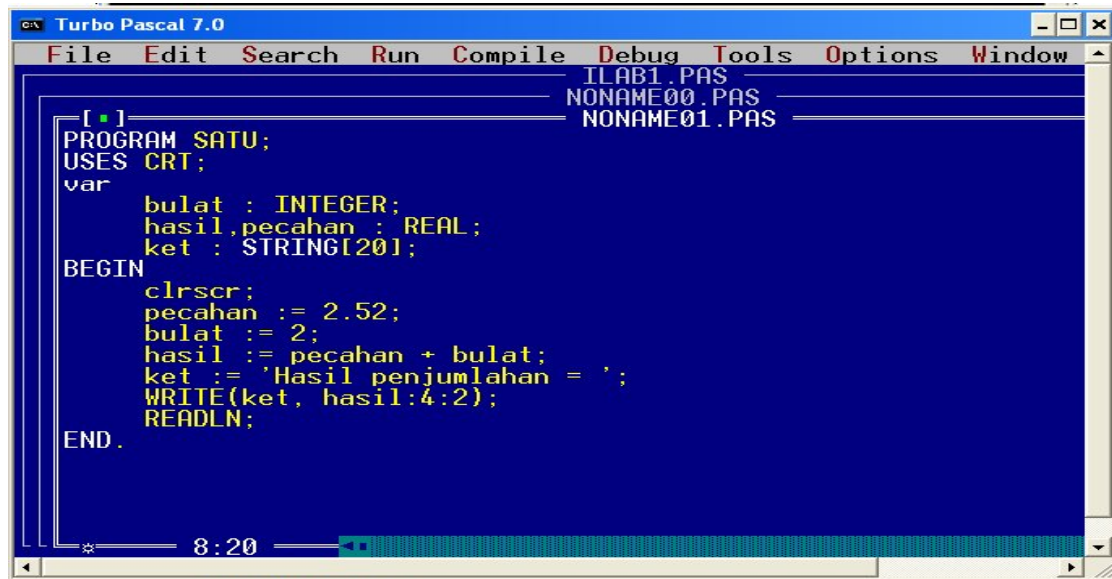
Contoh :

```
Const phi = 3.14;
```

Deklarasi tipe

Deklarasi ini digunakan untuk menyebutkan tipe setiap data yang akan digunakan pada program Pascal. Tipe data menentukan jangkauan nilai yang mungkin dari data yang digunakan

Contoh :

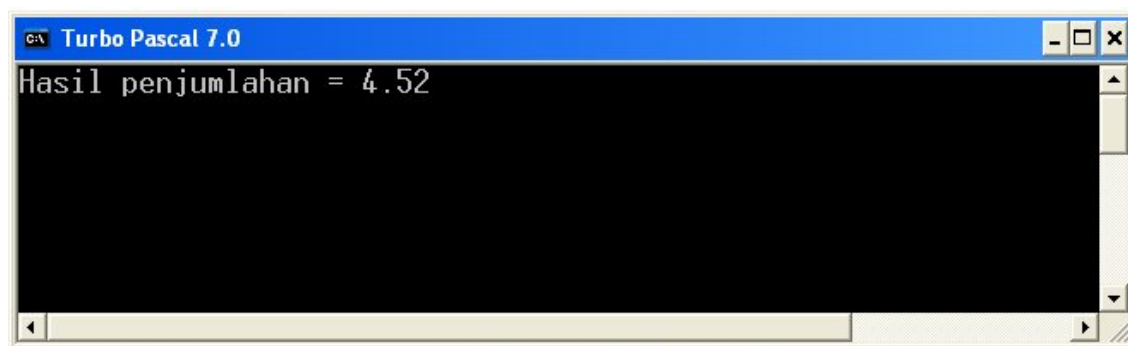


```

Turbo Pascal 7.0
File Edit Search Run Compile Debug Tools Options Window
[.] ILAB1.PAS
  NONAME00.PAS
  NONAME01.PAS
PROGRAM SATU;
USES CRT;
var
  bulat : INTEGER;
  hasil, pecahan : REAL;
  ket : STRING[20];
BEGIN
  clrscr;
  pecahan := 2.52;
  bulat := 2;
  hasil := pecahan + bulat;
  ket := 'Hasil penjumlahan = ';
  WRITE(ket, hasil:4:2);
  READLN;
END.
8:20

```

Output program, sbb : hasil penjumlahan = 4.52



```

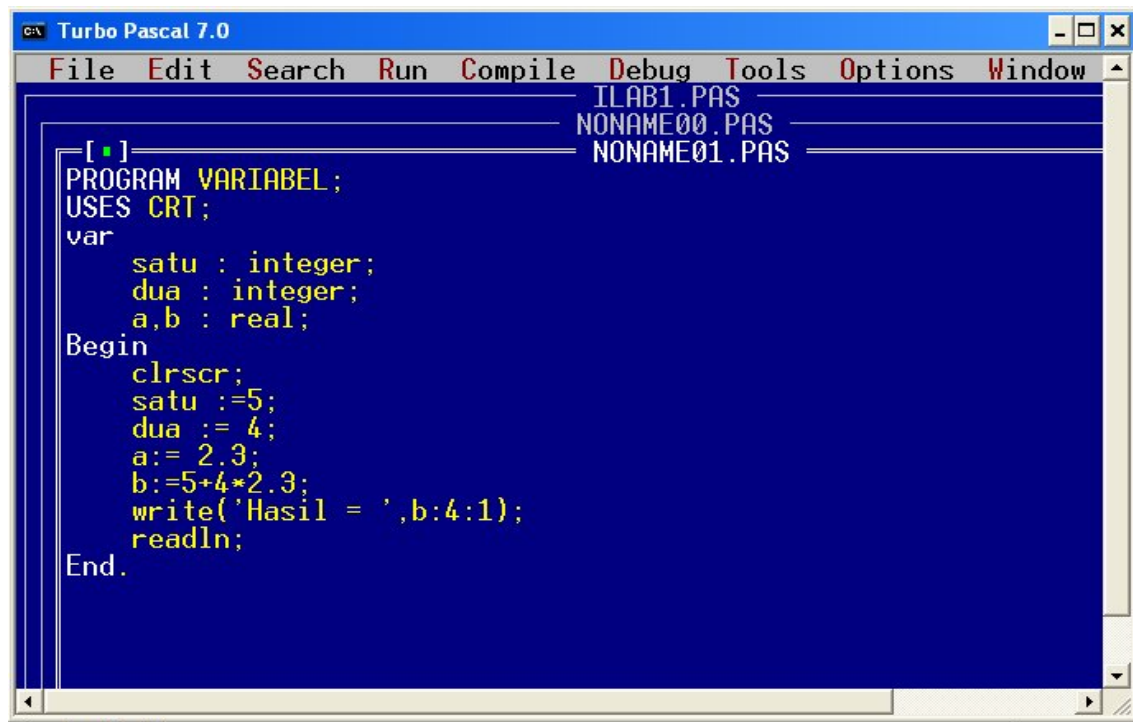
Turbo Pascal 7.0
Hasil penjumlahan = 4.52

```

Deklarasi variabel/perubah

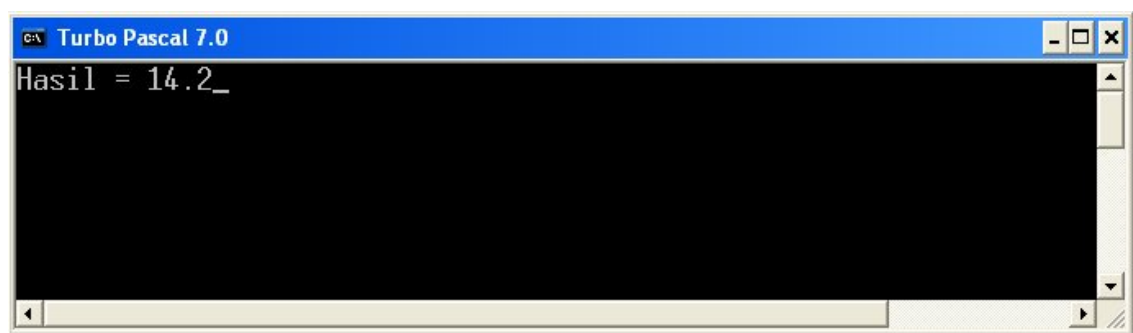
Deklarasi ini berisi data-data yang bisa berubah-ubah nilainya di dalam program. Deklarasi variabel harus di letakkan setelah deklarasi tipe (jika ada).

Contoh :



```
PROGRAM VARIABEL;  
USES CRT;  
var  
    satu : integer;  
    dua : integer;  
    a,b : real;  
Begin  
    clrscr;  
    satu :=5;  
    dua := 4;  
    a:= 2.3;  
    b:=5+4*2.3;  
    write('Hasil = ',b:4:1);  
    readln;  
End.
```

Output program : hasil = 14.2



```
Hasil = 14.2_
```

Program diatas bisa ditulis sbb :

```
VAR satu,dua : INTEGER;  
    a,b      : REAL;  
BEGIN  
    ...  
    statement;  
    ...  
END.
```

Contoh jika terdapat deklarasi tipe :

```
TYPE
    bilangan = integer;
VAR
    satu,dua,a    : bilangan;
    b             : real;
BEGIN
    ...
    statement;
    ...
END.
```

Deklarasi prosedur dan Fungsi

Program dapat dibagi menjadi beberapa bagian/subprogram, yang terdiri dari satu program utama dan satu / lebih program bagian (bisa berupa prosedur / fungsi). Deklarasi prosedur/ fungsi terletak pada subprogram yang menggunakannya

Tipe-tipe Data dalam Pascal

Tipe Data dapat terletak pada deklarasi variabel maupun pada deklarasi tipe. Pascal menyediakan beberapa macam tipe data, yang terdiri dari :

- Tipe data sederhana/skalar, terdiri dari :
 - Tipe data standar/predefinisi
 - bulat (integer)
 - real
 - karakter
 - string
 - logika (boolean)
 - Tipe data didefinisikan pemakai :
 - subjangkauan (subrange)
 - terbilang (enumerated)
 - Tipe data terstruktur, terdiri dari :
 - larik (array)
 - rekaman (record)
 - berkas (file)

- himpunan (set)
- Tipe data penunjuk (pointer)

Tipe data standar

Tipe data integer

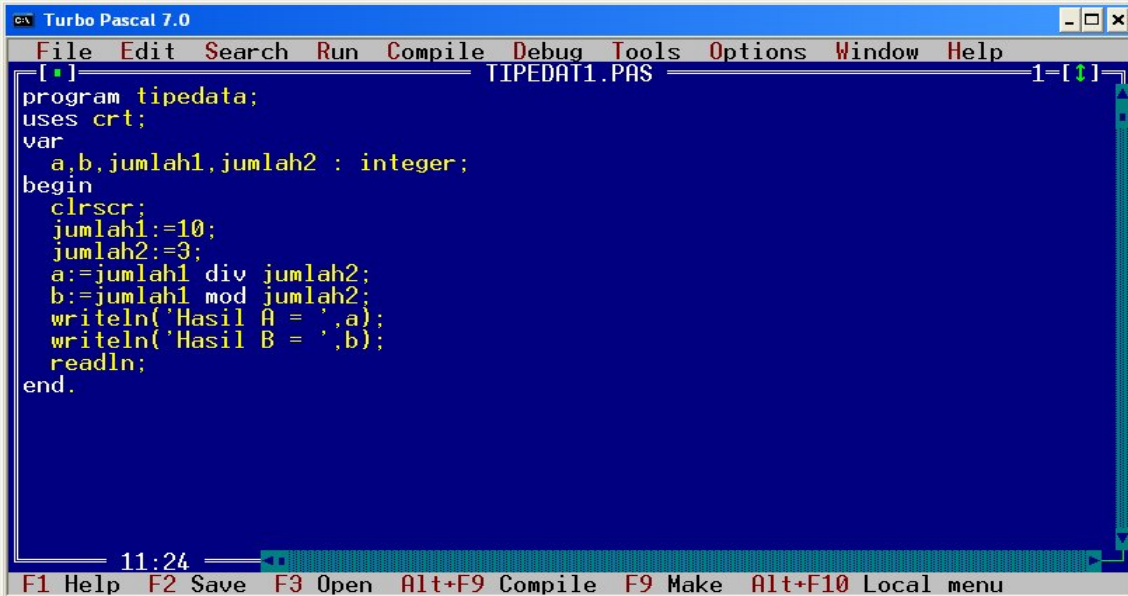
Tipe integer adalah bilangan yang tidak mempunyai titik desimal/bilangan pecahan. Integer terdiri dari beberapa tipe, yaitu :

- byte, dengan jangkauan nilai 0..255
- shortint, dengan jangkauan nilai -128..127
- integer, dengan jangkauan nilai -32768..32767
- word, dengan jangkauan nilai 0..65535
- longint, dengan jangkauan nilai -2147483648..2147483647

Operator yang dapat digunakan pada data tipe integer :

- +, penjumlahan
- , pengurangan
- *, perkalian
- div, pembagian
- mod, sisa pembagian

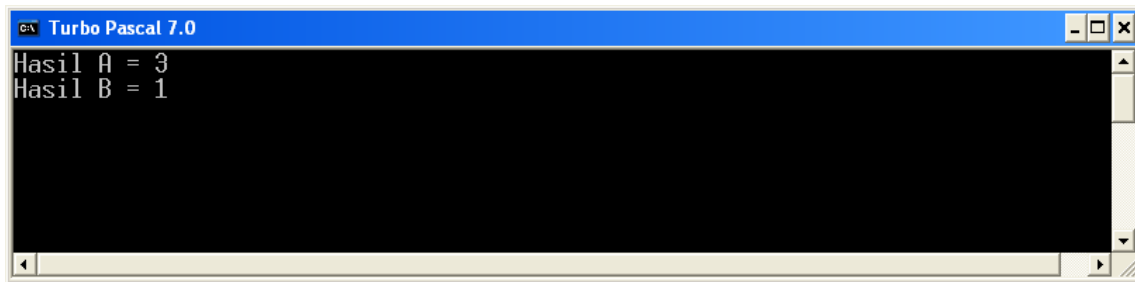
Contoh :



```

program tipedata;
uses crt;
var
  a,b,jumlah1,jumlah2 : integer;
begin
  clrscr;
  jumlah1:=10;
  jumlah2:=3;
  a:=jumlah1 div jumlah2;
  b:=jumlah1 mod jumlah2;
  writeln('Hasil A = ',a);
  writeln('Hasil B = ',b);
  readln;
end.
  
```

hasil program :



A screenshot of the Turbo Pascal 7.0 IDE. The window title is "Turbo Pascal 7.0". The main text area shows the output of a program: "Hasil A = 3" and "Hasil B = 1". The background is black, and the text is white.

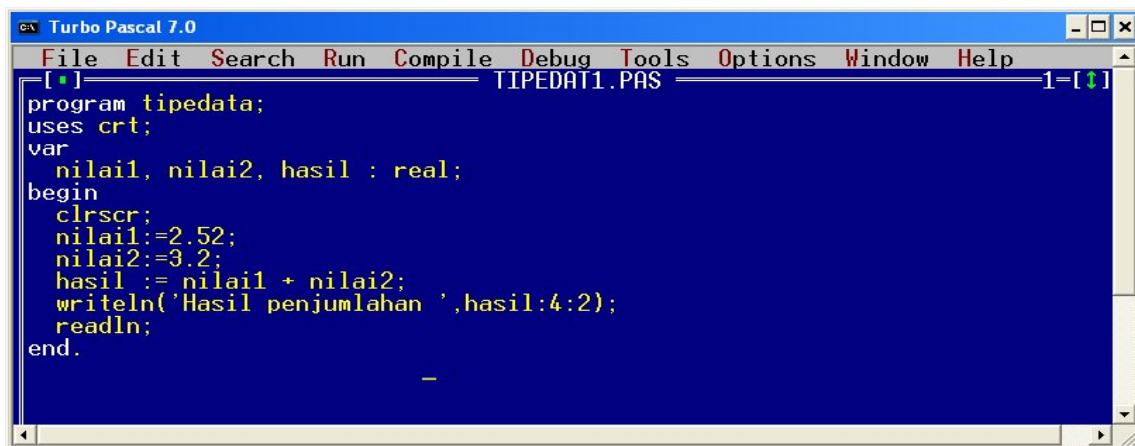
Tipe data real

Tipe real adalah bilangan yang mengandung pecahan, palingsedikit harus ada satu digit sebelum dan sesudah titik desimal.

Operator yang dapat digunakan pada data tipe real adalah :

- + penjumlahan
- pengurangan
- * perkalian
- / pembagian

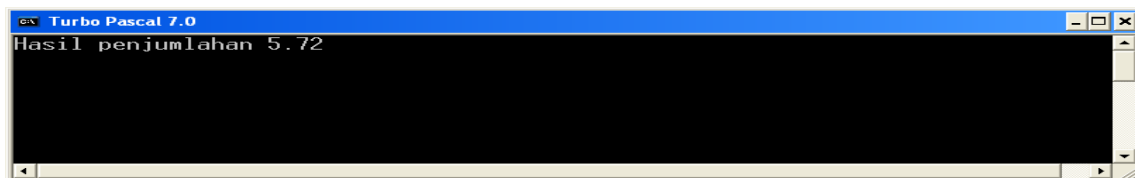
Contoh :



A screenshot of the Turbo Pascal 7.0 IDE. The window title is "Turbo Pascal 7.0". The menu bar includes File, Edit, Search, Run, Compile, Debug, Tools, Options, Window, and Help. The main text area shows the source code for a program named "TIPEDAT1.PAS". The code is as follows:

```
program typedata;  
uses crt;  
var  
  nilai1, nilai2, hasil : real;  
begin  
  clrscr;  
  nilai1:=2.52;  
  nilai2:=3.2;  
  hasil := nilai1 + nilai2;  
  writeln('Hasil penjumlahan ',hasil:4:2);  
  readln;  
end.
```

Output program, sbb :



A screenshot of the Turbo Pascal 7.0 IDE. The window title is "Turbo Pascal 7.0". The main text area shows the output of the program: "Hasil penjumlahan 5.72". The background is black, and the text is white.

Tipe data karakter

Nilai data karakter berupa sebuah karakter yang ditulis diantara tanda petik tunggal, misalnya : 'A', 'b', '@', dan sebagainya. Karakter yang dapat diterima oleh komputer :

huruf besar/kecil : A,B,C,...,Z / a,b,...,z

digit : 1,2,3,...,9

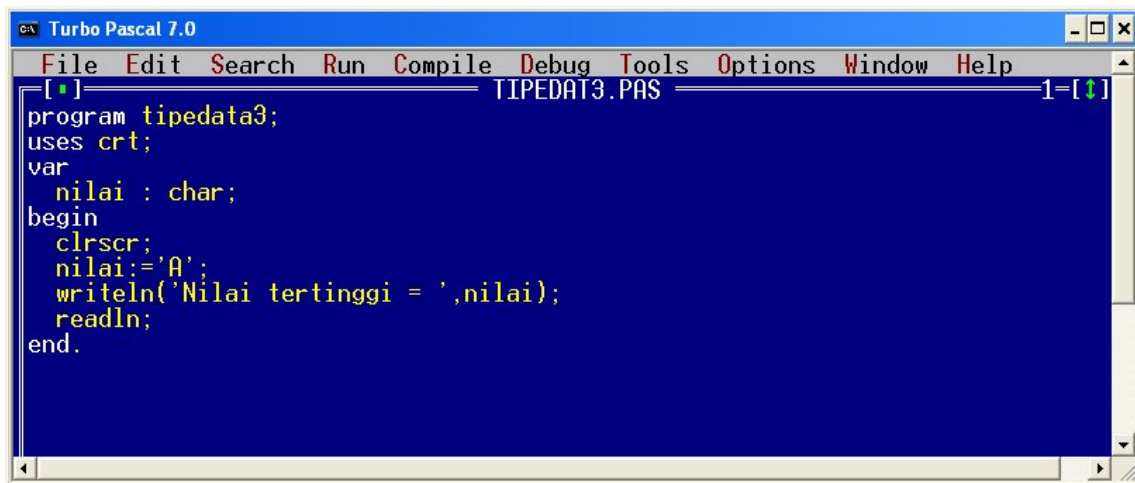
operator aritmatika : * / + -

tanda baca : , . ; : ? !

simbol khusus : \$ @ { } () [] % #

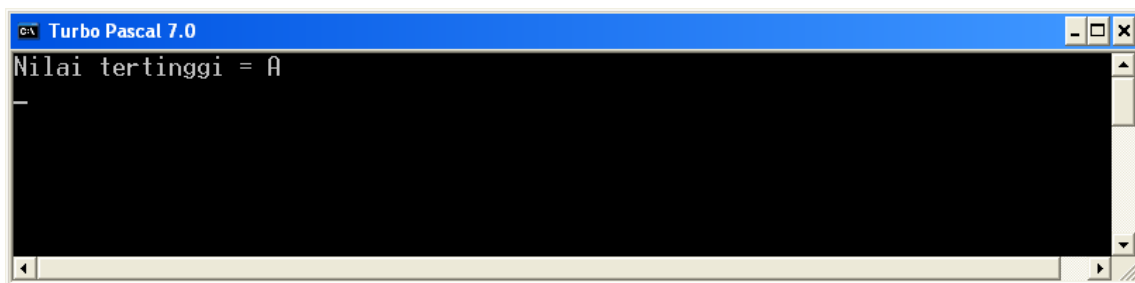
spasi

Contoh :



```
File Edit Search Run Compile Debug Tools Options Window Help
[.] TИPEDAT3.PAS 1=[↑]
program typedata3;
uses crt;
var
  nilai : char;
begin
  clrscr;
  nilai:='A';
  writeln('Nilai tertinggi = ',nilai);
  readln;
end.
```

hasilnya :

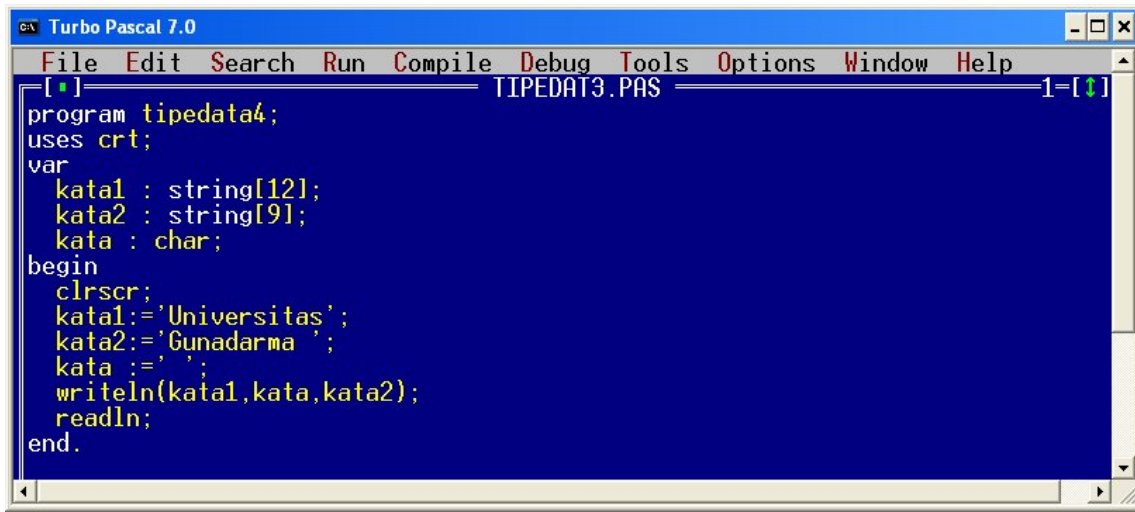


```
Turbo Pascal 7.0
Nilai tertinggi = A
```

Tipe data string

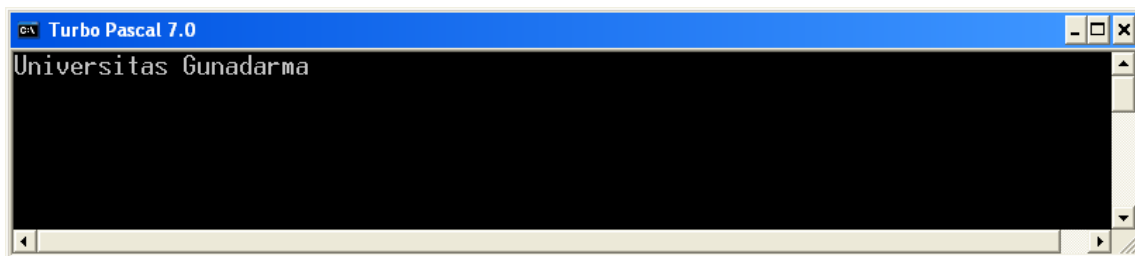
Nilai data string adalah satu atau lebih karakter yang terletak diantara tanda petik tunggal, misal : 'GUNADARMA'. Bila panjang dari suatu string di dalam deklarasi variabel tidak disebutkan, maka dianggap panjangnya 255 karakter.

Contoh :



```
program tippedata4;
uses crt;
var
  kata1 : string[12];
  kata2 : string[9];
  kata : char;
begin
  clrscr;
  kata1:='Universitas';
  kata2:='Gunadarma';
  kata := ' ';
  writeln(kata1,kata,kata2);
  readln;
end.
```

hasil :

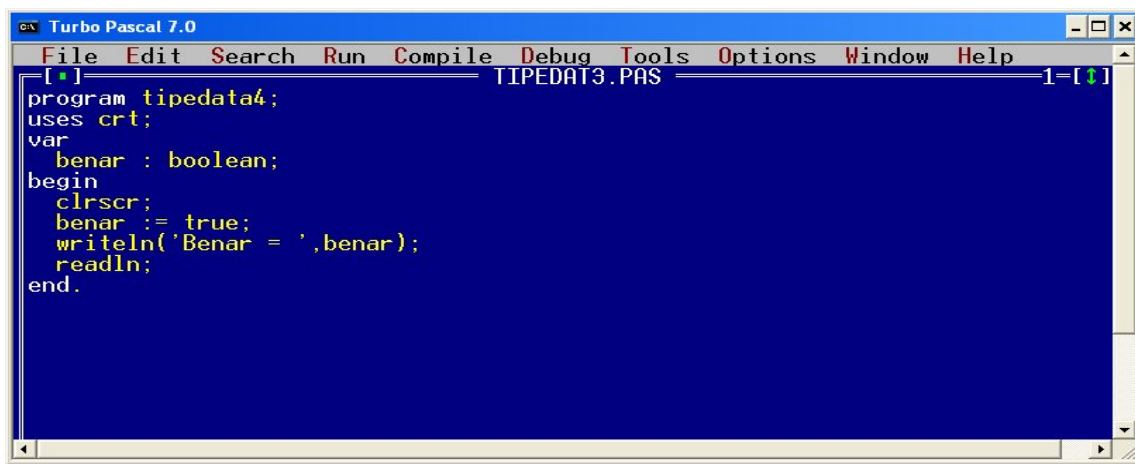


```
Universitas Gunadarma
```

Tipe data boolean

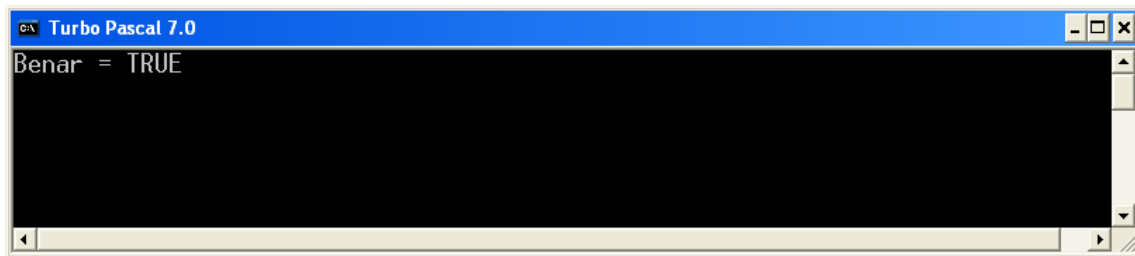
Data tipe boolean mempunyai dua nilai, yaitu True dan False.

Contoh :



```
program tippedata4;
uses crt;
var
  benar : boolean;
begin
  clrscr;
  benar := true;
  writeln('Benar = ',benar);
  readln;
end.
```

hasil :



Tipe data Terdefinisi

Tipe data subjangkauan

Tipe data ini adalah tipe data yang dapat didefinisikan sendiri oleh pemakai. Nilai data pada tipe ini mempunyai jangkauan tertentu.

Misalkan nilai ujian mempunyai harga 0 sampai 100, maka nilai ujian dapat didefinisikan sbb :

```
TYPE
    nilai = 0..100;
```

Contoh :

```
VAR sks      : 1..4;
    angkatan : 89..95;
    nilai    : 'A'..'E';
```

Tipe data terbilang

Tipe data ini juga dapat didefinisikan sendiri oleh pemakai. Disebut tipe terbilang karena semua nilai disebut satu persatu.

Contoh :

```
TYPE hari      = (Senin,Selasa,Rabu,Kamis,Jum'at,Sabtu,Minggu);
    hari_kerja = (Senin,Selasa,Rabu,Kamis,Jum'at);
    situasi    = (senang,gembira,sedih,susah);
```

Tipe data larik (array)

Larik (array) adalah kumpulan data yang mempunyai tipe data sejenis. Daftar nomor telpon, daftar kode mata kuliah, vektor, matrik merupakan contoh larik.

Contoh penulisan tipe larik berdimensi satu sbb :

```
CONST batas = 20;
```

```
VAR telpon : ARRAY[1..3] OF STRING[7];
```

```
{ larik dengan nama telpon mempunyai 3 data dengan tipe string }
```

```
nilai : ARRAY[1..5] OF INTEGER;
```

```
{ larik dengan nama nilai mempunyai 5 data dengan tipe integer }
```

```
gaji : ARRAY[1..batas] OF REAL;
```

```
{ larik dengan namagaji mempunyai 20 data dengan tipe real }
```

Contoh larik yang mempunyai tipe data terbilang atau subjangkauan :

```
TYPE batas = 0..100;
```

```
keadaan = (baru,lama,bagus,jelek);
```

```
VAR nilai : ARRAY[1..30] OF 'A'..'B';
```

```
{ larik dengan nama nilai mempunyai 30 data, dan pengisian data  
yang diperbolehkan hanya A, B, C, D, E }
```

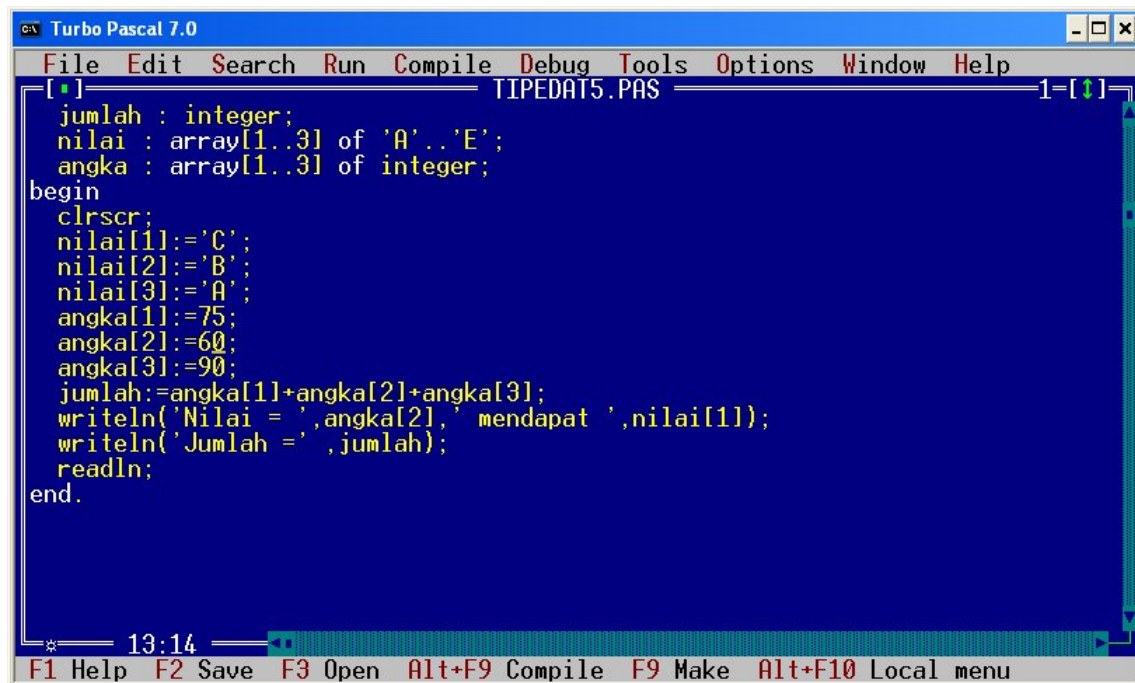
```
angka : ARRAY[1..50] OF batas;
```

```
{ larik dengan nama angka mempunyai 50 data, dan pengisian  
data yang diperbolehkan hanya 1,2,3,...,99,100 }
```

```
baju : ARRAY[1..10] OF keadaan;
```

```
{ larik dengan nama angka mempunyai 10 data,dan pengisian data  
yang diperbolehkan baru, lama, bagus, jelek }
```

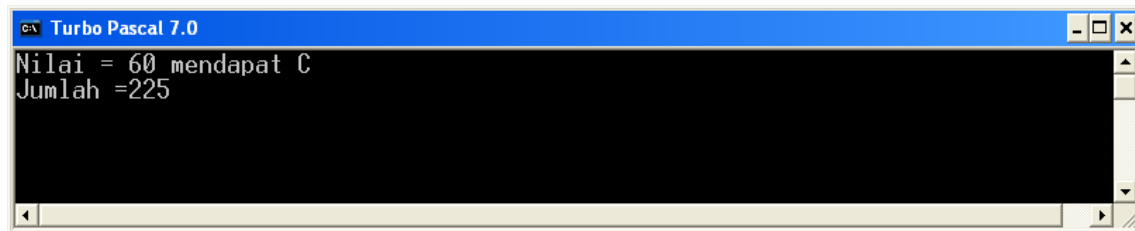
Contoh program :



```
File Edit Search Run Compile Debug Tools Options Window Help
[.] TIPE DAT5.PAS 1-[↑]
jumlah : integer;
nilai : array[1..3] of 'A'..'E';
angka : array[1..3] of integer;
begin
  clrscr;
  nilai[1]:='C';
  nilai[2]:='B';
  nilai[3]:='A';
  angka[1]:=75;
  angka[2]:=60;
  angka[3]:=90;
  jumlah:=angka[1]+angka[2]+angka[3];
  writeln('Nilai = ',angka[2],' mendapat ',nilai[1]);
  writeln('Jumlah = ',jumlah);
  readln;
end.
```

13:14
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu

hasil :



```
Turbo Pascal 7.0
Nilai = 60 mendapat C
Jumlah =225
```

Contoh penulisan tipe larik berdimensi dua sbb :

VAR tabel : ARRAY[1..3,1..2] OF BYTE; { larik tabel mempunyai 3 baris dan 2 kolom dengan tipe byte }

BEGIN tabel[1,1] := 5; { baris 1, kolom 1 }

tabel[1,2] := 7;

tabel[2,1] := 21; { baris 2, kolom 1 }

tabel[2,2] := 18;

tabel[3,1] := 8;

tabel[3,2] := 7;

WRITELN('BARIS 1 KOLOM 2 = ',tabel[1,2]);

END.

hasil : BARIS 1 KOLOM 2 = 7