

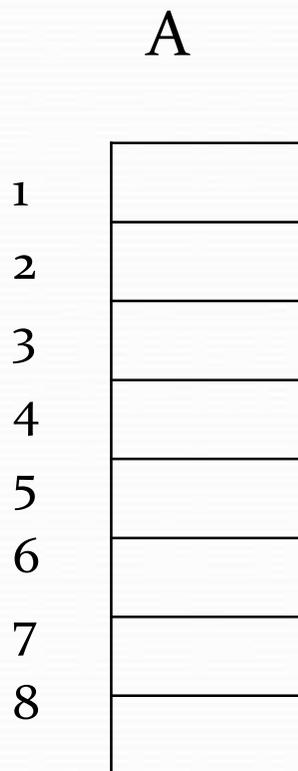
ARRAY / LARIK

Temu 14

Definisi Array / Larik

- Struktur data yang menyimpan sekumpulan elemen yang bertipe sama
- Setiap elemen diakses langsung melalui indeksnya
- Indeks larik harus tipe data yang sama yang menyatakan keterurutan, misalnya integer atau karakter

Cara Pendefinisian



- Sebuah larik yang bernama A dengan 8 buah elemen
- Sekumpulan kotak yang terurut (vertikal atau horizontal)
- Tiap kotak pada larik tersebut diberi indeks 1,2,3,...,8
- Setiap larik ditulis dengan notasi: $A[1], A[2], A[3], A[4], A[5], A[6], A[7], A[8]$
- Angka di dalam tanda kurung siku menyatakan indeks larik (notasi diatas sama dgn penulisan peubah ber-subscript $A_1, A_2, A_3, \dots, A_8$)

Mendeklarasikan Larik

- Larik adalah struktur data statis
- Jumlah elemen larik harus sudah diketahui sebelum program dieksekusi
- Jumlah elemen larik tidak dapat diubah, ditambah atau dikurangi selama pelaksanaan program
- Mendeklarasikan larik di dalam bagian deklarasi berarti:
 1. Mendefinisikan banyaknya elemen larik (ukuran larik), dan
 2. Mendefinisikan tipe elemen larik

Mendeklarasikan Larik di dalam bagian deklarasi

a. Sebagai Peubah, misalnya

- A adalah larik yang berukuran 50 buah elemen yang bertipe integer. Indeks larik dimulai dari 1

A : Array [1..50] of integer

- NamaMhs adalah larik yang berukuran 10 buah elemen yang bertipe string. Indeks larik dimulai dari 1

NamaMhs : Array [1..10] of string

- NilUjian adalah peubah larik yang berukuran 75 buah elemen yang bertipe real. Indeks larik dimulai dari 0

NilUjian : Array [0..74] of real

Mendeklarasikan Larik di dalam bagian deklarasi

b. Sebagai Tipe Bentukan, misalnya

- LarikInt didefinisikan sebagai nama peubah tipe baru untuk larik yang bertipe integer.
- Ukuran larik adalah 100 buah elemen

Deklarasi

type LarikInt : Array [1..100] of integer {tipe nama baru}

A : LarikInt {A adalah sebuah peubah larik integer dengan 100 elemen}

Mendeklarasikan Larik di dalam bagian deklarasi

- c. Mendefinisikan ukuran larik sebagai sebuah konstanta, misalnya
- LarikInt didefinisikan sebagai nama sebuah tipe bentukan untuk larik yang bertipe integer.
 - Ukuran maksimum larik adalah 100 buah elemen.
 - Ukuran maksimum larik dinyatakan sebagai konstanta, mendeklarasikan sebuah peubah yang bertipe larikInt, misalnya peubah A

Deklarasi

Const Nmaks = 100 {ukuran maksimum elemen larik}

type LarikInt : Array [1..Nmaks] of integer

A : LarikInt {A adalah sebuah peubah larik integer dengan 100 elemen}

Cara Mengacu Elemen Larik

- Elemen larik diacu melalui indeksnya
- Nilai indeks harus terdefinisi
- Contoh:
 - $A[4]$ {mengacu elemen keempat dari larik A}
 - $NamaMhs[2]$ {mengacu elemen kedua dari larik NamaMhs}
 - $A[i]$ {mengacu elemen ke-I dari larik A, asalkan nilai i sudah terdefinisi}
 - $NamaMhs[i+1]$ {asalkan nilai I sudah terdefinisi}

Contoh memanipulasi atau menggunakan elemen larik

- $A[4] \leftarrow 10$ {mengisi elemen keempat dari larik A dengan nilai 10}
- $\text{NamaMhs}[i] \leftarrow \text{'Achmad'}$ {mengisi elemen keempat dari larik A dengan nilai Achmad}
- $\text{Read}(A[i])$ {membaca elemen ke-I dari larik A}

if $A[i] < 10$ then

$A[i] \leftarrow A[i] + 10$

else

... {pernyataan lainnya}

Pemrosesan Larik

- Elemen larik tersusun di memori secara beruntun (sekuensial)
- Elemennya diproses secara beruntun melalui indeksnya yang terurut
- Memproses larik artinya mengunjungi (tranversal) setiap elemen larik dan memanipulasi dinilai didalamnya
- Kunjungan dimulai dari elemen pertama larik, berturut-turut pada elemen berikutnya sampai elemen terakhir dicapai, yaitu elemen dengan indeks terbesar

Contoh Algoritma Pemrosesan Larik

PROGRAM PemrosesanLarik

Deklarasi

const Nmaks = 100 {ukuran maksimum larik}

type LarikInt : array [1..Nmaks] of integer

A : LarikInt

i : integer {indeks larik}

Algoritma

inisialisasi

$i \leftarrow 1$ {mulai dari elemen pertama}

while $i \leq Nmaks$ do

 pemrosesan terhadap $A[i]$

$i \leftarrow i + 1$ {tinjau elemen berikutnya}

endwhile

{ $I > Nmaks$ }

Contoh 1# ArrayKonstan

Program ArrayKonstan

Deklarasi

Const

Hari : Array [1..7] of string = ('Senin','Selasa',...,
'Minggu')

NoHari : integer

Algoritma

Read(NoHari)

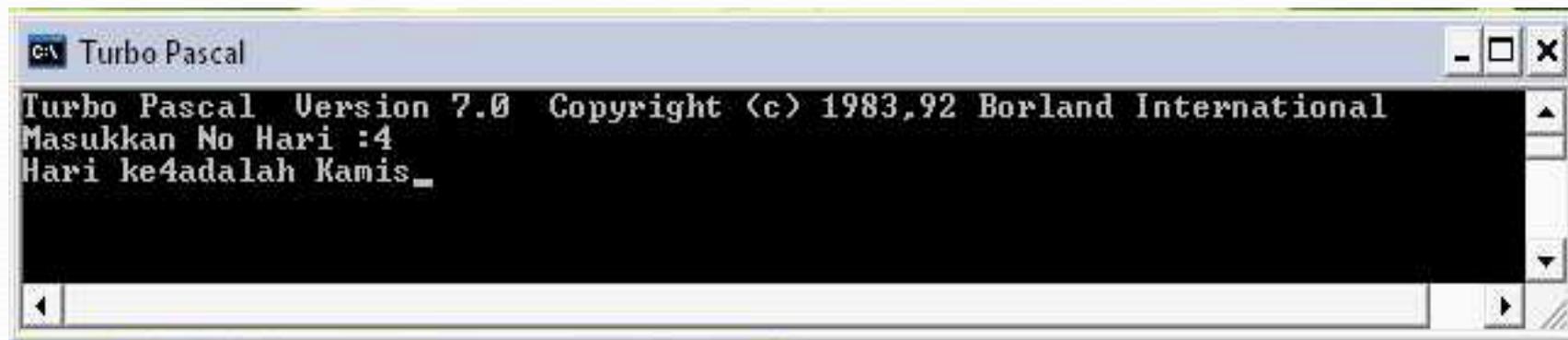
Write(Hari[NoHari])



The screenshot shows the Turbo Pascal IDE window titled "c:\ Turbo Pascal". The menu bar includes File, Edit, Search, Run, Compile, Debug, Tools, Options, Window, and Help. The editor displays the following Pascal code in a blue background:

```
NONAME00.PAS
Program ArrayKonstan;
Const
  Hari : Array [1..7] of string = ('Senin', 'Selasa', 'Rabu', 'Kamis',
  'Jumat', 'Sabtu', 'Minggu');
Var
  NoHari : integer;
Begin
  Write('Masukkan No Hari :');
  Readln(NoHari);
  Write('Hari ke',NoHari,'adalah ',Hari[NoHari]);
  readln;
End.
```

The status bar at the bottom shows the cursor at line 4, column 6, and lists keyboard shortcuts: F1 Help, F2 Save, F3 Open, Alt+F9 Compile, F9 Make, and Alt+F10 Local menu.



The screenshot shows the Turbo Pascal IDE window titled "c:\ Turbo Pascal" displaying the program's output. The text reads:

```
Turbo Pascal Version 7.0 Copyright (c) 1983,92 Borland International
Masukkan No Hari :4
Hari ke4adalah Kamis_
```

The output shows the program successfully executed, displaying the day name for the input number 4.

Contoh 2# Menghitung Rata-rata dalam Array

Program HitungRerataArray

Deklarasi

A : Array [1..10] of integer ;

i : integer;

jumlah : longint;

rerata : real;

Algoritma

Jumlah = 0

For i ← 1 to 10 do

 Read(A[i])

 Jumlah ← jumlah + A[i]

End For

rerata ← jumlah / i

write(jumlah)

write(rerata)

```
c:\ Turbo Pascal
File Edit Search Run Compile Debug Tools Options Window Help
ARRAYRER.PAS 1=[+]
Program ArrayRerata;
Var
  i : integer;
  jumlah : longint;
  rerata : real;
  A : Array[1..10] of integer;
Begin
  Jumlah := 0;
  Writeln('Menghitung Rata-rata 10 Buah Data');
  For i := 1 to 10 do
  Begin
    Write('Isikan Data ke ', i, '= ');
    Readln(A[i]);
    jumlah := jumlah + A[i];
  End;
  rerata := jumlah/i;
  Writeln('Jumlah = ', jumlah);
  Writeln('Rata-rata = ', rerata);
  Readln;
End.
12:39
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu
```

```
c:\ Turbo Pascal
Turbo Pascal Version 7.0 Copyright (c) 1983,92 Borland International
Menghitung Rata-rata 10 Buah Data
Isikan Data ke 1= 60
Isikan Data ke 2= 65
Isikan Data ke 3= 45
Isikan Data ke 4= 40
Isikan Data ke 5= 75
Isikan Data ke 6= 75
Isikan Data ke 7= 85
Isikan Data ke 8= 70
Isikan Data ke 9= 95
Isikan Data ke 10= 90
Jumlah = 700
Rata-rata = 7.0000000000E+01
```