**Array**

Array adalah tipe data terstruktur yang terdiri dari sejumlah komponen komponen yang mempunyai tipe sama. Komponen tersebut disebut sebagai komponen type, larik mempunyai jumlah komponen yang jumlahnya tetap. Banyaknya komponen dalam larik ditunjukkan oleh suatu index, dimana tiap komponen di array dapat diakses dengan menunjukkan nilai indexnya atau subskript.

Array dapat bertipe data sederhana seperti byte, word, integer, real, bolean, char, string dan tipe data scalar atau subrange. Tipe larik mengartikan isi dari larik atau komponen- komponenya mempunyai nilai dengan tipe data tersebut.

Contoh :

var

untai : array[1..50] of integer;

Pada contoh Array dengan nama untai telah dideklarasikan dengan tipe integer, dengan jumlah elemen maksimum 50 elemen, nilai dari elemen array tersebut diatas harus bertipe integer.

Contoh :

Program contoh\_array\_input;

uses crt;

var

bilangan : array[1..50] of integer;

begin clrscr; bilangan[1]:=3; bilangan[2]:=29; bilangan[3]:=30; bilangan[4]:=31; bilangan[5]:=23;

writeln('nilai varibel bilangan ke 3 =',bilangan[3]);

readln;

end.

Array juga dapat dideklarasikan bersama dengan tipe yang beragam seperti contoh dibawah ini :

Program contoh\_deklarasi\_array\_beragam;

uses crt;

var

NPM : array[1..20] of string[10];

nama : array[1..20] of string[25];

nilai : array[1..20] of real; umur : array[1..20] of byte; banyak,i : integer;

begin clrscr;

write('Isi berapa data array');readln(banyak);

for i := 1 to banyak do begin

write('NPM =');readln(npm[i]); write('Nama =');readln(nama[i]); write('Nilai=');readln(nilai[i]); write('umur =');readln(umur[i]); end;

{cetak varibel array}

writeln('NPM NAMA NILAI UMUR ');

for i:= 1 to banyak do begin

writeln(npm[i]:10,nama[i]:25,nilai[i]:3:2,' ',umur[i]:3);

end;

READLN; end.

Untuk deklarasi array dapat digunakan beberapa cara seperti beri kut ini : Type

Angka =string[20]; Var

nama : array [1..50] of angka;

begin

. end.

***Deklarasi tipe indeks subrange integer***

Indeks pada array dapat tipe skalar atau subrange, tetapi tidak bisa real.

Contoh:

var

nilai : array[1..10] of integer;

Pada contoh ini array nilai mempunyai 10 buah elemen yaitu dari 1 sampai 10. Array tersebut dapat dideklarasikan dengan type seperti berikut ini :

Type

skala = 1..10;

var

nilai : array [skala] of integer;

atau : Type

skala = 1..10;

Y = array[skala] of integer;

var

nilai : Y;

atau :

Type

Y = array[1..10] of integer;

var

nilai : Y;

Atau :

const

atas =1;

bawah = 5;

type

y = array[atas..bawah] of integer;

var

nilai : y;

**I. Deklarasi type indeks skalar**

Indeks dari larik dapat berupa tipe skalar. Contoh :

program deklarasi\_indeks\_array\_skalar;

uses crt;

var

jum : array[(jan,feb,mar,apr,mei)] of integer;

begin jum[jan]:=25; jum[feb]:=45; jum[mar]:=21; jum[apr]:=23; jum[mei]:=50;

writeln('Jumlah nilai bulan maret =',jum[mar]);

readln;

end.

dapat juga ditulis :

type

bln = (jan,feb,mar,apr,mei); Var

jum : array[bln] of integer;

atau :

type

bln =(jan,feb,mar,apr,mei);

var

jum : array[jan..mei] of integer;

**II. Deklarasi konstanta array**

Array tidak hanya dapat berupa suatu varibel yang dideklarasikan di bagian deklarasi

variabel, tetapi dapat juga berupa konstanta (const). Contoh :

program contoh\_deklarasi\_array\_konstan;

uses crt;

const

tetap : array[1..4] of integer=(7,10,21,20);

var

i : integer;

begin

for i:= 1 to 4 do

writeln('Nilai Konstan array ke ',i:2,' =',tetap[i]);

readln;

end.

Konstanta array dapat juga berupa ketetapan dalam bentuk karakter seperti berikut. Contoh :

program contoh\_konstan\_array\_char\_;

uses crt;

const

huruf : array[0..5] of char=('A','B','C','D','E','F'); VAR

i : integer;

begin

for i:= 0 to 5 do

writeln('Nilai konstan array ke',i:2,' = ',huruf[i]);

readln;

end.

Konstanta array dapat juga berupa string seperti berikut ini. Contoh :

program constanta\_array\_string;

uses crt;

type

A = array [1..5] of string;

const

Nama : A = ('basic','pascal','cobol','paradox','dbase');

var

I : integer;

begin

for i:= 1 to 5 do

writeln('Nilai array ke-',i:2,'= ',nama[i]);

readln;

end.

Dalam pascal string merupakan array dari elemen- elemen karakter seperti berikut : Contoh :

program string\_adalah\_array\_tipe\_char;

uses crt;

var

nama : string; i : integer; begin

nama:='Turbo Pascal';

for i:= 1 to length(nama) do

writeln('Elemen ',i,' dari ',Nama,'= ',nama[i]);

readln;

end.

Contoh program bilangan prima dengan menggunakan bantuan array. program mencari\_bilangan\_prima\_dengan\_array;

uses crt;

var

prima : array[1..100] of integer;

i,j : integer; bil : integer; begin

clrscr;

for i := 2 to 100 do

begin prima[i]:=i;

for j:= 2 to i-1 do begin

bil := (i mod j); { i dibagi j dicek apakah 0}

if bil = 0 then prima[i]:=0;

{jika habis dibagi,berarti bkn prima}

end;

if prima[i]<> 0 then write(prima[i],' ');

{cetak array yg prima}

end; readln; end.

Contoh pengurutan data dengan metode buble sort, yaitu dengan cara penukaran, dapat dilihat pada contoh dibawah ini :

Contoh program :

program penggunaan\_array\_untuk\_sortir\_buble\_sort;

uses crt;

var

nil1 : array[1..100] of integer;

n,i,j,dum : integer;

begin

clrscr;

write('mau isi berapa data acak (integer) ='); readln(n);

for i := 1 to n do begin

Write('Data Ke ',i,':');readln(nil1[i]);

end;

{\* penyapuan proses}

for i:= 1 to n-1 do begin

for j:= i to n do

begin

if nil1[j]<nil1[i] then begin

dum:=nil1[j]; nil1[j]:=nil1[i]; nil1[i]:=dum; end;

end; end; writeln;

writeln('Hasil Sortir'); for i := 1 to n do write(nil1[i]:3);

readln;

end.

**III. Array dua dimensi**

Di dalam pascal Array dapat berdimensi lebih dari satu yang disebut dengan array

dimensi banyak (Multidimensional array), disini akan dibahas array 2 dimensi saja. Array

2 dimensi dapat mewakili suatu bentuk tabel atau matrik, yaitu indeks pertama menunjukkan baris dan indeks ke dua menunjuk kan kolom dari tabel atau matrik.

Untuk mengetahui cara mendeklarasikan dari penggunaan aray dua dimensi dapat dilihat pada listing program dibawah ini .

Contoh :

Program deklarasi\_array\_dua\_dimensi;

uses crt;

var

tabel : array[1..3,1..2] of integer;

i,j : integer;

begin

clrscr; tabel[1,1]:=1; tabel[1,2]:=2; tabel[2,1]:=3; tabel[2,2]:=4; tabel[3,1]:=5; tabel[3,2]:=6;

for I := 1 to 3 do

begin

for J:= 1 to 2 do

begin

writeln('Elemen ',i,',',j,'= ',tabel[i,j]);

end; end; readln; end.

**IV. Alternatif deklarasi array dua dimensi**

Ada beberapa cara dalam mendeklarasikan array dua dimensi, beberapa cara tersebut

dapat dilihat dibawah ini : Contoh :

Var

tabel : array[1..3] of array[1..2] of byte;

atau :

type

matrik = array[1..3,1..2] of byte;

var

tabel : matrik;

atau :

Type

baris = 1..3;

kolom = 1..2;

matrik = array[baris,kolom] of byte;

var

tabel : matrik;

atau :

type

baris = 1..3;

kolom=1..2;

matrik=array[baris] of array[kolom] of byte;

var

tabel : matrik;

Dibawah ini akan diberikan listing program penggunaan array dua dimensi dalam aplikasi penjumlahan matrik :

Contoh:

Program Penjumlahan\_matrik;

uses crt; var matrik1,matrik2

, hasil : array[1..3,1..2] of integer;

i,j : integer;

begin clrscr;

{ input matrik ke satu } writeln(' Elemen matrik satu'); for i := 1 to 3 do

begin

for j := 1 to 2 do begin

write('Elemen baris -',i,' kolom -',j,'= ');

readln(matrik1[i,j]);

end;

end;

{input matrik ke dua}

writeln('input elemen matrik dua');

for i:= 1 to 3 do

begin

for j:= 1 to 2 do begin

write('Elemen baris -',i,' kolom -',j,'= ');

readln(matrik2[i,j]);

end;

end;

{proses penjumlahan tiap elemen}

for i := 1 to 3 do begin

for j:= 1 to 2 do

begin hasil[i,j]:=matrik1[i,j]+matrik2[i,j]; end;

end;

{proses cetak hasil} for i:= 1 to 3 do begin

for j:= 1 to 2 do begin

write(hasil[i,j]:6);

end; writeln; end; readln; end.

**V. Array sebagai parameter**

Array dapat digunakan sebagai parameter yang dikirimkan baik secara nilai (by value)

atau secara acuan (by reference) ke procedure atau ke function. Procedure yang menggunakan parameter berupa array harus dideklara sikan di dalam judul procedure yang menyebutkan parameternya bertipe array.

Contoh :

program contoh\_pengiriman\_parameter\_array\_di\_procedure;

uses crt;

const

garis ='---------------------------------------------------';

type

untai = array[1..10] of string[15];

bulat = array[1..10] of integer; huruf = array[1..10] of char; var

i,banyak : integer;

procedure proses(nama:untai;nilai:bulat);

var

ket : string; abjad : char; begin writeln(garis);

writeln('Nama Nilai Abjad Keterangan');

writeln(garis);

for i := 1 to banyak do begin

if nilai[i] > 90 then begin

abjad:='A';

ket :='Istimewa';

end;

if (nilai[i]<90) and (nilai[i]>70) then begin

abjad:='B';

ket :='Memuaskan';

end;

if (nilai[i]<70) and (nilai[i]>60) then begin

abjad:='C';

ket :='Cukup';

end;

if (nilai[i]<60) and (nilai[i]>45) then begin

abjad:='D';

ket :='Kurang';

end;

if nilai[i]< 45 then begin

abjad:='E';

ket :='Sangat kurang';

end;

writeln(nama[i]:15,' ',nilai[i]:4,' ',abjad,' ',ket:15);

end;

writeln(garis);

end;

procedure masuk\_data;

var

nama : untai; nilai : bulat; begin

write('banyak data =');readln(banyak);

for i:= 1 to banyak do begin

clrscr;

writeln('Data ke - ',i); write('Nama =');readln(nama[i]); write('Nilai =');readln(nilai[i]);

end; proses(nama,nilai); end;

{modul Utama} begin masuk\_data; readln;

end.

**Record**

Tipe data record merupakan tipe data terstruktur. Dalam penggunakan tipe data record dapat dikumpulkan beberapa item data yang masing- masing mempunyai tipe data berbeda- beda. Record dapat berisi beberapa field untuk sebuah subyek tertentu.

**I. Deklarasi record**

Diawali kata cadangan Record , lalu diikuti daftar field dan diakhiri kata

cadangan end; Contoh :

type

data\_pegawai = record kd\_peg : string[5]; nama : string[15]; alamat : string[20]; gaji : longint;

end;

var

pegawai : data\_pegawai;

atau langsung di deklarasikan di varibel :

var

pegawai : record

kd\_peg : string[5]; nama : string[15]; alamat : string[20]; gaji : longint;

end;

Cara menggunakan tiap field dari record untuk input, cetak dan proses adalah sebagai berikut :

Contoh :

Nama\_record.Nama\_field

program contoh\_record\_sederhana;

uses crt;

type

data\_pegawai = record kd\_peg : string[5]; nama : string[15];

var

alamat : string[20];

gaji : longint;

end;

pegawai : data\_pegawai;

begin clrscr;

write('Kode pegawai =');readln(pegawai.kd\_peg); write('Nama pegawai =');readln(pegawai.nama); write('Alamat pegawai =');readln(pegawai.alamat); write('Gaji pegawai =');readln(pegawai.gaji);

{cetak}

writeln('Kode pegawai :',pegawai.kd\_peg); writeln('Nama pegawai :',pegawai.nama); writeln('Alamat pegawai :',pegawai.alamat); writeln('Gaji pegawai :',pegawai.gaji); readln;

end.

**II. Statemen with**

Penggunaan statemen nama\_record.nama\_field seperti contoh sebelumnya dapat

diringkas menjadi : Contoh :

program contoh\_record\_menggunakan\_statmen\_with;

uses crt;

type

data\_pegawai = record kd\_peg : string[5]; nama : string[15]; alamat : string[20]; gaji : longint;

end;

var

pegawai : data\_pegawai;

begin clrscr;

with pegawai do

begin

write('Kode pegawai =');readln(kd\_peg); write('Nama pegawai =');readln(nama); write('Alamat pegawai =');readln(alamat); write('Gaji pegawai =');readln(gaji);

{cetak}

writeln('Kode pegawai :',kd\_peg); writeln('Nama pegawai :',nama); writeln('Alamat pegawai :',alamat); writeln('Gaji pegawai :',gaji);

end;

readln;

end.

Penjelasan :

Dengan menggunakan staremen with maka blok statemen berikutnya setelah statemen

With dapat menggunakan nama field tanpa menyebut kan nama recordnya lagi.

**III. Record dalam array**

Dalam contoh sebelunya penggunan tipe data record hanya dapat menyimpan satu

record. Untuk dapat menyimpan sejumlah record maka dapat digunakan array yang bertipe record yang sudah didifinisikan. Untuk itu dapat dilihat listing program berikut. Contoh :

program contoh\_record\_dalam\_array;

uses crt;

type

data\_pegawai = record kd\_peg : string[5]; nama : string[15]; alamat : string[20]; gaji : longint; end;

var

pegawai : array[1..10] of data\_pegawai;

i : integer;

begin clrscr;

for I:= 1 to 10 do begin

with pegawai[i] do begin

writeln('Record ke- ',i);

write('Kode pegawai =');readln(kd\_peg); write('Nama pegawai =');readln(nama); write('Alamat pegawai =');readln(alamat); write('Gaji pegawai =');readln(gaji); writeln;

end;

end;

{cetak}

writeln('Kode pegawai Nama Alamat gaji');

for i:= 1 to 10 do begin

with pegawai[i] do begin write(kd\_peg:5); write(nama:15); write(alamat:20); writeln(gaji:10); end;

end;

writeln('------------------------------------------');

readln;

end.

**IV. Field record bertipe array**

Jika dalam suatu record terdapat beberapa field yang sama tipenya dapat digunakan

array. Contoh ada data barang yang mempunyai struktur.

- Nama barang -> bertipe String

- Jumlah unit barang ke 1 -> bertipe Byte

- Jumlah unit barang ke 2-> bertipe Byte

- Jumlah unit barang ke 3-> bertipe Byte

Terlihat bahwa jumlah unit barang 1,2,3 bertipe sama. Dalam hal ini dapat digunakan array ber index 1.. 3 untuk mempersingkat filed jumlah unit barang.

Contoh :

program penggunaan\_field\_record\_tipe\_array;

uses crt;

type

data\_brg = record namaBrg : string[15]; unitBrg : array[1..3] of byte; end;

var

Barang : array[1..10] of data\_brg;

i,j,banyak, Jum1,jum2,jum3 : integer; begin

jum1 :=0; jum2 :=0; jum3 :=0;

write('Banyak record Max 10 =');readln(banyak);

for i:= 1 to banyak do begin

with barang[i] do begin

writeln('Record ke -',i);

write('Nama barang =');readln(namabrg);

for j:= 1 to 3 do begin

write('Unit barang ke- ',j,'= ');readln(unitbrg[j]);

end; end; end; clrscr;

writeln('---------------------------------'); writeln('Nama barang unit 1 unit2 unit3'); writeln('---------------------------------');

{ cetak data }

for i:= 1 to banyak do begin

with barang[i] do begin jum1:=jum1+unitbrg[1]; jum2:=jum2+unitbrg[2]; jum3:=jum3+unitbrg[3];

writeln(namabrg:15,unitbrg[1]:5,unitbrg[2]:5,unitbrg[3]:5);

end;

end;

writeln('---------------------------------'); writeln('Jumlah unit 1 =',jum1:6); writeln('Jumlah unit 2 =',jum2:6);

writeln('Jumlah unit 3 =',jum3:6);

readln;

end.

**V. Tipe data record dengan field tipe record**

Dalam Turbo Pascal tipe data record dapat didefinisikan juga sebagai field dari suatu

record. Artinya suatu record dapat juga mempunyai field yang merupakan record. Contoh: sebuah data pegawai mempunyai struktur sebagai berikut :

- Nama pegawai -> string

- Mulai masuk -> - Tgl

- Bln

- Thn

- Alamat pegawai -> - Jalan

- Kota

- Gaji -> - Gaji pokok

- Lembur

- Tunjangan

Maka dapat disusun program sebagai berikut : Contoh :

program penggunaan\_field\_tipe\_record;

uses crt;

type

masuk = record tgl : 1..31; bln : 1..12;

thn : integer;

end;

alamat = record

jalan : string[20]; kota : string[10]; end;

gajipeg = record pokok,tunjangan,lembur : real; end;

datapegawai = record

nama : string[20]; tglmasuk : masuk; almt : alamat;

gaji : gajipeg;

end;

var

pegawai : array [1..10] of datapegawai;

i,p,banyak : integer;

begin clrscr;

write('Banyak data record =');readln(banyak);

for i := 1 to banyak do begin

writeln('record ke -',i); with pegawai[i] do begin

write('nama pegawai :');readln(nama); write('Tanggal masuk:');readln(tglmasuk.tgl); write('Bulan Masuk :');readln(tglmasuk.bln); write('Tahun masuk :');readln(tglmasuk.thn); write('Alamat :');readln(almt.jalan); write('Kota :');readln(almt.kota); write('Gaji pokok :');readln(gaji.pokok); write('Tunjangan :');readln(gaji.tunjangan); write('Lembur :');readln(gaji.lembur);

end;

end;

{ cetak data }

for i := 1 to banyak do begin

writeln('record ke -',i);

with pegawai[i] do begin

writeln('nama :',nama); writeln('Tanggal masuk:',tglmasuk.tgl); writeln('Bulan Masuk :',tglmasuk.bln); writeln('Tahun masuk :',tglmasuk.thn); writeln('Alamat :',almt.jalan); writeln('Kota :',almt.kota); writeln('Gaji pokok :',gaji.pokok); writeln('Tunjangan :',gaji.tunjangan); writeln('Lembur :',gaji.lembur);

end;

end; readln; end.

**VI. Record bervariasi**

Record yang telah dibahas sebelumnya merupakan struktur record yang pasti, artinya

field-field di dalam record sudah tertentu dan pasti. Selain itu di program Pascal dapat juga dibuat suatu record yang mempunyai field yang tidak pasti atau dapat berubah, yang disebut sebagai record yang bervariasi. Dalam record yang bervariasi dapat

mengandung suatu field yang tergantung dari suatu kondisi. Dalam penerapanya dalam program hanya dapat diterima satu buah field yang bervariasi saja. Field bervariasi ini harus terletak dibawah field yang tetap.

Contoh :

Ada sebuah struktur data pegawai yang terdiri dari :

- Nama pegawai

- Alamat pegawai

- Umur

- Gaji -> untuk gaji dibedakan antara pegawai tetap dgn honorer

Untuk tetap - Tunjangan

- Lembur

- Gaji pokok

Untuk honorer - Gaji pokok

Deklarasi dan program :

type

Status\_pegawai = (Honorer,Tetap);

data\_pegawai = record

nama : string[15]; alamat : string[20]; umur : byte;

case status : status\_pegawai of honorer : (gaji\_h : real);

tetap : (gaji\_t : real; tunjangan : real; lembur : real);

end;

Contoh :

type

Status\_pegawai = (Honorer,Tetap);

data\_pegawai = record

nama : string[15]; alamat : string[20]; umur : byte;

case status : status\_pegawai of honorer : (gaji\_h : real);

tetap : (gaji\_t : real; tunjangan : real; lembur : real);

var

end;

pegawai : array[1..10] of data\_pegawai;

banyak,i : integer;

kode : char;

begin

{ input data}

write('Banyak Data max 10 :');readln(banyak);

for i := 1 to banyak do begin

with pegawai[i] do

begin

write('Nama pegawai =');readln(nama); write('Alamat =');readln(alamat); write('Umur =');readln(umur);

write('pegawai tetap(T) atau Honorer(H) ');readln(kode);

kode := upcase(kode);

case kode of

'H' : begin

Status:=Honorer;

write('Gaji didapat= ');readln(gaji\_h);

end;

'T' : begin

status:= Tetap;

write('gaji tetap = ');readln(gaji\_t); write('Tunjangan = ');readln(tunjangan); write('Lembur = ');readln(lembur);

end;

end;

end;

end;

{ cetak data}

writeln('----------------------------------------------------');

writeln(' Nama Alamat Umur Status Gaji Tunjangan Lembur');

writeln('----------------------------------------------------');

for i:= 1 to banyak do begin

with pegawai[i] do begin write(nama:15); write(alamat:20); write(umur:3,' '); case status of

honorer : writeln('Honorer',Gaji\_h:8:2); Tetap :

writeln('Tetap ',Gaji\_t:8:2,tunjangan:8:2,lembur:8:2);

end; end; end;

writeln('---------------------------------------------');

readln;

end.

program contoh\_record\_bervariasi;

uses crt;

type

status\_karyawan = (lajang,menikah,cerai);

data\_karyawan = record nama : string[15]; alamat : string[20]; gaji : integer;

case status : status\_karyawan of

lajang :();

menikah :(anakm : 0..20);

cerai :(anakc : 0..20; lagi :char);

end;

var

karyawan : array [1..10] of data\_karyawan;

i,banyak : integer;

sts : char; begin clrscr;

write('Jumlah data record :');readln(banyak);

for i := 1 to banyak do begin

with karyawan[i] do begin

write('Nama =');readln(nama); write('Alamat =');readln(alamat); write('Gaji =');readln(gaji);

write('status M=menikah L=lajang C=cerai');readln(sts);

if upcase(sts)='L' then begin

status:=lajang;

end;

if upcase(sts)='M' then begin status:=Menikah;

write('Jumlah anak= ');readln(anakm);

end;

if upcase(sts)='C' then begin

status:=Cerai;

write('Jumlah anak = ');readln(anakc);

write('Kawin lagi (Y/T) = ');readln(lagi);

end;

end;

end;

{ tampil}

for i := 1 to banyak do begin

with karyawan[i] do begin

write(nama); write(alamat); write(gaji);

case status of

lajang : writeln('lajang');

menikah : begin

writeln('menikah',' ',anakm:4);

end;

cerai : begin

writeln('cerai ', ' ',anakc:4,' ',lagi);

end;

end; end; end; readln; end.

program penggunaan\_field\_tipe\_record;

uses crt;

type

masuk = record tgl : 1..31; bln : 1..12;

thn : integer;

end;

alamat = record

jalan : string[20]; kota : string[10]; end;

gajipeg = record pokok,tunjangan,lembur : real; end;

datapegawai = record

nama : string[20]; tglmasuk : masuk; almt : alamat;

gaji : gajipeg;

var

end;

pegawai : array [1..10] of datapegawai;

i,p,banyak : integer;

begin clrscr;

write('Banyak data record =');readln(banyak);

for i := 1 to banyak do begin

writeln('record ke -',i); with pegawai[i] do begin

write('nama pegawai :');readln(nama);

write('Tanggal masuk:');readln(tglmasuk.tgl); write('Bulan Masuk :');readln(tglmasuk.bln); write('Tahun masuk :');readln(tglmasuk.thn); write('Alamat :');readln(almt.jalan); write('Kota :');readln(almt.kota); write('Gaji pokok :');readln(gaji.pokok); write('Tunjangan :');readln(gaji.tunjangan); write('Lembur :');readln(gaji.lembur);

end;

end;

{ cetak data }

for i := 1 to banyak do begin

writeln('record ke -',i); with pegawai[i] do begin

writeln('nama :',nama);

writeln('Tanggal masuk:',tglmasuk.tgl); writeln('Bulan Masuk :',tglmasuk.bln); writeln('Tahun masuk :',tglmasuk.thn); writeln('Alamat :',almt.jalan); writeln('Kota :',almt.kota); writeln('Gaji pokok :',gaji.pokok); writeln('Tunjangan :',gaji.tunjangan); writeln('Lembur :',gaji.lembur);

end; end; readln; end.