

STRUKTUR DATA

BAB IV

MATRIKS (ARRAY MULTI DIMENSI)

DEFINISI “MATRIKS”

Matriks adalah:

1. Kumpulan elemen yang bertipe sama.
2. Setiap elemen data dapat diakses secara langsung jika indeksnya diketahui.
3. Struktur data yang statis, artinya jumlah elemen dideklarasikan terlebih dulu.

Ordo Matiks

Matriks A yang terdiri dari **m baris dan n kolom** disebut matriks berordo $m \times n$. Ordo suatu matriks ditentukan oleh banyaknya baris dan kolom, maka bentuk umum matriks ditulis sebagai berikut :

$$A(m \times n) = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

Dengan m = banyak baris
 n = banyak kolom
 $m \times n$ = ordo matiks

Contoh:

$$a.A = \begin{bmatrix} 1 & 0 \\ 3 & 2 \end{bmatrix} \quad b.B = \begin{bmatrix} 3 & 2 & -1 \\ 1 & 0 & 4 \\ 5 & -2 & 8 \end{bmatrix} \quad c.C = \begin{bmatrix} 2 \\ 4 \end{bmatrix} \quad d.D = \begin{bmatrix} 4 & 1 & 0 \\ 7 & 8 & 9 \end{bmatrix}$$

Jawab :

- Ordo matriks A adalah 2×2
- Ordo matriks B adalah 3×3
- Ordo matriks C adalah 2×1
- Ordo matriks D adalah 2×3

**Matriks adalah struktur data yang mengacu [ada
Sebuah/sekumpulan elemen yang diakses melalui
indeks**

KEUNTUNGAN & KERUGIANNYA

KEUNTUNGAN

- 1. Paling mudah dioperasikan**
- 2. Ekonomis dalam pemakaian memori, bila semua elemen terisi**
- 3. Akses ke setiap elemen memerlukan waktu yang sama**

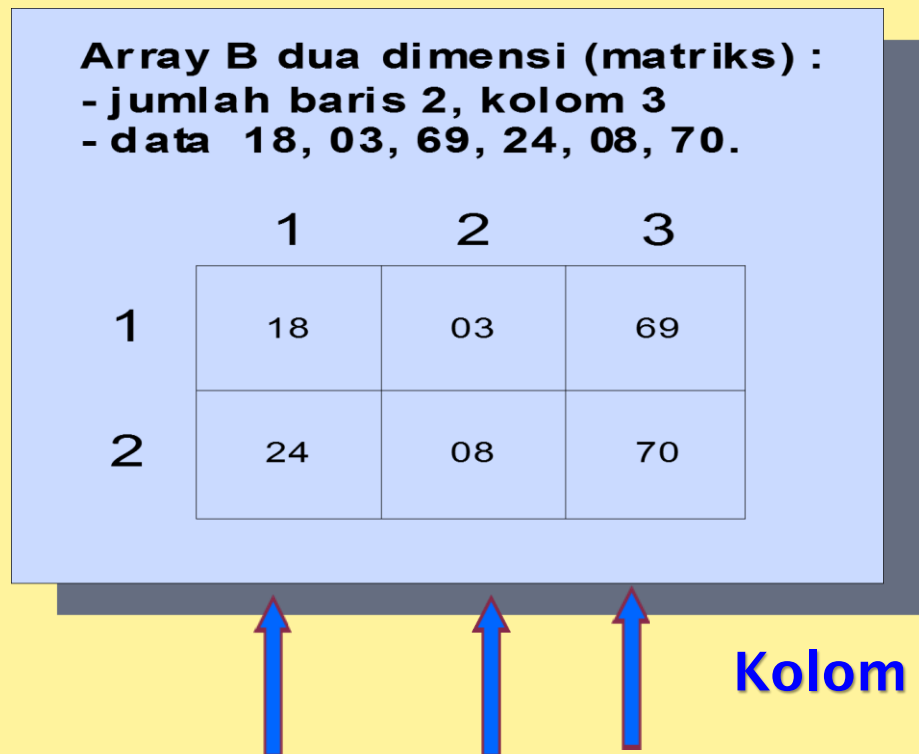
KERUGIAN

- 1. Memboroskan tempat jika banyak elemen yang tidak digunakan**

Array multi dimensi terdiri dari :

- Indeks Pertama : Baris (**row**)
- Indeks Kedua : Kolom (**column**).

Array jenis ini biasa digunakan untuk representasi dari **matriks** yang menyimpan data secara struktural/berurutan



Baris (row)

Elemen Matriks

B[1.1],B[1.2],B[1,3].
B[2,1],B[2,2],B[2,3]

Indeks Baris

B : 1, 2

Indeks Kolom

B : 1,2,3

Kolom (column)

Contoh :

Type

nama_array = ARRAY[bawah..atas, bawah..atas] of tipe_data;

var

variabel_array : nama_array;

atau dengan menggunakan statement var :

var

variabel_array : ARRAY[bawah..atas, bawah..atas] of tipe_data;

Penjelasan:

Bawah dan **Atas** menyatakan batas untuk array. tipe_data adalah merupakan tipe variabel yang dipunyai array (mis. Integer, char, real, dsb)

Contoh program sederhana array multi dimensi(2 dimensi) untuk matrix 3×3

Proses Matriks

1. Elemen Matriks diproses Baris demi Baris (Row Ordering)
2. Elemen Matriks diproses Kolom demi Kolom (Column Ordering)

Array B dua dimensi (matriks) :
- jumlah baris 2, kolom 3
- data 18, 03, 69, 24, 08, 70.

	1	2	3
1	18	03	69
2	24	08	70

Proses Matriks

Array B dua dimensi (matriks) :

- jumlah baris 2, kolom 3

- data 18, 03, 69, 24, 08, 70.

	1	2	3
1	18	3	69
2	24	8	70

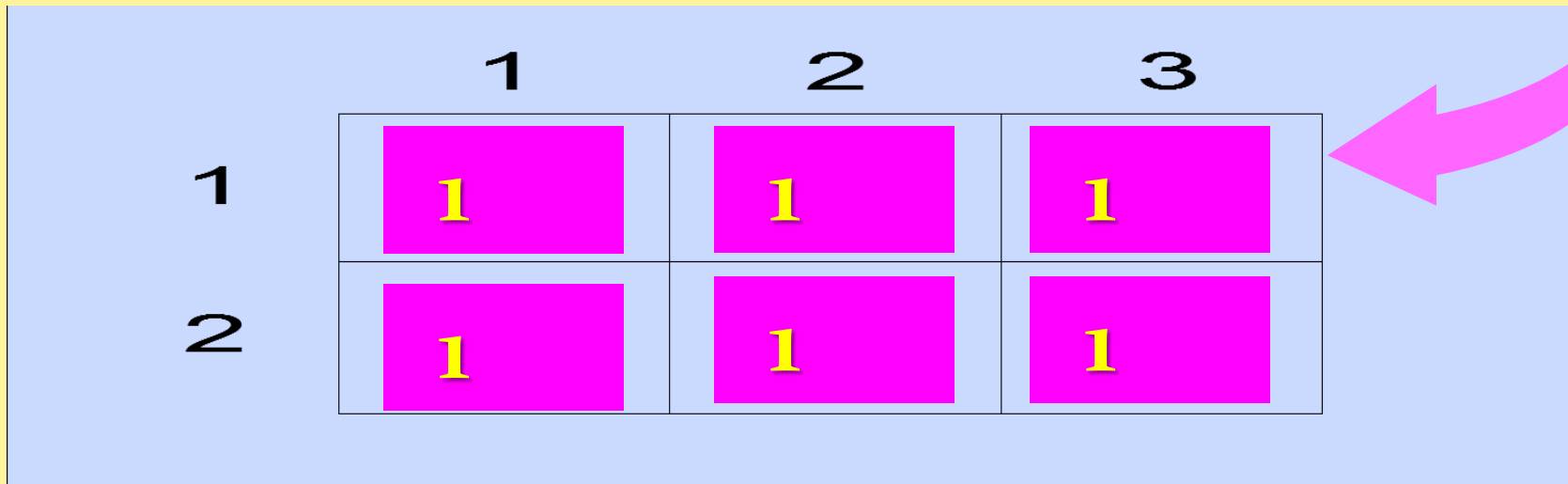
PROSES MATRIKS

- Array B dua dimensi (matriks) :**
- jumlah baris 2, kolom 3
 - data 18, 03, 69, 24, 08, 70.

	1	2	3
1	18	3	69
2	24	8	70

INISIALISASI

```
For Baris = 1 to 2 do  
  For Kolom = 1 to 3 do  
    A(Baris, Kolom) = 0  
  Endfor  
Endfor
```



	1	2	3
1	1	1	1
2	1	1	1

Isi dengan 1,2,3,4,5,6

Indeks = 1

For Baris = 1 to 2 do

For Kolom = 1 to 3 do

A(Baris, Kolom) = Indeks

Indeks = Indeks + 1

Endfor

Endfor

	1	2	3
1	1	2	3
2	4	5	6

Isi dengan 1,3,5,7,9,11

Indeks = ???

For Baris = 1 to 2 do

For Kolom = 1 to 3 do

A(Baris, Kolom) = ???

Indeks = ???

Endfor

Endfor

	1	2	3
1	1	3	5
2	7	9	13

Menjumlahkan setiap baris

```
For Baris = 1 to 2 do  
  TotalBaris = 0  
  For Kolom = 1 to 3 do
```

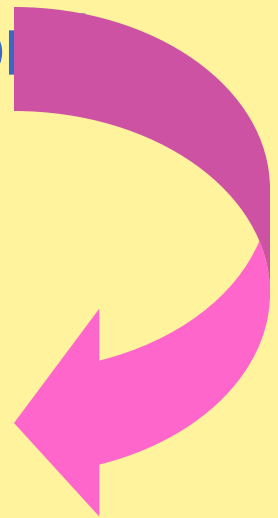
```
    TotalBaris = TotalBaris + A[Baris,Kolom]
```

```
  Endfor
```

```
  Print Total Baris
```

```
Endfor
```

	1	2	3	
1	18	3	69	90
2	24	8	70	102



Menjumlahkan Dua buah Matriks

$C = A + B$

For Baris = 1 to 2 do
For Kolom = 1 to 3 do

$C[\text{Baris}, \text{Kolom}] = A[\text{Baris}, \text{Kolom}] + B[\text{Baris}, \text{Kolom}]$

Endfor

Endfor

	1	2	3	
1	1 8	3	6 9	A
2	2 4	8	7 0	



	1	2	3	
1	1	2	3	B
2	4	5	6	

Mengalikan

```
For Baris = 1 to 2 do
```

```
  For Kolom = 1 to 3 do
```

```
    C[Baris, Kolom] = 0
```

```
    For K = 1 to P do
```

```
      C[Baris, Kolom] = C[B, K] + A[B, K] + B[K, K]
```

```
    Endfor
```

```
  Endfor
```

```
Endfor
```

	1	2	3
1	18	3	69
2	24	8	70

Jenis-Jenis Matriks

- Matriks Bujur Sangkar

Matriks yang jumlah baris dan jumlah kolomnya sama

Contoh :

$$\begin{bmatrix} 3 & 5 & 4 \\ 5 & 6 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

- Matriks Diagonal

Matriks bujur sangkar dimana unsur selain unsur diagonalnya adalah 0

Contoh :

$$\begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Matriks Identitas

Matriks diagonal yang unsur diagonalnya adalah 1

Contoh :

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Matriks Segitiga Atas

Matriks Bujur Sangkar yang semua unsur dibawah unsur diagonalnya bernilai 0

Contoh :

$$\begin{bmatrix} 5 & 9 & 3 \\ 0 & 1 & 7 \\ 0 & 0 & 8 \end{bmatrix}$$

▪ Matriks Segitiga Bawah

Matriks Bujur Sangkar yang semua unsur diatas unsur diagonalnya bernilai 0

Contoh :

$$\begin{bmatrix} 2 & 0 & 0 \\ 5 & 1 & 0 \\ 3 & 0 & 2 \end{bmatrix}$$

▪ Matriks Nol

Matriks yang semua unturnya bernilai Nol

Contoh :

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

- Matrik *transpose* A, dengan notasi A^t
Matriks yang diperoleh dengan mengubah baris matriks A menjadi kolom matriks pada matriks A^t

Contoh :

$$A = \begin{pmatrix} 2 & 1 \\ 3 & -2 \\ -1 & 0 \end{pmatrix} \quad \text{maka} \quad A^t = \begin{pmatrix} 2 & 3 & -1 \\ 1 & -2 & 0 \end{pmatrix}$$

Sifat *Tranpose*

1. $(A^t)^t = A$
2. $(AB)^t = B^t A^t$

- Matriks simetri

Matriks yang memenuhi hubungan $A = A^t$

Contoh :

$$\begin{pmatrix} 1 & -3 & 2 & 0 \\ -3 & 2 & 5 & -1 \\ 2 & 5 & 3 & -2 \\ 0 & -1 & -2 & 4 \end{pmatrix}$$

▪ Matrik Eselon Baris Tereduksi

Matriks yang mempunyai ciri-ciri sbb:

1. Pada baris tak nol maka unsur tak nol pertama adalah 1 (disebut 1 utama).
2. Pada baris yang berturutan baris yang lebih rendah memuat 1 utama yang lebih ke kanan.
3. Jika ada baris nol (baris yang semua unsurnya nol), maka ia diletakkan paling bawah.
4. Pada kolom yang memuat 1 utama, unsur yang lainnya adalah nol.

Catatan :

- Jika poin 1, 2, dan 3 dipenuhi, matriks dinamakan berbentuk *eselon baris*

Operasi Matriks

Penjumlahan Matriks

Syarat yang harus dipenuhi oleh keduanya adalah orde kedua matriks tersebut harus sama. Penjumlahan dua buah matriks akan menghasilkan sebuah matriks dengan ordo yang sama, dan setiap unsur didalamnya merupakan hasil penjumlahan dari unsur yang seletak pada kedua matriks tersebut.

Contoh :

Penjumlahan dua matriks berukuran 2 x 2 adalah sebagai berikut :

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a + e & b + f \\ c + g & d + h \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 6 & 8 \\ 10 & 12 \end{bmatrix}$$

• Perkalian Matriks Dengan Skalar

Contoh :

$$\text{Misalkan } C \in \mathfrak{R} \text{ dan } A = \begin{bmatrix} p & q \\ r & s \end{bmatrix}$$

$$\begin{aligned} C \times A &= C \begin{bmatrix} p & q \\ r & s \end{bmatrix} \\ &= \begin{bmatrix} Cp & Cq \\ Cr & Cs \end{bmatrix} \end{aligned}$$

Operasi Matriks

▪ Perkalian Matriks Dengan Matriks

Misalkan matriks $A_{m \times n}$ dan $B_{p \times q}$

Maka : - $A \times B$ bisa dilakukan jika $n = p$ dan hasilnya berorde $m \times q$

- $B \times A$ bisa dilakukan jika $q = m$ dan hasilnya berorde $p \times n$

Contoh :

$$A = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}_{2 \times 3}$$

dan

$$B = \begin{bmatrix} p & s \\ q & t \\ r & u \end{bmatrix}_{3 \times 2}$$

$$\text{Maka : } A \times B = \begin{bmatrix} ap + bq + cr & as + bt + cu \\ dp + eq + fr & ds + et + fu \end{bmatrix}_{2 \times 2}$$

Perhatikan bahwa unsur baris ke-2 kolom ke-1 dari AB merupakan jumlah dari hasil kali unsur-unsur pada baris ke-2 matriks A dengan unsur-unsur pada kolom ke-1 matriks B.

Matriks Invers

- Misalkan, A , B adalah matriks bujur sangkar dan berukuran sama dan I adalah matriks identitas.
- Jika $A \cdot B = I$ maka B merupakan invers dari A dengan notasi $B = A^{-1}$, dan sebaliknya.
 - Sifat *Invers*
 - $(A^{-1})^{-1} = A$
 - $(AB)^{-1} = B^{-1}A^{-1}$
 - Contoh: Diketahui

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 5 \end{bmatrix} \text{ dan } B = \begin{bmatrix} -5 & 2 \\ 3 & -1 \end{bmatrix}$$

- Terlihat bahwa $A \cdot B = B \cdot A = I$ maka B merupakan invers dari A dengan notasi $B = A^{-1}$, dan sebaliknya.

CONTOH

```
Program Menyusun_Kali_Matrik;  
Uses Wincrt;  
Var i,j,n:integer;  
Begin  
  Write('Masukkan Jumlah Perkalian: ');Readln(n);  
  Write('*':5);  
  For i:= 1 to n do  
    Write(i:5);  
  Writeln;  
  For i:= 1 to n do  
    Begin  
      Write(i:5);  
      For j:= 1 to n do  
        write(i*j:5);  
      Writeln;  
    End;  
End.
```

Pendeklarasian Matriks

1. Sebagai nama peubah.

DEKLARASI

```
M : array [1..5, 1..4] of integer
```

2. Sebagai tipe

DEKLARASI

```
type Mat : array[1..5, 1..4] of integer
```

```
M : Mat
```

3. Mendefinisikan ukuran maksimum matriks sebagai sebuah konstanta

DEKLARASI

```
const NbarisMaks = 20
```

```
const NkolomMaks = 20
```

```
M : array [1..NbarisMaks, 1..NKolomMaks] of integer
```

Pemrosesan Matriks

- Pemrosesan dengan menggunakan “for”

```
procedure ProsesMatriks1 (input M : MatriksInt, input Nbar,  
Nkol : integer)  
{Pemrosesan elemen matriks M[1..Nbar, 1..Nkol] per baris per  
kolom.}
```

```
{K.Awal : Matriks M sudah terdefinisi elemen-elemennya.}
```

```
{K.Akhir : Setiap elemen matriks M telah diproses.}
```

```
DEKLARASI
```

```
    i : integer
```

```
    j : integer
```

```
ALGORITMA:
```

```
    for i ← 1 to Nbar do
```

```
        for j ← 1 to Nkol do
```

```
            Proses(M[i, j])
```

```
        endfor
```

```
    endfor
```

▪ Pemrosesan dengan menggunakan “ while “

procedure ProsesMatriks2(input M : MatriksInt, input Nbar, Nkol : integer)
{Pemrosesan elemen matriks M[1..Nbar, 1..Nkol] per baris per kolom.}
{K.Awal : Matriks M sudah terdefinisi elemen-elemennya.}
{K.Akhir : Setiap elemen matriks M telah diproses.}

DEKLARASI

i : integer

j : integer

ALGORITMA:

i \leftarrow 1

while i \leq Nbar do

 j \leftarrow 1

 while j \leq Nkol do

 proses (M[i, j])

 j \leftarrow j+1

 endwhile

 i \leftarrow i+1

endwhile

▪ Pemrosesan dengan menggunakan “ repeat – until “

```
procedure ProsesMatriks3(input M : MatriksInt, input Nbar, Nkol : integer)
{Pemrosesan elemen matriks M[1..Nbar, 1..Nkol] per baris per kolom.}
{K.Awal : Matriks M sudah terdefinisi elemen-elemennya.}
{K.Akhir : Setiap elemen matriks M telah diproses.}
```

DEKLARASI

i : integer

j : integer

ALGORITMA:

i ← 1

repeat

 i ← 1

 repeat

 proses (M[i, j])

 j ← j+1

 until j>Nkol

 i ← i+1

until i >Nbar

STRUKTUR DATA

BAB V RECORD

DEFINISI

- Tipe data record merupakan tipe data terstruktur
- Tipe data record digunakan untuk menyimpan sejumlah data dengan nilai dengan tipe data yang berbeda dalam satu wadah.

PERBEDAAN RECORD DAN ARRAY

- Array (Larik) semua elemennya harus bertipe sama
- Record semua elemennya harus bertipe berbeda antara satu sama lainnya.

DEKLARASI PENULISAN

Type

Pengenal = Record

Namafield-1 : Type

Namafield-2 : Type

.....

Namafield-N : Type

End

Atau dapat juga dideklarasikan sebagai berikut :

Var

Pengenal = Record

Namafield-1 : Type

Namafield-2 : Type

.....

Namafield-N : Type

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 variabel :

```
var
```

```
    pegawai : record
```

```
        kd_peg : string[5];
```

```
        nama : string[15];
```

```
        alamat : string[20];
```

```
        gaji : longint;
```

```
end;
```

Contoh

type

```
data_pegawai = record
```

```
    kd_peg : string[9];
```

```
    nama : string[25];
```

```
    alamat : string[29];
```

```
    gaji : longint;
```

```
end;
```

var

```
pegawai : data_pegawai;
```

```
begin
```

```
    pegawai.kd_peg := '0213001';
```

```
    pegawai.nama := 'James Tenges';
```

```
    pegawai.alamat:= 'Jl. Sam Ratulangi No 56 Manado';
```

```
    pegawa.gaji:=3500000;
```

```
    writeln('Kode Pegawai :',pegawai.kd_peg);
```

```
    writeln('Nama :',pegawai.nama);
```

```
    writeln('Alamat :',pegawai.alamat);
```

```
    writeln('Gaji :',pegawai.gaji);
```

```
readln;
```

```
end.
```

Statement “ With “

- Digunakan untuk mempersingkat penulisan dalam pembacaan field,
- Penulisan :
 - **with namaRecord do**

Contoh

```
begin
clrscr;
  with pegawai do
  begin
    kd_peg := '0213001';
    nama := 'James Tenges';
    alamat:= 'Jl. Kyi Telingsing No 56
Kudus';
    gaji:=3500000;
  end;
```


Record dalam array

- Dalam contoh sebelumnya penggunaan tipe data record hanya dapat menyimpan satu record.
- Untuk dapat menyimpan sejumlah record maka dapat digunakan array yang bertipe record dan sudah didefinisikan

Contoh

```
type
  data_pegawai = record
    kd_peg : string[9];
    nama : string[25];
    alamat : string[29];
    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
```

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

Contoh

```
type
  data_brg = record
    namaBrg : string[15];
    unitBrg : array[1..3] of byte;
  end;
var
  Barang : array[1..10] of data_brg;
```

Tipe data “record” dengan field “tipe record”

- ❖ Dalam 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