

Algoritma & Pemrograman #10



by Sufajar Butsianto, MKom

Array

KELEBIHAN

- Struktur Data paling mudah
- Memori ekonomis, bila semua elemen terisi
- Waktu akses sama ke setiap elemen

KEKURANGAN

- ✦ Boros memori jika banyak elemen yang tidak digunakan
- ✦ Struktur Data Statis

Array Dimensi 2

- Sering kali digambarkan/dianalogikan sebagai sebuah matriks.
- Jika array berdimensi satu hanya terdiri dari 1 baris dan banyak kolom, array berdimensi dua terdiri dari banyak baris dan banyak kolom yang bertipe sama
- Gambar array berdimensi (baris x kolom = 3 x 4)

	0	1	2	3
0	5	20	1	11
1	4	7	67	-9
2	9	0	45	3

Deklarasi Array 2 Dimensi

□ `tipe_data nama_var_array[batas_baris][batas_kolom];`

Contoh:

□ `int matriks[3][4];`

□ `int matriks2[3][4] = { {5,20,1,11}, {4,7,67,-9},
{9,0,45,3} };`

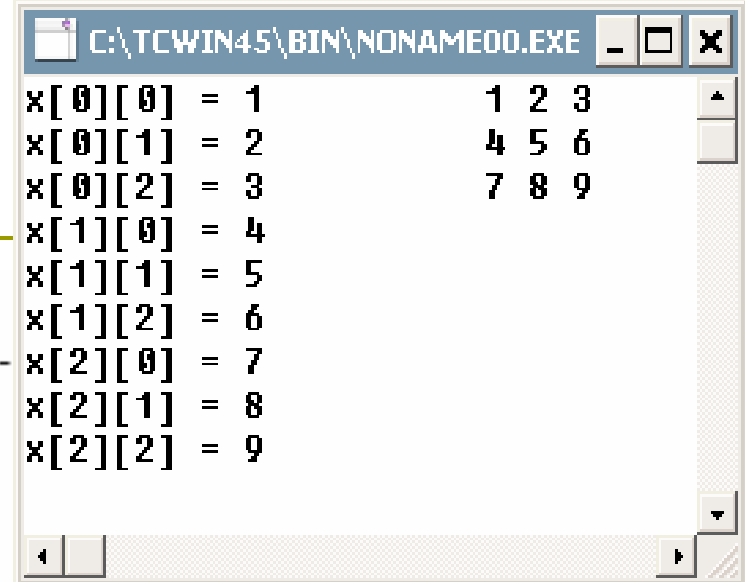
Contoh

Contoh:

```
#include <stdio.h>
#include <conio.h>

void main()
{ int x[3][3]={{1,2,3},{4,5,6},{7,8,9}},b,k;
  clrscr();
  for(b=0 ; b<3 ; b++)
    for(k=0 ; k<3 ; k++)
      printf("x[%d][%d] = %d\n",b,k,x[b][k]);

  for(b=0 ; b<3 ; b++)
    for(k=0 ; k<3 ; k++)
    {
      gotoxy(20+(k+1)*2,b+1);
      printf("%d",x[b][k]);
    }
  getch();
}
```



```
C:\TCWIN45\BIN\NONAME00.EXE
x[0][0] = 1      1 2 3
x[0][1] = 2      4 5 6
x[0][2] = 3      7 8 9
x[1][0] = 4
x[1][1] = 5
x[1][2] = 6
x[2][0] = 7
x[2][1] = 8
x[2][2] = 9
```

Contoh (2)

$$\mathbf{x} = \begin{pmatrix} 8 & 5 & 9 & 8 \\ 8 & 2 & 1 & 0 \end{pmatrix}$$

Deklarasi:

```
int x[2][4];
```

- `x[0][0]=8; x[0][1]=5; x[0][2]=9; x[0][3]=8;`
- `x[1][0]=8; x[1][1]=2; x[1][2]=1; x[1][3]=0;`

atau

- `int x[2][4]= {{8, 5, 9, 8},{8, 2, 1, 0}};`

Contoh (3)

□ `int matriks[3][5] = {{5,12,17,10,7},
 {15,6,25,2,19},
 {4,9,20,22,11}};`

□ Jika data yang diinputkan kurang dari deklarasi
`int matriks[3][5] = {{5,12,17,10,7},
 {15,6,25,2,19},
 {4,9 }}}`; //kurang 3 angka

Maka tiga data yang kurang akan diisi dengan karakter NULL atau berupa tanda `\0`

□ Jika data yang diinputkan lebih dari deklarasi
`int matriks[3][5] = {{5,12,17,10,7},
 {15,6,25,2,19},
 {4,9,20,22,11,14,19 }}}`; //lebih 2 angka

Matriks yang jumlah datanya lebih akan menyebabkan **ERROR**

Contoh 4

Matriks dengan deklarasi baris dinamis (contoh 1):

```
int matriks[][5] = {{5, 12, 17, 10, 7},  
                   {15, 6, 25, 2, 19},  
                   {4, 9, 20, 22, 11}};
```

Akan ditampilkan sebagai:

5	12	17	10	7
15	6	25	2	19
4	9	20	22	11

Matriks dengan deklarasi baris dinamis (contoh 2):

```
int matriks[][5] = {5, 12, 17, 10, 7, |  
                   15, 6, 25, 2, 19,  
                   4, 9, 20, 22, 11, 77, 88, 99};
```

⊕ Akan ditampilkan sebagai:

5	12	17	10	7
15	6	25	2	19
4	9	20	22	11
77	88	99	\0	\0

Contoh 5

```
char matriks[3][5] = { {'A', 'B', 'C', 'D', 'E'},  
                       {'F', 'G', 'H', 'I', 'J'},  
                       {'K', 'L', 'M', 'N', 'O'} };  
char matriks[3][5] = { "ABCDE",  
                       "FGHIJ",  
                       "KLMNO" } ;
```

Akan ditampilkan sebagai:

A	B	C	D	E
F	G	H	I	J
K	L	M	N	O

Jika jumlah nilai lebih banyak daripada deklarasi

```
char matriks[3][5] = { "ABCDEXXX",  
                       "FGHIJ",  
                       "KLMNO" } ;
```

Akan terjadi **ERROR!**

Contoh 6

Jika dideklarasikan:

```
char matriks[5][12] = { "Jakarta",  
                        "Bandung",  
                        "Surabaya",  
                        "Semarang",  
                        "Yogyakarta" };
```

Akan ditampilkan sebagai:

J	A	k	A	r	t	a	\0				
B	a	n	D	u	n	g	\0				
S	U	r	A	b	a	y	A	\0			
S	e	m	A	r	a	n	g	\0			
Y	O	g	Y	a	k	a	r	t	a	\0	

PROSES MATRIKS

Matriks

Program Proses_Matrik_BarisdemiBaris

KAMUS

```
#define M 2  
#define N 3  
int A[M][N];
```

ALGORITMA

```
For Baris ← 0 to M-1 do  
    For Kolom ← 0 to N-1 do  
        PROSES MATRIK  
    Endfor  
Endfor
```

Baris-demi-baris

1	2	3
4	5	6

```
#include <stdio.h>

void main() {
    int A[2][3] = {{1, 2, 3}, {4, 5, 6}};

    for (int baris=0; baris<2; baris++) {
        for (int kolom=0; kolom<3; kolom++) {
            printf("%d ", A[baris][kolom]);
        }
        printf("\n");
    }
}
```

PROSES MATRIKS

Matriks

Program Proses_Matrik_KolomdemiKolom

KAMUS

```
#define M 2
#define N 3
int A[M][N];
```

1	2	3
4	5	6

ALGORITMA

```
For Kolom ← 0 to N-1 do
    For Baris ← 0 to M-1 do
        PROSES MATRIK
    Endfor
Endfor
```

Kolom-demi-kolom

1	4
2	5
3	6

```
#include <stdio.h>
```

```
void main() {
```

```
    int A[2][3] = {{1, 2, 3}, {4, 5, 6}};
```

```
    for (int kolom=0; kolom<3; kolom++) {
```

```
        for (int baris=0; baris<2; baris++) {
```

```
            printf("%d ", A[baris][kolom]);
```

```
        }
```

```
        printf("\n");
```

```
    }
```

```
}
```

CONTOH PROSES

Matriks

ALGORITMA

```
For Baris  $\leftarrow$  0 to M-1 do  
  For Kolom  $\leftarrow$  0 to N-1 do  
    PROSES MATRIKS  
  Endfor  
Endfor
```

- ✓ Mengisi elemen matriks dengan 0 (inisialisasi)
- ✓ Mengisi elemen matriks dari piranti masukan
- ✓ Mencetak elemen matriks ke piranti keluaran



$A[\text{Baris}, \text{Kolom}] = 0$

Input $A[\text{Baris}, \text{Kolom}]$

Print $A[\text{Baris}, \text{Kolom}]$


INISIALISASI

Matriks

Algoritma:

```
For Baris = 0 to 1 do  
  For Kolom = 0 to 2 do  
    Matriks[Baris][Kolom] = 0  
  Endfor  
Endfor
```

	0	1	2
0	0	0	0
1	0	0	0



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

Matriks

Algoritma:

Indeks = 1

For Baris = 0 **to** 1 **do**

For Kolom = 0 **to** 2 **do**

Matriks[Baris][Kolom] = Indeks

Indeks = Indeks + 1

Endfor

Endfor

	0	1	2
0	1	2	3
1	4	5	6

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

Matriks

Algoritma:

Indeks = 1

For Baris = 0 to 1 do

For Kolom = 0 to 2 do

Matriks[Baris][Kolom] = Indeks

Indeks = Indeks+2

Endfor

Endfor

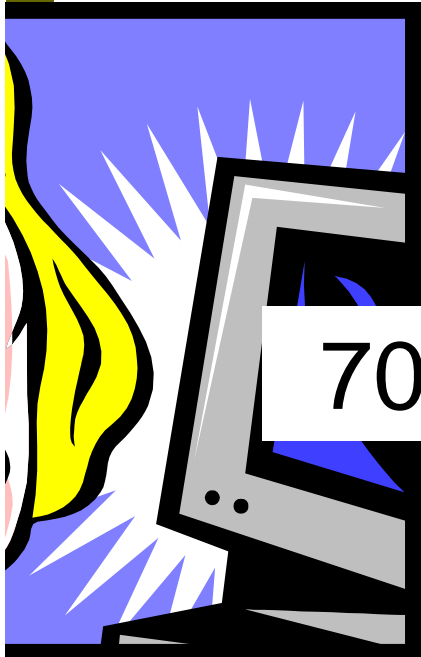
	0	1	2
0	1	3	5
1	7	9	11

CETAK ELEMEN

Matriks

Algoritma:

```
For Baris = 0 to 1 do  
    For Kolom = 0 to 2 do  
        print Matriks[Baris][Kolom]  
    Endfor  
Endfor
```



	0	1	2
0	18	3	69
1	24	8	70

PROSES LAINNYA

Matriks

```
For Baris = 0 to 1 do  
  For Kolom = 0 to 2 do  
    Matriks[Baris][Kolom] = ???  
    ???  
  Endfor  
Endfor
```

PROSES MATRIK DAPAT DIMODIFIKASI, sbb :

- **Menjumlahkan nilai** pada setiap baris
- **Membuat rata-rata** pada setiap baris atau setiap kolom
- Mencari nilai tertentu pada matrik
- **Menjumlahkan/Mengurangkan** dua buah matrik
- **Mengalikan** dua buah matrik, dan lain-lain

Menjumlahkan setiap baris

Matriks

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

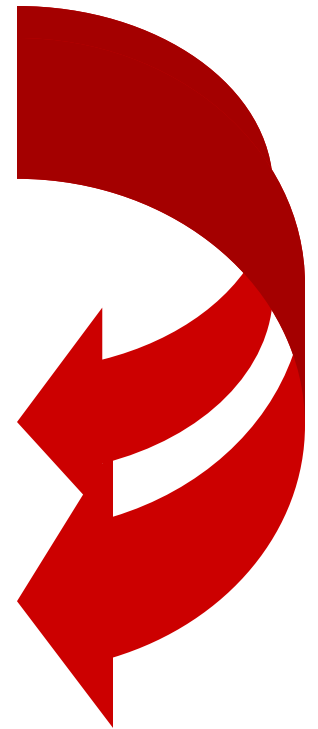
```
    TotalBaris = TotalBaris + Matriks[Baris][Kolom]
```

```
  Endfor
```

```
  Print Total Baris
```

```
Endfor
```

	0	1	2	
0	18	3	69	90
1	24	8	70	102



Tidak bisa kopi begitu saja

```
#include <stdio.h>
```

```
void main() {  
    int A[2][3] = {{1, 2, 3}, {4, 5, 6}};  
    int B[2][3];  
    B=A;  
}
```

Message

Compiling NONAME00.CPP:

Error NONAME00.CPP 6: Lvalue required in function main()

Warning NONAME00.CPP 8: 'B' is declared but never used in function main()

Warning NONAME00.CPP 8: 'A' is assigned a value that is never used in function main()

Mengkopikan array 2D ke 2D

```
for(int i=0;i<baris-1;i++){  
    for(int j=0;j<kolom-1;j++){  
        matrikshasil[i,j] = matriksasli[i,j];  
    }  
}
```

Kopi array 2D ke 1D

- Array $A[3][5] = \{\{1,2,3,4,5\},\{6,7,8,9,10\},\{11,12,13,14,15\}\}$;
- Array $B[15]$; (dari $3*5$)

Jawab-01. Cara-1

```
I=0;
while(I<=2)
{ J=0;
  while( J <= 4)
  { N=I*5 + J;
    B[N] = A[I][J];
    J++;
  }
  I++;
}
```

Dua instruksi ini dapat dijadikan satu tanpa menggunakan variabel N, menjadi :

$B[I*5 + J] = A[I][J];$

Perkembangan nilai I & J

I	J	N
0	0	0
	1	1
	2	2
	3	3
	4	4
1	0	5
	1	6
	2	7
	3	8
	4	9
2	0	10
	1	11
	2	12
	3	13
	4	14

$N = I*5 + J$

Kopi array 2D ke 1D (2)

```
#include <stdio.h>

void main() {
    int A[3][5] = {{1,2,3,4,5},{6,7,8,9,10},{11,12,13,14,15}};
    int B[15];
    int n=0;
    for (int i=0;i<3;i++) {
        for (int j=0;j<5;j++) {
            B[n]=A[i][j];
            n++;
        }
    }
    for (i=0;i<15;i++) {
        printf("%d ",B[i]);
    }
}
```

Jawab-01. Cara-2

```
N=0;
I=0;
while(I<=2)
{ J=0;
  while( J <= 4)
  { B[N] = A[I][J];
    N++;
    J++;
  }
  I++;
}
```

Kopi Array 1D ke 2D

Misalkan terdapat array 1 dimensi sebagai berikut: A[15]

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O

Disediakan array 2 dimensi B[3][5] sebagai berikut sehingga diharapkan array A dapat dikopikan ke array B seperti berikut ini:

		0	1	2	3	4
	0	A	B	C	D	E
B	1	F	G	H	I	J
	2	K	L	M	N	O

Contoh (6)

Ilustrasi proses :

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
masuk ke baris-0					masuk ke baris-1					masuk ke baris-2				

```
#include <stdio.h>

void main(){
    char A[3][5];
    char B[15]={'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O'};
    int n=0;
    for (int i=0;i<3;i++){
        for (int j=0;j<5;j++){
            A[i][j]=B[n];
            n++;
        }
    }
    for (i=0;i<3;i++){
        for (int j=0;j<5;j++){
            printf("%c ",A[i][j]);
        }
        printf("\n");
    }
}
```

Operasi Matriks

- Penginputan Matriks
- Pencetakan Matriks
- Transpose Matriks
- Ambil Diagonal Matriks
- Penjumlahan Matriks
- Pengurangan Matriks
- Perkalian Matriks

Input Matriks

```
#include <stdio.h>
void main() {
    int matriks[4][4]; //matriks 4 x 4
    for(int i=0;i<4;i++){
        for(int j=0;j<4;j++){
            printf("Elemen ke [%d,%d]",(i+1),(j+1));
            scanf("%d",matriks[i,j]);
        }
    }
}
```

Bagaimana kalau dibuat fungsi?

Cetak Matriks

```
#include <stdio.h>
void main(){
    int matriks[4][4]; //matriks 4 x 4
    for(int i=0;i<4;i++){
        for(int j=0;j<4;j++){
            printf("%d ",matriks[i,j]);
        }
        printf("\n");
    }
}
```

Bagaimana kalau dibuat fungsi?

Transpose

- Transpose adalah elemen baris matriks akan menjadi kolom matriks dan sebaliknya kolom matriks akan menjadi baris matriks.

Matriks Awal :

```
1 2 3
4 5 6
7 8 9
```

Hasil Transpose:

```
1 4 7
2 5 8
3 6 9
```

```
for(int i=0;i<4;i++){
    for(int j=0;j<4;j++){
        matriksHasil[i,j] = matriksA[j,i];
    }
}
```

Diagonal Matriks

□ Diagonal Matriks

- Matriks harus bujur sangkar!
- Diagonal adalah elemen matriks yang baris dan kolomnya sama!

Matriks awal:

```
1 2 3
4 5 6
7 8 9
```

Diagonal : 1 5 9

```
for(int i=0;i<4;i++){
    for(int j=0;j<4;j++){
        if (i==j){
            printf("%d",matriks[i,j]);
        }
    }
}
```


Penjumlahan Matriks

- Agar kedua matriks dapat dijumlahkan harus memiliki jumlah baris dan kolom yang sama.
- Inputkan matriks A dan matriks B
- Siapkan matriks C untuk menampung hasil penjumlahan matriks A dan B sesuai dengan elemen-elemennya.
- Elemen matriks A $[0,0]$ dijumlahkan dengan elemen matriks B $[0,0]$ juga dan disimpan di elemen matriks C $[0,0]$ juga. Begitu seterusnya
...

Penjumlahan Matriks (2)

Array A

1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5

+

Array B

2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5

```
for(int i=0;i<4;i++){  
    for(int j=0;j<4;j++){  
        matriksC[i,j] = matriksA[i,j] + matriks[i,j];  
    }  
}
```

Hasil

3	4	5	6	7
4	5	6	7	8
5	6	7	8	9
6	7	8	9	10

Pengurangan 2 buah Matriks

- Agar kedua matriks dapat dikurangkan harus memiliki jumlah baris dan kolom yang sama.
- Inputkan matriks A dan matriks B
- Siapkan matriks C untuk menampung hasil pengurangan matriks A dan B sesuai dengan elemen-elemennya.
- Elemen matriks A $[0,0]$ dikurangkan dengan elemen matriks B $[0,0]$ juga dan disimpan di elemen matriks C $[0,0]$ juga. Begitu seterusnya
...

Mengurangkan $C = A - B$

Dua buah Matriks

Algoritma:

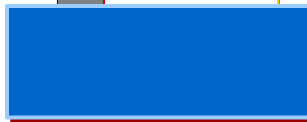
For Baris = 0 to 1 do

For Kolom = 0 to 2 do

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

Endfor
Endfor

	0	1	2
0	18	3	69
1	24	8	70



	0	1	2
0	1	2	3
1	4	5	6

Perkalian Matriks dengan Konstanta

Matriks awal:

1 2 3
3 4 5
6 7 8

Diakalikan dengan konstanta 3 menjadi:

3 6 9
9 12 15
18 21 24

Misal matriks sebagai berikut:

3	4	5	6	7
4	5	6	7	8
5	6	7	8	9
6	7	8	9	10

Perkalian Matriks dengan Konstanta (2)

```
for(int i=0;i<4;i++){  
    for(int j=0;j<4;j++){  
        matriksHasil[i,j] = 3*matriks[j,i];  
    }  
}
```

Matriks hasil:

9	12	15	18	21
12	15	18	21	24
15	18	21	24	27
18	21	24	27	30

Perkalian 2 buah Matriks

- Kedua matriks harus memiliki bentuk $m \times n$ untuk matriks A dan $n \times o$ untuk matriks B
- Sehingga matriks hasil akan memiliki bentuk $m \times o$
- Sehingga : $(m \times o) = (m \times n) \times (n \times o)$
- Inputkan matriks A
- Inputkan matriks B
- Tampung hasil perkalian matriks pada matriks C

Perkalian Matriks

Misalkan terdapat $A[2][3]$ dan $B[3][5]$ sebagai berikut:

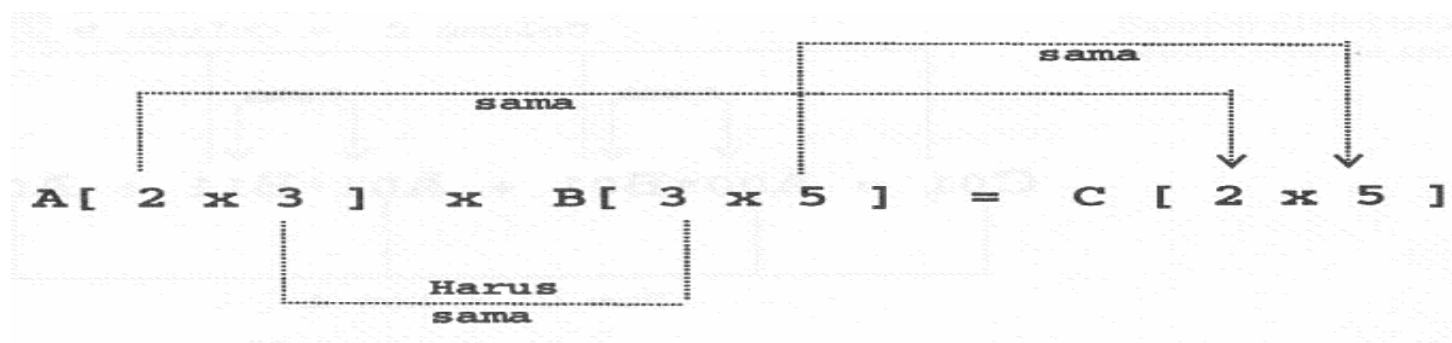
	0	1	2
A[0]	2	4	3
A[1]	3	2	5

	0	1	2	3	4
B[0]	3	2	5	7	4
B[1]	2	4	6	3	2
B[2]	3	3	2	5	4

Proses Perkalian

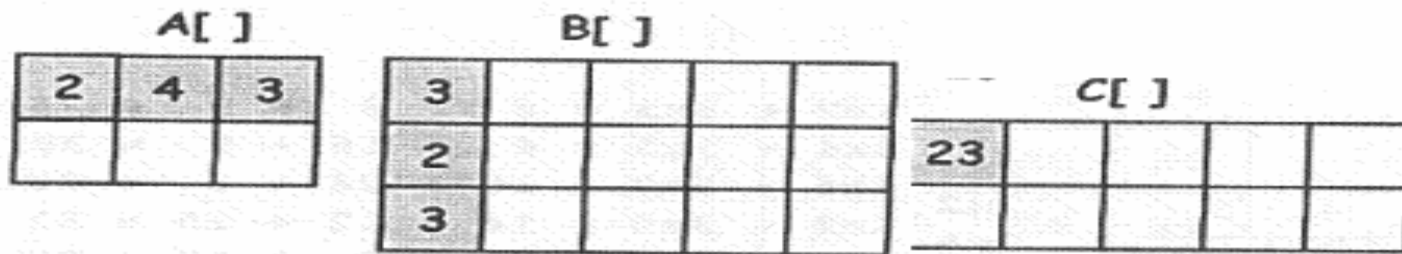
$$\begin{aligned} 2 \times 3 + 4 \times 2 + 3 \times 3 &= 6 + 8 + 9 = 23 \\ 2 \times 2 + 4 \times 4 + 3 \times 3 &= 4 + 16 + 9 = 29 \\ 2 \times 5 + 4 \times 6 + 3 \times 2 &= 10 + 24 + 6 = 40 \\ 2 \times 7 + 4 \times 3 + 3 \times 5 &= 14 + 12 + 15 = 31 \\ 2 \times 4 + 4 \times 2 + 3 \times 4 &= 8 + 8 + 12 = 28 \end{aligned}$$

$$\begin{aligned} 3 \times 3 + 2 \times 2 + 5 \times 3 &= 9 + 4 + 15 = 28 \\ 3 \times 2 + 2 \times 4 + 5 \times 3 &= 6 + 8 + 15 = 29 \\ 3 \times 5 + 2 \times 6 + 5 \times 2 &= 15 + 12 + 10 = 37 \\ 3 \times 7 + 2 \times 3 + 5 \times 5 &= 21 + 6 + 25 = 52 \\ 3 \times 4 + 2 \times 2 + 5 \times 4 &= 12 + 4 + 20 = 36 \end{aligned}$$

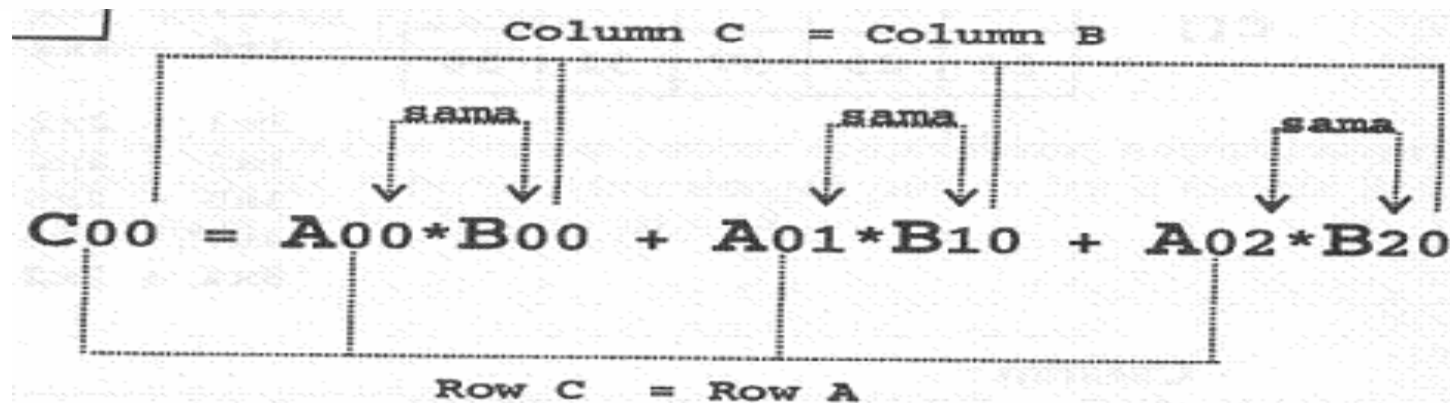


Proses Pertama

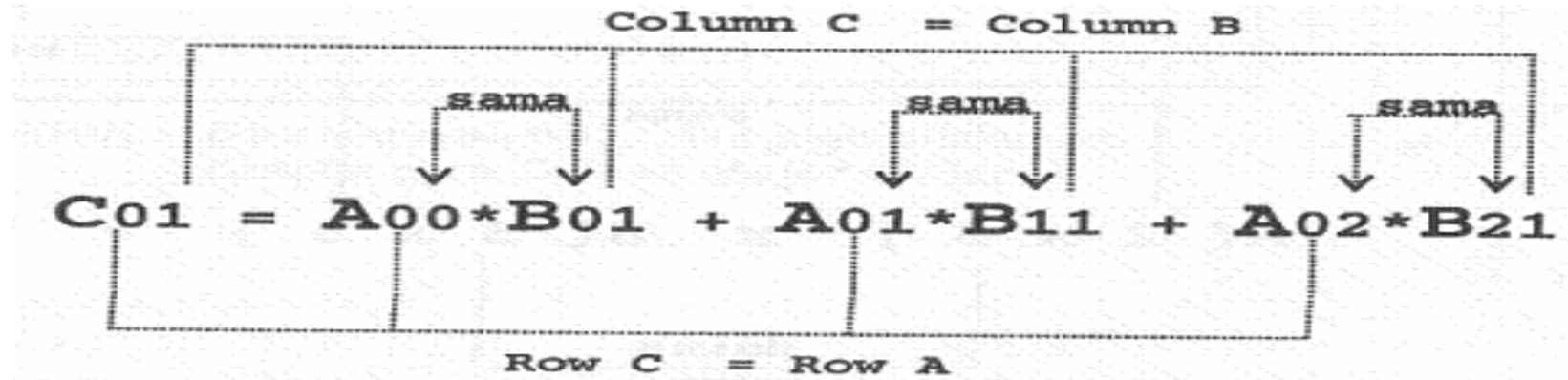
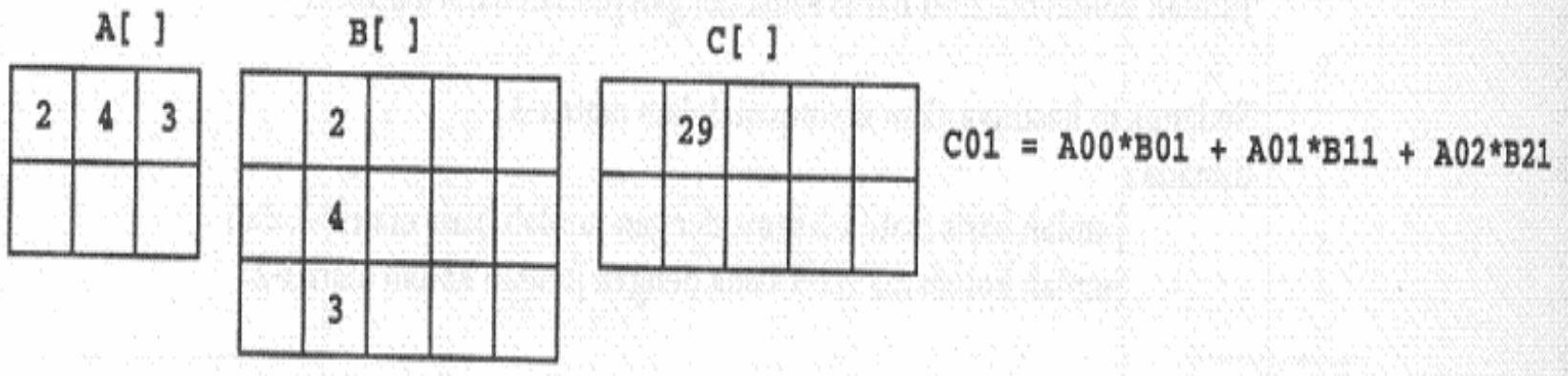
Urutan Proses Matriks:
Pertama

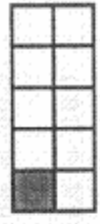


$$C_{00} = A_{00} * B_{00} + A_{01} * B_{10} + A_{02} * B_{20}$$
$$23 = 2 * 3 + 4 * 2 + 3 * 3$$



Proses kedua

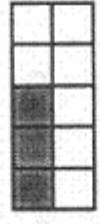




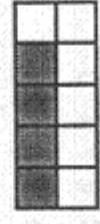
$$C_{00} = A_{00} \cdot B_{00} + A_{01} \cdot B_{10} + A_{02} \cdot B_{20}$$



$$C_{01} = A_{00} \cdot B_{01} + A_{01} \cdot B_{11} + A_{02} \cdot B_{21}$$



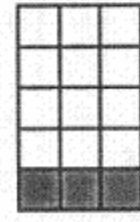
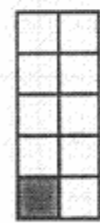
$$C_{02} = A_{00} \cdot B_{02} + A_{01} \cdot B_{12} + A_{02} \cdot B_{22}$$



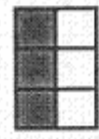
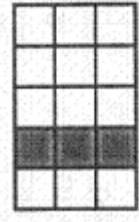
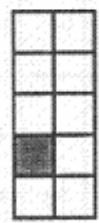
$$C_{03} = A_{00} \cdot B_{03} + A_{01} \cdot B_{13} + A_{02} \cdot B_{23}$$



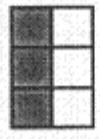
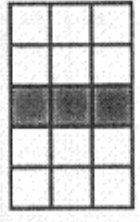
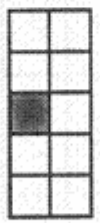
$$C_{04} = A_{00} \cdot B_{04} + A_{01} \cdot B_{14} + A_{02} \cdot B_{24}$$



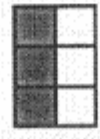
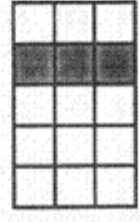
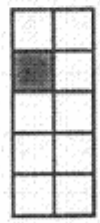
1.



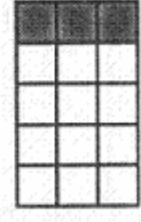
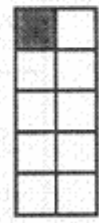
2.



3.



4.



5.

Sampai disini telah terisi satu baris pertama array C

$$C_{10} = A_{10} \cdot B_{00} + A_{11} \cdot B_{10} + A_{12} \cdot B_{20}$$

6.

$$C_{11} = A_{10} \cdot B_{01} + A_{11} \cdot B_{11} + A_{12} \cdot B_{21}$$

7.

$$C_{12} = A_{10} \cdot B_{02} + A_{11} \cdot B_{12} + A_{12} \cdot B_{22}$$

8.

$$C_{13} = A_{10} \cdot B_{03} + A_{11} \cdot B_{13} + A_{12} \cdot B_{23}$$

9.

$$C_{14} = A_{10} \cdot B_{04} + A_{11} \cdot B_{14} + A_{12} \cdot B_{23}$$

10.

Sampai disini telah terisi semua elemen array C

Rumus Umum

$$C_{I4} = A_{I0} * B_{04} + A_{I1} * B_{14} + A_{I2} * B_{24}$$

K dipakai untuk :
 COL. pada array A dan
 ROW pada array B
 dan untuk suatu nilai I & J,
 nilai K bergerak dari 0 s.d 2

J dipakai untuk :
 COL. pada array C dan B
 Untuk suatu nilai I,
 nilai J bergerak dari 0 s.d. 4

I dipakai untuk :
 ROW pada array C dan A
 I bergerak dari 0 s.d. 1

$$C [I, J] = \sum (A [I, K] * B [K, J])$$

Untuk suatu nilai I & J,
 nilai K bergerak dari 0
 sampai 2

Nilai I
 bergerak dari 0
 sampai 1

Sewaktu I masih = 0,
 J bergerak dari 0 sampai 4
 Sewaktu I = 1,
 J bergerak lagi dari 0 sampai 4

Program

Dengan program sebagai berikut:

```
for(i=0;i<2;i++){  
    for(j=0;j<5;j++){  
        C[i,j] = 0;  
        for(k=0;k<3;k++){  
            C[i,j] = C[i,j] + A[i,k] * B[k,j];  
        }  
    }  
}
```



Next: Manipulasi String
