

SIX

ARRAY (Con't) & Manipulasi Karakter-String

Pembahasan Soal-soal TTS

Array 2 dimensi dan Matriks

- **Array 2 dimensi numerik**

```
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**

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

- **Array 2 dimensi karakter**

```
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!**

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	

PENYALINAN ARRAY

Array 2 dimensi

Misalkan terdapat array 2 dimensi sebagai berikut A[3][5]

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

Cara menyalin array 2 dimensi ke matriks 2 dimensi lainnya:

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

Cara menyalin array 2 dimensi ke matriks 1 dimensi:

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

$N = I*5 + J$

Atau

Jawab-01. Cara-2

```

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

Sehingga pada array 1 dimensi menjadi:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Dari baris - 0					Dari baris - 1					Dari baris - 2				

Array 1 dimensi

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
1	F	G	H	I	J
2	K	L	M	N	O

Cara :

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				

Jawab-01. Cara-1

```

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

```

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

```

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

$$N = I*5 + J$$

Jawab-01. Cara-2

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

Suatu matriks

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

selanjutnya dapat dideklarasikan sebagai berikut:

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

atau diklarasikan dengan memberikan nilai elemennya sbb:

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

Selanjutnya larik 2 dimensi x dapat digambarkan sebagai berikut:

```
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
```

Operasi-operasi matriks

1. Menginputkan Matriks

Untuk setiap elemen matriks disimpan nilainya sesuai dengan tipe data yang dideklarasikan kepadanya.

Contoh program:

```
#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]);
    }
  }
}
```

```
}
```

2. Mencetak Matriks

Matriks dicetak per elemen dengan menggunakan looping.

```
#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");
    }
}
```

3. Perjumlahan Matriks

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

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

4. Pengurangan Matriks

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

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

-1	0	1	2	3
-2	-1	0	1	2
-3	-2	-1	0	1
-4	-3	-2	-1	0

5. Transpose matriks

- o 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
```

- o Siapkan matriks hasil untuk menampung hasil transpose

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

6. Ambil diagonal matriks

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]);
        }
    }
}
```

7. Menghitung max, min, rata, dan total

Misal matriks sebagai berikut:

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

```
for(int i=0;i<4;i++){
    for(int j=0;j<4;j++){
        total = 0;
        if (i==0 && j==0){
            maks = matriks[i,j];
            min = matriks[i,j];
        } else {
            if (maks<matriks[i,j]) maks = matriks[i,j];
            if (min>matriks[i,j]) min = matriks[i,j];
        }
        total += matriks[i,j];
    }
}
rata = total / (4*4);
```

Maka Total = 123
Maks = 10
Min = 3

8. Perkalian matriks dengan konstanta tertentu

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

```
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

9. Perkalian 2 matriks

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

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

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

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

Sedangkan Array C sebagai array hasil $C[2][5]$:

		1	2	3	4	5
C[]	1	23	29	40	31	28
	2	28	29	37	52	36

Dan proses perhitungannya adalah:

$$2 \times 3 + 4 \times 2 + 3 \times 3 = 6 + 8 + 9 = 23$$

$$2 \times 2 + 4 \times 4 + 3 \times 3 = 4 + 16 + 9 = 29$$

$$2 \times 5 + 4 \times 6 + 3 \times 2 = 10 + 24 + 6 = 40$$

$$2 \times 7 + 4 \times 3 + 3 \times 5 = 14 + 12 + 15 = 31$$

$$2 \times 4 + 4 \times 2 + 3 \times 4 = 8 + 8 + 12 = 28$$

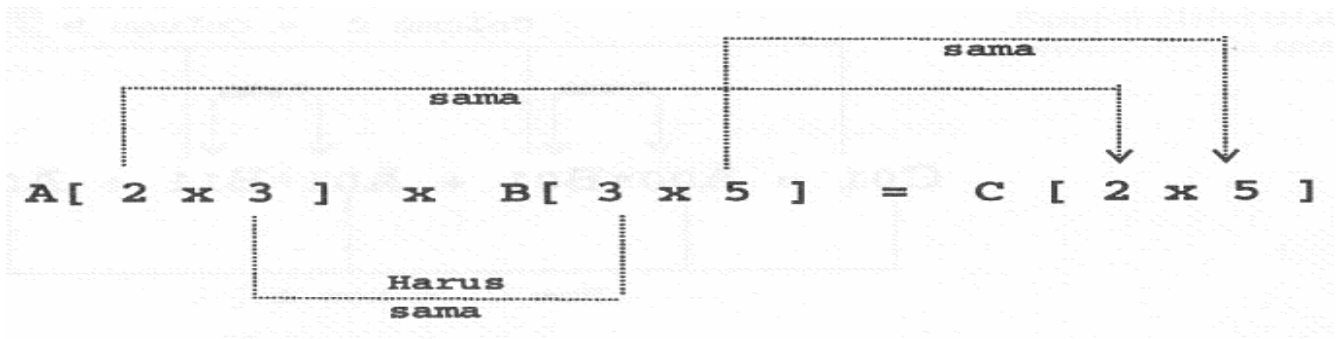
$$3 \times 3 + 2 \times 2 + 5 \times 3 = 9 + 4 + 15 = 28$$

$$3 \times 2 + 2 \times 4 + 5 \times 3 = 6 + 8 + 15 = 29$$

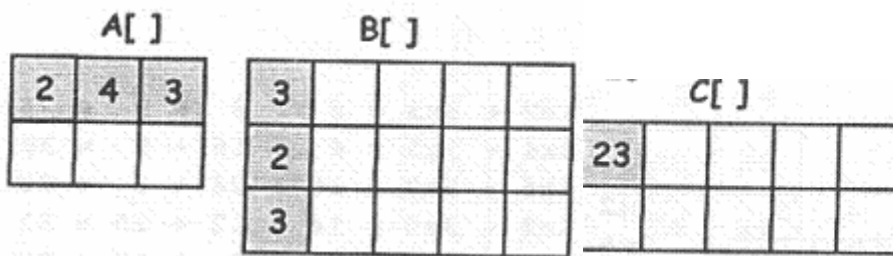
$$3 \times 5 + 2 \times 6 + 5 \times 2 = 15 + 12 + 10 = 37$$

$$3 \times 7 + 2 \times 3 + 5 \times 5 = 21 + 6 + 25 = 52$$

$$3 \times 4 + 2 \times 2 + 5 \times 4 = 12 + 4 + 20 = 36$$

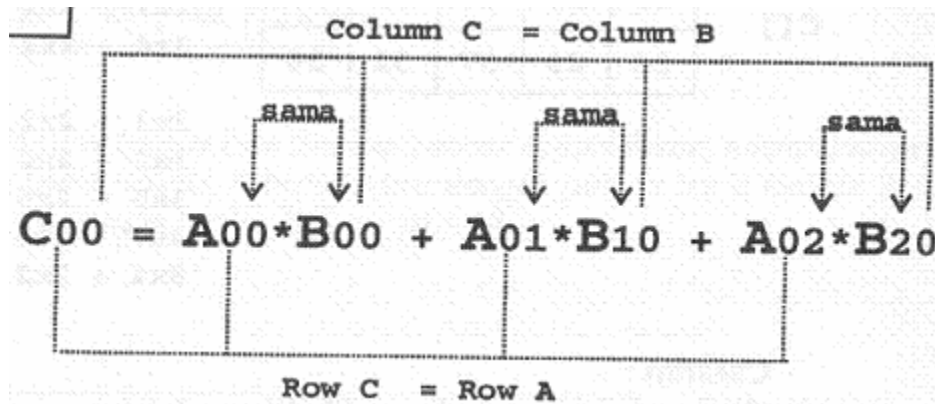


Urutan Proses Matriks:
Pertama

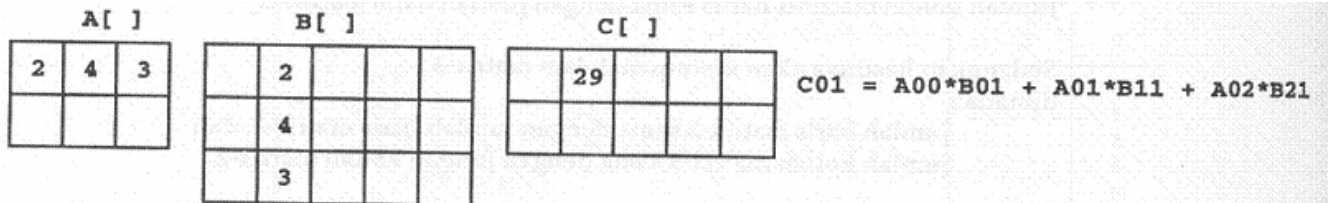


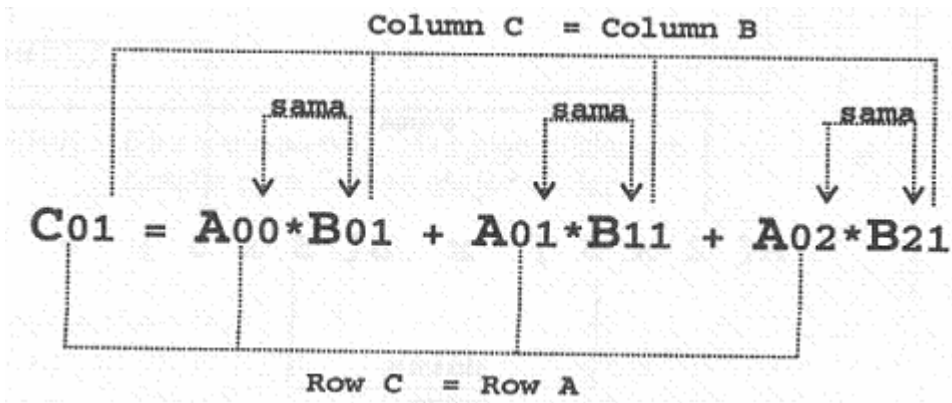
$$C_{00} = A_{00} \cdot B_{00} + A_{01} \cdot B_{10} + A_{02} \cdot B_{20}$$

$$23 = 2 \cdot 3 + 4 \cdot 2 + 3 \cdot 3$$



Kedua





Dan seterusnya:

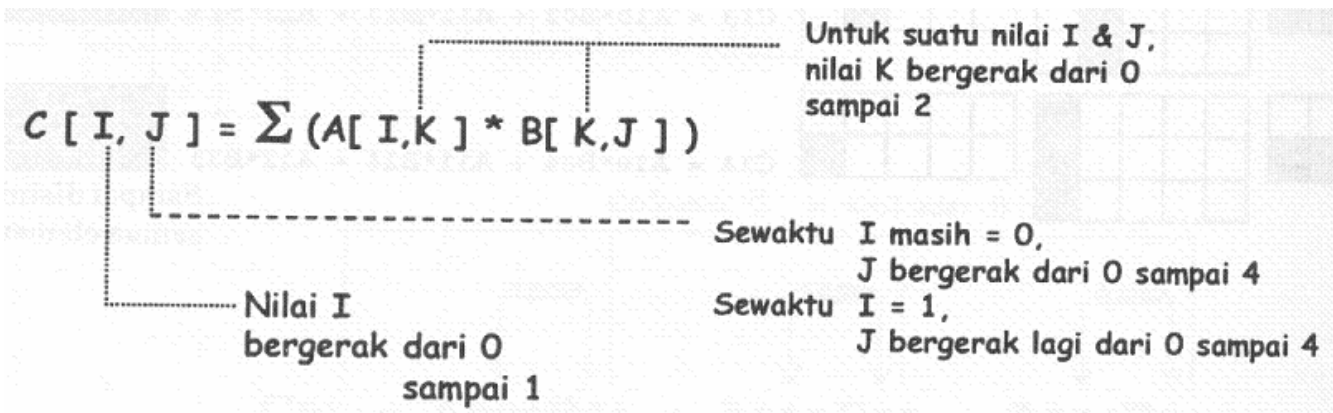
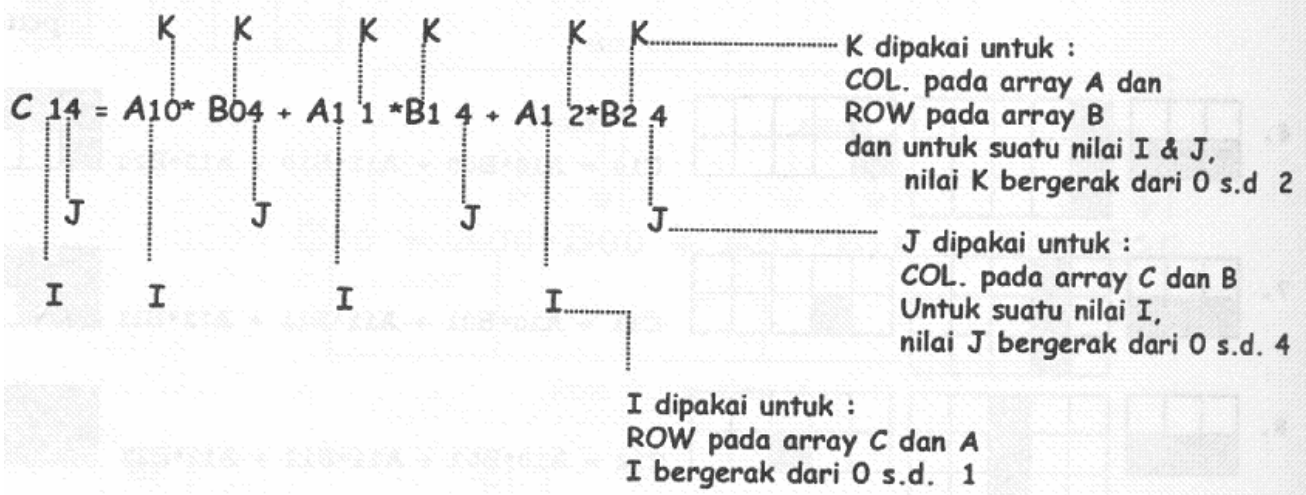
Hasil Lengkapnya adalah:

1.				$C_{00} = A_{00} \cdot B_{00} + A_{01} \cdot B_{10} + A_{02} \cdot B_{20}$	
2.				$C_{01} = A_{00} \cdot B_{01} + A_{01} \cdot B_{11} + A_{02} \cdot B_{21}$	
3.				$C_{02} = A_{00} \cdot B_{02} + A_{01} \cdot B_{12} + A_{02} \cdot B_{22}$	
4.				$C_{03} = A_{00} \cdot B_{03} + A_{01} \cdot B_{13} + A_{02} \cdot B_{23}$	
5.				$C_{04} = A_{00} \cdot B_{04} + A_{01} \cdot B_{14} + A_{02} \cdot B_{24}$	

Sampai disini telah terisi satu baris pertama array C

6.				$C_{10} = A_{10} \cdot B_{00} + A_{11} \cdot B_{10} + A_{12} \cdot B_{20}$	
7.				$C_{11} = A_{10} \cdot B_{01} + A_{11} \cdot B_{11} + A_{12} \cdot B_{21}$	
8.				$C_{12} = A_{10} \cdot B_{02} + A_{11} \cdot B_{12} + A_{12} \cdot B_{22}$	
9.				$C_{13} = A_{10} \cdot B_{03} + A_{11} \cdot B_{13} + A_{12} \cdot B_{23}$	
10.				$C_{14} = A_{10} \cdot B_{04} + A_{11} \cdot B_{14} + A_{12} \cdot B_{24}$	

Sampai disini telah terisi semua elemen array C



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];
    }
  }
}

```

KARAKTER

- Suatu nilai karakter mempunyai lebar nilai 1 byte.
- Konstanta karakter ditulis dalam tanda petik tunggal, sedang variabel karakter ditulis dengan menggunakan kata kunci **char**.
- Yang termasuk ke dalam karakter adalah huruf-huruf alfabet, tanda baca, angka '0', '1', ..., '9', dan karakter-karakter khusus seperti '&', '^', '%', '#', '@', dan sebagainya.
- Karakter kosong (*null*) adalah karakter yang panjangnya nol ,dan dilambangkan dengan " (petik kosong).
- Satu-satunya operasi yang dapat dilakukan terhadap tipe karakter adalah operasi perbandingan.

- Tipe data **char** diperlakukan seperti tipe data numerik.
- Operator perbandingan/ hubungan yang berlaku untuk tipe karakter adalah:

```

== (sama dengan)
!= (tidak sama dengan)
< (lebih kecil)
> (lebih besar)
>= (lebih besar atau sama dengan)
<= (lebih kecil atau sama dengan)

```

- Operasi dengan operator perbandingan menghasilkan nilai *boolean*. Misalnya, bila a dan b adalah variabel bertipe karakter, maka operasi a < b akan menghasilkan nilai benar atau salah bergantung pada keterurutan nilai a dan b pada sistem pengkodeannya (ASCII).
- Dalam kode ASCII 8 bit dikenal 256 macam karakter (0-255).
- Sedangkan kode ASCII 7 bit hanya mempunyai 128 macam karakter, dari nilai kode dalam desimal 0 sampai dengan 127.
- Kode ASCII dengan nilai kode 0 sampai dengan 31 dan 127 termasuk dalam status karakter-karakter kontrol yang tidak dapat dicetak dalam printer (*non-printable characters*)
- Karakter dengan kode ASCII 32 adalah karakter spasi.
- Karakter dengan kode ASCII 32 sampai 126 termasuk dalam status karakter-karakter yang dapat dicetak di printer

Simbol karakter ASCII:

0 :	18 : ↑	37 : %	56 : 8	75 : K	94 : ^	113 : q
1 : ☺	19 : !!	38 : &	57 : 9	76 : L	95 : _	114 : r
2 : ☹	20 : ¶	39 : '	58 : :	77 : M	96 : `	115 : s
3 : ♥	21 : §	40 : (59 : ;	78 : N	97 : a	116 : t
4 : ♦	22 : —	41 :)	60 : <	79 : O	98 : b	117 : u
5 : ♣	23 : ↓	42 : *	61 : =	80 : P	99 : c	118 : v
6 : ♠	24 : ↑	43 : +	62 : >	81 : Q	100 : d	119 : w
7 :	25 : ↓	44 : ,	63 : ?	82 : R	101 : e	120 : x
8 :	26 : →	45 : -	64 : @	83 : S	102 : f	121 : y
9 :	27 : ←	46 : .	65 : A	84 : T	103 : g	122 : z
10 :	28 : L	47 : /	66 : B	85 : U	104 : h	123 : {
	29 : ↔	48 : 0	67 : C	86 : V	105 : i	124 :
11 : ♂	30 : ▲	49 : 1	68 : D	87 : W	106 : j	125 : }
12 : ♀	31 : ▼	50 : 2	69 : E	88 : X	107 : k	126 : ~
13 :	32 :	51 : 3	70 : F	89 : Y	108 : l	127 : △
14 : 🎵	33 : !	52 : 4	71 : G	90 : Z	109 : m	
15 : ☀	34 : "	53 : 5	72 : H	91 : [110 : n	
16 : ▶	35 : #	54 : 6	73 : I	92 : \	111 : o	
17 : ◀	36 : \$	55 : 7	74 : J	93 :]	112 : p	

Nama Makro	Kelompok Kode ASCII untuk nilai benar	Keterangan
isascii(c)	0-127	Karakter berkode ASCII 0 sampai 127 (ASCII 127 bit)
iscntrl(c)	0-31 dan 127	Karakter-karakter kontrol
isspace(c)	9, 10, 13, 32	Karakter spasi, tab dan Enter (whitespace character)
isgraph(c)	33-126	Karakter-karakter yang dapat dicetak selain spasi
isprint(c)	33-126	Semua karakter yang dapat dicetak
ispunct(c)	33-47;58-64;91-996;123-126	Karakter-karakter yang dapat dicetak kecuali spasi, huruf dan angka
isalnum(c)	348-57;65-90;97-122	Karakter-karakter huruf dan angka
isalpha(c)	65-90	Karakter-karakter huruf
islower(c)	97-122	Karakter-karakter huruf kecil
isupper(c)	65-90	Karakter-karakter huruf besar
isdigit(c)	48-57	Karakter-karakter angka

- Program berikut menunjukkan penggunaan makro **isspace(c)**.
- Program berikut ini akan menerima data karakter yang dimasukkan berulang-ulang sampai dimasukkan nilai spasi, tab atau enter (*whitespace character*).

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

int main()
{
    int Karakter;
    do {
        Karakter = getche ();
        if(isspace(Karakter)) break;
    } while(1);
    return 0;
}
```

Contoh lain:

```
#include<conio.h>
#define Iisspace_j(c) ((c==' ' || c=='\r' || c=='\t')? 1: 0)

void main()
{
    int Karakter;
    do
    {
        Karakter = getche();
        if(Iisspace_j(Karakter)) break;
    }
    while (1);
}
```

Program diatas mendefinisikan suatu “makro” (semacam program kecil yang memiliki nama tertentu dan mengembalikan nilai tertentu. Makro bisa sudah disediakan oleh C atau dibuat sendiri) yang dibuat sendiri, bernama **Iisspace**.

Contoh lain

```
#include<stdio.h>
#include<conio.h>
#include<ctype.h>
void main()
{
    int Karakter;
    printf("Tekan Spasi, Enter atau Tab untuk keluar program....\n");
    do
    {
        Karakter = getche();
        if(isspace(Karakter)) break;
        if(!isalpha (Karakter))
            printf(" adalah angka\n");
        else
            printf(" bukan angka tetapihuruf\n");
    } while(1);
}
```

Jika program ini dijalankan dan ditekan tombol-tombol karakter '0','a' dan Enter, maka akan dihasilkan:

```
Tekan Spasi, Enter atau Tab untuk keluar program....
0 adalah angka
a bukan angka tetapi huruf
```

- Untuk mengkonversikan nilai karakter ada dua fungsi pustaka, yaitu fungsi pustaka **tolower()** dan fungsi pustaka **toupper()**.
- Sedangkan makro-makro untuk proses konversi ini bernama **_tolower(c)** dan **_toupper(c)**.
- Fungsi pustaka **tolower()** dan makro **_tolower(c)** digunakan untuk mengkonversikan dari nilai karakter huruf besar (*uppercase*) ke nilai karakter huruf kecil (*lowercase*)
- Sedangkan fungsi pustaka **toupper()** dan makro **_toupper(c)** digunakan untuk mengkonversikan dari nilai karakter huruf kecil (*lowercase*) ke nilai karakter huruf besar (*uppercase*)

```
#include<stdio.h>
#include<conio.h>
#include<ctype.h>
void main()
{
    int Karakter;
    printf("Tekan Spasi, Enter atau Tab untuk keluar program....\n");
    do {
        printf("Inputkan Karakter yang anda inginkan ?");
        Karakter = getche();
        if(isspace(Karakter)) break;
        if(isalpha(Karakter))
            if(isupper(Karakter))
                printf(" huruf kecilnya: %c\n",tolower(Karakter));
            else
                printf(" huruf besarnya: %c\n",toupper(Karakter));
        else
            printf(" bukan huruf\n");
    }
```

```

        printf(" adalah bukan huruf\n");
    } while(1);
}

```

Hasil eksekusi program tersebut apabila tombol-tombol karakter '9','R','y', dan Enter adalah:

```

Tekan Spasi, Enter atau Tab untuk keluar program....
Inputkan Karakter yang anda inginkan? 9 adalah bukan huruf
Inputkan Karakter yang anda inginkan? R huruf kecilnya adalah r
Inputkan Karakter yang anda inginkan? y huruf besarnya adalah Y

```

Berikut ini adalah contoh suatu makro bernama `Todigit_j(c)` yang dibuat untuk mengkonversikan nilai karakter ke nilai numerik.

```

#include<iostream.h>
#define Todigit_j(c) (c-'0')
void main()
{
    char Huruf;
    int Numerik;
    Huruf = getch();
    Numerik = Todigit_j(Huruf);
    printf("Huruf '%c' diubah ke digit menjadi %d",Huruf,Numerik);
}

```

Jika program ini dijalankan akan diperoleh hasil:

```
Huruf '7' diubah ke digit menjadi 7
```

STRING

- Nilai String adalah kumpulan dari nilai-nilai karakter yang berurutan dalam bentuk satu dimensi, nilai string ini haruslah ditulis didalam tanda petik dua (") misalnya: "*ini string*".
- Suatu nilai string disimpan di memori dengan diakhiri oleh nilai `'\0'` (*null*), misalnya nilai string "UKDW" disimpan dimemori dalam bentuk

U	K	D	W	'\0'
---	---	---	---	------

Dengan mengetahui nilai string diakhiri oleh nilai `'\0'`, maka akhir nilai dari suatu string dapat dideteksi. Berikut ini contoh program:

Contoh 1:

```

#include <stdio.h>
main()
{
    char string[100]="String";
    int K;
}

```



```

    for(K=0;string[K]!='\0';K++) printf("%c\n",string[K]);
}

```

Program ini jika dijalankan akan memunculkan hasil:

```

s
t
r
i
n
g

```

Contoh 2:

```

#include <stdio.h>
#include <ctype.h>
void main()
{
    char string[100]="StRiNg";
    int K;
    for(K=0;string[K]!='\0';K++) printf("%c\n",toupper(string[K]));
}

```

Program ini jika dijalankan akan memunculkan hasil:

```

S
T
R
I
N
G

```

- Bahasa C menggunakan fungsi-fungsi pustaka yang disediakan untuk mengoperasikan suatu nilai string yang dimasukkan dalam file header **string.h**
- Untuk menentukan panjang suatu nilai string, kita membutuhkan sebuah fungsi pustaka bernama **strlen()** yang berada juga di dalam file header string.h

```

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

void main()
{
    char a[100];
    clrscr();
    printf("Masukkan kalimat apapun yang anda sukai (max 100 huruf):");
    gets(a);
    printf("panjang huruf adalah: %d karakter\n",strlen(a));
    getch();
}

```

- Dalam bahasa C, untuk menyalin nilai suatu string tidak dapat langsung menuliskannya seperti halnya kompiler lain, sehingga proses menyalin atau

mengerjakan suatu nilai string ke variabel string yang lain diperlukan suatu fungsi pustaka yang bernama **strcpy()**.

```
#include <stdio.h>
#include <string.h>
void main()
{
    char a[100];
    char b[]="STRING";

    strcpy(a,b);
    printf("string pertama: %s\n",a);
    printf("string kedua : %s\n",b);
}
```

Program ini jika dijalankan akan memunculkan hasil:

```
String pertama: STRING
String kedua : STRING
```

Fungsi pustaka **strcat()** dipakai untuk menggabungkan nilai string.

```
#include<stdio.h>
#include<string.h>
void main()
{
    char string1[]="Kami kelompok ";
    char string2[]=" belajar algoritma dan pemrograman";
    strcat(string1,string2);
    printf("Jadi gabungannya adalah: %s\n",string1);
}
```

- Untuk membandingkan dua nilai string tidak bisa menggunakan operator hubungan, karena operator tersebut tidak untuk operasi string. Untuk membandingkan dua nilai string kita gunakan fungsi pustaka **strcmp()**
- Hasil perbandingannya akan menghasilkan **nilai lebih kecil dari nol** jika string pertama **lebih kecil dari string kedua**
- Hasil perbandingan akan **sama dengan nol** jika nilai string pertama **sama** dengan nilai string kedua
- Nilai **string pertama lebih besar dari nilai string kedua** menghasilkan perbandingan **lebih besar dari nol**

Berikut adalah contoh program perbandingan dua nilai string

```
#include<stdio.h>
#include<string.h>
void main()
{
    char string1[]="CDEFG";
    char string2[]="Cdefg";
    int Hasil;
    Hasil=strcmp(string1,string2);
    if(Hasil== 0)
        printf("String1 sama dengan String2\n");
    else
```

```

    if(Hasil<0)
        printf("String1 lebih kecil dari String2\n");
    else
        printf("String1 lebih besar dari String2\n");
}

```

Jika program di atas dijalankan akan menghasilkan:

```
String1 lebih kecil dari String2
```

- Dalam bahasa C disediakan suatu fungsi pustaka yaitu **strchr()** untuk mencari nilai suatu karakter yang ada di suatu string.
- Hasil dari fungsi ini adalah alamat letak dari karakter pertama di nilai string yang sama dengan karakter yang dicari.

Contoh:

```

#include <stdio.h>
#include <string.h>
void main()
{
    char string [20]="contoh program";
    char *hasil;
    hasil=strchr(string,'o');
    printf("%s\n",hasil);
}

```

- Fungsi **strchr()** di atas untuk mencari karakter "o", nilai karakter ini ada di nilai string yang dicari sehingga fungsi strchr() akan memberikan hasil alamat dari karakter "o" tersebut dan alamat ini kemudian disimpan di variabel pointer hasil
- Jika variabel hasil ditampilkan dengan menggunakan kode format untuk nilai string yaitu (%s), maka mulai dari alamat karakter tersebut sampai dengan akhir nilai stringnya akan ditampilkan sehingga hasil yang akan diperoleh yaitu: **"ontoh program"**
- Bahasa C tidak menyediakan suatu fungsi untuk membuang spasi pada suatu string, sehingga perlu fungsi-fungsi buatan sendiri
- Ada berbagai jenis fungsi untuk membuang spasi dalam suatu string yaitu fungsi yang hanya membuang nilai spasi di awal string, fungsi yang hanya membuang spasi di akhir string, dan fungsi yang membuang semua spasi yang ada di suatu string
- Dibawah ini akan dicontohkan suatu fungsi yang akan membuang semua spasi yang ada dalam suatu string

```

#include <stdio.h>
void buang_spasi(char *s);
void main()
{
    char string[20]="coba    program";
    buang_spasi(string);
    printf("%s\n",string);
}

```

```

void buang_spasi(char*s)
{
    int a,b;
    for(a=0; s[a];)

```

```

{
    if(s[a]==' ')
        for(b=a; s[b];b++)
            s[b]=s[b+1];
    if(s[a]!=' ') a++;
}
}

```

- Fungsi **buang_spasi()** digunakan untuk membuang nilai-nilai blank/(spasi) di awal suatu string.
- Proses ini dilakukan dengan menyeleksi karakter urutan pertama (indeks ke nol). Jika karakter ini adalah blank(spasi) maka karakter-karakter selanjutnya digeser ke arah kiri sehingga urutan karakternya diturunkan satu.
- Proses ini dilakukan berulang ulang sampai dijumpai karakter dengan indeks nol yang merupakan akhir dari string tersebut
- Pada kompiler C, tidak disediakan fungsi pustaka untuk mengambil bagian dari suatu string, maka pada fungsi berikut akan dibuat suatu fungsi untuk mengambil bagian dari suatu string, yang membutuhkan 4 buah argumen, yaitu berupa variabel string sebagai penerima hasil, variabel string sebagai yang akan disalin karakternya, nilai numerik posisi awal string dan nilai numerik akan banyaknya karakter yang diambil

Contoh program:

```

#include <stdio.h>
void Ambil_bagian ( char* x,char* y,int a, int b);
void main()
{
    char x[40];
    char y[]="universitaskristendutawacana"; //panggil procedure
    Ambil_bagian(x,y,5,9);
    printf("%s\n",x);
}

//x adalah string penampung hasil
//y adalah string sumber
//a adalah posisi mulai pengkopian
//b adalah posisi akhir pengkopian
void Ambil_bagian ( char* x,char* y,int a, int b)
{
    int p,q,r;
    for (r=0 ; y[r];r++); //r digunakan untuk mengetahui panjang string y
    if (a>=0 && a<=r-1) //kalau mulai masih diantara panjang string y
    {
        for (p=a, q=0; ( q< b && y[p]); p++,q++)
            x[q] = y[p];
        x[b] = '\0';
    }
    else
        x[0]='\0';
}

```

Hasil : rsitaskri

Fungsi ini mengambil karakter ke-5 sejumlah 9 karakter dari "universitaskristendutawacana".

PR1

Buatlah program penjumlahan, pengurangan, dan perkalian matriks lengkap dari mulai menginputkan data matriks pertama ($m \times n$), dimana baris dan kolom matriks pertama diinputkan user, dan matriks kedua ($n \times p$), dimana baris dan kolom matriks kedua juga diinputkan oleh user, kemudian nilai-nilai dari kedua matriks ditampilkan bersama dengan hasil penjumlahan, pengurangan, dan perkaliannya. Perhatikan juga error handling untuk baris dan kolom yang ada

PR2

Buatlah program yang meminta inputan data karakter dari user yang disimpan ke dalam array 1 dimensi.

Kemudian buatlah menu dan program untuk menu seperti berikut:

1. Input karakter
2. Cari karakter
3. Hapus karakter
4. Ubah karakter tertentu
5. Tampilkan karakter-karakter tersebut
6. Statistik karakter (jumlah vokal dan konsonan)
7. Exit

PR 1 dan PR 2 dikumpulkan ke email saya (anton@ukdw.ac.id) maksimal tanggal 29 maret 2006 pukul 13.30 dalam bentuk file ZIP yang berisi file source code dari pr1.cpp dan pr2.cpp yang telah dibuat dengan menggunakan Turbo C++ 4.5.

Jika file hasil sama akan diberi nilai 0 untuk kedua atau lebih pekerjaan yang sama tersebut!

NEXT WEEK: Procedure dan Function