

## # FIVE

### ARRAY / LARIK

#### Soal-soal minggu lalu:

1. Buat program untuk menghitung banyaknya bilangan genap dan ganjil serta total dari bilangan-bilangan genap dan ganjil yang ada dari n buah bilangan.
2. Buat program untuk menampilkan deret bilangan prima dari 1 sampai dengan n, dimana n adalah inputan dari user.
3. Mencetak Tulisan "<x> UKDW Almamaterku <y>" sebanyak 10 kali, dengan <x> adalah nomor urut (mulai 1 sampai 10), sedangkan <y> adalah nomor urut terbalik (mulai 10 sampai 1)
4. Buatlah tabel harga fotokopian dari 1 – 100 lembar, dimana harga perlembar adalah 80 rupiah!
5. Buatlah program untuk menampilkan deret sebagai berikut:

```
1    2    3    4    5
6    7    8    9    10
11   12   13   14   15
16   17   18   19   20
```

Dengan 1 buah loop saja!

6. Buatlah program dengan looping untuk menampilkan hasil seperti berikut:

P	Q	P or Q	P and Q	Not P	P xor Q
1	1	1	1	0	0
1	0	0	0	0	1
0	1	1	0	1	1
0	0	1	0	1	0

7. Buatlah program untuk menampilkan deret seperti:

Untuk n = 5

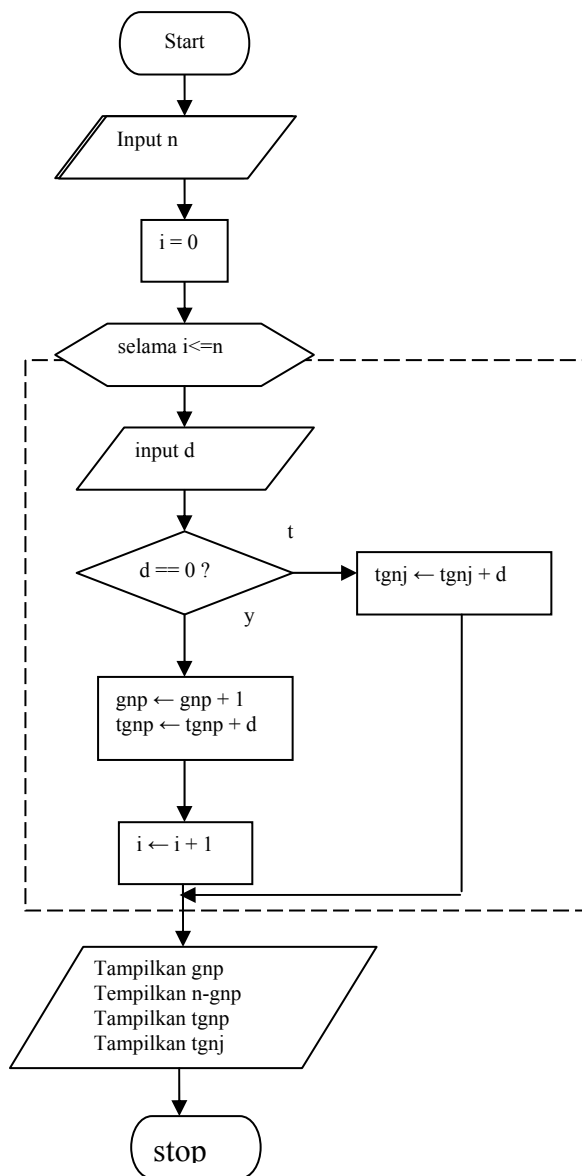
X O X O X

X O X O  
X O X  
X O  
X

8. Buatlah program untuk menampilkan bilangan fibonacci pada deret ke-n!  
Bilangan fibonacci adalah bilangan seperti: 1 1 2 3 5 8 13 ... dst  
Jadi jika inputan  $n = 7$ , maka hasil adalah 13

### Jawaban PR

1.



```

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

void main(){
    clrscr();
    int n,d,gnp=0,tgnp=0,tgnj=0;
    printf("masukkan n = ");scanf("%d",&n);
    for(int i=0;i<n;i++){
        printf("data ke-%d = ",(i+1));scanf("%d",&d);
        if(d % 2 == 0) {
            gnp++;
            tgnp += d;
        }else tgnj += d;
    }
    printf("jumlah bilangan genap = %d\n",gnp);
    printf("jumlah bilangan ganjil = %d\n",(n-gnp));
    printf("total bilangan genap = %d\n",tgnp);
    printf("total bilangan ganjil = %d\n",tgnj);
    getch();
}

```

2.

```

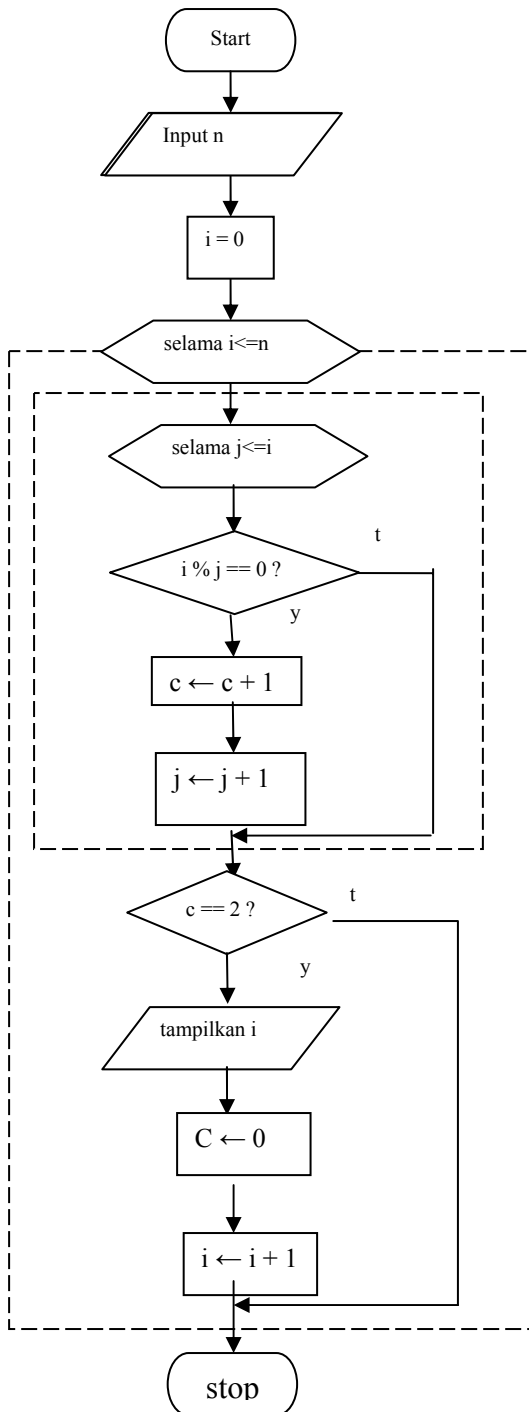
#include <stdio.h>
#include <conio.h>

void main(){
    clrscr();
    int n,c=0;
    printf("masukkan n = ");scanf("%d",&n);
    printf("deret bilangan prima adalah \n");
    for(int i=1;i<=n;i++){
        for(int j=1;j<=i;j++){
            if(i % j == 0){
                c++;
            }
        }
        if(c == 2) printf("%d\t",i);
        c=0;
    }
}

```

```
    getch();
```

```
}
```



3.

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

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

```

void main(){
    clrscr();
    for(int i=1;i<=10;i++){
        printf("%d UKDW Almamaterku %d\n",i,(11-i));
    }
    getch();
}

```

4.

```

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

```

```

void main(){
    clrscr();
    printf("Lembar\tHarga\n");
    printf("-----\n");
    for(int i=1;i<=100;i++){
        printf("%d\t%d\n",i,(i*80));
    }
    getch();
}

```

5.

```

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

```

```

void main(){
    int i,j;
    for(i=1;i<=20;i++){
        if(i % 5 == 0)
            printf("%d\n",i);
        else
            printf("%d\t",i);
    }
}

```

6.

```

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

```

```

void main(){
    int i,j;
    printf("P\tQ\tP or Q\tP and Q\tNot P\tP xor Q\n");
    printf("=====\n");
    for(i=1;i>=0;i--){
        for(j=1;j>=0;j--){
            printf("%d\t%d\t%d\t%d\t%d\t%d\n",i,j,(i || j),(i && j),!i,(i ^
j));
        }
    }
}

```

7.

```

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

```

```

void main(){
    int i,j,n;
    printf("masukkan n = ");scanf("%d",&n);
    for(i=1;i<=n;i++){
        for(j=1;j<=((n+1)-i);j++){
            if(j % 2 == 0) printf("O "); else printf("X ");
        }
        printf("\n");
    }
}

```

8.

```

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

```

```

void main(){
    int i,j,n,f=1,f2=1, fibo;
    printf("masukkan n = ");scanf("%d",&n);
    printf("Deret Fibonacci -> 1 ");
    for(i=1;i<=n-2;i++){
        printf("%d ",f2);
        fibo=f+f2;
    }
}

```

```

        f = f2;
        f2 = fibo;
    }
    printf("%d\n",fibo);
    printf("jawab = %d",fibo);
}

```

## PENDAHULUAN

- Selama ini kita menggunakan satu variabel untuk menyimpan 1 buah nilai dengan tipe data tertentu.
- Misalnya :  

```
int a1, a2, a3, a4, a5;
```

Deklarasi variabel diatas digunakan untuk menyimpan 5 data integer dimana masing-masing variabel diberi nama a1, a2, a3, a4, dan a5.
- Jika kita memiliki 10 data, 100 data integer bahkan mungkin data yang ingin kita proses tidak kita ketahui atau bersifat dinamis? Kita tidak mungkin menggunakan variabel seperti diatas.
- Di dalam C dan pemrograman yang lain, terdapat suatu fasilitas untuk menyimpan data-data yang bertipe data sama dengan suatu nama tertentu.

## DEFINISI ARRAY

- Array adalah suatu tipe data terstruktur yang berupa sejumlah data sejenis (bertipe data sama) yang jumlahnya tetap dan diberi suatu nama tertentu.
- Elemen-elemen array tersusun secara sekuensial di dalam memori sehingga memiliki alamat yang berdekatan.
- Array dapat berupa array 1 dimensi, 2 dimensi, bahkan n-dimensi.
- Elemen-elemen array bertipe data sama tapi bisa bernilai sama atau berbeda-beda.

<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	indeks
<b>8</b>	<b>10</b>	<b>6</b>	<b>-2</b>	<b>11</b>	<b>7</b>	<b>1</b>	<b>100</b>	value
<b>ffea</b>	<b>ffeb</b>	<b>ffec</b>	<b>ffed</b>	<b>ffef</b>	<b>fffa</b>	<b>fffb</b>	<b>fffc</b>	alamat

## CARA PENGAKSESAN ELEMEN ARRAY

- Elemen-elemen array dapat diakses oleh program menggunakan suatu indeks tertentu
- Pengaksesan elemen array dapat dilakukan berurutan atau random berdasarkan indeks tertentu secara langsung.
- Pengisian dan pengambilan nilai pada indeks tertentu dapat dilakukan dengan mengeset nilai atau menampilkan nilai pada indeks yang dimaksud.
- Dalam C, tidak terdapat error handling terhadap batasan nilai indeks, apakah indeks tersebut berada di dalam indeks array yang sudah didefinisikan atau belum. Hal ini merupakan tanggung jawab programmer. Sehingga jika programmer mengakses indeks yang salah, maka nilai yang dihasilkan akan berbeda atau rusak karena mengakses alamat memori yang tidak sesuai.

## DEKLARASI ARRAY 1 DIMENSI

`tipe_data nama_var_array[ukuran];`

`tipe_data` : menyatakan jenis tipe data elemen larik (int, char, float, dll)

`nama_var_array` : menyatakan nama variabel yang dipakai.

`ukuran` : menunjukkan jumlah maksimal elemen larik.

### Contoh:

```
char huruf[9];
int umur[10];
int kondisi[2] = {0,1}
int arr_dinamis[] = {1,2,3}
```

`char huruf[9]:` berarti akan memesan tempat di memori komputer sebanyak 9 tempat dengan indeks dari 0-8, dimana semua elemennya bertipe data karakter semuanya. Kalau satu karakter berukuran 1 byte, berarti membutuhkan memori sebesar 9 byte.

`int umur[10]:` berarti akan memesan tempat di memori komputer sebanyak 10 tempat dengan indeks dari 0-9, dimana semua elemennya bertipe data integer semuanya. Kalau satu integer berukuran 4 bytes, berarti membutuhkan memori sebesar  $4 \times 10 = 20$  bytes.

`int kondisi[2]:` berarti akan memesan tempat di memori komputer sebanyak 2 tempat dengan indeks 0-1, dimana semua elemennya bertipe data integer semuanya. Dan pada contoh di atas isi elemen-elemennya yang



sebanyak 2 buah diisi sekaligus (diinisialisasi) yaitu pada elemen kondisi[0] bernilai 0, dan elemen kondisi[1] bernilai 1.

`int arr_dinamis[]`:berarti mendeklarasikan array dengan ukuran maksimum array tidak diketahui, namun ukuran tersebut diketahui berdasarkan inisialisasi yaitu sebanyak 3 elemen, yang isinya 1,2, dan 3.

Kita tidak dapat mendeklarasikan array dinamis tanpa inisialisasi.

## **PENJELASAN ARRAY 1 DIMENSI**

- Tanda [] disebut juga “elemen yang ke- „. Misalnya kondisi[0] berarti elemen yang ke nol.
- Array yang sudah dipesan, misalnya 10 tempat tidak harus diisi semuanya, bisa saja hanya diisi 5 elemen saja, baik secara berurutan maupun tidak. Namun pada kondisi yang tidak sepenuhnya terisi tersebut, tempat pemesanan di memori tetap sebanyak 10 tempat, jadi tempat yang tidak terisi tetap akan terpesan dan dibiarkan kosong.

## **BERBAGAI CONTOH ARRAY 1 DIMENSI**

### **Contoh 1 (variabel array dan variabel biasa)**

```
//Dengan variabel biasa:  
int x1=3,x2=5,x3=2,x4=7,x5=9;  
  
//Dengan larik:  
int x[5]={3,5,2,7,9};
```

Bagaimana jika kita ingin menghitung total dari variabel biasa?

```
total = x1 + x2 + x3 + x4 + x5;
```

### **Contoh 2 (menginputkan dan menampilkan array)**

```

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

void main()
{ int nilai[5], x;
  clrscr();

  printf("Memasukkan nilai : \n");
  for(x=0;x<5;x++)
  {
    printf("Nilai Angka : "); scanf("%d",&nilai[x]);
  }
  printf("\n");

  printf("Membaca nilai : \n");
  for(x=0;x<5;x++)
  {
    printf("Nilai Angka : %d",nilai[x]);
  }

  getch();
}

```

```

C:\ Command Prompt (2) - tc
Memasukkan nilai :
Nilai Angka 4
Nilai Angka 7
Nilai Angka 3
Nilai Angka 9
Nilai Angka 6

Membaca nilai :
Nilai Angka 4
Nilai Angka 7
Nilai Angka 3
Nilai Angka 9
Nilai Angka 6

```

### Contoh 3 (manipulasi array 1 dimensi)

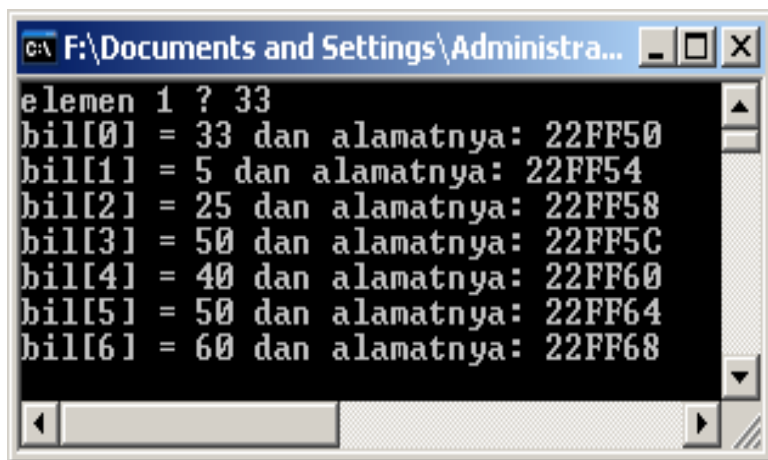
```

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

int main(){
  int bil[7],i;
  printf("elemen 1 ? ");scanf("%d",&bil[0]);
  bil[1] = 5;
  bil[2] = bil[1] + 20;
  for(i=4;i<7;i++) bil[i] = i*10;
  bil[3] = bil[bil[1]];
  for(i=0;i<7;i++) printf("bil[%d] = %d dan alamatnya: %X\n",i,bil[i],&bil[i]);
  getch();
  return 0;}

```

**Hasilnya:**



```
C:\F:\Documents and Settings\Administra...
elemen 1 ? 33
bil[0] = 33 dan alamatnya: 22FF50
bil[1] = 5 dan alamatnya: 22FF54
bil[2] = 25 dan alamatnya: 22FF58
bil[3] = 50 dan alamatnya: 22FF5C
bil[4] = 40 dan alamatnya: 22FF60
bil[5] = 50 dan alamatnya: 22FF64
bil[6] = 60 dan alamatnya: 22FF68
```

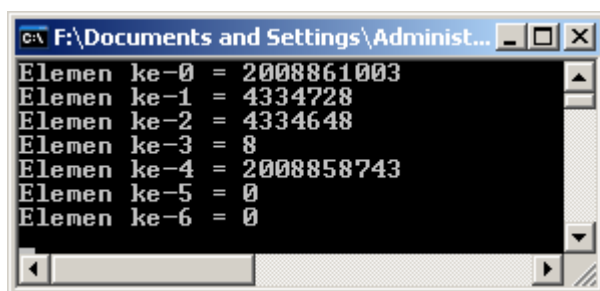
Terlihat bahwa alamat array berurutan dengan jarak antar alamat adalah 4 bytes (integer berukuran 4 bytes)

#### Contoh 4 (tanpa inisialisasi langsung ditampilkan)

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

int main(){
    int bil[7]; //tanpa inisialisasi
    for(int i=0;i<7;i++){
        printf("Elemen ke-%i = %d\n",i,bil[i]);
    }
    getch();
    return 0;
}
```

Hasilnya:



```
C:\F:\Documents and Settings\Administ...
Elemen ke-0 = 2008861003
Elemen ke-1 = 4334728
Elemen ke-2 = 4334648
Elemen ke-3 = 8
Elemen ke-4 = 2008858743
Elemen ke-5 = 0
Elemen ke-6 = 0
```

#### Contoh 5 (inisialisasi dengan 0)



```

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

int main(){
    int bil[7] = {0}; //inisialisasi 0
    for(int i=0;i<7;i++){
        printf("Elemen ke-%i = %d\n",i,bil[i]);
    }
    getch();
    return 0;
}

```

Hasilnya:

```

C:\ F:\Documents a...
Elemen ke-0 = 0
Elemen ke-1 = 0
Elemen ke-2 = 0
Elemen ke-3 = 0
Elemen ke-4 = 0
Elemen ke-5 = 0
Elemen ke-6 = 0

```

Contoh 6 (inisialisasi hanya 2 elemen pertama)

```

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

int main(){
    int bil[7] = {2,5};
    for(int i=0;i<7;i++){
        printf("Elemen ke-%i = %d\n",i,bil[i]);
    }
    getch();
    return 0;
}

```

Hasilnya:

```

C:\ F:\Documents and...
Elemen ke-0 = 2
Elemen ke-1 = 5
Elemen ke-2 = 0
Elemen ke-3 = 0
Elemen ke-4 = 0
Elemen ke-5 = 0
Elemen ke-6 = 0

```

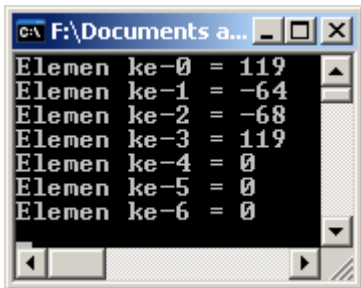
Contoh 7 (karakter yang tidak diinisialisasi)



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

int main(){
    char h[5];
    for(int i=0;i<5;i++){
        printf("Elemen ke-%i = %c\n",i,h[i]);
    }
    getch();
    return 0;
}
```

Hasilnya:



### ARRAY DIMENSI DUA

- 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

### ILUSTRASI ARRAY 2 DIMENSI

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 BENTUK UMUM 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} };
```

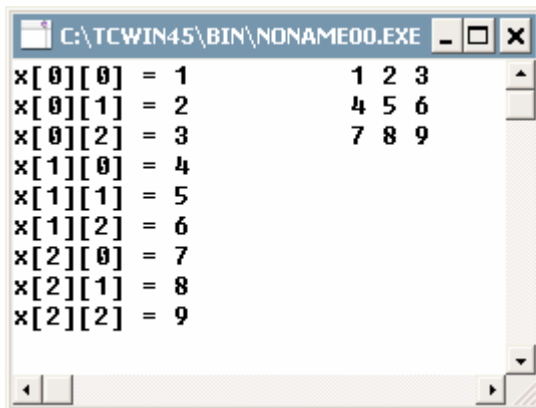
## Contoh:

```
#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();
}
```

## Tampilan program:



```
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
```

Larik Dimensi Dua dapat mewakili bentuk suatu matriks, contoh:

Suatu matriks

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

selanjutnya dapat dideklarasikan sebagai berikut:

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

atau dideklarasikan 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
```

## Larik String

- Tipe data string pada bahasa C merupakan kumpulan dari tipe data char. Nilai dari string tunggal dapat dibuat dari larik karakter berdimensi satu.
- Berarti larik string tunggal, dapat dibuat dari larik karakter berdimensi satu, dan larik string dimensi satu dapat dibentuk dari larik karakter berdimensi dua.
- Dan untuk larik string berdimensi X, dapat dibuat dari larik karakter berdimensi X+1.

## Bentuk umum:

Deklarasi suatu string (array of character) di C:

`char nama[50];` berarti mendeklarasikan array dimensi satu yang isinya elemen bertipe data karakter semua sebanyak 50 (dari 0-49)

Deklarasi array of string (jadi array berdimensi 2):

`char nama[10][50];` berarti mendeklarasikan array of string yang berjumlah 10 data bertipe string (array of character), dimana masing-masing string mampu menampung karakter maksimal sebanyak 50 karakter.

Contoh:

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

void main()
{
    int x,y;
    char Bulan[12][3] = {"Jan", "Peb", "Mar", "Apr", "Mei", "Jun",
                        "Jul", "Ags", "Sep", "Okt", "Nop", "Des"};

    clrscr();
    for(x=0;x<12;x++)
    {
        for(y=0;y<3;y++)
            printf ("%c",Bulan[x][y]);
        printf(" ");
    }
    getch();
}
```

Hasilnya:



```
Command Prompt (2) - tc
Jan Peb Mar Apr Mei Jun Jul Ags Sep Okt Nop Des _
```

### Contoh 2:

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

void main()
{
    int x,y;
    char Bulan[12][4] = {"Jan", "Peb", "Mar", "Apr", "Mei", "Jun",
                        "Jul", "Ags", "Sep", "Okt", "Nop", "Des"};

    clrscr();
    for(x=0;x<12;x++)
    {
        printf ("%s ",Bulan[x]);
    }
    getch();
}
```

Hasilnya:



```
Command Prompt (2) - tc
Jan Peb Mar Apr Mei Jun Jul Ags Sep Okt Nop Des _
```

### Contoh 3:

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

void main()
{
    int x,y;
    char hari[7][10] = {"Minggu", "Senin", "Selasa", "Rabu", "Kamis", "Jumat",
                       "Sabtu"};

    clrscr();
    for(x=0;x<7;x++)
    {
        printf ("%s ",hari[x]);
    }
    getch();
}
```

Hasilnya:





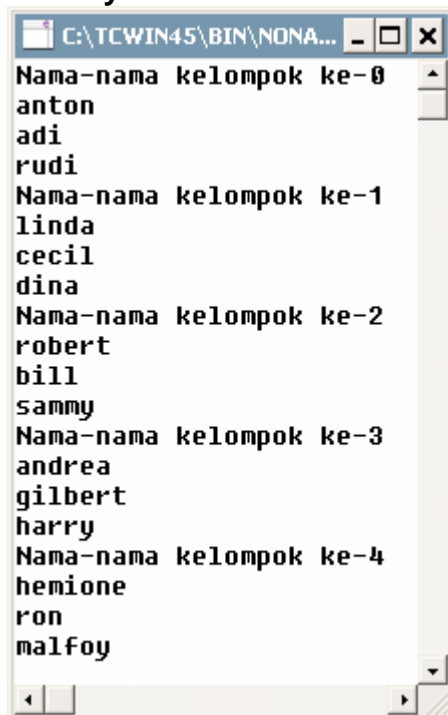
#### Contoh 4:

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

void main()
{
    int x,y;
    char nama[5][3][15] = {"anton", "adi", "rudi"},
        {"linda", "cecil", "dina"},
        {"robert", "bill", "sammy"},
        {"andrea", "gilbert", "harry"},
        {"hemione", "ron", "malfoy"};

    clrscr();
    for(x=0;x<5;x++)
    {
        printf("Nama-nama kelompok ke-%d\n",x);
        for(y=0;y<3;y++)
            printf ("%s\n",nama[x][y]);
    }
    getch();
}
```

#### Hasilnya:



## Soal-soal Latihan

1. Buatlah program untuk memasukkan n buah nilai dalam array integer dan kemudian Tampilkan kembali data-data array tersebut, jumlah totalnya dan rata-rata-nya!
2. Buatlah program untuk menganalisa sebuah string inputan yang diinputkan oleh user dan kemudian tampilkan:
  - o Berapa jumlah karakter yang berupa karakter vokal (a,i,u,e,o)
  - o Berapa jumlah karakter yang non vokal dan karakter-karakter lainnya
3. Kembangkan soal 3 untuk menghitung berapa jumlah vokal "A", "I", "U", "E", dan "O" masing-masing!
4. Buatlah program untuk membalik kata, misalnya kata "anton" akan menjadi "notna" dengan menggunakan array of char!
5. Buatlah program untuk menyalin isi array A ke array B yang kosong dengan jumlah elemen sama.

### Ilustrasinya:

```
Array: int A[5];
Indeks    0 1  2 3 4
Nilai     5 10 6 0 4
```

```
Array: int B[5];
Indeks    0 1  2 3 4
Nilai     _ _  _ _ _
```

Kemudian array B akan berisi sama dengan isi array A:

```
Array: int B[5];
Indeks    0 1  2 3 4
Nilai     5 10 6 0 4
```

6. Kembangkan soal 5, dimana array B akan berisi elemen-elemen array A tapi dengan posisi terbalik, data pertama di array A akan berada di data terakhir di array B!

```
Array: int B[5];
Indeks    0 1 2  3 4
Nilai     4 0 6 10 5
```

7. Kembangkan soal no-5, elemen B diisi dengan data yang dari array A yang nilainya berupa bilangan genap dan lebih besar dari nol pada indeks yang sama dengan array A!

```
Array: int B[5];
Indeks    0 1  2 3 4
Nilai     _ 10 6 _ 4
```

8. Kembangkan soal no-7 dimana array B akan diisi dengan data dari array A yang nilainya berupa bilangan genap, lebih besar dari nol, tapi indeksnya berurutan!

```
Array: int B[5];
Indeks    0  1 2 3 4
Nilai     10 6 4 _ _
```

**AFTER TTS:** Array 2 dimensi dan manipulasi String!