

Algoritma & Pemrograman #9



by antonius rachmat c, s.kom

Latihan

- ❑ Buatlah fungsi untuk menghitung frekuensi kemunculan huruf dalam sebuah kalimat!
 - Misal `int hitung_frekuensi(string input, huruf)`
- ❑ Buatlah fungsi untuk menentukan bilangan terkecil dari n buah bilangan yang diinputkan
 - Misal `int terkecil()`
- ❑ Buatlah fungsi untuk mengubah nilai ke huruf (A, B, C, D, dan E)
 - `char nilai2huruf(int angka)`
 - Buatlah fungsi untuk mengubah nilai huruf ke bobotnya
 - ❑ `int nilai2bobot(char huruf)`

-
- ❑ Buatlah fungsi untuk mengubah jam, menit, detik menjadi detik!
 - ❑ Buatlah fungsi untuk menjumlahkan deret: $1+3+5+7+\dots +n$.
 - ❑ Buatlah fungsi untuk mengetahui kuadran suatu koordinat!
 - ❑ Buatlah fungsi untuk menyederhanakan b/c menjadi $A_{b/c}$

Review Fungsi

- ▣ Argumen fungsi: formal dan aktual
- ▣ Fungsi by Value
- ▣ Fungsi by Reference (?)

Pendahuluan

- ❑ Selama ini kita menggunakan satu variabel untuk menyimpan 1 buah nilai dengan tipe data tertentu.
 - `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.
 - Bagaimana jika kita ingin menghitung total dari variabel biasa?
$$\text{total} = x1 + x2 + x3 + x4 + x5 + \dots + xn;$$

Array

- ❑ Di dalam C dan pemrograman yang lain, terdapat suatu fasilitas untuk menyimpan data-data yang bertipe data sama dengan suatu nama tertentu = ARRAY/LARIK
- ❑ 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.

Bentuk Array

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

Array (2)

- ❑ 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

```
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.

- Tipe data sejenis
- Ada indeks yang teratur dan berurutan
- Bersifat statis, harus diketahui ukurannya terlebih dahulu

Contoh dan Arti

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.

Contoh dan Arti (2)

Contoh:

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

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 Lebih Lanjut

- ❑ 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.

CONTOH PROSES

Array / Larik

ALGORITMA

For Indeks \leftarrow 1 to N do

PROSES LARIK

Endfor

✓ Mengisi elemen larik dengan 0
(inisialisasi)

✓ Mengisi elemen larik dari piranti
masukan

✓ Mencetak elemen larik ke piranti
keluaran

A[Indeks]=0

Input A[Indeks]

Print A[Indeks]

INISIALISASI

Array / Larik

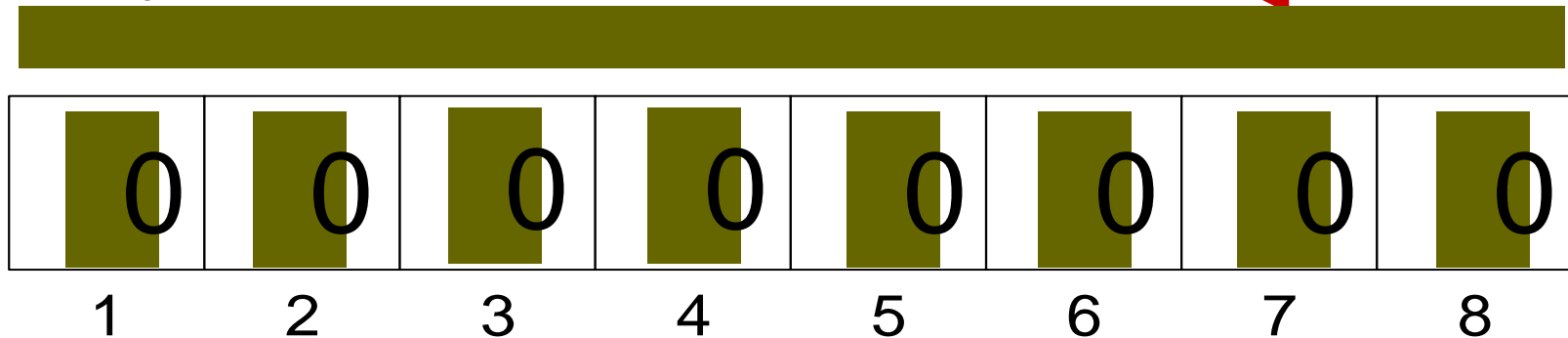
ALGORITMA

For Indeks \leftarrow 1 to 8 do

$A[\text{Indeks}] = 0$

Endfor

Array A satu dimensi :



INPUT ELEMEN

Array / Larik

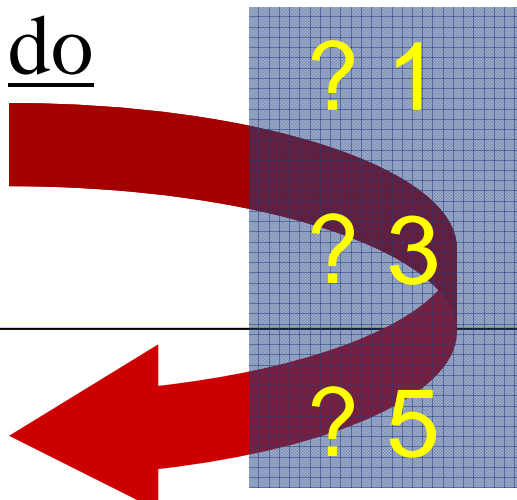
ALGORITMA

For Indeks \leftarrow 1 to 8 do

Input *A[Indeks]*

Endfor

Array A satu dimensi :



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 3 | 5 | 7 | 2 | 9 | 4 | 7 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

CETAK ELEMEN

Array / Larik

ALGORITMA

For Indeks \leftarrow 1 to 8 do
 Print *A[Indeks]*
Endfor



Array A satu dimensi

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| 1 | 3 | 5 | 7 | 2 | 9 | 4 | 7 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

Contoh dalam C

Contoh 1 (menginputkan dan menampilkan)

```
#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();
}
```



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

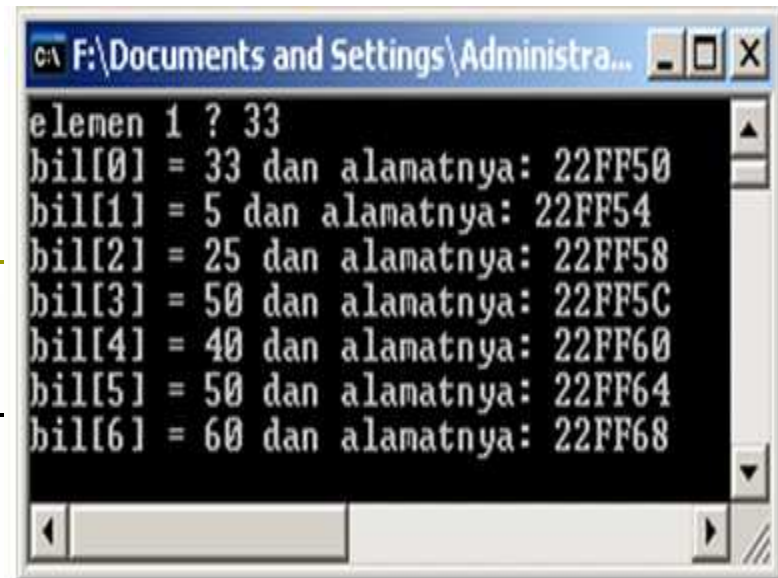
Membaca nilai :
Nilai Angka 4
Nilai Angka 7
Nilai Angka 3
Nilai Angka 9
Nilai Angka 6
```

Contoh (2)

Contoh 2 (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;}
```



```
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
```

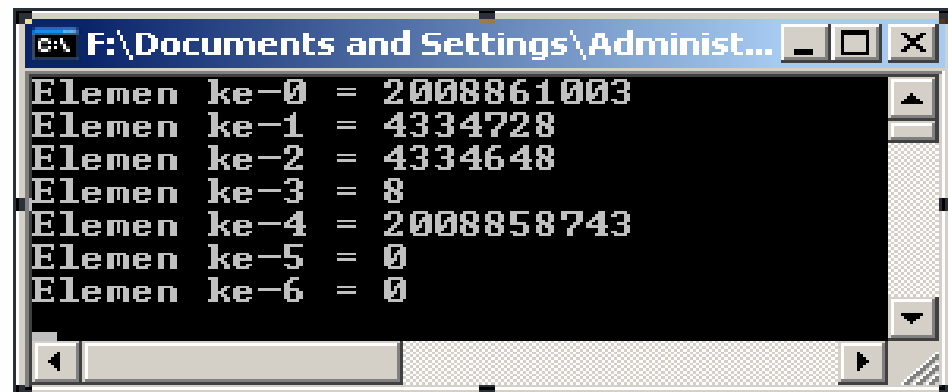
Terlihat bahwa alamat array berurutan dengan jarak antar alamat adalah 4 bytes (integer berukuran 4 bytes)

Contoh (3)

Contoh 3 (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;
}
```



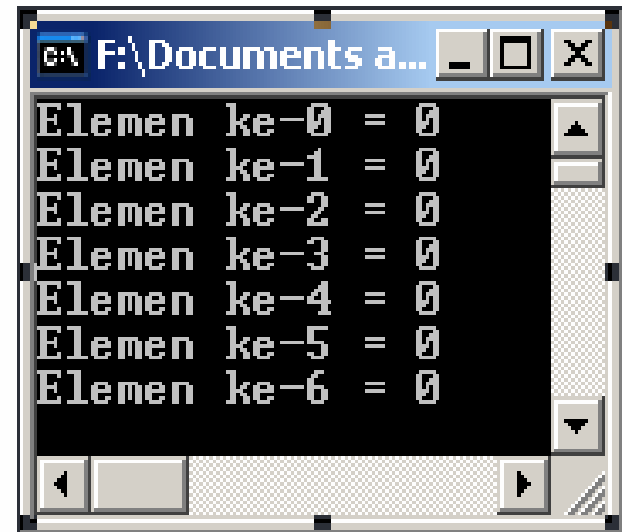
```
F:\Documents and Settings\Administ...
Elemen ke-0 = 2008861003
Elemen ke-1 = 4334728
Elemen ke-2 = 4334648
Elemen ke-3 = 8
Elemen ke-4 = 2008858743
Elemen ke-5 = 0
Elemen ke-6 = 0
```

Contoh (4)

Contoh 4 (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;
}
```



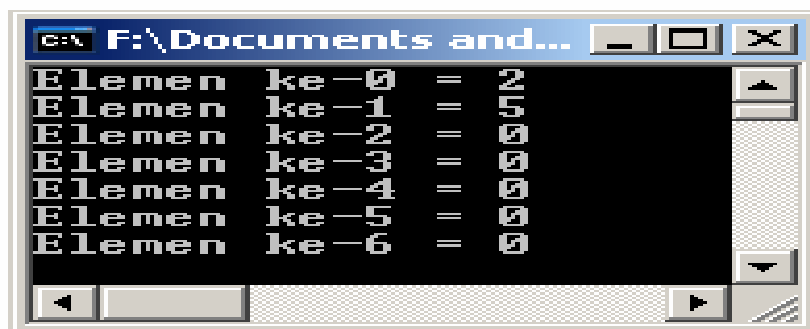
Contoh (5)

Contoh 5 (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:



```
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
```

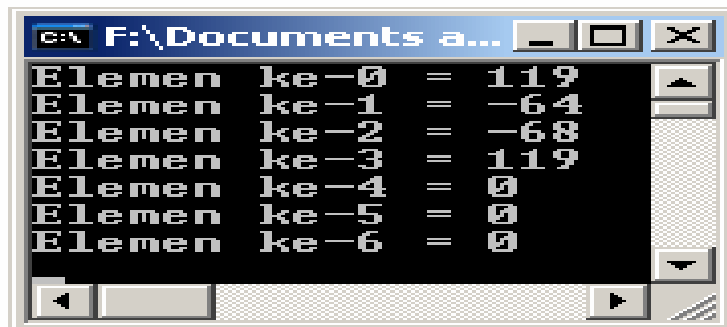
Contoh (6)

Contoh 6 (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:



```
cmd F:\Documents a...
Elemen ke-0 = 119
Elemen ke-1 = -64
Elemen ke-2 = -68
Elemen ke-3 = 119
Elemen ke-4 = 0
Elemen ke-5 = 0
Elemen ke-6 = 0
```

Pengiriman Parameter fungsi berupa array dimensi 1

- Pengiriman parameter berupa array dimensi 1 bersifat by reference
 - Karena yang dikirimkan adalah alamat dari elemen pertama array, bukan seluruh nilai elemen-elemen array.
 - Alamat pertama array ditunjukkan oleh nama arraynya, tidak perlu dituliskan indeksinya.
 - Bentuk ini akan terlihat pada argumen di parameter aktual.

Jika diganti: antonius
rachmat chrismanto

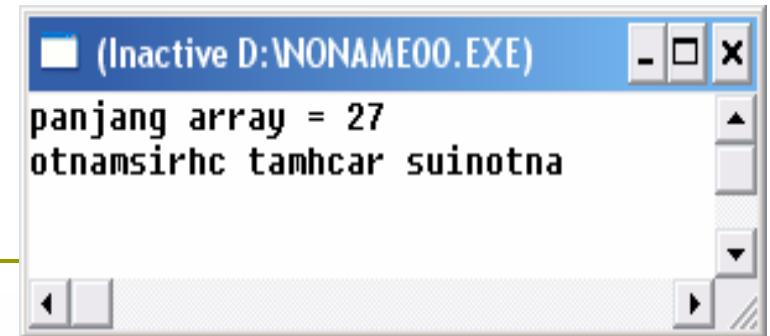
Contoh

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

```
void cetak_mundur(char S[]) {  
    int i,n;  
    for (n=0;n<S[n];n++);  
    printf("panjang array = %d\n",n);  
    for (i=n-1;i>=0;i--) {  
        printf("%c",S[i]);  
    }  
}
```

```
void main() {  
    char str[50]="anton";  
    cetak_mundur(str);  
}
```

panjang array = 5
notna

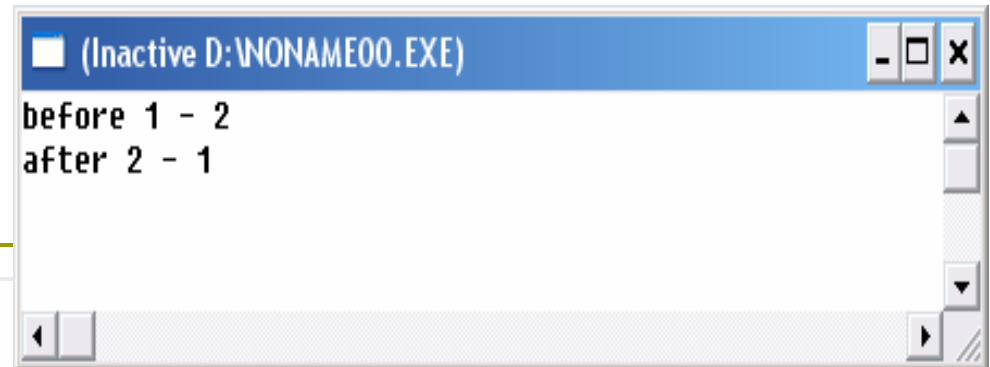


Contoh (2)

```
#include <stdio.h>

void tukar(int array[2]) {
    int t=array[0];
    array[0]=array[1];
    array[1]=t;
}

void main(){
    int array[2] = {1,2};
    printf("before %d - %d\n",array[0],array[1]);
    tukar(array);
    printf("after %d - %d",array[0],array[1]);
}
```



Soal

- 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
```

Soal

- Kembangkan soal tadi, 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
```

Soal

- Kembangkan soal no 1, 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
```

Soal

- Kembangkan soal no-1 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 | _ | _ |
|-------|----|---|---|---|---|

PR

- ❑ Buatlah program dengan array untuk menghitung total, rata-rata, maximum, minimum!
- ❑ Kemudian tambahkan kemampuan:
 - Mencari bilangan max dan min kedua
 - Frekuensi kemunculan suatu bilangan
- ❑ Tambahkan kemampuan:
 - Untuk mencari data
 - Untuk menghapus suatu data
- ❑ Dibuat dalam menu dan fungsi



NEXT: *Array 2 Dimensi*
