



# **PENGANTAR KOMPUTER & SOFTWARE II**

---

## **Array (Part I)**

# Tujuan

---

- Mahasiswa mampu memahami makna dan penggunaan array sebagai struktur data koleksi objek.
- Mahasiswa dapat menggunakan notasi array yang sesuai dengan benar.
- Mahasiswa dapat memahami persoalan yang dapat dikonversi menjadi program sederhana yang mengandung array.

# Mengolah 3 data

- Tuliskan program yang menerima 3 nama, lalu menampilkan semua kombinasi pasangan nama.
- Contoh keluaran:

```
Ali
Budi
Caca
Ali - Budi
Ali - Caca
Budi - Caca
```

```
int main () {
    // KAMUS
    string nama1, nama2, nama3;
    //ALGORITMA
    cin >> nama1;
    cin >> nama2;
    cin >> nama3;
    cout << nama1 " - " nama2 << endl;
    cout << nama1 " - " nama3 << endl;
    cout << nama2 " - " nama3 << endl;
}
```

# Mengolah 10 data

- Tuliskan program yang menerima 10 nama, lalu menampilkan semua kombinasi pasangan nama.
- Contoh keluaran:

```
Nama-1: Ali
Nama-2: Budi
...
Nama-9: Ina
Nama-10: Jaja
Ali – Budi
Ali – Caca
...
Ina - Jaja
```

```
int main () {
    // KAMUS
    string nama1, nama2, nama3, nama4, nama5;
    string nama6, nama7, nama8, nama9, nama10;
    //ALGORITMA
    cin >> nama1;
    cin >> nama2;
    ... // lanjutkan sendiri!!
    cin >> nama10;

    cout << nama1 << " - " << nama2 << endl;
    cout << nama1 << " - " << nama3 << endl;
    ... // lanjutkan sendiri!!
    cout << nama9 << " - " << nama10 << endl;
}
```

# Bagaimana kalau...

Anda diminta menampilkan semua kombinasi pasangan nama yang mungkin dari ...

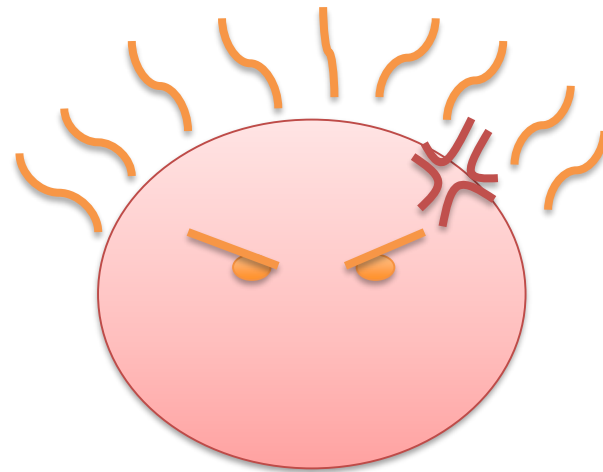
**100 nama ???**

**1000 nama ???**

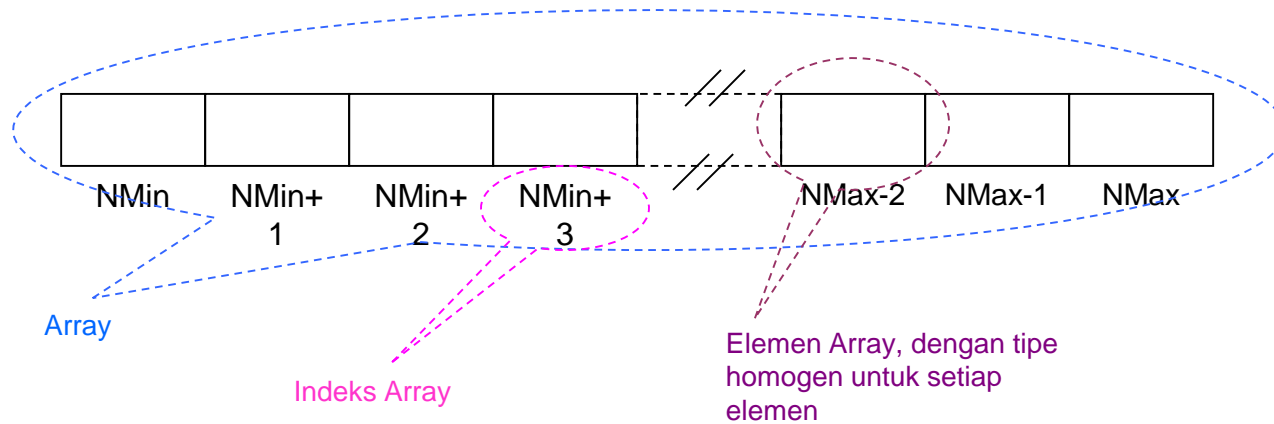
**10000 nama ???**

**1000000 nama ???**

....



# Array / Tabel / Vektor / Larik



- Type array adalah type yang mengacu kepada sebuah atau sekumpulan elemen melalui indeks
- Merepresentasikan sekumpulan informasi yang bertipe sama dan disimpan dengan urutan yang sesuai dengan definisi indeks

# Array dalam C/C++ (1)

- Variabel dapat dideklarasikan ber-type **array** dari suatu **type** tertentu
- Setiap elemen array diakses dengan alamat berupa **indeks** yang bertype integer
- Cara deklarasi: `<type> <namatabel>[<ukuran>];`
- Contoh: `int TabInt[10];`

Array bernama **TabInt** dengan setiap elemen bertype **integer**, dengan ukuran **10** elemen, dengan alamat setiap elemen array (indeks) adalah dari **indeks ke-0 s.d. 9**

# Contoh Deklarasi Array yang lain

```
int main() {  
    // KAMUS  
    int TabJumlahHari[12]; // indeks 0..11  
    float TabNilai[15];   // indeks 0..14  
    char TabHuruf[100];   // indeks 0..99  
    string TabKata[100];  // indeks 0..99  
    Point TabTitik[20];   // indeks 0..19, Point  
    terdefinisi  
    // ALGORITMA  
    ...  
}
```

- Elemen dari array dapat diakses langsung jika dan hanya jika indeks terdefinisi
- Cara mengacu sebuah elemen:  
TabInt[2]  
TabInt[i] jika i terdefinisi



# Mengisi Array

- Mengisi array merupakan aktifitas memberi nilai elemen array
  - Pemberian nilai satu elemen, contoh: **TabInt[0]=31;**
  - Pemberian nilai beberapa elemen, contoh:  

```
for (i=0;i<10;i++) {  
    TabInt[i]=i*10;  
}
```
- Hati-hati!
  - Jangan mengakses elemen yang **indeks-nya berada di luar definisi**, misalnya **TabInt[10]** → index ke-10 tidak terdefinisi untuk TabInt
  - Jangan membaca elemen yang **belum** diisi nilainya

# Membaca Array

- Cara akses elemen:  
`<namatabel>[<indeks>]`
- Contoh: `int TabInt[10];`

1	2	4	-1	100	2	0	-1	3	9
0	1	2	3	4	5	6	7	8	9

`cout << TabInt[4];` // akan tercetak: **100**

`int x = TabInt[0] + TabInt[5];` // x bernilai **3**

`TabInt[10] ???` // Berada di luar range, tidak terdefinisi!!

# Mengisi dan membaca isi Array

- Elemen array yang telah diberi nilai dapat diakses kembali
- Contoh berikut menampilkan semua isi array ke layar

```
// File: isibacaarray.cpp
// Mengisi array dan menampilkan
// seluruh elemen pada array
#include <iostream>
using namespace std;
int main ()
{ // KAMUS
  int TabInt[10]; int i;

  // Algoritma

  // mengisi array
  for (i=0; i<10; i++) {
    TabInt[i]=i*10;
  }
  // membaca dan menuliskan isi
  // array ke layar
  for (i=0; i<10; i++) {
    cout << TabInt[i] << endl;
  }
  return 0;
}
```

# Latihan 1: Mengisi Array

---

Buatlah sebuah program yang meminta input 5 nama mahasiswa kemudian menampilkan kelima nama tersebut dalam satu baris dengan spasi.

**Contoh tampilan hasil:**

Nama 1: Adi //input

Nama 2: Budi //input

Nama 3: Alex //input

Nama 4: Wati //input

Nama 5: Santi //input

Adi Budi Alex Wati Santi //output

# Solusi Latihan 1

```
#include <iostream>
using namespace std;

int main(){
    //KAMUS
    int i;
    string Nama[5];

    //MENGINISI ARRAY
    for(i=1;i<=5;i++){
        cout << "Nama " << i << " : ";
        cin >> Nama[i-1];
    }

    //MENAMPILKAN
    for(i=0;i<5;i++){
        cout << Nama[i] << " ";
    }

    return 0;
}
```

# Tugas 1

---

Buatlah sebuah program yang meminta input 5 nama mahasiswa, kemudian menampilkan nama mahasiswa sesuai nomor urut yang di-input user.

**Contoh tampilan hasil:**

Nama 1: Adi //input

Nama 2: Budi //input

Nama 3: Alex //input

Nama 4: Wati //input

Nama 5: Santi //input

Nomor Urut: 3 //input

Alex //output

# Tugas 2

Buatlah sebuah program yang meminta input berupa N nama dari user kemudian menampilkan kombinasi pasangan-pasangan nama dari N nama tersebut! Buatlah program yang sesederhana mungkin!

**Contoh tampilan hasil:**

N = 3  *//(Input)*

Nama 1 = Adi  *//(Input)*

Nama 2 = Budi  *//(Input)*

Nama 3 = Wati  *//(Input)*

Adi – Budi  *//(Output)*

Adi – Wati  *//(Output)*

Budi – Wati  *//(Output)*