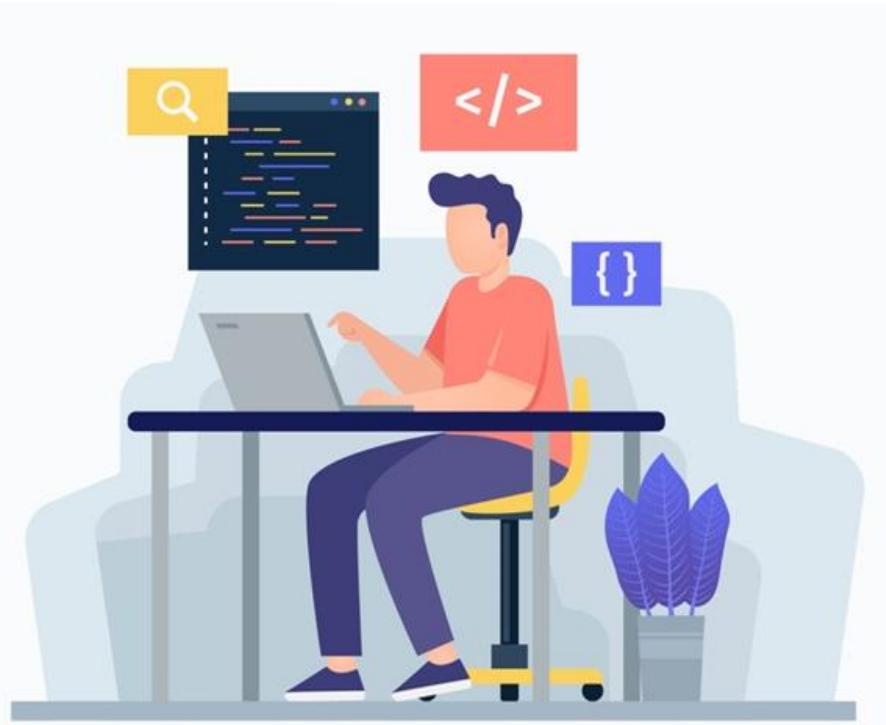




**PROGRAM STUDI  
TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS DIAN NUSWANTORO**

Mata Kuliah  
**Dasar Pemrograman**



# Array

TIM DASAR PEMROGRAMAN  
TEKNIK INFORMATIKA S1  
UNIVERSITAS DIAN NUSWANTORO

# Capaian Pembelajaran

1. Menjelaskan konsep array secara umum
2. Menjelaskan skema pemrosesan sekuensial pada array 1 dimensi
3. Membuat kode program yang memanfaatkan array 1 dimensi

# Array di C++

- Di C++, cara membangkitkan array adalah :
  - Deklarasi: <tipe> <nama>[<ukuran>];
  - Deklarasi & Inisialisasi langsung: <tipe> <nama>[<ukuran>]={<val1>,...,<val-n>}
- Akses setiap nilai:
  - <nama>[<indeks>]

# Deklarasi Array

**tipe\_data nama\_var\_array [ukuran\_indeks]**

- tipe\_data : menyatakan jenis tipe data elemen (int, char, float, dll)
- nama\_var\_array : menyatakan nama variabel yang dipakai.
- ukuran : menunjukkan jumlah indeks maksimal elemen array.
- Contoh: **int nilai[10];**

# Lanjutan Array

## □ Inisialisasi array

```
int b[4] = {1, 2, 3, 4};
```

## □ Array n-dimensi

```
int b[<s1>] [<s2>] ... [<sn>]
```

s<sub>i</sub> Harus konstanta

## □ Indeks array di hitung dari 0 sampai (s<sub>i</sub>-1)

## □ C/C++ tidak menangani “out-of-range exception” artinya akan memberikan nilai yang tidak jelas

```
int b[4] = {1, 2, 3, 4};  
for (int i = 0; i < 4; i++)  
{  
    printf("%d\n", b[i]);  
}  
printf("%d\n", b[10]);
```

b[10] = ?

# Array Assignment

```
int a[4] = {1, 2, 3, 4};
```

```
int a[4] = {1};
```

```
int a[] = {1, 2, 3, 4};
```

```
int a[4];
a[0] = 1;
a[1] = 2;
a[2] = 3;
a[3] = 4;
```

```
int a[4];
memset(a, 0, 4*sizeof(int));
```

a[0], a[1], a[2],  
a[3] = ?

# Array 2D Assignment

```
int a[3][2];
a[0][0] = 1;
a[0][1] = 2;
a[1][0] = 3;
a[1][1] = 4;
a[2][0] = 5;
a[2][1] = 6;
```

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

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

```
int a[3][2];
memset(a, 0, 6*sizeof(int));
```

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

# Memahami dimana array disimpan

- Penting untuk melihat bahwa array disimpan pada suatu memory di komputer ini dengan “per byte”.
- Simbol & ketika mengotuputkan akan mengembalikan referensi lokasi variabel tersebut disimpan di memory.
- Karena integer membutuhkan alokasi 4 byte. Sehingga array i dimana bertipe array of integer dengan panjang 5 memiliki alokasi sebesar  $4 \times 5$  yaitu 20. **(bukti array kontingu)**
- Lihat urutan angkanya, setiap blok ada selisih 4. (alamat ditulis dengan hexadesimal 0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f)

 D:\Latih

```
0x61feac  
0x61feb0  
0x61feb4  
0x61feb8  
0x61febcd
```

```
#include <iostream>  
  
using namespace std;  
  
int main()  
{  
//kamus  
    int i[5];  
//algoritma  
    cout << &i[0] << endl;  
    cout << &i[1] << endl;  
    cout << &i[2] << endl;  
    cout << &i[3] << endl;  
    cout << &i[4] << endl;  
  
    return 0;  
}
```

# Pertanyaan

1. Dari kode slide sebelumnya, Apa yang terjadi jika memanggil: `cout << &i[5] << endl;` setelah perintah `cout << &i[4] << endl;` ?
2. Dari kode slide sebelumnya, Berapa alokasi memory yang dibutuhkan jika variabel i bertipe **short**?
3. Diberikan kode sebagai berikut: →
4. apa outputnya?
5. Kenapa output `i[0]` dan `i[3]` pasti 100 dan 15?
6. Kenapa selain `i[0]` dan `i[3]` outputnya angka tidak jelas?

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     //kamus
8     int i[5];
9     i[0] = 100;
10    i[3] = 15;
11
12    cout << i[0] << endl;
13    cout << i[1] << endl;
14    cout << i[2] << endl;
15    cout << i[3] << endl;
16    cout << i[4] << endl;
17
18    return 0;
19 }
```

# Jawaban

1. Outputnya alamat memory juga dengan alamat selanjutnya. (tidak error)
2. 10 karena setiap blok memory kapasitasnya 2 byte dan panjang arraynya ada 5 sehingga  $2 \times 5 = 10$
3. Silahkan di coba sendiri ^-^ biar tau outputnya
4. Karena  $i[0]$  dan  $i[3]$  sudah di assign nilai 100 dan 15
5. Karena selain  $i[0]$  dan  $i[3]$  belum di assign sesuatu nilai, jadi outputnya angka tidak jelas

# Memset

- Dengan memberikan pustaka string.h atau cstring.
- Bisa memberikan nilai default pada setiap elemen array atau tipe string yang kita miliki.
- Contoh kode yang memberikan nilai default 0 pada semua elemen array i.

```
#include <iostream>
#include <cstring>

using namespace std;

int main()
{
//kamus
    int i[5];
//algoritma
    cout << sizeof(i) << endl;
    memset(i, 0, sizeof(i));
    cout << i[0]<< endl;
    cout << i[1]<< endl;
    cout << i[2]<< endl;
    cout << i[5]<< endl; // akan muncul angka tidak jelas, knp?

    return 0;
}
```

# Skema Pemrosesan Sekuensial

- Looping adalah jawabannya!

```
#include <iostream>

using namespace std;

int main()
{
    //kamus
    int arr[5]; // deklarasi
    //algoritma
    // input
    for(int i=0;i<5;i++)
    {
        cin >> arr[i];
    }
    // output
    for(int i=0;i<5;i++)
    {
        cout << arr[i] << endl;
    }

    return 0;
}
```

```
#include <iostream>

using namespace std;

int main()
{
    //kamus
    int arr[5]; // deklarasi
    //algoritma
    // input dan output sekaligus
    for(int i=0;i<5;i++)
    {
        cin >> arr[i];
        cout << arr[i] << endl;
    }

    return 0;
}
```

# Menyiapkan panjang array dengan input user

- Pastikan bahwa batasan bertipe integer atau konstanta.

```
#include <iostream>

using namespace std;

int main()
{
    //kamus
    int N;
    cin >> N;
    int arr[N]; // deklarasi
    //algoritma
    // input dan output sekaligus
    for(int i=0;i<N;i++)
    {
        cin >> arr[i];
        cout << arr[i] << endl;
    }

    return 0;
}
```

# Latihan

- Buat program dengan array untuk menyimpan nama hari dalam 1 minggu!

```
#include <iostream>
#include <cstring>

using namespace std;

int main()
{
    //kamus
    string arrNamaHari[7];
    int i;
    //algoritma
    i =0;
    // inputkan hari
    while(i<7)
    {
        cin >> arrNamaHari[i];
        i++;
    }
    i =0;
    // outputkan hari
    while(i<7)
    {
        cout << arrNamaHari[i] << endl;
        i++;
    }

    return 0;
}
```

## Program NamaHari

### Kamus

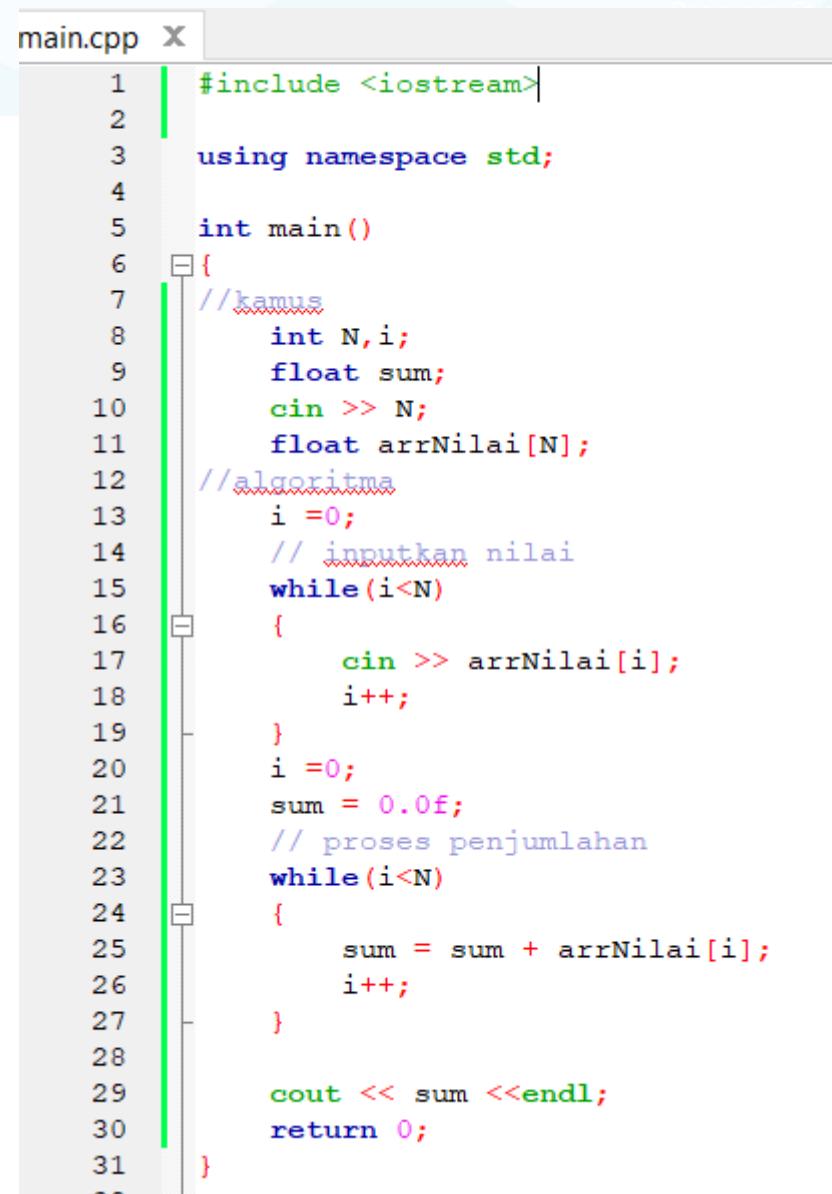
arrNamaHari: array [1...7] of string  
i : integer

### Algoritma

i  $\leftarrow$  0  
{inputkan hari}  
while i<7 do  
    input(arrNamaHari<sub>i</sub>)  
    i  $\leftarrow$  i+1  
i  $\leftarrow$  0  
{outputkan hari}  
while i<7 do  
    output(arrNamaHari<sub>i</sub>)  
    i  $\leftarrow$  i+1

# Latihan

- Hitung sejumlah nilai float didalam suatu array yang setiap elemennya di inputkan oleh user!



```
main.cpp X
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     //kamus
8     int N,i;
9     float sum;
10    cin >> N;
11    float arrNilai[N];
12    //algoritma
13    i =0;
14    // inputkan nilai
15    while(i<N)
16    {
17        cin >> arrNilai[i];
18        i++;
19    }
20    i =0;
21    sum = 0.0f;
22    // proses penjumlahan
23    while(i<N)
24    {
25        sum = sum + arrNilai[i];
26        i++;
27    }
28
29    cout << sum << endl;
30    return 0;
31 }
```

## Dari soal sebelumnya hitung rata-rata?

- Di baris kode ke-29 diganti cout << sum/N << endl;

# Pencarian elemen pada Array

- Diketahui array dari bilangan bulat yang terdefinisi: int arrNilai[10]={12,3,4,1,2,3,4,20,10,11};
- Buat variabel K dimana variabel ini akan diassign oleh inputan integer dari user.
- Jika k merupakan elemen dari arrNilai maka outputkan “k ketemu”
- Jika tidak outputkan “k tidak ketemu”

```
#include <iostream>

using namespace std;

int main()
{
    //kamus
    int arrNilai[10]={12,3,4,1,2,3,4,20,10,11};
    int k;
    char ket;
    //algoritma
    cin >> k;
    ket = 't';
    for(int i=0;i<10;i++)
    {
        if(k==arrNilai[i])
        {
            ket = 'y';
            break;
        }
    }
    if(ket == 'y')
    {
        cout << k << "ketemu";
    }
    else
    {
        cout << k << "tidak ketemu";
    }
    return 0;
}
```

# Pencarian elemen terbesar atau terkecil

- Buat variabel penampung nilai terbesar atau terkecil
- Assign variabel sebelumnya tadi dengan elemen pertama dari array
- Lakukan penelusuran setiap elemen dari array, lakukan:
- Bandingkan elemen array yang sekarang di cari dengan variabel penampung.
- Jika memenuhi maka assign element array sekarang ke variabel penampung
- Setelah penelusuran selesai maka variabel penampung akan menyimpan nilai elemen terbesar atau terkecil dan outputkan

# Referensi

## Utama:

1. Bjarne Stroustrup, 2014, Programming: Principles and Practice Using C++ (Second Edition), Addison-Wesley Professional

## Pendukung:

1. Introduction to Computer Science and Programming in Python, MIT  
<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016>
2. Introduction to Computer Science and Programming, MIT  
<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-00sc-introduction-to-computer-science-and-programming-spring-2011/index.htm>



# TERIMA KASIH

ANY QUESTIONS?