



- **Array di C++**

- Struktur data yang menyimpan kumpulan elemen dengan **tipe data yang sama**
- Elemen disimpan dalam **lokasi memori yang berurutan**
- Setiap elemen dapat diakses menggunakan **index**
- Index dimulai dari **0** (zero-based indexing)
- Ukuran array ditentukan saat deklarasi dan tidak bisa diubah

- **Mengapa Menggunakan Array?**

- **Efisiensi:** Menyimpan banyak data tanpa membuat variabel terpisah
- **Organisasi:** Mengelompokkan data yang berkaitan
- **Kemudahan:** Akses dan manipulasi data dengan mudah menggunakan looping
- **Fundamental:** Dasar untuk struktur data yang lebih kompleks

- **Contoh:**

```
// Tanpa array - kurang efisien  
int nilai1 = 85;  
int nilai2 = 90;  
int nilai3 = 78;  
int nilai4 = 92;  
int nilai5 = 88;
```

# Konsep Array - Visualisasi dan Karakteristik

- **Visualisasi Array:**

```
Array: nilai[5]

Index:      [0]   [1]   [2]   [3]   [4]
          +-----+-----+-----+-----+-----+
Nilai:      | 85  | 90  | 78  | 92  | 88  |
          +-----+-----+-----+-----+-----+
Memori:     1000  1004  1008  1012  1016  (alamat memori)
```

- **Sintaks Dasar Deklarasi Array:**

```
tipe_data nama_array[ukuran];
```

- **Contoh Deklarasi Array:**

```
// Array integer dengan 5 elemen
int nilai[5];

// Array double dengan 10 elemen
double suhu[10];

// Array char dengan 50 elemen
char huruf[50];

// Array string dengan 3 elemen
string nama[3];

// Array boolean dengan 7 elemen
bool status[7];
```

- **Penjelasan Komponen:**

- **tipe\_data**: Tipe data elemen array (int, double, char, string, dll)
- **nama\_array**: Identifier/nama variabel array
- **ukuran**: Jumlah elemen yang dapat disimpan (harus konstanta atau literal)

- **Aturan Penting dalam Deklarasi:**

```
// ? BENAR - ukuran menggunakan konstanta literal
int data[100];

// ? BENAR - ukuran menggunakan const variable
const int UKURAN = 50;
int data[UKURAN];

// ? SALAH - ukuran menggunakan variabel biasa (non-const)
int n = 20;
int data[n]; // Error di beberapa compiler

// ? SALAH - ukuran tidak boleh negatif atau nol
int data[-5]; // Error!
int data[0]; // Error!
```

- **Metode 1: Inisialisasi Saat Deklarasi (Lengkap)**

```
// Menginisialisasi semua elemen
int angka[5] = {10, 20, 30, 40, 50};
string buah[3] = {"Apel", "Jeruk", "Mangga"};
double nilai[4] = {85.5, 90.0, 78.5, 92.0};
```

- **Metode 2: Inisialisasi Sebagian**

```
// Elemen yang tidak diinisialisasi akan otomatis diisi 0
int angka[5] = {10, 20};
// Hasilnya: {10, 20, 0, 0, 0}

int data[10] = {1};
// Hasilnya: {1, 0, 0, 0, 0, 0, 0, 0, 0, 0}
```

- **Metode 3: Inisialisasi Tanpa Ukuran Eksplisit**

```
// Compiler otomatis menentukan ukuran dari jumlah elemen
int nilai[] = {85, 90, 78, 92, 88}; // ukuran = 5
string hari[] = {"Senin", "Selasa", "Rabu"}; // ukuran = 3
```

#### • Metode 4: Inisialisasi Setelah Deklarasi

```
int nilai[3];
nilai[0] = 85;
nilai[1] = 90;
nilai[2] = 78;
```



- **syntax:**

```
nama_array[index]
```

- **Contoh Akses Elemen:**

```
int nilai[5] = {85, 90, 78, 92, 88};

// Membaca nilai elemen
cout << nilai[0]; // Output: 85
cout << nilai[2]; // Output: 78
cout << nilai[4]; // Output: 88

// Mengubah nilai elemen
nilai[1] = 95; // Ubah elemen index 1 dari 90 menjadi 95
nilai[3] = 100; // Ubah elemen index 3 dari 92 menjadi 100
```

- **Akses dengan Variabel Index:**

```
int data[5] = {10, 20, 30, 40, 50};
int i = 2;

cout << data[i];      // Output: 30 (elemen index 2)
cout << data[i+1];    // Output: 40 (elemen index 3)
cout << data[i*2];    // Output: 50 (elemen index 4)
```

### • Operasi dengan Elemen Array:

```
int angka[3] = {5, 10, 15};

// Aritmatika
int hasil = angka[0] + angka[1]; // hasil = 15
angka[2] = angka[0] * 3;         // angka[2] = 15

// Perbandingan
if (angka[0] < angka[1]) {
    cout << "Elemen pertama lebih kecil" << endl;
}
```

- **Cara Mengetahui Ukuran Array:**

```
int nilai[5] = {85, 90, 78, 92, 88};

// Menggunakan sizeof
int ukuran = sizeof(nilai) / sizeof(nilai[0]);
// sizeof(nilai) = total bytes array
// sizeof(nilai[0]) = bytes per elemen
// Hasilnya: ukuran = 5

cout << "Ukuran array: " << ukuran << endl;

- Metode 1: For Loop Biasa
int data[5] = {10, 20, 30, 40, 50};
int n = sizeof(data) / sizeof(data[0]);

// Loop dengan index
for (int i = 0; i < n; i++) {
    cout << "Index " << i << ": " << data[i] << endl;
}

- Metode 2: Range-based For Loop (Foreach - C++11)
int angka[5] = {10, 20, 30, 40, 50};
```

```
// Foreach - otomatis iterasi semua elemen
for (int nilai : angka) {
    cout << nilai << " ";
}
cout << endl;
// Output: 10 20 30 40 50
```

## Contoh: Array Suhu Harian

Slide 7

- **Studi Kasus:**

- Menyimpan suhu selama 7 hari (1 minggu)
- Menampilkan suhu per hari
- Mengidentifikasi hari dengan suhu tertinggi

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

int main() {
    // Array suhu dalam Celsius untuk 7 hari
    double suhu[7] = {28.5, 29.0, 27.5, 30.2, 31.0, 28.8, 27.0};
    string hari[7] = {"Senin", "Selasa", "Rabu", "Kamis",
                     "Jumat", "Sabtu", "Minggu"};

    // Menampilkan suhu per hari
    cout << "Suhu Mingguan:" << endl;
    cout << "======" << endl;
    for (int i = 0; i < 7; i++) {
        cout << hari[i] << ": " << suhu[i] << "°C" << endl;
    }
}
```

```

    }

    // Mencari hari dengan suhu tertinggi
    int indexTertinggi = 0;
    for (int i = 1; i < 7; i++) {
        if (suhu[i] > suhu[indexTertinggi]) {
            indexTertinggi = i;
        }
    }

    cout << "\nHari terpanas: " << hari[indexTertinggi];
    cout << " (" << suhu[indexTertinggi] << "°C)" << endl;

    return 0;
}

```

### • Output Program:

```

Suhu Mingguan:
=====
Senin: 28.5°C
Selasa: 29°C
Rabu: 27.5°C
Kamis: 30.2°C
Jumat: 31°C
Sabtu: 28.8°C
Minggu: 27°C

```

Hari terpanas: Jumat (31°C)

# Input Array dengan Looping

Slide 8

- **Sintaks Input Array dengan For Loop:**

```
for (int i = 0; i < ukuran; i++) {  
    cin >> nama_array[i];  
}
```

- **Contoh Program Input Nilai:**

```
#include <iostream>  
using namespace std;  
  
int main() {  
    const int JUMLAH = 5;  
    int nilai[JUMLAH];  
  
    // Input nilai menggunakan for loop  
    cout << "Masukkan " << JUMLAH << " nilai:" << endl;  
    for (int i = 0; i < JUMLAH; i++) {  
        cout << "Nilai ke-" << (i+1) << ": ";  
        cin >> nilai[i];  
    }
```



```
}

// Verifikasi input
cout << "\nData yang diinput:" << endl;
for (int i = 0; i < JUMLAH; i++) {
    cout << "nilai[" << i << "] = " << nilai[i] << endl;
}

return 0;
}
```

#### • Output Program:

Masukkan 5 nilai:

Nilai ke-1: 85

Nilai ke-2: 90

Nilai ke-3: 78

Nilai ke-4: 92

Nilai ke-5: 88

Data yang diinput:

nilai[0] = 85

nilai[1] = 90

nilai[2] = 78

nilai[3] = 92

nilai[4] = 88

# Output Array dengan Looping

Slide 9

- **Sintaks Output Array dengan For Loop:**

```
for (int i = 0; i < ukuran; i++) {  
    cout << nama_array[i];  
}
```

- **Contoh 1: Output Horizontal**

```
#include <iostream>  
using namespace std;  
  
int main() {  
    int angka[5] = {10, 20, 30, 40, 50};  
  
    // Output horizontal dengan spasi  
    cout << "Array: ";  
    for (int i = 0; i < 5; i++) {  
        cout << angka[i] << " ";  
    }  
    cout << endl;
```

```
    return 0;
}
// Output: Array: 10 20 30 40 50
```

### • Contoh 2: Output Vertikal dengan Index

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

int main() {
    string nama[4] = {"Alice", "Bob", "Charlie", "Diana"};

    // Output vertikal dengan nomor urut
    cout << "Daftar Nama:" << endl;
    for (int i = 0; i < 4; i++) {
        cout << (i+1) << ". " << nama[i] << endl;
    }

    return 0;
}
```

### Output:

```
Daftar Nama:
1. Alice
```

2. Bob
3. Charlie
4. Diana

## Contoh: Input/Output Array Lengkap

Slide 10

- **Program Lengkap: Sistem Data Mahasiswa**

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

int main() {
    const int JUMLAH = 3;
    string nama[JUMLAH];
    int umur[JUMLAH];

    // INPUT DATA
    cout << "=== INPUT DATA MAHASISWA ===" << endl;
    for (int i = 0; i < JUMLAH; i++) {
        cout << "\nMahasiswa ke-" << (i+1) << ":" << endl;
        cout << "Nama: ";
        getline(cin, nama[i]); // getline untuk input string dengan spasi
        cout << "Umur: ";
        cin >> umur[i];
        cin.ignore(); // Membersihkan buffer
    }

    // OUTPUT DATA
```

```

cout << "\n=== DATA MAHASISWA ===" << endl;
cout << "No.\tNama\t\tUmur" << endl;
cout << "===== " << endl;
for (int i = 0; i < JUMLAH; i++) {
    cout << (i+1) << "\t" << nama[i] << "\t\t" << umur[i] << endl;
}

// ANALISIS DATA
int totalUmur = 0;
for (int i = 0; i < JUMLAH; i++) {
    totalUmur += umur[i];
}
double rataUmur = (double)totalUmur / JUMLAH;

cout << "\nRata-rata umur: " << rataUmur << " tahun" << endl;

return 0;
}

```

- **Operasi Umum pada Array:**

- **Pencarian (Searching):** Mencari nilai tertentu atau karakteristik khusus
- **Pengurutan (Sorting):** Mengurutkan elemen
- **Agregasi:** Menghitung sum, average, max, min
- **Filtering:** Memfilter elemen berdasarkan kondisi
- **Transformasi:** Mengubah nilai elemen

- **Operasi yang Akan Dipelajari:**

1. **Mencari Nilai Maksimum (Max)**
2. **Mencari Nilai Minimum (Min)**
3. **Menghitung Total/Jumlah (Sum)**
4. **Menghitung Rata-rata (Average)**

## 5. Kombinasi operasi di atas



- **Algoritma Mencari Nilai Maksimum:**

1. Anggap elemen pertama adalah nilai maksimum
2. Loop dari elemen kedua hingga akhir
3. Jika ada elemen lebih besar dari max saat ini, update max
4. Setelah loop selesai, max berisi nilai terbesar

- **Implementasi Kode:**

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

int main() {
    int nilai[8] = {75, 92, 68, 88, 95, 70, 85, 90};
    int n = 8; // jumlah elemen

    // Inisialisasi max dengan elemen pertama
    int max = nilai[0];
```

```

int indexMax = 0;

// Loop untuk mencari nilai maksimum
for (int i = 1; i < n; i++) {
    if (nilai[i] > max) {
        max = nilai[i];
        indexMax = i;
    }
}

// Tampilkan hasil
cout << "Array nilai: ";
for (int i = 0; i < n; i++) {
    cout << nilai[i] << " ";
}
cout << endl;

cout << "\nNilai maksimum: " << max << endl;
cout << "Berada di index: " << indexMax << endl;
cout << "Siswa ke-" << (indexMax + 1) << endl;

return 0;
}

```

# Menghitung Sum (Total)

Slide 13

- **Algoritma Menghitung Sum:**

1. Inisialisasi variabel sum dengan 0
2. Loop melalui semua elemen array
3. Tambahkan setiap elemen ke sum
4. Sum berisi total dari semua elemen

- **Implementasi Kode:**

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

int main() {
    int penjualan[5] = {150, 200, 175, 225, 180};
    int n = 5;

    // Inisialisasi sum dengan 0
    int totalPenjualan = 0;
```

```
// Loop untuk menghitung total
for (int i = 0; i < n; i++) {
    totalPenjualan += penjualan[i];
    // totalPenjualan = totalPenjualan + penjualan[i];
}

// Tampilkan hasil
cout << "Penjualan per hari:" << endl;
for (int i = 0; i < n; i++) {
    cout << "Hari " << (i+1) << ": " << penjualan[i] << " unit" << endl;
}

cout << "\nTotal penjualan: " << totalPenjualan << " unit" << endl;

return 0;
}
```

# Menghitung Average (Rata-rata)

Slide 14

- **Algoritma Menghitung Average:**

1. Hitung sum/total dari semua elemen
2. Bagi sum dengan jumlah elemen
3. Hasilnya adalah rata-rata (average)

- **Implementasi Kode:**

```
#include <iostream>
#include <iomanip> // untuk setprecision
using namespace std;

int main() {
    int nilai[6] = {85, 90, 78, 92, 88, 95};
    int n = 6;

    // Hitung total
    int total = 0;
    for (int i = 0; i < n; i++) {
```

```

        total += nilai[i];
    }

    // Hitung rata-rata
    double rataRata = (double)total / n;

    // Tampilkan hasil
    cout << "Nilai ujian:" << endl;
    for (int i = 0; i < n; i++) {
        cout << "Siswa " << (i+1) << ": " << nilai[i] << endl;
    }

    cout << "\nTotal nilai: " << total << endl;
    cout << "Jumlah siswa: " << n << endl;
    cout << fixed << setprecision(2); // 2 angka desimal
    cout << "Rata-rata: " << rataRata << endl;

    return 0;
}

```