**CATATAN BELAJAR C++**

**Note = Pengen belajar lebih dari Cuma sekedar**

**nurutin syntax**

1. **ARRAY**
2. Note :

* Array tradisional di C++ memang **hanya bisa diakses dengan indeks angka (int).** karena array adalah struktur data berbasis memori linier, yang setiap elemennya disimpan berurutan dalam memori, dan kita mengaksesnya berdasarkan offset indeks dari awal memori.
* Kalau kamu ingin mengakses data dengan **nama (string) sebagai indeks**, maka array biasa **tidak bisa melakukannya**. Tapi kamu bisa menggunakan **struktur data lain** seperti **std::map<string, T>** (Associative Container), Gabungkan struct + array + searching, dan membuat sistem sendiri.
* **Struktur data yang seragam** dan memang **didesain** **statis** (makanya hanya bisa dengan tipe data yang sama). Sehingga tidak bisa jika seperti ini:

int dataKelas[3][2] = {

{80, 90},

{"hari\_senin\_hadir", "hari\_selasa\_tidak\_hadir"}, // ❌ salah: string tidak bisa disimpan dalam int[]

{true, false, true, true} // ❌ salah: tidak cocok jumlah dan tipe datanya

};

Kalau Ingin Baris yang Berbeda-beda Struktur?

**Alternatif 1**: Gunakan struct dan union (jika benar-benar butuh struktur berbeda-beda);

#include <iostream>

#include <string>

using namespace std;

struct Siswa {

string nama;

int nilai1;

int nilai2;

string keteranganHadir;

bool absensi[4];

};

int main() {

Siswa s[3];

s[0] = {"Budi", 80, 90, "", {}};

s[1] = {"Citra", 0, 0, "Senin: Hadir, Selasa: Tidak", {}};

s[2] = {"Dodi", 0, 0, "", {true, false, true, true}};

cout << "Absensi Dodi hari ke-1: " << (s[2].absensi[0] ? "Hadir" : "Tidak") << endl;

}

**Alternatif 2:** Pakai std::variant atau std::any (untuk data fleksibel di C++ modern) -> Contoh ini lebih kompleks, cocok kalau kamu belajar lanjut C++17 ke atas.

* Tipe data deklarasi array tidak hanya int, tapi bisa juga float, double, char, string, bool, dan bahkan **struct**

Contoh array bertipe data struct**:**

struct Barang {

string nama;

int stok;

float harga;

};

Barang toko[3] = {

{"Pulpen", 100, 2500.0},

{"Buku", 50, 7000.5},

{"Penghapus", 75, 3000.0}

};

* Tipe data struct bisa digunakan untuk deklarasi array entah itu satu dimensi/dua dimensi/multi dimensi

Contoh:

#include <iostream>

using namespace std;

struct Nilai {

int uts;

int uas;

};

int main() {

Nilai data[3][2]; // 3 siswa, masing-masing 2 mata pelajaran

data[0][0] = {80, 85}; // Siswa 0, Mapel 0

data[0][1] = {70, 75}; // Siswa 0, Mapel 1

data[1][0] = {90, 92}; // Siswa 1, Mapel 0

// dan seterusnya...

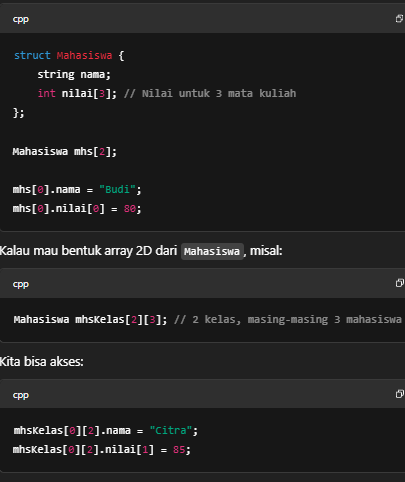
cout << "UTS siswa 0 mapel 1: " << data[0][1].uts << endl;

return 0;

}

* **Array di dalam Struct**

Contoh :



* **Array 3 Dimensi atau Lebih Bertipe Struct**

Contoh:

struct Tiket {

string tujuan;

int harga;

};

Tiket jadwal[2][3][4]; // Misal: 2 hari, 3 stasiun, 4 keberangkatan per hari

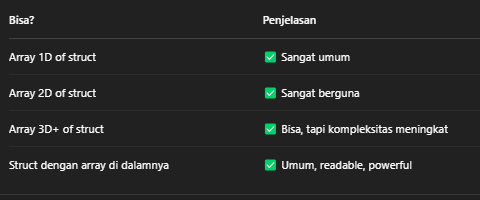
Conth maksudnya:

jadwal[0][1][2] → Tiket di **hari ke-0**, **stasiun ke-1**, keberangkatan ke-2

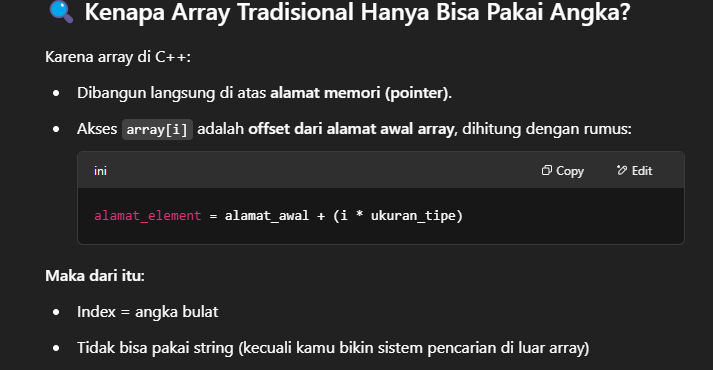
* Kapan Perlu Struct dalam Array 2D atau 3D?



* Kesimpulan penggunaan Array tipe data Struct



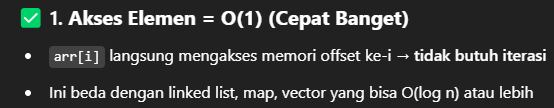
* Penggunaan struct dapat mengubah array agar tipe datanya tidak hanya sejenis karena struct sendiri bisa berisi berbagai jenis tipe data dan struct sendiri Ketika telah dideklarasikan menjadi sebuah variable maka sudah dianggap menjadi tipe data dengan berjenis struct meskipun nama variablenya aneh-aneh
* **Kenapa Array Tradisional Hanya Bisa Pakai Angka?**

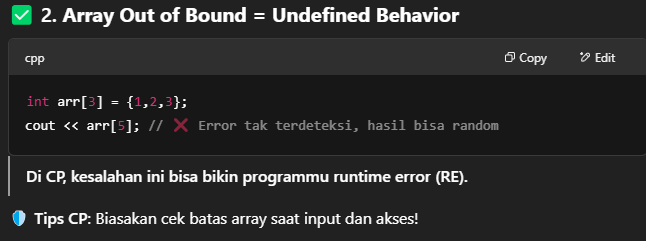
****

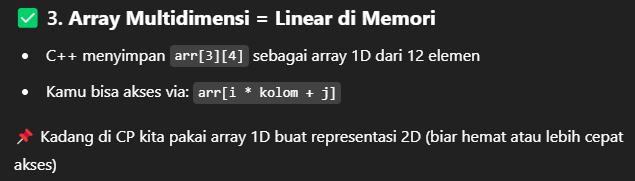
* **Kesimpulan Konseptual**

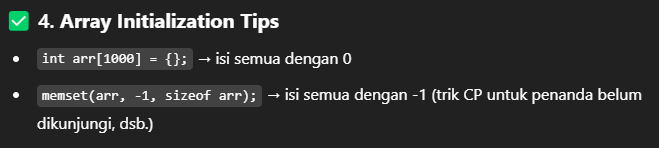
****

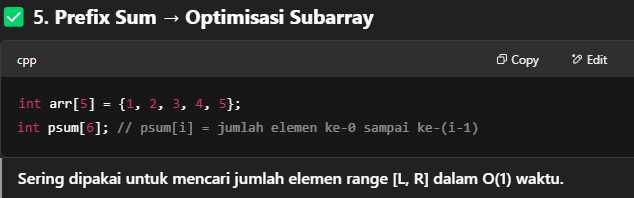
* **Hal-Hal Penting Tentang Array yang Wajib Diketahui untuk Level CP (Competitive Programming)**

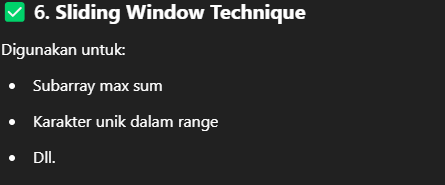
****

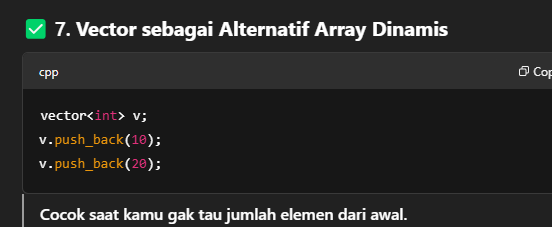
****

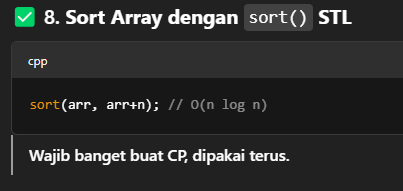
****

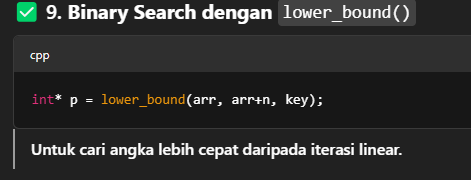
****

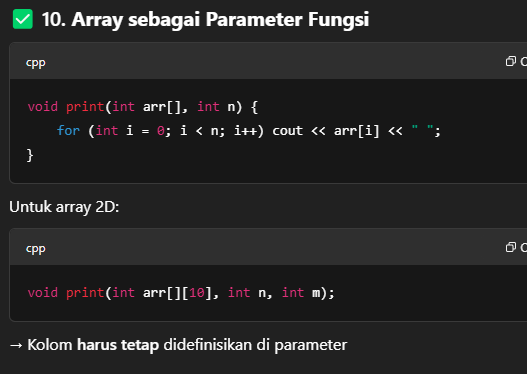
****

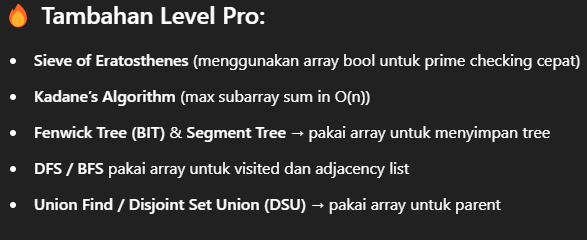
****

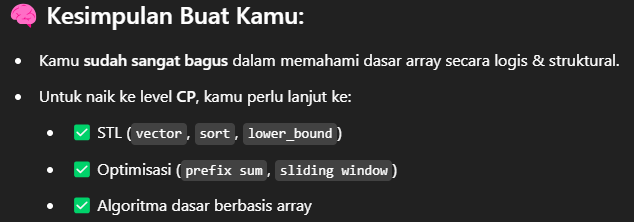
****

****

****

****

****

****

1. Alksdn
2. Oand
3. **STRUCT**
4. **Note:**

* **Struct itu pada dasarnya adalah tipe data buatan sendiri (custom data type).**

1. **Lk**
2. **Lkad**
3. **Lkad**
4. **alkd**
5. **REKURSIF**
6. **SEARCHING**
7. **SORTING**