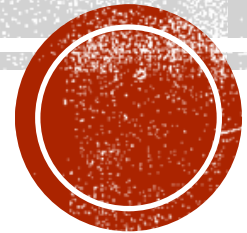


# ARRAY & STRING

Pertemuan 3

Tutor : Kak Vanya



# APA SAJA YANG AKAN KITA PELAJARI ?

1. Array 1D dan 2D
2. Array
3. Array Looping (For loop & Foreach)
4. Array Vector

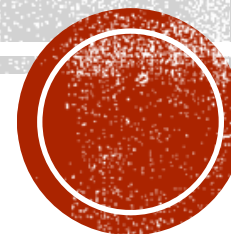


# TUJUAN PEMBELAJARAN

1. Memahami konsep dan kegunaan Array dan String dalam C++.
2. Membuat, mengakses, dan memodifikasi data dalam array dan string.
3. Menggunakan array untuk menyimpan banyak data dan string untuk memanipulasi teks.
4. Mampu menyelesaikan permasalahan dasar menggunakan array dan string.



# ARRAY BIASA (1D)



# APA ITU ARRAY ?

- **Array** adalah struktur data yang menyimpan banyak data dengan tipe yang sama, dalam satu variabel dan diakses menggunakan indeks.
- **Elemen array diakses melalui indeks** (mulai dari 0). Karena bahasa komputer, angka itu dimulai dari 0.
- Array **digunakan jika kita ingin menyimpan banyak** nilai tanpa membuat variabel 1 per 1.
- **Array terdiri dari 1 dimensi, 2 dimensi atau multi dimensi.**
- Contoh bentuk :

```
int nilai[3] = {70, 80, 90};
```



# CIRI-CIRI ARRAY

1. Tipe data harus sama.
2. Ukuran tetap (kecuali pakai dynamic memory)
3. Indeks dimulai dari 0.
4. Cocok untuk menyimpan data seperti nilai siswa, daftar item, dll.
5. Compiler juga sudah pintar untuk menentukan berapa jumlah array tanpa perlu mengisi berapa memori atau jumlah yg dapat ditampung arraynya.



Value	1					Variabel Biasa
	1	2	3	4	5	Array
Index	0	1	2	3	4	





```
1 // cara membuat Array
2 string mobil[5];
3
4 // cara membuat array string
5 string mobil[5] = {"Volvo", "BMW", "Ford", "Mazda", "Toyota"};
6
7 // cara membuat array integer
8 int myNum[3] = {10, 20, 30};
9
```



# BENTUK ARRAY



```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      string mobil[5] = {"Toyota", "Honda", "BMW", "Suzuki", "Wuling"};
6      cout << mobil[0] << endl; // Menampilkan elemen pertama
7      cout << mobil[3] << endl; // Menampilkan elemen keempat
8      return 0;
9  }
10
```





```
1  #include <iostream> // Mengimpor library untuk input/output standar
2  using namespace std; // Menghindari keharusan menulis std:: di setiap fungsi C++
3
4  int main() {
5      // Mendeklarasikan array bertipe string dengan 5 elemen
6      string mobil[5];
7
8      // Mengisi elemen array satu per satu
9      mobil[0] = "Lahkokgini"; // Elemen pertama (indeks ke-0)
10     mobil[1] = "Tokoma";      // Elemen kedua (indeks ke-1)
11     mobil[2] = "Willy";       // Elemen ketiga (indeks ke-2)
12     mobil[3] = "BVMW";        // Elemen keempat (indeks ke-3)
13     mobil[4] = "Suzoku";       // Elemen kelima (indeks ke-4)
14
15     // Menampilkan judul output ke layar
16     cout << "Isi array mobil:" << endl;
```

# ARRAY FOR LOOP & SIZEOF

- Array for loop ini digunakan untuk memanggil semua data yang di dalam array berdasarkan indeks atau berapa jumlah elemen yang sudah diisi dalam array yang sudah di deklarasikan.
- Daripada menggunakan manual manggil 1 per 1 semisal isi array nya banyak kan ribet ya.
- Dalam looping array kita juga menggunakan **sizeof** untuk menentukan berapa jumlah array yg perlu kita output dan menyesuaikan dengan tipe datanya.  
(rekomen)





```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int myNumbers[5] = {10, 20, 30, 40, 50};
6      int getArrayLength = sizeof(myNumbers) / sizeof(myNumbers[0]);
7
8      cout << getArrayLength;
9
10     return 0;
11 }
```



```
1  for(int i = 0; i < sizeof(murid)/sizeof(*murid); i++){  
2      cout << murid[i] << '\n';  
3  }
```





```
1  #include <iostream>
2
3  using namespace std;
4
5  int main(){
6  string cars[5] = {"Volvo", "BMW", "Ford", "Mazda", "Tesla"};
7
8  // Loop through strings
9  for (int i = 0; i < 5; i++) {
10     cout << cars[i] << "\n";
11 }
12
13     return 0;
14 }
15
```



# FOREACH LOOP ARRAY

- Foreach ini digunakan untuk melakukan looping berdasarkan jumlah elemen yang ada.
- Biasa digunakan dalam struktur data, vector dan list.
- Untuk sintaks nya :

```
for (type variableName : arrayName) {  
  
    // code block to be executed  
  
}
```



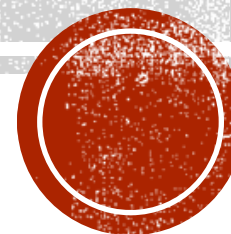


```
1  #include <iostream>
2
3  using namespace std;
4
5  int main(){
6
7      string murid[] = {"Spongebob", "Patrick", "Squidward", "Mr.Krab", "Ms.Puff", "Mr.Krab"};
8
9      int nilai[] = {70, 65, 78, 50, 100, 20};
10
11     for(string murid : murid){
12         cout << murid << '\n';
13     }
14
15     cout << "=====\n";
16
17     for(int nilai : nilai){
18         cout << nilai << '\n';
19     }
20
21     return 0;
22 }
```





# ARRAY VECTOR



# ARRAY VECTOR

- Array biasa tidak dapat terus menerus menambahkan data.
- Sedangkan vector dia itu resizable dalam arti bisa menambahkan data dan menghapus data tanpa ada batas jumlah array.



Aspek	Array (bawaan C++)	Vector (STL)
Deklarasi	<code>int a[5];</code>	<code>vector&lt;int&gt; v;</code>
Ukuran	<b>Tetap</b> (fixed)	<b>Dinamis</b> (bisa bertambah/berkurang)
Alokasi Memori	Statis	Dinamis (heap)
Penambahan Data	Tidak bisa setelah deklarasi	Bisa dengan <code>push_back()</code>
Penghapusan Elemen	Tidak bisa	Bisa dengan <code>pop_back()</code> , <code>erase()</code> , <code>clear()</code>
Fungsi Pendukung	Terbatas	Banyak (melalui STL)
Keamanan Akses	Rentan (tidak ada pengecekan index)	Aman (gunakan <code>at(index)</code> untuk cek batas)
Flexibilitas	Rendah	Tinggi
Performa	Lebih cepat (karena langsung di memory)	Sedikit lebih lambat (karena dinamis)
Header yang Dibutuhkan	Tidak perlu ( <code>&lt;iostream&gt;</code> cukup)	Perlu <code>#include &lt;vector&gt;</code>
Contoh Penggunaan	Data statis, ukuran diketahui	Data input user, list dinamis, data besar





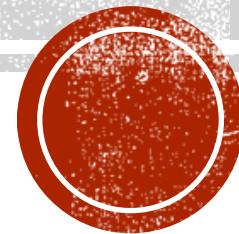
```
1  #include <iostream>
2  #include <vector> // Including the vector library
3  using namespace std;
4
5  int main() {
6      // A vector with 3 elements
7      vector<string> cars = {"Volvo", "BMW", "Ford"};
8
9      // Adding another element to the vector
10     cars.push_back("Tesla");
11     cars.push_back("Toyota");
12
13     for (string car : cars) {
14         cout << car << "\n";
15     }
16     return 0;
17 }
18
```



Perintah	Fungsi
<code>push_back(x)</code>	Menambahkan elemen x ke akhir vector
<code>pop_back()</code>	Menghapus elemen terakhir
<code>at(i)</code>	Mengakses elemen ke-i (lebih aman daripada [])
<code>front()</code>	Mengembalikan elemen pertama
<code>back()</code>	Mengembalikan elemen terakhir
<code>size()</code>	Mengembalikan jumlah elemen dalam vector
<code>empty()</code>	Mengembalikan true jika vector kosong
<code>clear()</code>	Menghapus seluruh isi vector



# ARRAY MULTIDIMENSI



# APA ITU ARRAY MULTIDIMENSI ?

- Array multidimensi adalah array dari array. Setiap dimensi menambahkan level struktur baru – misalnya array 2D adalah array dari array 1D.
- `letters[0] = {"A","B","C","D"}`
- `letters[1] = {"E","F","G","H"}`



```
1  string letters[2][4];
2  // Array dengan 2 baris, 4 kolom
3
4  string letters[2][4] = {
5      { "A", "B", "C", "D" },
6      { "E", "F", "G", "H" }
7  };
8
```





```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      // Array 2D: 2 baris (i) dan 3 kolom (j)
6      int angka[2][3] = {
7          {1, 2, 3},    // Baris ke-0
8          {4, 5, 6}     // Baris ke-1
9      };
10
11     // Menampilkan isi array dengan indeks
12     for (int i = 0; i < 2; i++) {        // Baris
13         for (int j = 0; j < 3; j++) {    // Kolom
14             cout << "angka[" << i << "][" << j << "] = " << angka[i][j] << endl;
15         }
16     }
17
18     return 0;
19 }
20
```

