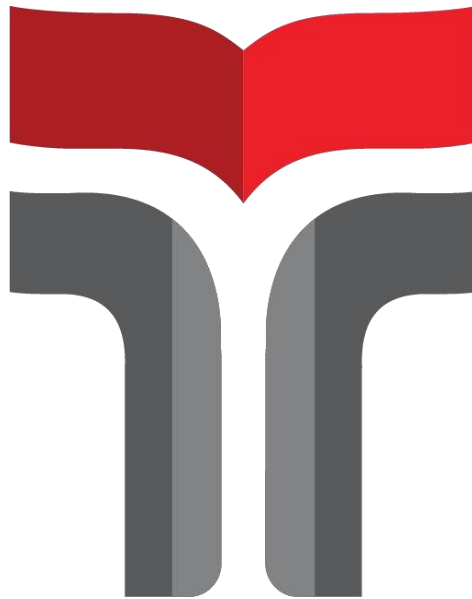


LAPORAN PRAKTIKUM  
STRUKTUR DATA 2  
"Pengenal C++: Pointer & Array"



Oleh:

NAMA: Ammar Dzaki Nandana

NIM: 2311104071

KELAS: SE 07 B

DOSEN: Wahyu Andi Saputra

PRODI S1 REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO  
2023/2024

## I. TUJUAN

- Memahami konsep dasar array dalam pemrograman C++.
- Memahami konsep pointer dan penggunaannya dalam mengakses elemen array.
- Mempelajari bagaimana array dan pointer saling berhubungan di memori komputer.
- Menerapkan pointer untuk manipulasi data dalam array.

## II. DASAR TEORI

### 2.1 Array

Array adalah struktur data yang digunakan untuk menyimpan sekumpulan data dengan tipe yang sama. Setiap elemen dalam array diidentifikasi oleh indeksnya, di mana indeks dimulai dari 0. Array memberikan cara yang mudah untuk menyimpan dan mengakses sejumlah besar data secara berurutan.

- Deklarasi Array: `int arr[5];` // Array dengan 5 elemen bertipe integer
- Akses Elemen Array Setiap elemen dapat diakses menggunakan indeks: `arr[0] = 10;` // Menyimpan nilai 10 pada elemen pertama

### 2.2 Pointer

Pointer adalah variabel yang menyimpan alamat memori dari variabel lain. Pointer sangat berguna dalam pemrograman karena memungkinkan manipulasi langsung dari nilai yang disimpan di lokasi memori tertentu

- Deklarasi Pointer: `int *ptr;` Pointer di atas dapat menyimpan alamat dari variabel bertipe int.
- Mendapatkan Alamat Memori: Operator `&` digunakan untuk mendapatkan alamat memori dari sebuah variabel: `int var = 10; ptr = &var;` // Pointer ptr menyimpan alamat dari var
- Mengakses Nilai dari Alamat yang Ditunjuk: Operator `*` digunakan untuk mengakses nilai dari alamat yang ditunjuk oleh pointer: `int value = *ptr;` // Mengambil nilai dari variabel yang ditunjuk ptr

### 2.3 Hubungan Array dan Pointer

Dalam C++, array dan pointer sangat erat hubungannya. Nama array sebenarnya merupakan pointer tetap yang menunjuk ke elemen pertama dari array tersebut. Dengan menggunakan pointer, kita bisa mengakses elemen-elemen dalam array.

- Contoh

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

int *ptr = arr; // Pointer ptr menunjuk ke elemen pertama dari array arr

// Mengakses elemen array melalui pointer

std::cout << *(ptr + 1); // Output: 2 (elemen kedua dari array)
```

Pointer dapat digunakan untuk melakukan iterasi melalui array dan mengakses atau memanipulasi elemen-elemen array. Dengan memanfaatkan aritmatika pointer, kita dapat berpindah-pindah antara elemen-elemen array berdasarkan jarak memori antar elemen.

## Macam-Macam Array

### 1. Array Satu Dimensi

Array satu dimensi adalah bentuk paling sederhana dari array yang terdiri dari satu baris elemen. Array ini menyimpan sekumpulan data dalam satu baris atau satu kolom, dan elemen-elemen tersebut dapat diakses menggunakan satu indeks.

Deklarasi: `int arr[5];` // Array satu dimensi dengan 5 elemen

Contoh Penggunaan: `int arr[5] = {1, 2, 3, 4, 5};` // Array berisi 5 elemen

`std::cout << arr[2];` // Mengakses elemen ketiga (nilai: 3)

#### Tipe-Tipe Penggunaan Array Satu Dimensi:

> Array Statis: Ukuran array sudah ditentukan saat deklarasi dan tidak bisa diubah selama program berjalan: `int arr[5] = {1, 2, 3, 4, 5};`

> Array Dinamis: Ukuran array dapat ditentukan secara dinamis selama program berjalan, menggunakan pointer dan alokasi memori dinamis (new di C++): `int* arr = new int[n];`  
// Array dengan ukuran n yang ditentukan saat runtime

### 2. Array Dua Dimensi

Array dua dimensi adalah array yang terdiri dari baris dan kolom, yang dapat dilihat sebagai tabel atau matriks. Array dua dimensi membutuhkan dua indeks untuk mengakses elemen: satu untuk baris dan satu untuk kolom.

- Deklarasi: `int arr[3][4];` // Array 2D dengan 3 baris dan 4 kolom

```
- Contoh Penggunaan: int arr[2][3] = {
    {1, 2, 3},
    {4, 5, 6}
};
```

```
std::cout << arr[1][2]; // Mengakses elemen baris kedua, kolom ketiga (nilai: 6)
```

### Perbedaan Utama antara Array Satu Dimensi dan Dua Dimensi

- Array Satu Dimensi hanya membutuhkan satu indeks untuk mengakses elemen, sedangkan Array Dua Dimensi membutuhkan dua indeks.
- Array satu dimensi digunakan untuk menyimpan data dalam bentuk linear (misalnya daftar nilai), sementara array dua dimensi cocok untuk data dalam bentuk tabel atau matriks (misalnya data tabel nilai siswa).

## III. GUIDED

```
1 #include <iostream>
2
3 using namespace std;
4
5 int penjumlahan(int a, int b){
6     return a + b;
7 }
8 void great(string name){
9     cout << "Hello," << name << "!" << endl;
10 }
11
12 int main(){
13     int hasil = penjumlahan(5, 3);
14
15     cout << "hasil " << hasil << endl;
16
17     great("Ammar");
18
19
20 //     int nilai[5]={1,2,3,4,5};
21 //     cout << nilai[0];
22 //     cout << nilai[1];
23 //     cout << nilai[2];
24 //     cout << nilai[3];
25 //     cout << nilai[4];
26
27
28 //     int nilai[5]={1,2,3,4,5};
29 //
30 //     for(int i=0; i<5; i++){
31 //         cout << nilai[i] << endl;
32 //     }
33
34
35 // array 2 dimensi
36 //int nilai[3][4] = {
37 //    {1,2,3,4},
38 //    {5,6,7,8},
39 //    {9,10,11,12}
40 //};
41 //
42 //for(int i=0; i<3; i++){
43 //    for(int j=0; j<4; j++){
44 //        cout << nilai[i][j] << " ";
45 //    }
46 //    cout << endl;
47 // }
48 //
49
50 // pointer
51 //int x,y;
52 //int *px;
53 //x = 87;
54 //px = &x;
55 //y= *px;
56
57 //cout << "Alamat x= " << &x << endl;
58 //cout << "Isi px= " << *px << endl;
59 //cout << "Isi x= " << x << endl;
60 //cout << "Nilai yang ditunjukkan px= " << *px << endl;
61 //cout << "Nilai y= " << y << endl;
62 //getch();
63
64 // pointer & string
65
66 }
67
68 }
```

## IV. UNGUIDED

1.

```
1  #include <iostream>
2  #include <vector>
3
4  int main() {
5      int n;
6      std::vector<int> array, genap, ganjil;
7
8      std::cout << "Masukkan jumlah elemen array: ";
9      std::cin >> n;
10
11     std::cout << "Masukkan " << n << " elemen array: ";
12     for (int i = 0; i < n; ++i) {
13         int temp;
14         std::cin >> temp;
15         array.push_back(temp);
16
17         if (temp % 2 == 0) {
18             genap.push_back(temp);
19         } else {
20             ganjil.push_back(temp);
21         }
22     }
23
24     std::cout << "Data array = ";
25     for (int i = 0; i < n; ++i) {
26         std::cout << array[i] << " ";
27     }
28     std::cout << std::endl;
29
30     std::cout << "Nomor genap = ";
31     for (int i = 0; i < genap.size(); ++i) {
32         std::cout << genap[i] << " ";
33     }
34     std::cout << std::endl;
35
36     std::cout << "Nomor ganjil = ";
37     for (int i = 0; i < ganjil.size(); ++i) {
38         std::cout << ganjil[i] << " ";
39     }
40     std::cout << std::endl;
41
42     return 0;
43 }
```

2.

```
1  #include <iostream>
2
3  int main() {
4      int x, y, z;
5
6      std::cout << "Masukkan ukuran dimensi pertama (x): ";
7      std::cin >> x;
8      std::cout << "Masukkan ukuran dimensi kedua (y): ";
9      std::cin >> y;
10     std::cout << "Masukkan ukuran dimensi ketiga (z): ";
11     std::cin >> z;
12
13     int array[x][y][z];
14
15     std::cout << "Masukkan elemen array 3D:" << std::endl;
16     for (int i = 0; i < x; ++i) {
17         for (int j = 0; j < y; ++j) {
18             for (int k = 0; k < z; ++k) {
19                 std::cout << "array[" << i << "][" << j << "][" << k << "] = ";
20                 std::cin >> array[i][j][k];
21             }
22         }
23     }
24
25     std::cout << "Array 3D yang diinput adalah:" << std::endl;
26     for (int i = 0; i < x; ++i) {
27         for (int j = 0; j < y; ++j) {
28             for (int k = 0; k < z; ++k) {
29                 std::cout << "array[" << i << "][" << j << "][" << k << "] = " << array[i][j][k] << std::endl;
30             }
31         }
32     }
33
34     return 0;
35 }
```

3.

```
1  #include <iostream>
2  #include <limits>
3
4  void findMax(int array[], int n) {
5      int max = array[0];
6      for (int i = 1; i < n; i++) {
7          if (array[i] > max) {
8              max = array[i];
9          }
10     }
11     std::cout << "Nilai maksimum: " << max << std::endl;
12 }
13
14 void findMin(int array[], int n) {
15     int min = array[0];
16     for (int i = 1; i < n; i++) {
17         if (array[i] < min) {
18             min = array[i];
19         }
20     }
21     std::cout << "Nilai minimum: " << min << std::endl;
22 }
23
24 void findAverage(int array[], int n) {
25     int sum = 0;
26     for (int i = 0; i < n; i++) {
27         sum += array[i];
28     }
29     double average = static_cast<double>(sum) / n;
30     std::cout << "Nilai rata-rata: " << average << std::endl;
31 }
32
33 int main() {
34     int n;
35
36     std::cout << "Masukkan jumlah elemen array: ";
37     std::cin >> n;
38
39     int array[n];
40
41     std::cout << "Masukkan elemen array:" << std::endl;
42     for (int i = 0; i < n; i++) {
43         std::cin >> array[i];
44     }
45
46     int pilihan;
47     do {
48
49         std::cout << "\nMenu:\n";
50         std::cout << "1. Cari nilai maksimum\n";
51         std::cout << "2. Cari nilai minimum\n";
52         std::cout << "3. Cari nilai rata-rata\n";
53         std::cout << "4. Keluar\n";
54         std::cout << "Pilih opsi: ";
55         std::cin >> pilihan;
56
57         switch (pilihan) {
58             case 1:
59                 findMax(array, n);
60                 break;
61             case 2:
62                 findMin(array, n);
63                 break;
64             case 3:
65                 findAverage(array, n);
66                 break;
67             case 4:
68                 std::cout << "Keluar dari program." << std::endl;
69                 break;
70             default:
71                 std::cout << "Pilihan tidak valid. Coba lagi." << std::endl;
72         }
73     } while (pilihan != 4);
74
75     return 0;
76 }
77
```

## **V. KESIMPULAN**

Kesimpulannya, array merupakan struktur data penting yang digunakan untuk menyimpan sekumpulan nilai dengan tipe data yang sama. Array satu dimensi digunakan untuk menyimpan data secara linear dan hanya memerlukan satu indeks untuk mengakses elemennya, cocok untuk aplikasi seperti daftar nilai atau kumpulan data sederhana. Sementara itu, array dua dimensi lebih kompleks, digunakan untuk menyimpan data dalam bentuk tabel atau matriks, dan memerlukan dua indeks untuk mengakses setiap elemen, seperti data yang berbentuk grid atau tabel. Kedua jenis array ini bisa digunakan dalam bentuk statis, di mana ukuran array ditentukan saat deklarasi, atau dinamis, yang memungkinkan ukuran array ditentukan saat runtime. Array dua dimensi umumnya digunakan dalam aplikasi yang membutuhkan representasi data multidimensi, seperti grafik atau perhitungan matriks.





## VI. UNGUIDED

NO. 1

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      float bil1, bil2;
7
8      // Input dua buah bilangan dari user
9      cout << "Masukkan bilangan pertama: ";
10     cin >> bil1;
11     cout << "Masukkan bilangan kedua: ";
12     cin >> bil2;
13
14     // Tampilkan hasil penjumlahan
15     cout << "Hasil penjumlahan: " << bil1 + bil2 << endl;
16
17     // Tampilkan hasil pengurangan
18     cout << "Hasil pengurangan: " << bil1 - bil2 << endl;
19
20     // Tampilkan hasil perkalian
21     cout << "Hasil perkalian: " << bil1 * bil2 << endl;
22
23     // Tampilkan hasil pembagian
24     // Pengecekan untuk menghindari pembagian dengan nol
25     if (bil2 != 0) {
26         cout << "Hasil pembagian: " << bil1 / bil2 << endl;
27     } else {
28         cout << "Pembagian tidak bisa dilakukan karena bilangan kedua adalah nol." << endl;
29     }
30
31     return 0;
32 }
33
34
35 //2311104071_AMMAR DZAKI NANDANA
36
```

## NO. 2

```
1 #include <iostream>
2 #include <string>
3
4 using namespace std;
5
6 // Fungsi untuk mengubah angka menjadi teks
7 string angkaKeTeks(int angka)
8 {
9     string satuan[] = {"", "satu", "dua", "tiga", "empat", "lima", "enam", "tujuh", "delapan", "sembilan"};
10    string belasan[] = {"sepuluh", "sebelas", "dua belas", "tiga belas", "empat belas", "lima belas", "enam belas", "tujuh belas", "delapan belas", "sembilan belas"};
11    string puluhan[] = {"", "", "dua puluh", "tiga puluh", "empat puluh", "lima puluh", "enam puluh", "tujuh puluh", "delapan puluh", "sembilan puluh"};
12
13    // Jika angka adalah 0
14    if (angka == 0)
15    {
16        return "nol";
17    }
18
19    // Jika angka adalah 100
20    if (angka == 100)
21    {
22        return "seratus";
23    }
24
25    // Jika angka di bawah 10
26    if (angka < 10)
27    {
28        return satuan[angka];
29    }
30
31    // Jika angka di antara 10 dan 19 (belasan)
32    if (angka >= 10 && angka < 20)
33    {
34        return belasan[angka - 10];
35    }
36
37    // Jika angka di antara 20 dan 99 (puluhan)
38    if (angka >= 20 && angka < 100)
39    {
40        return puluhan[angka / 10] + (angka % 10 != 0 ? " " + satuan[angka % 10] : "");
41    }
42
43    return "";
44 }
45
46 int main()
47 {
48     int angka;
49
50     // Input dari pengguna
51     cout << "Masukkan angka antara 0 sampai 100: ";
52     cin >> angka;
53
54     // Memastikan input valid
55     if (angka < 0 || angka > 100)
56     {
57         cout << "Masukkan angka yang valid (0 sampai 100)." << endl;
58     }
59     else
60     {
61         // Output hasil dalam bentuk teks
62         cout << angka << ": " << angkaKeTeks(angka) << endl;
63     }
64
65     return 0;
66 }
67 //2311104071_AMMAR DZAKI NANDANA
```

## NO. 3

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7
8     // Input dari pengguna
9     cout << "Masukkan angka: ";
10    cin >> n;
11
12    // Loop untuk mencetak pola
13    for (int i = n; i >= 1; i--) {
14        // Bagian kiri: menurun dari i hingga 1
15        for (int j = i; j >= 1; j--) {
16            cout << j << " ";
17        }
18
19        // Cetak tanda bintang *
20        cout << " * ";
21
22        // Bagian kanan: menaik dari 1 hingga i
23        for (int j = 1; j <= i; j++) {
24            cout << " " << j;
25        }
26
27        // Baris baru untuk pola berikutnya
28        cout << endl;
29    }
30
31    // Output terakhir hanya tanda *
32    cout << " * " << endl;
33
34    return 0;
35 }
36 #include <iostream>
37 #include <string>
38
39 using namespace std;
40
41 // Fungsi untuk mengubah angka menjadi teks
42 string angkaKeTeks(int angka)
43 {
44     string satuan[] = {"", "satu", "dua", "tiga", "empat", "lima", "enam", "tujuh", "delapan", "sembilan"};
45     string belasan[] = {"sepuluh", "sebelas", "dua belas", "tiga belas", "empat belas", "lima belas", "enam belas", "tujuh belas", "delapan belas", "sembilan belas"};
46     string puluhan[] = {"", "dua puluh", "tiga puluh", "empat puluh", "lima puluh", "enam puluh", "tujuh puluh", "delapan puluh", "sembilan puluh"};
47
48     // Jika angka adalah 0
49     if (angka == 0)
50     {
51         return "nol";
52     }
53
54     // Jika angka adalah 100
55     if (angka == 100)
56     {
57         return "seratus";
58     }
59
60     // Jika angka di bawah 10
61     if (angka < 10)
62     {
63         return satuan[angka];
64     }
65
66     // Jika angka di antara 10 dan 19 (belasan)
67     if (angka >= 10 && angka < 20)
68     {
69         return belasan[angka - 10];
70     }
71
72     // Jika angka di antara 20 dan 99 (puluhan)
73     if (angka >= 20 && angka < 100)
74     {
75         return puluhan[angka / 10] + (angka % 10 != 0 ? " " + satuan[angka % 10] : "");
76     }
77
78     return "";
79 }
80
81 int main()
82 {
83     int angka;
84
85     // Input dari pengguna
86     cout << "Masukkan angka antara 0 sampai 100: ";
87     cin >> angka;
88
89     // Memastikan input valid
90     if (angka < 0 || angka > 100)
91     {
92         cout << "Masukkan angka yang valid (0 sampai 100)." << endl;
93     }
94     else
95     {
96         // Output hasil dalam bentuk teks
97         cout << angka << ": " << angkaKeTeks(angka) << endl;
98     }
99
100    return 0;
101 }
102 //2311104071_AMMAR DZAKI NANDANA
```

## **VII. KESIMPULAN**

Penginstalan Code::Blocks beserta kompilasi MinGW (untuk pengguna Windows) telah berhasil dilakukan. IDE ini siap digunakan untuk pengembangan program dalam bahasa C atau C++. Dengan menggunakan Code::Blocks, proses pengembangan menjadi lebih mudah karena tersedianya fitur debugging dan kompilasi otomatis.