

LAPORAN PRAKTIKUM
Modul 2
“PENGENALAN BAHASA C++ (BAGIAN KEDUA)”



Disusun Oleh:
NAMA : Ganes Gemi Putra
NIM : 2311104075
Kelas : SE-07-02

Dosen :
WAHYU ANDI SAPUTRA

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
PURWOKERTO
2024

1. Tujuan

Memahami penggunaan pointer dan alamat memori
Mengimplementasikan fungsi dan prosedur dalam program

2. Landasan Teori

1. Array:

- Array adalah kumpulan data yang memiliki tipe data yang sama, dengan indeks yang digunakan untuk mengakses setiap elemen.
- Array dapat berupa satu dimensi, dua dimensi, atau multidimensi. Modul ini menjelaskan cara deklarasi dan penggunaan array di C++ beserta contoh kodenya.

2. Pointer:

- Pointer adalah variabel yang menyimpan alamat memori dari variabel lain. Modul ini menjelaskan bagaimana pointer bekerja, bagaimana mendeklarasikannya, dan bagaimana menggunakannya untuk mengakses dan memodifikasi data yang ada di lokasi memori yang ditunjuk.

3. Hubungan antara Array dan Pointer:

- Terdapat hubungan erat antara array dan pointer dalam C++. Operasi yang dilakukan pada array bisa juga dilakukan dengan menggunakan pointer.

4. String:

- Dalam bahasa C, string adalah kumpulan karakter yang disimpan dalam array karakter, diakhiri dengan karakter null ('\\0'). Modul ini menjelaskan cara deklarasi dan manipulasi string menggunakan pointer.

5. Fungsi dan Prosedur:

- Modul ini juga menjelaskan bagaimana fungsi dan prosedur bekerja di C++. Fungsi merupakan blok kode yang memiliki nilai balik, sedangkan prosedur adalah fungsi yang tidak mengembalikan nilai (menggunakan `void`).

6. Parameter dalam Fungsi:

- Ada tiga cara melewatkan parameter ke fungsi: call by value (pemanggilan dengan nilai), call by pointer (pemanggilan dengan pointer), dan call by reference (pemanggilan dengan referensi).

3. Guided

```
main.cpp x
1  #include <iostream>
2  #include <conio.h>
3
4  using namespace std;
5  int main() {
6      /*
7       int nilai[5]={1,2,3,4,5};
8
9       cout << nilai[0] << nilai[1] << nilai[2] << nilai[3] << nilai[4] << endl;
10
11      for (int i=0;i<5;i++) {
12          cout << nilai[i] << endl;
13      }
14
15      int nilai[3][4] = {{1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}};
16      for (int i=0;i<3;i++) {
17          for (int j=0;j<4;j++) {
18              cout << nilai[i][j] << " ";
19          }
20          cout << endl;
21      }
22      */
23      int x,y;
24      int *px;
25      x=87;
26      px = &x;
27      y = *px;
28
29      cout << "alamat x: " << &x << endl;
30      cout << "isi px:" << *px << endl;
31      cout << "nilai X:" << x << endl;
32      cout << "alamat yang ditunjuk px: " << *px << endl;
33      cout << "nilai y:" << y << endl;
34      getch();
35  }
```

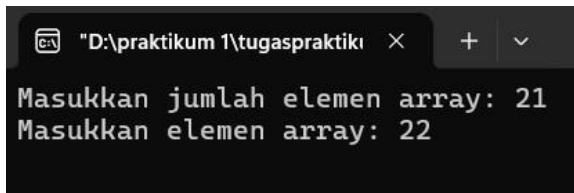
Hasil :

```
D:\LAPRAKCPP\Guided\bin\D x + v
alamat x: 0x61fe10
isi px:0x61fe10
nilai X:87
alamat yang ditunjuk px: 87
nilai y:87
|
```

4. Unguided 1.

```
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  int main() {
6      int n;
7      cout << "Masukkan jumlah elemen array: ";
8      cin >> n;
9
10     vector<int> arr(n);
11     vector<int> genap, ganjil;
12
13     // Input array
14     cout << "Masukkan elemen array: ";
15     for (int i = 0; i < n; i++) {
16         cin >> arr[i];
17         if (arr[i] % 2 == 0) {
18             genap.push_back(arr[i]);
19         } else {
20             ganjil.push_back(arr[i]);
21         }
22     }
23
24     // Display full array
25     cout << "Data Array : ";
26     for (int i = 0; i < n; i++) {
27         cout << arr[i] << " ";
28     }
29     cout << endl;
30
31     // Display even numbers
32     cout << "Nomor Genap : ";
33     for (int i = 0; i < genap.size(); i++) {
34         cout << genap[i] << " ";
35     }
36     cout << endl;
37
38     // Display odd numbers
39     cout << "Nomor Ganjil : ";
40     for (int i = 0; i < ganjil.size(); i++) {
41         cout << ganjil[i] << " ";
42     }
43     cout << endl;
44
45     return 0;
46 }
47
```

Hasil :

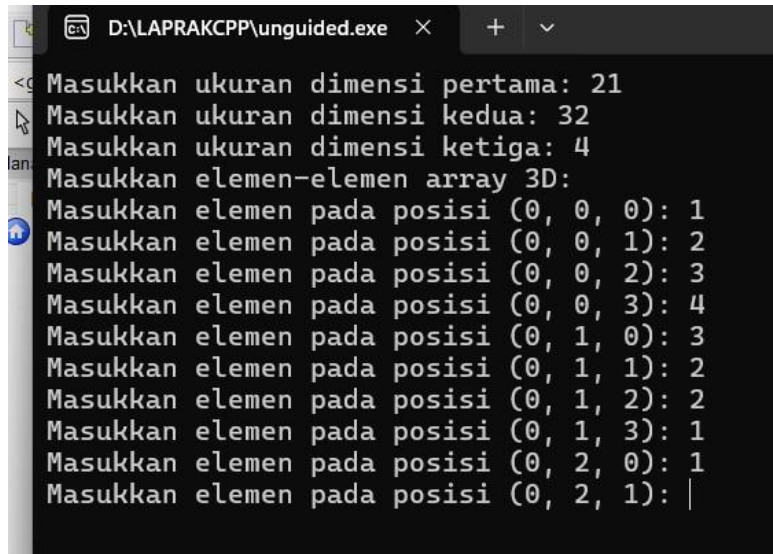


```
"D:\praktikum 1\tugaspraktik" × + ∨
Masukkan jumlah elemen array: 21
Masukkan elemen array: 22
```

2.

```
53 int main() {
54     int x, y, z;
55
56     // Input untuk ukuran array tiga dimensi
57     cout << "Masukkan ukuran dimensi pertama: ";
58     cin >> x;
59     cout << "Masukkan ukuran dimensi kedua: ";
60     cin >> y;
61     cout << "Masukkan ukuran dimensi ketiga: ";
62     cin >> z;
63
64     // Membuat array tiga dimensi secara dinamis
65     int ***array3D = new int**[x];
66     for (int i = 0; i < x; ++i) {
67         array3D[i] = new int*[y];
68         for (int j = 0; j < y; ++j) {
69             array3D[i][j] = new int[z];
70         }
71     }
72
73     // Input elemen-elemen array
74     cout << "Masukkan elemen-elemen array 3D: " << endl;
75     for (int i = 0; i < x; ++i) {
76         for (int j = 0; j < y; ++j) {
77             for (int k = 0; k < z; ++k) {
78                 cout << "Masukkan elemen pada posisi (" << i << ", " << j << ", " << k << "): ";
79                 cin >> array3D[i][j][k];
80             }
81         }
82     }
83
84     // Menampilkan array 3D
85     cout << "Array 3D adalah: " << endl;
86     for (int i = 0; i < x; ++i) {
87         for (int j = 0; j < y; ++j) {
88             for (int k = 0; k < z; ++k) {
89                 cout << "Elemen di (" << i << ", " << j << ", " << k << "): " << array3D[i][j][k] << endl;
90             }
91         }
92     }
93
94     // Membebaskan memori yang dialokasikan untuk array 3D
95     for (int i = 0; i < x; ++i) {
96         for (int j = 0; j < y; ++j) {
97             delete[] array3D[i][j];
98         }
99     }
100     delete[] array3D[i];
101 }
```

Hasil :



The screenshot shows a Windows command prompt window titled "D:\LAPRAKCPP\unguided.exe". The program prompts the user to enter the dimensions of a 3D array and its elements. The user has entered the following values:

- Masukkan ukuran dimensi pertama: 21
- Masukkan ukuran dimensi kedua: 32
- Masukkan ukuran dimensi ketiga: 4
- Masukkan elemen-elemen array 3D:
- Masukkan elemen pada posisi (0, 0, 0): 1
- Masukkan elemen pada posisi (0, 0, 1): 2
- Masukkan elemen pada posisi (0, 0, 2): 3
- Masukkan elemen pada posisi (0, 0, 3): 4
- Masukkan elemen pada posisi (0, 1, 0): 3
- Masukkan elemen pada posisi (0, 1, 1): 2
- Masukkan elemen pada posisi (0, 1, 2): 2
- Masukkan elemen pada posisi (0, 1, 3): 1
- Masukkan elemen pada posisi (0, 2, 0): 1
- Masukkan elemen pada posisi (0, 2, 1): |

```

107 |
108 |
109 | #include <iostream>
110 | using namespace std;
111 |
112 | // Fungsi untuk mencari nilai maksimum
113 | int cariMaksimum(int arr[], int n) {
114 |     int maks = arr[0];
115 |     for (int i = 1; i < n; ++i) {
116 |         if (arr[i] > maks) {
117 |             maks = arr[i];
118 |         }
119 |     }
120 |     return maks;
121 | }
122 |
123 | // Fungsi untuk mencari nilai minimum
124 | int cariMinimum(int arr[], int n) {
125 |     int min = arr[0];
126 |     for (int i = 1; i < n; ++i) {
127 |         if (arr[i] < min) {
128 |             min = arr[i];
129 |         }
130 |     }
131 |     return min;
132 | }
133 |
134 | // Fungsi untuk menghitung nilai rata-rata
135 | float cariRataRata(int arr[], int n) {
136 |     int total = 0;
137 |     for (int i = 0; i < n; ++i) {
138 |         total += arr[i];
139 |     }
140 |     return static_cast<float>(total) / n;
141 | }
142 |
143 | int main() {
144 |     int n, pilihan;
145 |
146 |     // Input jumlah elemen array
147 |     cout << "Masukkan jumlah elemen array: ";
148 |     cin >> n;
149 |
150 |     int arr[n];
151 |
152 |     // Input elemen array
153 |     cout << "Masukkan elemen-elemen array:" << endl;
154 |     for (int i = 0; i < n; ++i) {
155 |         cout << "Elemen ke-" << i + 1 << ": ";

```

Hasil :

```

Masukkan jumlah elemen array: 24
Masukkan elemen-elemen array:
Elemen ke-1: 1
Elemen ke-2: 8
Elemen ke-3: 0
Elemen ke-4: |

```

5. Kesimpulan 1. Pemahaman Dasar C++:

Modul ini membantu memperdalam pemahaman tentang konsep dasar pemrograman C++ seperti array, pointer, string, dan fungsi. Setiap konsep dijelaskan dengan detail mulai dari deklarasi hingga implementasi.

2. Penggunaan Array:

Array adalah struktur data yang penting dan sering digunakan dalam pemrograman untuk menyimpan dan mengakses data berurutan. Pemrogram dapat menggunakan array satu dimensi, dua dimensi, atau multidimensi sesuai kebutuhan penyimpanan data.

3. Peran Pointer:

Pointer merupakan konsep esensial dalam C++ karena memberikan kemampuan untuk mengakses dan mengubah data secara langsung di memori. Pointer juga memainkan peran penting dalam operasi dengan array dan fungsi.

4. Hubungan Antara Array dan Pointer:

Pointer dapat digunakan untuk mengelola array dengan lebih fleksibel, termasuk mengakses elemen-elemen array secara dinamis dan melakukan operasi manipulasi data.

5. Fungsi dan Prosedur:

Pemecahan program menjadi beberapa fungsi atau prosedur menjadikan kode lebih modular, terstruktur, dan mudah dikembangkan. Pemahaman tentang parameter formal dan aktual serta cara melewatkan parameter (by value, by pointer, by reference) penting untuk efisiensi pengelolaan data.

6. Penerapan Praktis:

Modul ini menekankan pada penerapan teori dalam pembuatan program, baik untuk tugas-tugas pengelolaan data sederhana seperti mencari nilai maksimum atau rata-rata dari array, maupun pemahaman alokasi memori melalui penggunaan pointer.

Secara keseluruhan, pemahaman terhadap konsep-konsep ini membantu meningkatkan keterampilan dalam pemrograman C++ dengan fokus pada efisiensi pengelolaan memori dan data [\[5†source\]](#) .