

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL II
PENGENALAN BAHASA C++**



Disusun Oleh :

NAMA : Damanik, Yohaanes Geovan Ondova

NIM : 103112400022

Dosen

FAHRUDIN MUKTI WIBOWO

**PROGRAM STUDI STRUKTUR DATA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

A. Dasar Teori

- Array adalah kumpulan data dengan nama yang sama dan setiap elemen bertipe data sama. Untuk mengakses setiap komponen / elemen array berdasarkan indeks dari setiap elemen.
 1. Array Satu Dimensi: Kumpulan data dengan nama dan tipe yang sama, diakses berdasarkan indeks (dimulai dari 0). Contoh deklarasi: `int nilai[10];`
 2. Array Dua Dimensi: Mirip tabel, digunakan untuk menyimpan data dalam bentuk tabel. Diakses dengan dua indeks. Contoh: `int data_nilai[4][3];`
 3. Array Berdimensi Banyak: Array dengan indeks lebih dari dua, lebih kompleks untuk dibayangkan.
- Pointer
 1. Data dan Memori: Semua data disimpan di memori (RAM), yang dapat diibaratkan *array* 1 dimensi dengan setiap *cell* memiliki alamat unik (*address*). Simbol `&` digunakan untuk mendapatkan alamat memori suatu variabel.
 2. Pointer dan Alamat: Variabel *pointer* menyimpan alamat memori variabel lain, sehingga dapat mengakses nilai variabel yang ditunjuknya. Deklarasi *pointer*: `type *nama_variabel;` (misal: `int *p_int;`). Simbol `*` di depan nama *pointer* digunakan untuk mendapatkan nilai dari variabel yang ditunjuknya.
 3. Pointer dan Array: Ada hubungan erat antara *array* dan *pointer*. Nama *array* tanpa indeks (misal: `a`) secara implisit adalah *pointer* ke elemen pertama *array* tersebut (`&a[0]`). *Pointer* juga bisa digunakan untuk mengakses elemen *array* dengan aritmatika *pointer* (misal: `*(pa+i)` untuk elemen `a[i]`).
 4. Pointer dan String: *String* dalam C++ pada dasarnya adalah *array* karakter yang diakhiri dengan karakter null (`\0`). *Pointer* karakter (`char *`) dapat menunjuk ke awal suatu *string*. Penting untuk membedakan antara *array* karakter yang dapat diubah isinya (`char amessage[] = "..."`) dan *pointer* ke konstanta *string* yang tidak dapat diubah isinya (`char *pmessage = "..."`).
- Fungsi adalah blok dari kode yang dirancang untuk melaksanakan tugas khusus dengan tujuan: Program menjadi terstruktur, sehingga mudah dipahami dan mudah dikembangkan. Program dibagi menjadi beberapa modul yang kecil. Dapat mengurangi pengulangan kode (duplikasi kode) sehingga menghemat ukuran program.
 1. Tujuan Fungsi: Membuat program terstruktur, mudah dipahami, dikembangkan, dan mengurangi duplikasi kode.

2. Bentuk Umum: tipe_keluaran nama_fungsi (daftar_parameter) { blok pernyataan fungsi; .
3. Nilai Balik Fungsi: Fungsi dapat mengembalikan sebuah nilai.

- Prosedur (Fungsi Void)

1. Dalam C++, prosedur adalah fungsi yang tidak mengembalikan nilai (*void*). Fungsi ini melakukan tugas tertentu tanpa memberikan *return value kepada pemanggil*.

- Parameter Fungsi

2. Parameter Formal: Variabel dalam daftar parameter saat mendefinisikan fungsi
3. Parameter Aktual: Parameter (bisa variabel, konstanta, atau ungkapan) yang digunakan saat memanggil fungsi

B. Guided (berisi screenshot source code & output program disertai penjelasannya)

Guided 1

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

int main(){
    int x, y;
    int *px;
    x = 87;
    px = &x;
    y = *px;

    //Menampilkan informasi
    cout << "Alamat x  = " << &x << endl;
    cout << "Isi px    = " << px << endl;
    cout << "Isi x     = " << x << endl;
    cout << "Nilai *px  = " << *px << endl;
    cout << "Nilai y    = " << y << endl;

    return 0;
}
```

Output

```
PS C:\Users\Lenovo\Documents\STRUKTUR DATA MODUL 2\Guided1> cd "c:\Users\Lenovo\Documents\STRUKTUR DATA MODUL 2\Guided1" ; if ($?) { .\Guided1 }
Alamat x    = 0x27eb7ffe60
Isi px      = 0x27eb7ffe60
Isi x       = 87
Nilai *px   = 87
Nilai y     = 87
PS C:\Users\Lenovo\Documents\STRUKTUR DATA MODUL 2\Guided1> █
```

Deskripsi: Program ini mendemonstrasikan konsep dasar pointer dalam bahasa C++ dengan mendeklarasikan variabel integer x dan y serta pointer px yang menunjuk ke tipe data integer. Pertama, nilai 87 dimasukkan ke dalam variabel x, kemudian pointer px diatur untuk menyimpan alamat memori dari x menggunakan operator reference (&), dan selanjutnya nilai yang ditunjuk oleh px (yaitu nilai x) di-assign ke variabel y menggunakan operator dereference (*). Kelima pernyataan cout kemudian menampilkan informasi yang konsisten: alamat x dan isi px akan sama (menunjukkan lokasi memori yang sama), sementara nilai x, *px, dan y juga akan sama (yaitu 87), membuktikan bahwa pointer px berhasil mengakses dan memanipulasi data secara tidak langsung melalui alamat memori.

Guided 2

```
#include <iostream>
#define MAX 5

using namespace std;

int main(){
    int i, j;
    float nilai[MAX]; //Array 1 dimensi
    static int nilai_tahun[MAX][MAX] = {
        {0, 2, 2, 2, 0},
        {0, 1, 1, 1, 0},
        {0, 3, 3, 3, 0},
        {4, 4, 0, 0, 4},
        {5, 0, 0, 0, 5}
    };

    //Input data array 1 dimensi
    cout << "=== Input Nilai Siswa ===\n";
    for (i = 0; i < MAX; i++){
        cout << "Masukan nilai ke-" << i + 1
        << ": ";
        cin >> nilai[i];
    }

    //Menampilkan isi array 1 dimensi
    cout << "\n=== Data Nilai Siswa ===\n";
    for (i = 0; i < MAX; i++){
```

```

        cout << "Nilai ke-" << i + 1 << " = "
        << nilai[i]
        << endl;
    }

    //Menampilkan isi array 2 dimensi
    cout << "\n=== Nilai Tahunan ===\n";
    for (i = 0; i < MAX; i++) {
        for (j = 0; j < MAX; j++){
            cout << nilai_tahun[i][j] << " ";
        }
        cout << endl;
    }

    return 0;
}

```

Output

```

PS C:\Users\Lenovo\Documents\STRUKTUR DATA MODUL 2\Guided1> cd "c:\Users\Lenovo\Documents\STRUKTUR DATA MODUL 2\Guided1" & g++ nerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
=== Input Nilai Siswa ===
Masukan nilai ke-1: 89
Masukan nilai ke-2: 90
Masukan nilai ke-3: 99
Masukan nilai ke-4: 98
Masukan nilai ke-5: 97

=== Data Nilai Siswa ===
Nilai ke-1 = 89
Nilai ke-2 = 90
Nilai ke-3 = 99
Nilai ke-4 = 98
Nilai ke-5 = 97

=== Nilai Tahunan ===
0 2 2 2 0
0 1 1 1 0
0 3 3 3 0
4 4 0 0 4
5 0 0 0 5
PS C:\Users\Lenovo\Documents\STRUKTUR DATA MODUL 2\Guided2>

```

Deskripsi:

Program ini mendemonstrasikan penggunaan array satu dimensi dan dua dimensi dalam C++. Program menginput 5 nilai siswa ke dalam array satu dimensi nilai melalui input pengguna, kemudian menampilkan semua nilai yang telah diinputkan. Selanjutnya, program menampilkan array dua dimensi nilai_tahun berukuran 5x5 yang telah diinisialisasi dengan nilai-nilai tertentu. Program ini secara efektif menunjukkan perbedaan antara array dinamis yang diisi secara interaktif dan array statis yang nilainya sudah ditentukan, serta cara mengakses dan menampilkan elemen-elemen dari kedua jenis array tersebut menggunakan perulangan bersarang.

Guided 3

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

//Mendeklarasikan prototype fungsi
int maks3(int a, int b, int c);

int main(){
    int x, y, z;
    cout << "Masukan nilai bilangan ke-1 = ";
    cin >> x;
    cout << "Masukan nilai bilangan ke-2 = ";
    cin >> y;
    cout << "Masukan nilai bilangan ke-3 = ";
    cin >> z;

    cout << "Nilai maksimumnya adalah = "
    << maks3(x, y, z);
    return 0;
}

//Badan fungsi
int maks3(int a, int b, int c) {
    int temp_max = a;
    if (b > temp_max)
        temp_max = b;
    if (c > temp_max)
        temp_max = c;
    return temp_max;
}
```

Output

```
PS C:\Users\Lenovo\Documents\STRUKTUR DATA MODUL 2\Guided3> cd "c:\Users\Lenovo\Documents\STRUKTUR DATA MODUL 2\Guided3" & gcc nerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Masukan nilai bilangan ke-1 = 7
Masukan nilai bilangan ke-2 = 8
Masukan nilai bilangan ke-3 = 9
Nilai maksimumnya adalah = 9
PS C:\Users\Lenovo\Documents\STRUKTUR DATA MODUL 2\Guided3> |
```

Deskripsi:

Program ini digunakan untuk mencari nilai terbesar dari tiga bilangan bulat yang diinputkan pengguna. Program terdiri dari fungsi maks3() yang mengembalikan nilai maksimum dengan menggunakan operator ternary untuk perbandingan, dan fungsi main() yang menangani input tiga bilangan sekaligus serta menampilkan hasilnya. Program ini mendemonstrasikan konsep fungsi, parameter, dan operator kondisional dalam pemrograman C++ dengan kode yang ringkas dan efisien.

Guided 4

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

//Prototype prosedur
void tulis(int x);

int main(){
    int jum;
    cout << "Jumlah baris kata = ";
    cin >> jum;
    tulis(jum);
    return 0;
}

//Badan Prosedur
void tulis(int x) {
    for (int i = 0; i < x; i++) {
        cout << "Baris ke-" << i + 1 << endl;
    }
}
```

Output

```
PS C:\Users\Lenovo\Documents\STRUKTUR DATA MODUL 2\Guided3> cd "c:\Users\Lenovo\Documents\STRUKTUR DATA MODUL 2\Guided4" & g++ nerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Jumlah baris kata = 7
Baris ke-1
Baris ke-2
Baris ke-3
Baris ke-4
Baris ke-5
Baris ke-6
Baris ke-7
PS C:\Users\Lenovo\Documents\STRUKTUR DATA MODUL 2\Guided4> |
```

Deskripsi:

Program ini mendemonstrasikan penggunaan prosedur (fungsi tanpa nilai kembali) dalam C++ untuk menampilkan sejumlah baris teks secara berulang. Program meminta pengguna memasukkan jumlah baris yang ingin ditampilkan, kemudian memanggil prosedur tulis() dengan parameter jumlah tersebut. Di dalam prosedur tulis, sebuah loop for akan mengeksekusi perintah output sebanyak nilai parameter yang diberikan, menampilkan teks "Baris ke-n" untuk setiap iterasi dimana n merupakan nomor urut baris. Program ini mengilustrasikan konsep modularisasi kode dengan menggunakan prosedur untuk tugas berulang dan passing parameter antara fungsi main dengan prosedur.

C. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)

Unguided 1

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

int main() {
    string nama[5];
    float nilai[5][3];
    float rata[5];
    int terbaik = 0;

    cout << "=== Input Data Mahasiswa ===" << endl;

    for (int i = 0; i < 5; i++) {
        cout << "\nNama mahasiswa ke-" << i + 1 << ": ";
        cin >> nama[i];
        float total = 0;
        for (int j = 0; j < 3; j++) {
            cout << "Nilai mata kuliah ke-" << j + 1 << ": ";
            cin >> nilai[i][j];
            total = total + nilai[i][j];
        }
        rata[i] = total / 3;
    }

    for (int i = 1; i < 5; i++) {
        if (rata[i] > rata[terbaik]) {
            terbaik = i;
        }
    }

    cout << "\n=== Daftar Nilai Mahasiswa ===" << endl;
    for (int i = 0; i < 5; i++) {
        cout << "\nNama: " << nama[i] << endl;
        cout << "Nilai: ";
        for (int j = 0; j < 3; j++) {
            cout << nilai[i][j] << " ";
        }
        cout << "\nRata-rata: " << rata[i];
        if (i == terbaik) {
            cout << " <-- Terbaik";
        }
        cout << endl;
    }

    cout << "\nMahasiswa dengan rata-rata tertinggi adalah: "
        << nama[terbaik] << " dengan nilai " << rata[terbaik] << endl;

    return 0;
}
```



```
}
```

Output

```
PS C:\Users\Lenovo\Documents\STRUKTUR DATA MODUL 2\Unguided1> cd "c:\Users\Lenovo\Do
d1.cpp -o unguided1 } ; if ($?) { .\unguided1 }
=== Input Data Mahasiswa ===

Nama mahasiswa ke-1: Damanik
Nilai mata kuliah ke-1: 88
Nilai mata kuliah ke-2: 88
Nilai mata kuliah ke-3: 89

Nama mahasiswa ke-2: Yohanes
Nilai mata kuliah ke-1: 89
Nilai mata kuliah ke-2: 89
Nilai mata kuliah ke-3: 90

Nama mahasiswa ke-3: Geovan
Nilai mata kuliah ke-1: 78
Nilai mata kuliah ke-2: 77
Nilai mata kuliah ke-3: 76

Nama mahasiswa ke-4: Ondova
Nilai mata kuliah ke-1: 89
Nilai mata kuliah ke-2: 78
Nilai mata kuliah ke-3: 77

Nama mahasiswa ke-5: Yosep
Nilai mata kuliah ke-1: 89
Nilai mata kuliah ke-2: 90
Nilai mata kuliah ke-3: 98
```

```
=== Daftar Nilai Mahasiswa ===
```

```
Nama: Damanik  
Nilai: 88 88 89  
Rata-rata: 88.3333
```

```
Nama: Yohanes  
Nilai: 89 89 90  
Rata-rata: 89.3333
```

```
Nama: Geovan  
Nilai: 78 77 76  
Rata-rata: 77
```

```
Nama: Ondova  
Nilai: 89 78 77  
Rata-rata: 81.3333
```

```
Nama: Yosep  
Nilai: 89 90 98  
Rata-rata: 92.3333 <-- Terbaik
```

```
Mahasiswa dengan rata-rata tertinggi adalah: Yosep dengan nilai 92.3333  
PS C:\Users\Lenovo\Documents\STRUKTUR DATA MODUL 2\Unguided1> █
```

Deskripsi:

Program ini mengelola data nilai 5 mahasiswa dengan menginput nama dan 3 nilai mata kuliah masing-masing, kemudian menghitung rata-rata setiap mahasiswa dan menentukan mahasiswa dengan nilai rata-rata tertinggi menggunakan array satu dimensi untuk nama dan rata-rata, serta array dua dimensi untuk menyimpan nilai per mata kuliah, yang akhirnya menampilkan seluruh data beserta penanda mahasiswa terbaik.

Unguided 2

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

int main() {
    int N;
    cout << "Masukkan jumlah elemen (N): ";
    cin >> N;

    int *ptr;
    ptr = new int[N];

    cout << "\nMasukkan " << N << " bilangan:\n";
    for (int i = 0; i < N; i++) {
        cout << "Elemen ke-" << i + 1 << ": ";
        cin >> *(ptr + i);
    }
}
```

```

    int jumlah = 0;
    int maks = *ptr;
    int min = *ptr;

    for (int i = 0; i < N; i++) {
        int nilai = *(ptr + i);
        jumlah = jumlah + nilai;

        if (nilai > maks)
            maks = nilai;
        if (nilai < min)
            min = nilai;
    }

    cout << "\n=== Hasil Perhitungan ===" << endl;
    cout << "Jumlah semua bilangan = " << jumlah << endl;
    cout << "Nilai maksimum = " << maks << endl;
    cout << "Nilai minimum = " << min << endl;

    delete[] ptr;

    return 0;
}

```

Output

```

PS C:\Users\Lenovo\Documents\STRUKTUR DATA MODUL 2\Unguided1> cd "c:\Users\Lenovo\Documents\STRUKTUR DATA MODUL 2\Unguided2" & gcc d2.cpp -o unguided2 } ; if ($?) { .\unguided2 }
Masukkan jumlah elemen (N): 5

Masukkan 5 bilangan:
Elemen ke-1: 1
Elemen ke-2: 2
Elemen ke-3: 3
Elemen ke-4: 4
Elemen ke-5: 5

=== Hasil Perhitungan ===
Jumlah semua bilangan = 15
Nilai maksimum = 5
Nilai minimum = 1
PS C:\Users\Lenovo\Documents\STRUKTUR DATA MODUL 2\Unguided2>

```

Deskripsi:

Program ini mengalokasikan memori dinamis untuk sebuah array integer berdasarkan input pengguna (N), kemudian menerima sejumlah N bilangan, dan melakukan perhitungan untuk menentukan jumlah total, nilai maksimum, dan nilai minimum dari bilangan-bilangan tersebut menggunakan pointer aritmatika, sebelum akhirnya menghapus alokasi memori untuk mencegah kebocoran memori.

Unguided 3

```

#include <iostream>
using namespace std;

float hitungRata(int nilai[], int n) {
    float total = 0;
    for (int i = 0; i < n; i++) {
        total = total + nilai[i];
    }
    return total / n;
}

void cariNilai(int nilai[], int n, int &maks, int &min) {
    maks = nilai[0];
    min = nilai[0];
    for (int i = 1; i < n; i++) {
        if (nilai[i] > maks)
            maks = nilai[i];
        if (nilai[i] < min)
            min = nilai[i];
    }
}

int main() {
    int N;
    cout << "Masukkan jumlah siswa: ";
    cin >> N;

    int nilai[N];
    for (int i = 0; i < N; i++) {
        cout << "Nilai siswa ke-" << i + 1 << ": ";
        cin >> nilai[i];
    }

    float rata = hitungRata(nilai, N);

    int tertinggi, terendah;
    cariNilai(nilai, N, tertinggi, terendah);

    cout << "\n=== Hasil Perhitungan ===" << endl;
    cout << "Rata-rata kelas : " << rata << endl;
    cout << "Nilai tertinggi : " << tertinggi << endl;
    cout << "Nilai terendah : " << terendah << endl;

    return 0;
}

```

Output

```
PS C:\Users\Lenovo\Documents\STRUKTUR DATA MODUL 2\Unguided2> cd "c:\Users\Lenovo\Documents\STRUKTUR DATA MODUL 2\Unguided3" & gcc d3.cpp -o unguided3 } ; if ($?) { .\unguided3 }
Masukkan jumlah siswa: 5
Nilai siswa ke-1: 88
Nilai siswa ke-2: 89
Nilai siswa ke-3: 90
Nilai siswa ke-4: 91
Nilai siswa ke-5: 96

=== Hasil Perhitungan ===
Rata-rata kelas : 90.8
Nilai tertinggi : 96
Nilai terendah : 88
PS C:\Users\Lenovo\Documents\STRUKTUR DATA MODUL 2\Unguided3>
```

Deskripsi:

Program ini mengelola nilai sejumlah siswa dengan menghitung statistik dasar menggunakan fungsi terpisah. Program meminta input jumlah siswa (N) dan nilai masing-masing siswa, kemudian memanfaatkan fungsi `hitungRata()` untuk menghitung nilai rata-rata kelas dan fungsi `cariNilai()` yang menggunakan parameter `reference` untuk menentukan nilai tertinggi dan terendah, yang akhirnya menampilkan ketiga hasil perhitungan tersebut..

Unuited 4

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

void tampilSegitiga(int n) {
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= i; j++) {
            cout << j << " ";
        }
        cout << endl;
    }
}

int main() {
    int n;
    cout << "Masukkan nilai n: ";
    cin >> n;

    cout << "\nPola segitiga angka:\n";
    tampilSegitiga(n);

    return 0;
}
```

Output

```
PS C:\Users\Lenovo\Documents\STRUKTUR DATA MODUL 2\Guided3> cd "c:\Users\Lenovo\Documents\STRUKTUR DATA MODUL 2\Guided3" & gcc nerFile.cpp -o tempCodeRunnerFile ; if ($?) { .\tempCodeRunnerFile }
Jumlah baris kata = 7
Baris ke-1
Baris ke-2
Baris ke-3
Baris ke-4
Baris ke-5
Baris ke-6
Baris ke-7
PS C:\Users\Lenovo\Documents\STRUKTUR DATA MODUL 2\Guided4> █
```

Deskripsi:

Program ini menampilkan pola segitiga angka dimana pengguna memasukkan nilai n yang menentukan tinggi segitiga, kemudian fungsi `tampilSegitiga()` akan mencetak pola angka dengan perulangan bersarang dimana setiap baris ke- i menampilkan deret angka dari 1 hingga i , sehingga membentuk segitiga angka yang semakin lebar ke bawah.

D. Kesimpulan

Praktikum Modul II berhasil memperkuat pemahaman tentang tiga konsep inti dalam C++: array, pointer, dan fungsi. Berdasarkan berbagai latihan yang dilakukan, array terbukti optimal untuk mengorganisir data dalam bentuk terstruktur seperti pengelolaan nilai siswa. Pointer dipahami sebagai variabel yang menyimpan alamat memori, memungkinkan manipulasi data secara dinamis dan tidak langsung. Sementara itu, fungsi dan prosedur berperan penting dalam menciptakan program yang modular dengan meminimalisir pengulangan kode, sebagaimana terlihat dalam program penghitungan statistik nilai dan pembuatan pola. Ketiga konsep ini membentuk fondasi essential untuk pengembangan kode C++ yang terorganisir, dan efisien.

E. Referensi

https://informatika.stei.itb.ac.id/~rinaldi.munir/PTI/2013-2014/KU1072_Array_CPP.pdf

<https://elektro.um.ac.id/wp-content/uploads/2016/04/Dasar-Pemrograman-Modul-7-Pointer.pdf>