

**LAPORAN PRAKTIKUM  
STRUKTUR DATA**

**MODUL II  
PENGENALAN BAHASA C++**



**Disusun Oleh :**

NAMA : Devi Nurliana

NIM : 103112400144

**Dosen**

WAHYU ANDI SAPUTRA

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

## A. Dasar Teori

C++ adalah pengembangan dari bahasa C yang dibuat oleh Bjarne Stroustrup sekitar tahun 1980-an. C++ disebut bahasa multi-paradigma, artinya bisa dipakai dengan gaya prosedural (pakai fungsi biasa), berorientasi objek (pakai class dan object), atau bahkan gabungan keduanya. C++ punya dasar-dasar seperti variabel, operator percabangan (if, switch), perulangan (for, while), dan bisa memakai class untuk membuat objek.

## B. Guided

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;
}
```

Screenshot Output

```
-MIEngine-Pid-pirpcpfq.e5f' '
Alamat x =0x5ffee4
Isi px =0x5ffee4
Isi x =87
Nilai *px =87
Nilai y =87
```

Deskripsi

Program ini digunakan untuk menunjukkan cara kerja pointer dalam bahasa C++. Di dalam program, variabel x diberi nilai awal 87, kemudian sebuah pointer bernama px digunakan untuk menyimpan alamat memori dari variabel x dengan perintah px = &x. Setelah itu, nilai yang tersimpan pada alamat tersebut diakses menggunakan operator dereferensi \*px dan hasilnya disimpan ke variabel y. Program menampilkan lima informasi utama, yaitu alamat variabel x, isi

dari pointer px (yang merupakan alamat x), nilai variabel x, nilai dari px, serta nilai y. Dari hasil tampilan tersebut, dapat dipahami bahwa pointer berfungsi menyimpan alamat suatu variabel, dan operator () memungkinkan kita untuk mengambil nilai yang tersimpan pada alamat tersebut.

## Guided 2

```
#include <iostream> //array
#define MAX 5 // UKURAN ARRAY DITETAPKAN SEBAGAI

using namespace std;

int main() {
    int i, j;
    float nilai[MAX]; // Array 1 dimensi
    static int nilai_tahun[MAX][MAX] = { // nilai tetap
        {0, 2, 2, 0, 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 (perulangan for)
    cout << "\n==== Data Nilai Siswa ===\n";
    for (i = 0; i < MAX; i++) {
        cout << "Masukkan 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;
}
}
```

### Screenshoot Output

```
== Data Nilai Siswa ==
Masukkan Nilai ke-1: 90
Masukkan Nilai ke-2: 98
Masukkan Nilai ke-3: 98
Masukkan Nilai ke-4: 76
Masukkan Nilai ke-5: 56

== Data Nilai Siswa ==
Nilai ke-1 = 90
n== Nilai Tahunan ==
0 2 2 0 0
0 1 1 1 0
0 3 3 3 0
4 4 0 0 4
5 0 0 0 5
```

### Deskripsi

Program ini berfungsi untuk menunjukkan cara penggunaan array satu dimensi dan dua dimensi dalam bahasa C++. Di dalam program, dideklarasikan konstanta MAX dengan nilai 5 sebagai ukuran tetap array. Array satu dimensi nilai[MAX] digunakan untuk menampung input nilai dari pengguna, sedangkan array dua dimensi nilaiTahun[MAX][MAX] berisi data statis yang merepresentasikan nilai-nilai tahunan. Program terlebih dahulu meminta pengguna memasukkan lima nilai ke dalam array satu dimensi melalui perulangan for, kemudian menampilkan kembali nilai-nilai tersebut ke layar. Setelah itu, program mencetak isi array dua dimensi nilaiTahun dalam bentuk tabel menggunakan dua perulangan bersarang. Pada akhirnya, program memperlihatkan bagaimana data dapat disimpan dan ditampilkan baik dalam bentuk array satu dimensi (baris tunggal) maupun array dua dimensi (bentuk matriks).

### Guided 3

```
#include <iostream> // input output program
using namespace std;

// Mendeklarasikan PROTOTYPE FUNGSI
int maks3(int a, int b, int c);

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

```

cout << "Masukkan nilai bilangan ke-3 = ";
cin >> z;

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

// Badan Fungsi ( diolah untuk dibandingkan )
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;
}

```

#### Screenshoot Output

```

Masukkan nilai bilangan ke-1 = 4
Masukkan nilai bilangan ke-2 = 2
Masukkan nilai bilangan ke-3 = 6
Nilai maksimumnya adalah = 6

```

#### Deskripsi

Program ini digunakan untuk menentukan nilai terbesar dari tiga bilangan bulat yang dimasukkan oleh pengguna. Program memanfaatkan konsep fungsi dengan parameter formal untuk membandingkan ketiga bilangan dan mengembalikan nilai yang paling besar. Pada awal program, fungsi maks3() dideklarasikan sebagai prototype, kemudian dipanggil di dalam fungsi main(). Fungsi maks3() menerima tiga parameter (a, b, dan c) dan menggunakan percabangan if untuk menemukan bilangan dengan nilai tertinggi. Nilai sementara disimpan di dalam variabel temp\_max sebelum akhirnya dikembalikan ke fungsi main() dan ditampilkan ke layar.

#### Guided 4

```

#include <iostream>
using namespace std;

// Prototype Prosedur
void tulis(int x);

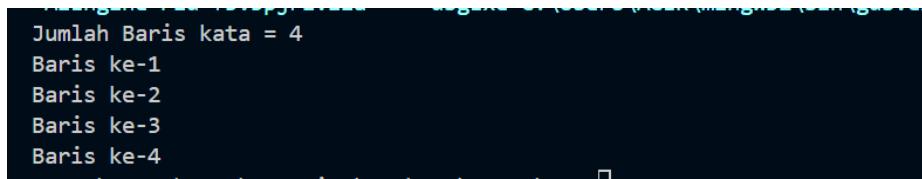
int main() {
    int jum;
    cout << "Jumlah Baris kata = ";

```

```
    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;
    }
}
```

Screenshoot Output



```
Jumlah Baris kata = 4
Baris ke-1
Baris ke-2
Baris ke-3
Baris ke-4
```

### Deskripsi

Program ini merupakan contoh sederhana penggunaan prosedur dalam bahasa C++. Program meminta pengguna memasukkan jumlah baris yang ingin dicetak, kemudian memanggil prosedur tulis() yang menerima jumlah tersebut sebagai parameter. Di dalam prosedur, digunakan perulangan for untuk menampilkan setiap baris secara berurutan dengan format "Baris ke-1", "Baris ke-2", dan seterusnya hingga jumlah yang diminta. Konsep yang diperlihatkan mencakup penerimaan input dari pengguna, pemanggilan prosedur dengan parameter, dan penggunaan looping untuk mengeksekusi perintah berulang. Program ini bermanfaat untuk memahami bagaimana fungsi/prosedur dapat memisahkan logika tertentu dari fungsi utama (main) sehingga kode lebih terstruktur dan mudah dibaca.

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

Unguided 1

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

int main() {
    const int MAHASISWA = 5;
    const int MATAKULIAH = 3;

    string nama[MAHASISWA];
    float nilai[MAHASISWA][MATAKULIAH];
    float rata[MAHASISWA];
    int baik = 0;

    cout << "Nilai Mahasiswa \n";
    cout << "===== \n";

    for (int i = 0; i < MAHASISWA; i++) {
        cout << "\nNama Mahasiswa ke-" << i + 1 << " : ";
        cin >> nama[i];

        float total = 0;
        for (int j = 0; j < MATAKULIAH; j++) {
            cout << " Nilai Mata Kuliah ke:" << j + 1 << " : ";
            cin >> nilai[i][j];
            total += nilai[i][j];
        }
        rata[i] = total / MATAKULIAH;
    }

    for (int i = 1; i < MAHASISWA; i++) {
        if (rata[i] > rata[bahik]) {
            baik = i;
        }
    }

    cout << "\n\nDaftar Nilai Mahasiswa\n";
    cout << "===== \n";
    cout << left << setw(12) << "Nama"
        << setw(10) << "Mata Kuliah 1"
        << setw(10) << "Mata Kuliah 2"
        << setw(10) << "Mata Kuliah 3"
        << setw(10) << "Rata-Rata"
```

```

        << "Keterangan \n";
cout << "-----\n";

for (int i = 0; i < MAHASISWA; i++) {
    cout << left << setw(12) << nama[i];
    for (int j = 0; j < MATAKULIAH; j++) {
        cout << setw(10) << nilai[i][j];
    }
    cout << setw(10) << fixed << setprecision(2) << rata[i];
    if (i == baik)
        cout << "Mahasiswa Terbaik";
    cout << endl;
}

cout << "===== \n";
cout << "Mahasiswa dengan rata-rata tertinggi: " << nama[bahik]
    << " (" << rata[bahik] << ") \n";

return 0;
}

```

## Screenshots Output

```

Nilai Mahasiswa
=====
Nama Mahasiswa ke-1 : Devi
Nilai Mata kuliah ke:1 : 80
Nilai Mata kuliah ke:2 : 60
Nilai Mata kuliah ke:3 : 87

Nama Mahasiswa ke-2 : Lia
Nilai Mata kuliah ke:1 : 90
Nilai Mata kuliah ke:2 : 85
Nilai Mata kuliah ke:3 : 90

Nama Mahasiswa ke-3 : Ana
Nilai Mata kuliah ke:1 : 88
Nilai Mata kuliah ke:2 : 99
Nilai Mata kuliah ke:3 : 77

Nama Mahasiswa ke-4 : Nurli
Nilai Mata kuliah ke:1 : 90
Nilai Mata kuliah ke:2 : 88
Nilai Mata kuliah ke:3 : 91

Nama Mahasiswa ke-5 : Vinur
Nilai Mata kuliah ke:1 : 76
Nilai Mata kuliah ke:2 : 87
Nilai Mata kuliah ke:3 : 65

```

Daftar Nilai Mahasiswa					
Nama	Mata Kuliah 1	Mata Kuliah 2	Mata Kuliah 3	Rata-Rata	Keterangan
Devi	88	60	87	75.67	
Lia	90.00	85.00	90.00	88.33	
Ana	88.00	99.00	77.00	88.00	
Nurli	98.00	88.00	91.00	89.67	Mahasiswa Terbaik
Vinur	76.00	87.00	65.00	76.00	

Mahasiswa dengan rata-rata tertinggi: Nurli (89.67)

Deskripsi:

Program ini berfungsi untuk mengolah dan menampilkan data nilai 3 mata kuliah dari 5 mahasiswa menggunakan array dua dimensi, dengan langkah kerja sebagai berikut:

### 1. Inisialisasi data

Program mendefinisikan konstanta MAHASISWA = 5 dan MATAKULIAH = 3.

Dua array dibuat: nama[MAHASISWA] untuk menyimpan nama mahasiswa dan nilai[MAHASISWA][MATAKULIAH] untuk menyimpan nilai 3 mata kuliah setiap mahasiswa.

Selain itu, dibuat juga array rata[MAHASISWA] untuk menyimpan rata-rata nilai serta variabel terbaik untuk menentukan mahasiswa dengan rata-rata tertinggi.

### 2. Input data mahasiswa

Program meminta pengguna memasukkan nama dan 3 nilai mata kuliah untuk masing-masing mahasiswa.

Setiap kali nilai dimasukkan, program menghitung jumlah totalnya lalu menyimpan rata-rata ke dalam array rata[i].

### 3. Menentukan mahasiswa terbaik

Setelah semua data dimasukkan, program membandingkan nilai rata-rata setiap mahasiswa untuk mencari rata-rata tertinggi.

Jika rata[i] lebih besar dari nilai rata-rata sebelumnya, indeks terbaik akan diperbarui dengan posisi mahasiswa tersebut.

### 4. Menampilkan hasil dalam bentuk tabel

Program menampilkan tabel berisi:

Nama mahasiswa

Nilai ketiga mata kuliah

Nilai rata-rata

Keterangan “Mahasiswa Terbaik” bagi yang memiliki rata-rata tertinggi.

Format tampilan tabel diatur menggunakan fungsi setw dari library <iomanip> agar kolom sejajar

dan mudah dibaca.

#### 5. Menampilkan mahasiswa dengan rata-rata tertinggi

Setelah tabel selesai dicetak, program menampilkan nama mahasiswa dengan rata-rata tertinggi beserta nilai rata-ratanya pada baris terakhir.

#### 6. Selesai

Program berakhir dan mengembalikan nilai 0.

### Unguided 2

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

void hitungPointer(int *ptr, int N, int &jumlah, int &maks, int &min);

int main() {
    int N;
    cout << "Jumlah semua elemen array: ";
    cin >> N;

    int *ptr = new int[N];

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

    int jumlah, maks, min;

    hitungPointer(ptr, N, jumlah, maks, min);
```

```

cout << "\nHasil Perhitungan:\n";
cout << "Jumlah dari keseluruhan bilangan : " << jumlah << endl;
cout << "Nilai maksimum : " << maks << endl;
cout << "Nilai minimum : " << min << endl;

delete[] ptr;
return 0;
}

void hitungPointer(int *ptr, int N, int &jumlah, int &maks, int &min) {
    jumlah = 0;
    maks = *ptr;
    min = *ptr;

    for (int i = 0; i < N; i++) {
        jumlah += *(ptr + i);
        if (*(ptr + i) > maks)
            maks = *(ptr + i);
        if (*(ptr + i) < min)
            min = *(ptr + i);
    }
}

```

### Screenshots Output

```

Jumlah semua elemen array: 4

Masukkan 4 bilangan bulat:
Elemen ke-1 : 2
Elemen ke-2 : 4
Elemen ke-3 : 6
Elemen ke-4 : 8

Hasil Perhitungan:
Jumlah dari keseluruhan bilangan : 20
Nilai maksimum : 8
Nilai minimum : 2

```

Deskripsi:

Program ini berfungsi untuk menyimpan dan menghitung data bilangan bulat menggunakan pointer ke array, dengan langkah kerja sebagai berikut:

#### 1. Input jumlah elemen array

Program meminta pengguna memasukkan jumlah elemen array (N). Nilai N akan digunakan untuk menentukan ukuran array dinamis yang dibuat dengan pointer menggunakan perintah new int[N].

#### 2. Input data melalui pointer

Program meminta pengguna memasukkan N bilangan bulat. Setiap elemen disimpan ke dalam array melalui pointer dengan ekspresi  $*(\text{ptr} + i)$  yang menunjukkan posisi alamat memori dari setiap elemen.

### 3. Pemanggilan fungsi hitungPointer

Setelah semua data dimasukkan, program memanggil fungsi hitungPointer(ptr, N, jumlah, maks, min) untuk melakukan perhitungan jumlah, nilai maksimum, dan nilai minimum dari seluruh elemen array.

### 4. Perhitungan di dalam fungsi

Di dalam fungsi hitungPointer, nilai awal maksimum dan minimum diambil dari elemen pertama  $*\text{ptr}$ . Kemudian dilakukan perulangan for untuk menjumlahkan seluruh elemen dengan jumlah  $+= *(\text{ptr} + i)$  serta membandingkan nilai setiap elemen guna menentukan nilai maksimum dan minimum dengan operasi pointer aritmatika.

### 5. Menampilkan hasil perhitungan

Setelah fungsi selesai dijalankan, program menampilkan hasil berupa jumlah keseluruhan bilangan, nilai maksimum, dan nilai minimum ke layar.

### 6. Penghapusan memori dan selesai

Setelah semua proses selesai, program menghapus alokasi memori dinamis menggunakan `delete[]` ptr, kemudian program berakhir dan mengembalikan nilai 0.

## Unguided 3

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

float hitungRerata(float data[], int jumlah);
void cariNilaiEkstrem(float data[], int jumlah, float &tertinggi, float &terendah);

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

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

```

float rataRata, tertinggi, terendah;
rataRata = hitungRerata(nilai, jumlahSiswa);
cariNilaiEkstrem(nilai, jumlahSiswa, tertinggi, terendah);

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

return 0;
}

float hitungRerata(float data[], int jumlah) {
    float total = 0;
    for (int i = 0; i < jumlah; i++) {
        total += data[i];
    }
    return total / jumlah;
}

void cariNilaiEkstrem(float data[], int jumlah, float &tertinggi, float &terendah) {
    tertinggi = data[0];
    terendah = data[0];

    for (int i = 1; i < jumlah; i++) {
        if (data[i] > tertinggi)
            tertinggi = data[i];
        if (data[i] < terendah)
            terendah = data[i];
    }
}

```

## Screenshots Output

```

Masukkan jumlah siswa: 5
Masukkan nilai siswa ke-1: 90
Masukkan nilai siswa ke-2: 80
Masukkan nilai siswa ke-3: 70
Masukkan nilai siswa ke-4: 80
Masukkan nilai siswa ke-5: 90

--- Hasil Perhitungan Nilai ---
Rata-rata kelas : 82
Nilai tertinggi : 90
Nilai terendah : 70

```

Deskripsi:

Program ini berfungsi untuk menghitung rata-rata nilai kelas serta menentukan nilai tertinggi dan terendah menggunakan fungsi dengan parameter formal berupa array, dengan langkah kerja sebagai berikut:

#### 1. Input nilai siswa

Program meminta pengguna memasukkan jumlah siswa (N), lalu membuat array nilai[N] untuk menyimpan nilai ujian masing-masing siswa. Selanjutnya, pengguna memasukkan nilai setiap siswa satu per satu melalui perulangan for.

#### 2. Pemanggilan fungsi hitungRerata

Program memanggil fungsi hitungRerata(data, jumlah) yang menggunakan parameter formal berupa array dan ukuran array. Fungsi ini menjumlahkan seluruh nilai dalam array kemudian membaginya dengan jumlah siswa untuk mendapatkan rata-rata kelas, yang kemudian dikembalikan ke fungsi main.

#### 3. Pemanggilan fungsi cariNilaiEkstrem

Program memanggil fungsi cariNilaiEkstrem(data, jumlah, tertinggi, terendah) yang juga menggunakan parameter formal array dan ukuran array. Fungsi ini membandingkan setiap elemen array untuk menentukan nilai tertinggi dan terendah, yang hasilnya disimpan pada variabel referensi tertinggi dan terendah.

#### 4. Menampilkan hasil perhitungan

Program menampilkan hasil perhitungan berupa rata-rata kelas, nilai tertinggi, dan nilai terendah ke layar dengan format yang rapi.

#### 5. Selesai

Program kemudian berakhir dan mengembalikan nilai 0.

### Unguided 4

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

void mencetakSegitiga(int n);

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

    mencetakSegitiga(n);
    return 0;
}

void mencetakSegitiga(int n) {
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= i; j++) {
```

```
        cout << j << " ";
    }
    cout << endl;
}
}
```

### Screenshots Output



```
Masukkan nilai n: 6
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
```

#### Deskripsi:

Program ini berfungsi untuk menampilkan pola segitiga angka menggunakan prosedur, dengan langkah kerja sebagai berikut:

##### 1. Input nilai n

Program meminta pengguna memasukkan sebuah bilangan bulat n melalui perintah cin >> n. Nilai n akan menentukan tinggi segitiga angka yang akan dicetak.

##### 2. Pemanggilan prosedur mencetakSegitiga

Setelah nilai n dimasukkan, program memanggil prosedur mencetakSegitiga(n) untuk menampilkan pola segitiga sesuai nilai yang diberikan oleh pengguna.

##### 3. Proses pencetakan pola

Di dalam prosedur mencetakSegitiga, digunakan dua perulangan bersarang.

Perulangan luar menggunakan for (int i = 1; i <= n; i++) yang berfungsi mengatur jumlah baris pola. Perulangan dalam menggunakan for (int j = 1; j <= i; j++) untuk mencetak angka dari 1 sampai i pada setiap baris. Setelah satu baris selesai dicetak, program mengeksekusi cout << endl untuk pindah ke baris berikutnya.

##### 4. Selesai

Setelah semua baris tercetak sesuai nilai n, program berakhir dan mengembalikan nilai 0.

### D. Kesimpulan

Dari praktikum yang telah dilaksanakan dapat disimpulkan bahwa setiap latihan memiliki tujuan berbeda namun saling berkaitan dalam memahami konsep dasar C++. Praktikum ini melatih penggunaan pointer, array satu dan dua dimensi, fungsi dengan parameter formal, prosedur, serta perulangan bersarang. Selain itu, peserta juga mempelajari cara membandingkan beberapa bilangan untuk menentukan nilai tertinggi menggunakan fungsi dengan parameter formal. Seluruh kegiatan bertujuan untuk memperkuat pemahaman dalam pengolahan data dan penulisan kode yang terstruktur serta efisien.

## E. Referensi

1. Deitel, H. M., & Deitel, P. J. (2013). C++ How to Program (8th Edition). Pearson Education.
2. Malik, D. S. (2011). C++ Programming: From Problem Analysis to Program Design (6th Edition). Cengage Learning.
3. Stroustrup, B. (2013). The C++ Programming Language (4th Edition). Addison-Wesley.
4. Sutanta, E. (2011). Pemrograman C++ untuk Pemula dan Mahir. Andi Publisher.
5. Nugroho, A. (2018). Logika dan Algoritma Pemrograman dengan C++. Deepublish.