

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL 2
PENGENALAN BAHASA C++
(BAGIAN KEDUA)**



Disusun Oleh :

NAMA : Muhammad Omar Nativ
NIM : 103112430063

Dosen

Fahrudin Mukti Wibowo, S.Kom., M.Eng.

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

A. Dasar Teori

C++ merupakan bahasa pemrograman tingkat tinggi yang dikembangkan dari bahasa C dan mendukung pemrograman berorientasi objek maupun terstruktur. Pada modul ini, pembahasan difokuskan pada konsep dasar seperti array, pointer, string, serta fungsi dan prosedur.

Array digunakan untuk menyimpan sekumpulan data dengan tipe yang sama secara berurutan dalam memori dan diakses menggunakan indeks. Pointer berfungsi untuk menyimpan alamat memori suatu variabel, sehingga memungkinkan manipulasi data secara langsung dan efisien, serta memiliki keterkaitan erat dengan array maupun string dalam pengelolaan data. String sendiri pada dasarnya merupakan array dari karakter yang diakhiri dengan simbol null ('\0'). Fungsi dalam C++ digunakan untuk memecah program menjadi bagian-bagian kecil agar lebih terstruktur dan mudah dipelihara, sedangkan prosedur atau fungsi void digunakan untuk melaksanakan tugas tertentu tanpa mengembalikan nilai.

Selain itu, konsep parameter juga penting dipahami, di mana C++ menyediakan tiga cara pengiriman data, yaitu call by value, call by pointer, dan call by reference, yang masing-masing memiliki pengaruh berbeda terhadap variabel yang diproses di dalam 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;

    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;
}

```

Screenshots Output

```

nativ@omarnadip telkom-c++ % cd "/Users/nativ/telkom-c++/Modul 02/Guided/" && g++ guided1.cpp -o guided1 && "/Users/nativ/telkom-c++/Modul 02/Guid
ed/"guided1
Alamat x   = 0x16f0f6d0c
Isi px    = 0x16f0f6d0c
Isi x     = 87
Nilai *px = 87
Nilai y   = 87
nativ@omarnadip Guided %

```

Deskripsi:

Program di atas merupakan contoh penggunaan pointer dalam C++. Variabel x bertipe int diberi nilai 87, kemudian pointer px diisi dengan alamat memori dari x menggunakan operator &. Yang berarti, px menunjuk ke lokasi memori tempat nilai x disimpan. Selanjutnya, variabel y diberi nilai *px, yaitu nilai yang ditunjuk oleh pointer px, sehingga y juga bernilai 87. Saat program dijalankan, akan ditampilkan alamat memori x, isi dari pointer px (yang sama dengan alamat x), nilai x, nilai yang ditunjuk oleh px, dan nilai y.

Guided 2

```
#include <iostream>
#define MAX 5
using namespace std;

int main()
{
    int i, j;
    float nilai_total, rata_rata;
    float nilai[MAX];

    static int nilai_tahun[MAX][MAX] =
    {
        {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}};

    cout << "\n=== input nilai siswa ===\n";
    for (i = 0; i < MAX; i++)
    {
        cout << "masukkan nilai ke-" << i + 1 << " = ";
        cin >> nilai[i];
    }

    cout << "\ndata nilai siswa :\n";
    for (i = 0; i < MAX; i++)
    {
        cout << "nilai k-" << i + 1 << " = " << nilai[i] << endl;
    }

    cout << "\nnilai tahunan : \n";
    for (i = 0; i < MAX; i++)
    {
        for (j = 0; j < MAX; j++)
            cout << nilai_tahun[i][j] << " ";
        cout << endl;
    }
}
```

Screenshots Output

```
nadiv@omarnadip Guided % cd "/Users/nadiv/telkom-c++/Modul 02/Guided/" && g++ guided2.cpp -o guided2 && "/Users/nadiv/telkom-c++/Modul 02/Guided/"
guided2

=== input nilai siswa ===
masukkan nilai ke-1 = 90
masukkan nilai ke-2 = 95
masukkan nilai ke-3 = 48
masukkan nilai ke-4 = 99
masukkan nilai ke-5 = 10

data nilai siswa :
nilai k-1 = 90
nilai k-2 = 95
nilai k-3 = 48
nilai k-4 = 99
nilai k-5 = 10

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
nadiv@omarnadip Guided %
```

Deskripsi:

Program di atas merupakan contoh penggunaan array satu dimensi dan dua dimensi dalam C++. Array `nilai[MAX]` digunakan untuk menyimpan lima nilai input dari pengguna, sedangkan array dua dimensi `nilai_tahun[MAX][MAX]` berisi data tetap yang merepresentasikan nilai siswa dalam beberapa periode atau tahun. Program dimulai dengan meminta pengguna memasukkan lima nilai, kemudian menampilkan kembali nilai-nilai yang telah diinput. Setelah itu, program menampilkan isi dari array dua dimensi `nilai_tahun` dalam bentuk tabel 5×5.

Guided 3

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

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 << endl;
    cout << "nilai maksimumnya adalah = " << maks3(x, y, z) << endl;
}

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);
}
```

Screenshots Output

```
nadiv@omarnadip Guided % cd "/Users/nadiv/telkom-c++/Modul 02/Guided/" && g++ guided3.cpp -o guided3 && "/Users/nadiv/telkom-c++/Modul 02/Guided/"
guided3
masukkan nilai bilangan ke-1 = 98
masukkan nilai bilangan ke-2 = 45
masukkan nilai bilangan ke-3 = 90

nilai maksimumnya adalah = 98
nadiv@omarnadip Guided %
```

Deskripsi:

Program di atas merupakan contoh penggunaan fungsi dalam C++. Fungsi maks3() digunakan untuk menentukan nilai terbesar dari tiga bilangan yang dimasukkan oleh pengguna. Pada bagian main(), program meminta input tiga bilangan integer (x, y, z), kemudian memanggil fungsi maks3(x, y, z) untuk mencari nilai maksimum di antara ketiganya.

Di dalam fungsi, nilai awal temp_max diisi dengan a, lalu dibandingkan secara berurutan dengan b dan c menggunakan struktur kontrol if. Jika salah satu nilai lebih besar, maka temp_max diperbarui. Setelah semua perbandingan selesai, fungsi mengembalikan nilai terbesar ke fungsi main() untuk ditampilkan di layar.

Guided 4

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

void tulis(int x);

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

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

Screenshots Output

```
nadiv@omarnadip Guided % cd "/Users/nadiv/telkom-c++/Modul 02/Guided/" && g++ guided4.cpp -o guided4 && "/Users/nadiv/telkom-c++/Modul 02/Guided/"
guided4
jumlah baris kata = 9
baris ke-1
baris ke-2
baris ke-3
baris ke-4
baris ke-5
baris ke-6
baris ke-7
baris ke-8
baris ke-9
nadiv@omarnadip Guided %
```

Deskripsi:

Program diatas merupakan contoh penggunaan prosedur dalam C++. Fungsi tulis() bertipe void, artinya fungsi tersebut tidak mengembalikan nilai, melainkan hanya menjalankan perintah tertentu. Pada bagian main(), program meminta pengguna untuk memasukkan jumlah baris yang ingin ditampilkan, kemudian memanggil fungsi tulis(jum) dengan nilai input tersebut sebagai argumen. Di dalam fungsi tulis(), digunakan perulangan for untuk mencetak kalimat "baris ke-" diikuti dengan nomor urutan baris sesuai jumlah yang dimasukkan.

- 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;

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

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

    cout << "\nDaftar Nilai Mahasiswa:\n";
    cout << "Nama\tNilai1\tNilai2\tNilai3\tRata-rata\n";

    for (int i = 0; i < 5; i++) {
        cout << nama[i] << "\t";
        for (int j = 0; j < 3; j++) {
            cout << nilai[i][j] << "\t";
        }
        cout << rata[i];
    }
```

```

        if (i == terbaik) cout << " <- Terbaik";
        cout << endl;
    }

    cout << "Mahasiswa terbaik: " << nama[terbaik] << " dengan rata-rata " <<
rata[terbaik] << endl;
}

```

Screenshots Output

```

nativ@omarnadip Unguided % cd "/Users/nativ/telkom-c++/Modul 02/Unguided/" && g++ soal1.cpp -o soal1 && "/Users/nativ/telkom-c++/Modul 02/Unguided/"
./soal1
Masukkan nama mahasiswa ke-1: nativ
Nilai mata kuliah 1: 99
Nilai mata kuliah 2: 99
Nilai mata kuliah 3: 99

Masukkan nama mahasiswa ke-2: bella
Nilai mata kuliah 1: 100
Nilai mata kuliah 2: 97
Nilai mata kuliah 3: 99

Masukkan nama mahasiswa ke-3: tapir
Nilai mata kuliah 1: 87
Nilai mata kuliah 2: 92
Nilai mata kuliah 3: 91

Masukkan nama mahasiswa ke-4: meja
Nilai mata kuliah 1: 99
Nilai mata kuliah 2: 95
Nilai mata kuliah 3: 78

Masukkan nama mahasiswa ke-5: papan
Nilai mata kuliah 1: 100
Nilai mata kuliah 2: 100
Nilai mata kuliah 3: 86

Daftar Nilai Mahasiswa:
Nama  Nilai1 Nilai2 Nilai3 Rata-rata
nativ 99 99 99 99 <- Terbaik
bella 100 97 99 98.6667
tapir 87 92 91 90
meja 99 95 78 90.6667
papan 100 100 86 95.3333
Mahasiswa terbaik: nativ dengan rata-rata 99
nativ@omarnadip Unguided %

```

Deskripsi:

Program di atas merupakan contoh penggunaan array dua dimensi untuk menyimpan dan mengolah data nilai mahasiswa. Program meminta pengguna untuk memasukkan nama lima mahasiswa dan tiga nilai mata kuliah untuk masing-masing mahasiswa. Setiap kali nilai dimasukkan, program menghitung total nilai lalu mencari rata-rata dengan membaginya dengan jumlah mata kuliah, yaitu 3. Setelah semua data di input, program membandingkan seluruh nilai rata-rata untuk menentukan mahasiswa dengan rata-rata tertinggi.

Selanjutnya, program menampilkan tabel daftar nilai berisi nama, tiga nilai mata kuliah, dan rata-rata masing-masing mahasiswa. Mahasiswa dengan rata-rata tertinggi diberi tanda "<- Terbaik" di samping datanya. Terakhir, program menampilkan hasil akhir berupa nama mahasiswa terbaik beserta nilai rata-ratanya.

Unguided 2

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

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

    int *ptr = new int[N];

    cout << "Masukkan " << N << " bilangan:\n";
    for (int i = 0; i < N; i++) {
        cin >> *(ptr + i);
    }

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

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

    cout << "\nHasil Perhitungan:\n";
    cout << "Jumlah = " << jumlah << endl;
    cout << "Maksimum = " << maks << endl;
    cout << "Minimum = " << min << endl;

    delete[] ptr;
}
```

Screenshots Output

```
nadiv@omarnadip Unguided % cd "/Users/nadiv/telkom-c++/Modul 02/Unguided/" && g++ soal2.cpp -o soal2 && "/Users/nadiv/telkom-c++/Modul 02/Unguided/"soal2
Masukkan jumlah bilangan (N): 3
Masukkan 3 bilangan:
98
75
21

Hasil Perhitungan:
Jumlah = 194
Maksimum = 98
Minimum = 21
nadiv@omarnadip Unguided %
```

Deskripsi:

Program di atas merupakan contoh penggunaan pointer dan pointer aritmatika dalam C++ untuk mengakses elemen-elemen array secara langsung melalui alamat memori. Program dimulai dengan meminta pengguna memasukkan jumlah bilangan N, kemudian mengalokasikan memori secara dinamis menggunakan `new` untuk menampung N bilangan bulat. Setiap elemen array diisi melalui pointer dengan ekspresi `*(ptr + i)`, yang berarti mengakses nilai pada alamat memori ke-i dari pointer `ptr`.

Setelah semua bilangan dimasukkan, program melakukan perulangan untuk menghitung jumlah seluruh bilangan, serta mencari nilai maksimum dan minimum dengan menggunakan operasi pointer yang sama. Hasil perhitungan kemudian ditampilkan ke layar. Di akhir program, digunakan perintah `delete[] ptr` untuk menghapus alokasi memori dinamis agar tidak terjadi kebocoran memori.

Unguided 3

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

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

void cariNilai(float nilai[], int n, float &maks, float &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;

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

    float rata = hitungRata(nilai, N);
    float maks, min;
    cariNilai(nilai, N, maks, min);

    cout << "\nHasil Perhitungan:" << endl;
    cout << "Rata-rata kelas = " << rata << endl;
    cout << "Nilai tertinggi = " << maks << endl;
    cout << "Nilai terendah = " << min << endl;
}
```

Screenshots Output

```
nadiv@omarnadip Unguided % cd "/Users/nadiv/telkom-c++/Modul 02/Unguided/" && g++ soal3.cpp -o soal3 && "/Users/nadiv/telkom-c++/Modul 02/Unguided/"soal3
Masukkan jumlah siswa: 3
Nilai siswa ke-1 : 89
Nilai siswa ke-2 : 100
Nilai siswa ke-3 : 96

Hasil Perhitungan:
Rata-rata kelas = 95
Nilai tertinggi = 100
Nilai terendah = 89
nadiv@omarnadip Unguided %
```

Deskripsi:

Program di atas menggunakan fungsi dengan parameter formal berupa array untuk mengolah nilai ujian beberapa siswa. Fungsi `hitungRata()` menghitung rata-rata dari seluruh elemen array, sedangkan fungsi `cariNilai()` mencari nilai tertinggi dan terendah menggunakan parameter referensi (&). Di dalam `main()`, pengguna memasukkan jumlah siswa dan nilai masing-masing, lalu program menampilkan hasil berupa rata-rata kelas, nilai tertinggi, dan nilai terendah.

Unguided 4

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

void tampilPola(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;

    tampilPola(n);
}
```

Screenshots Output



```
nadiv@omarnadip Unguided % cd "/Users/nadiv/telkom-c++/Modul 02/Unguided/" && g++ soal4.cpp -o soal4 && "/Users/nadiv/telkom-c++/Modul 02/Unguided/"soal4
Masukkan nilai n: 5
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
nadiv@omarnadip Unguided %
```

Deskripsi:

Program di atas menggunakan prosedur (fungsi void) untuk menampilkan pola segitiga angka berdasarkan input pengguna. Prosedur tampilPola() menerima satu parameter berupa bilangan bulat n dan menggunakan dua perulangan untuk mencetak angka dari 1 hingga i pada setiap baris. Di dalam fungsi main(), pengguna diminta memasukkan nilai n, kemudian prosedur dipanggil untuk menampilkan hasilnya. Jika nilai n adalah 5, maka program akan mencetak pola segitiga angka dari 1 hingga 5 secara bertingkat.

D. Kesimpulan

Pada modul 2 ini, saya mempelajari konsep yang lebih lanjut dalam pemrograman C++, yaitu penggunaan array satu dan dua dimensi, pointer, serta fungsi dan prosedur. Melalui latihan-latihan dalam modul ini, saya jadi lebih memahami cara mengolah data menggunakan array, memanfaatkan pointer untuk mengakses dan memproses nilai melalui alamat memori, serta membagi program menjadi bagian-bagian yang lebih terstruktur menggunakan fungsi dan prosedur. Pembelajaran ini membantu saya memahami logika program yang lebih efisien dan terorganisir.

E. Referensi

<https://algorit.ma/blog/apa-itu-c-2022/>

<https://id.wikipedia.org/wiki/C%2B%2B>

<http://educba.com/star-patterns-in-c-plus-plus/>

<https://www.petanikode.com/cpp-array/>