

LAPORAN PRAKTIKUM STRUKTUR DATA

MODUL I PENGENALAN CODE BLOCKS



Disusun Oleh :
Nadia Tambunan
103122400005

Dosen
Diah Septiani S.Kom M.Cs

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

A. Dasar Teori

Algoritma pemrograman dengan C++ merupakan serangkaian langkah logis yang dirancang untuk menyelesaikan masalah melalui penggunaan bahasa pemrograman C++ yang mendukung paradigma pemrograman berorientasi objek. Bahasa ini memungkinkan implementasi algoritma secara efisien dan terstruktur menggunakan konsep dasar seperti percabangan, perulangan, fungsi, dan manipulasi data dengan array, pointer, serta struktur data lainnya. Selain itu, C++ memberikan fleksibilitas dalam pengelolaan memori dan modularisasi kode, sehingga algoritma yang dihasilkan dapat bersifat efektif dan optimal untuk berbagai aplikasi komputer. Pendekatan ini membuat C++ sangat baik untuk pembelajaran algoritma karena memberikan pemahaman mendalam baik dari sisi teori maupun praktek pemrograman modern yang relevan hingga saat ini.

B. Guided

Source Code Guide 1 & 2

```
#include <iostream> // Library standar untuk input/output (cout, cin, dll)
#include <conio.h> // Library lama, dipakai untuk fungsi getch()
#define MAX 5 // constanta yg biasanya untuk array

using namespace std; // Supaya kita bisa pakai cout tanpa harus nulis std::cout

int main() // Nadia Tambunan 103122400005
{
    // ini bahasan variable
    int x,y; // Deklarasi variabel biasa, tipe integer
    int *px; // Deklarasi pointer ke integer (px bisa menyimpan alamat variabel int)
    x = 87; // Isi variabel x dengan nilai 87
    px = &x; // Isi pointer px dengan alamat dari x (px menunjuk ke x)
    y = *px; // Ambil nilai dari alamat yang ditunjuk px (y = isi dari x = 87)

    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;

    // INI BAHASAN ARRAY
    int i,j;
    float nilai[MAX]; // Array 1 dimensi, ukurannya MAX
    static int nilai_tahun[MAX][MAX] = { // Array 2 Dimensi (5x5)
        {0, 2, 2, 0, 0},
        {0, 1, 1, 1, 0},
```

```

    {0, 3, 3, 3, 0},
    {4, 4, 0, 0, 4},
    {5, 2, 2, 0, 5}
};

// Input data Array 1 dimensi
for (i = 0; i < MAX; i++) {
    cout << "\nMasukkan nilai ke-" << i + 1 << " : ";
    cin >> nilai[i]; // cin buat input
};

// Menampilkan isi array 1 dimensi
cout << "\nData nilai siswa:\n";
for (i = 0; i < MAX; i++) {
    cout << "Nilai ke-" << i + 1 << " = " << nilai[i] << endl;
} // Nadia Tambunan 103122400005

// Menampilkan isi array 2 dimensi
cout << "\nNilai Tahunan : \n";
for(i = 0; i < MAX; i++) {
    for(j = 0; j < MAX; j++) {
        cout << nilai_tahun[i][j] << " ";
    }
    cout << endl; // Pindah baris setelah 1 baris selesai ditampilkan
}

getch(); // Tunggu user menekan tombol apa saja sebelum program berhenti
return 0; // Program selesai dengan status "berhasil"
}

```

Screenshots Output

```
Alamat x = 0x5ffeb8
Isi px = 0x5ffeb8
Isi x = 87
Nilai *px = 87
Nilai y = 87

Masukkan nilai ke-1 : 4

Masukkan nilai ke-2 : 2

Masukkan nilai ke-3 : 3

Masukkan nilai ke-4 : 8

Masukkan nilai ke-5 : 1

Data nilai siswa:
Nilai ke-1 = 4
Nilai ke-2 = 2
Nilai ke-3 = 3
Nilai ke-4 = 8
Nilai ke-5 = 1

Nilai Tahunan :
0 2 2 0 0
0 1 1 1 0
0 3 3 3 0
4 4 0 0 4
5 2 2 0 5
PS C:\Users\LENOVO\Documents\====NGODING====\Belajar CPP> Nadia Tambunan 103122400005
```

Deskripsi

Pada pertemuan pertama ini, saya mereview dasar algoritma pemrograman, namun menggunakan C++ (dulunya pake java soalnya). Di sini isi kodingan saya tidak jauh-jauh dari variabel, pointer, dan array. Saya memahami bahwa variabel menyimpan nilai, pointer menyimpan alamat memori, dan array digunakan untuk menyimpan data sejenis baik dalam satu dimensi maupun dua dimensi (kalo beda jenis, namanya Tuple). Materi ini membantu saya mengingat kembali bagaimana data disimpan dan diakses di memori. Hal ini sejalan dengan penelitian Donyina & Heckel (2025) yang menekankan pentingnya visualisasi untuk mempermudah pemahaman pointer, serta Yao dkk. (2022) yang menyoroti perlunya metode pengajaran pemrograman yang efektif agar konsep dasar seperti ini lebih mudah dipahami mahasiswa.

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

Unguided Soal 1

```
struct nilaiSTD {  
    double CLO1;  
    double CLO2;  
    double CLO3;  
    double CLO4; // Nadia Tambunan 103122400005  
    double nilaiAkhir;  
    string indeksNilai;  
};
```

Unguided Soal 2

```
nilaiSTD mhs;
```

Unguided Soal 3

```
cout << "Masukkan nilai CLO-1 : ";  
cin >> mhs.CLO1;  
cout << "Masukkan nilai CLO-2 : ";  
cin >> mhs.CLO2;  
cout << "Masukkan nilai CLO-3 : ";  
cin >> mhs.CLO3;  
cout << "Masukkan nilai CLO-4 : ";  
cin >> mhs.CLO4; // Nadia Tambunan 103122400005
```

Screenshots Output

```
Masukkan nilai CLO-1 : 80  
Masukkan nilai CLO-2 : 90  
Masukkan nilai CLO-3 : 70  
Masukkan nilai CLO-4 : 60
```

Unguided Soal 4

```
nilaiSTD buatNilaiSTD(double CLO1, double CLO2, double CLO3,  
double CLO4) {}
```

Unguided Soal 5

```
nilaiSTD buatNilaiSTD(double CLO1, double CLO2, double CLO3,
double CLO4) {
    nilaiSTD data;
    data.CLO1 = CLO1;
    data.CLO2 = CLO2;
    data.CLO3 = CLO3;
    data.CLO4 = CLO4;

    // no. 5
    data.nilaiAkhir = hitungNilaiAkhir(CLO1, CLO2, CLO3, CLO4);
    data.indeksNilai = tentukanIndeks(data.nilaiAkhir);

    return data;
}
```

Unguided Soal 6

```
double hitungNilaiAkhir(double CLO1, double CLO2, double CLO3,
double CLO4) {
    return (0.3 * CLO1) + (0.3 * CLO2) + (0.2 * CLO3) + (0.2 *
CLO4);
}
```

Unguided Soal 7

```
string tentukanIndeks(double nilaiAkhir) {
    if (nilaiAkhir >= 85) return "A";
    else if (nilaiAkhir >= 80) return "AB";
    else if (nilaiAkhir >= 70) return "B";
    else if (nilaiAkhir >= 65) return "BC";
    else if (nilaiAkhir >= 55) return "C";
    else if (nilaiAkhir >= 40) return "D";
    else return "E";
}
```

Screenshots Output

```
Masukkan nilai CLO-1 : 80
Masukkan nilai CLO-2 : 90
Masukkan nilai CLO-3 : 70
Masukkan nilai CLO-4 : 60

HASIL PERHITUNGAN
Nilai Akhir : 77
Indeks Nilai : B
PS C:\Users\LENOVO\Documents\====Mata Kuliah====\Semester 3\Struktur Data\Tugas\Pertemuan 1\Unguided> nadia tambunan 103122400005
```

Deskripsi

Kode-kode tersebut diawali dengan kamus (deklarasi variabel), dan Algoritma (baris kode berisi perintah). susunan tersebut akhirnya membangun sebuah program yang dapat digunakan :>

E. Kesimpulan

Pertemuan pertama ini intinya adalah mengingat kembali bagaimana konsep Dasar Algoritma Pemrograman (bedanya, sekarang pake C++. dulu mah pake Java). Kira-kira, udah review tentang variabel yang bertugas menyimpan data, membuat program dimana sistem bisa menerima input user, dan sebagainya.

F. Referensi

Riyanti, & Yahfizham. (2023). Konsep dasar algoritma pemrograman. *Jurnal Arjuna: Publikasi Ilmu Pendidikan, Bahasa dan Matematika*, 1(6), 268–277. <https://doi.org/10.61132/arjuna.v1i6.333>

Donyina, A., & Heckel, R. (2025). Pedagogy of teaching pointers in the C programming language using graph transformations. ResearchGate. <https://www.researchgate.net/publication/390214273>

Yao, D., Zhang, X., & Liu, Y. (2022). Teaching reform in C programming course from the perspective of sustainable development: Construction and 9-year practice of “Three Classrooms–Four Integrations–Five Combinations” teaching model. *Sustainability*, 14(22), 15226. <https://doi.org/10.3390/su142215226>