

LAPORAN PROJECT UTS

STRUKTUR DATA



Tema :

Sistem penjadwalan tugas menggunakan stack dengan fitur Menambah tugas baru ke dalam stack, Menampilkan tugas terakhir yang ditambahkan, Menghapus tugas terakhir

Mata Kuliah : Struktur Data

Dosen Pengampu : Riza Akhsani Setyo Prayoga, S.Kom., M.MT.
(0730049402)

Kelompok 2 :

1. Fitri Romadhona (23050974179)
2. Amelia Nadhiva Rinda Putri (23050974171)

UNIVERSITAS NEGERI SURABAYA
FAKULTAS TEKNIK
JURUSAN TEKNIK INFORMATIKA
PRODI S1 PENDIDIKAN TEKNOLOGI INFORMASI 2024

LEMBAR KONTRIBUSI

A. Biodata dan Kontribusi Anggota 1

Nama Lengkap	Fitri Romadhona
NIM	23050974179
Program Studi	S1 Pendidikan Teknologi Informasi
Email	Fitri.179@mhs.unesa.ac.id

B. Kontribusi Anggota 1

- Membuat Gambaran Project
- Membuat dan Menyusun Source Code
- Membuat Flowchart
- Menyusun Laporan

C. Biodata Anggota 2

Nama Lengkap	Amelia Nadhiva Rinda Putri
NIM	23050974171
Program Studi	S1 Pendidikan Teknologi Informasi
Email	amelia.171@mhs.unesa.ac.id

D. Kontribusi Anggota 2

- Mencari Referensi Project
- Membuat dan Menyusun Source Code

BAB I GAMBARAN PROJECT

Proyek ini merupakan implementasi sederhana dari Sistem Penjadwalan Tugas yang menggunakan struktur data stack untuk menyimpan dan mengelola tugas-tugas. Stack adalah salah satu struktur data linear yang bekerja dengan prinsip LIFO (Last In, First Out), artinya elemen terakhir yang dimasukkan ke dalam stack akan menjadi elemen pertama yang dikeluarkan. Dalam sistem penjadwalan tugas yang membutuhkan prioritas berdasarkan urutan tugas yang ditambahkan, di mana tugas yang terakhir ditambahkan harus diselesaikan terlebih dahulu. Sistem ini dirancang menggunakan bahasa pemrograman C++,

Sistem penjadwalan ini memiliki beberapa fitur utama:

1. Menambah Tugas.
2. Menampilkan Tugas Terakhir.
3. Menghapus Tugas Terakhir.
4. Menampilkan Semua Tugas.

BAB II SOURCE CODE , PENJELASAN dan FLOWCHART

1. Source Code dan Penjelasan

```
#include <iostream>
#include <stack>
#include <string>

using namespace std;

class TaskScheduler {
private:
    stack<string> tasks;

public:
    // Menambah tugas baru ke dalam stack
    void addTask(const string& task) {
        tasks.push(task);
        cout << "Tugas \"" << task << "\" berhasil ditambahkan.\n";
    }

    // Menampilkan tugas terakhir yang ditambahkan
    void showLastTask() {
        if (!tasks.empty()) {
            cout << "Tugas terakhir: " << tasks.top() << endl;
        } else {
            cout << "Tidak ada tugas yang tersedia.\n";
        }
    }

    // Menghapus tugas terakhir
    void removeLastTask() {
        if (!tasks.empty()) {
            cout << "Tugas \"" << tasks.top() << "\" dihapus.\n";
            tasks.pop();
        } else {
            cout << "Tidak ada tugas yang tersedia untuk dihapus.\n";
        }
    }

    // Menampilkan semua tugas
    void showAllTasks() {
        if (tasks.empty()) {
            cout << "Tidak ada tugas yang tersedia.\n";
            return;
        }

        cout << "Daftar tugas:\n";
        stack<string> tempStack = tasks; // Membuat copy stack
        while (!tempStack.empty()) {
            cout << "- " << tempStack.top() << endl;
            tempStack.pop();
        }
    }
};

int main() {
```

```

TaskScheduler scheduler;

// Menambah tugas otomatis
scheduler.addTask("Tugas 1: Grafika Komputer");
scheduler.addTask("Tugas 2: Algoritma");
scheduler.addTask("Tugas 3: Struktur Data");
scheduler.addTask("Tugas 3: Basis Data");

// Menampilkan tugas terakhir
scheduler.showLastTask();

// Menghapus tugas terakhir
scheduler.removeLastTask();

// Menampilkan semua tugas
scheduler.showAllTasks();

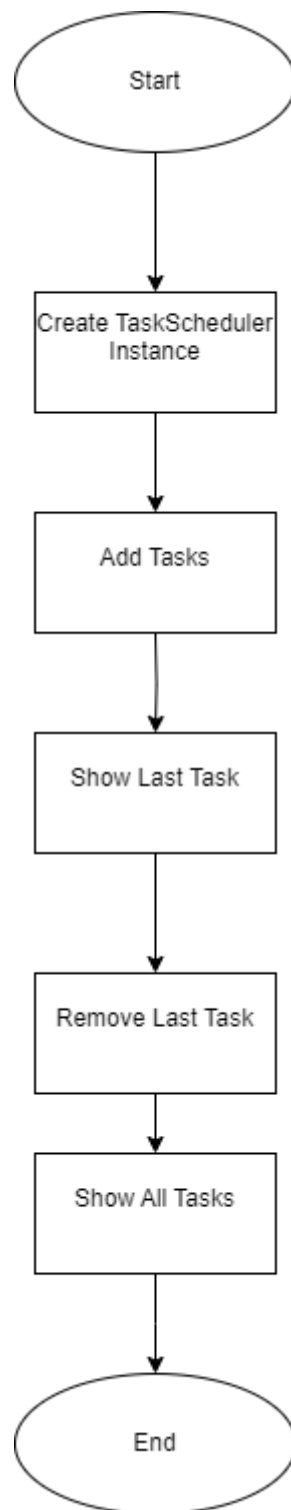
return 0;
}

```

2. Penjelasan

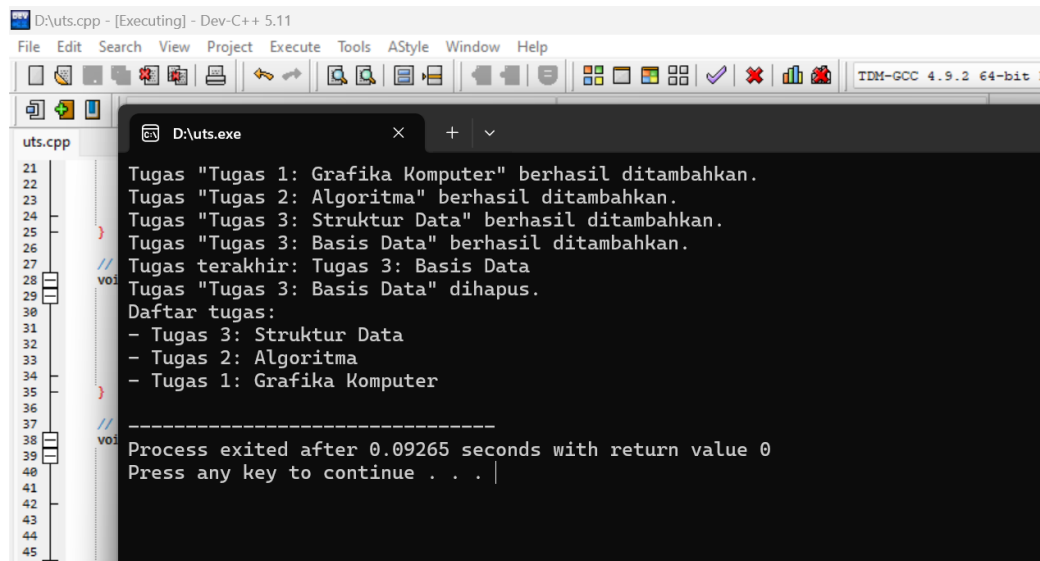
1. Mengimplementasikan kelas TaskScheduler, yang memiliki atribut privat tasks, yaitu sebuah stack dari string untuk menyimpan daftar tugas.
2. Metod addTask(const string& task): Untuk menambahkan tugas baru ke dalam stack dan menampilkan pesan konfirmasi.
3. Metod showLastTask(): Menampilkan tugas terakhir yang ditambahkan. Jika tidak ada tugas, akan menampilkan pesan bahwa tidak ada tugas yang tersedia.
4. Metod removeLastTask(): Menghapus tugas terakhir dari stack dan menampilkan pesan konfirmasi. Jika tidak ada tugas, akan menampilkan pesan yang sesuai.
5. Metod showAllTasks(): Menampilkan semua tugas yang ada di stack. Jika stack kosong, akan menampilkan pesan bahwa tidak ada tugas yang tersedia.
6. Metode main() menguji fungsionalitas sistem dengan menambahkan beberapa tugas, menampilkan tugas terakhir, menghapus tugas terakhir, dan kemudian menampilkan semua tugas yang ada.

3. Flowchart



BAB III HASIL , PENJELASAN dan KESIMPULAN

1. Hasil Project



```
D:\uts.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
D:\uts.exe
Tugas "Tugas 1: Grafika Komputer" berhasil ditambahkan.
Tugas "Tugas 2: Algoritma" berhasil ditambahkan.
Tugas "Tugas 3: Struktur Data" berhasil ditambahkan.
Tugas "Tugas 3: Basis Data" berhasil ditambahkan.
Tugas terakhir: Tugas 3: Basis Data
Tugas "Tugas 3: Basis Data" dihapus.
Daftar tugas:
- Tugas 3: Struktur Data
- Tugas 2: Algoritma
- Tugas 1: Grafika Komputer
-----
Process exited after 0.09265 seconds with return value 0
Press any key to continue . . .
```

2. Penjelasan

- Menambah Tugas: Program akan menambahkan empat tugas ke dalam stack dan mengonfirmasi setiap penambahan.
- Menampilkan Tugas Terakhir: Program akan menampilkan tugas terakhir yang ditambahkan (dalam hal ini, "Tugas 3: Basis Data").
- Menghapus Tugas Terakhir: Program akan menghapus tugas terakhir (yaitu "Tugas 3: Basis Data") dan memberikan konfirmasi tentang penghapusan.
- Menampilkan Semua Tugas: Program akan menampilkan daftar tugas yang tersisa dalam stack setelah penghapusan, yaitu:
 - Tugas 1: Grafika Komputer
 - Tugas 2: Algoritma
 - Tugas 3: Struktur Data

3. Kesimpulan

Dalam proyek Sistem Penjadwalan Tugas Menggunakan Stack, kami mengimplementasikan sebuah sistem yang sederhana namun efisien untuk mengelola tugas-tugas menggunakan struktur informasi stack. Stack bekerja dengan prinsip LIFO (Final In, To begin with Out), yang memungkinkan tugas terakhir yang ditambahkan menjadi prioritas untuk diselesaikan terlebih dahulu. Dengan fitur-fitur yang diimplementasikan seperti menambah tugas, menampilkan tugas terakhir, dan menghapus tugas terakhir, sistem ini menawarkan solusi praktis untuk mengatur tugas berdasarkan urutan prioritas.