IPB University

Oleh : P2-Kelompok 9

STRUKTUR DATA

PROJECT MANAGEMENT USING DATA STRUCTURES: UNORDERED MAP, QUEUE, VECTOR, AND HASHING

MULAI PRESENTASI

IPB University

ANGGOTA:

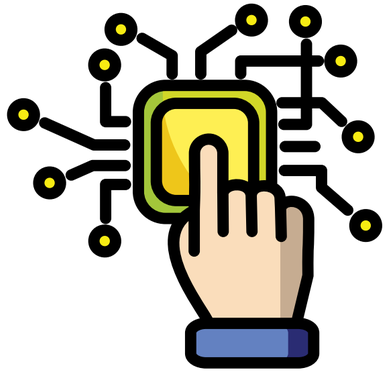
Abyan Fidriansyah
G6401221043

Muhammad Hafidz Rizki
G6401221046

Muh Farid FB
G6401221060

PENDAHULUAN

Kepentingan Pengelolaan Proyek dalam Era Digital



Peran Aplikasi Manajemen Proyek

FITUR APLIKASI MANAJEMEN PROYEK

- Pencatatan dan pemantauan progress proyek.
- Manajemen tugas dan penugasan tim.
- Komunikasi dan kolaborasi antar anggota tim.
- Analisis kinerja tim dan individu.

Oleh : P2-Kelompok 9

METODE

01.

Unordered_Map

Struktur data asosiatif yang menyimpan pasangan kunci dan nilai, di mana setiap kunci harus unik.

02.

Queue

Struktur data di mana data dimasukkan dari satu ujung dan dikeluarkan dari ujung lainnya.

03.

Vector

Array 1 dimensi/vektor, adalah struktur data yang menyimpan sekumpulan nilai dengan urutan tertentu.

04.

Hashing

Algoritma yang mengubah teks atau pesan menjadi rangkaian karakter acak yang memiliki karakter yang sama.

HASIL DAN PEMBAHASAN

01. **Pencatatan dan Pemantauan Progress Proyek**

```
void createProject() {
    Project project;
    cout << "Enter Project ID (max 3 alphanum char): ";
    cin >> project.id;
    cin.ignore();
    cout << "Enter Project Name: ";
    getline(cin, project.name);
    cout << "Enter Project Description: ";
    getline(cin, project.description);
    cout << "Enter Start Date (YYYY-MM-DD): ";
    cin >> project.startDate;
    cout << "Enter Deadline (YYYY-MM-DD): ";
    cin >> project.deadline;
    projects.push_back(project);
    saveProjects();
    cout << "Project created successfully.\n";
}

void deleteProject() {
    cout << "List of Projects:\n";
    for (size_t i = 0; i < projects.size(); ++i) {
        cout << i + 1 << ". [" << projects[i].id << "] " << projects[i].name << "\n";
    }
    cout << "Enter Project ID to delete: ";
    string projectId;
    cin >> projectId;
    auto it = remove_if(projects.begin(), projects.end(), [&projectId](const Project& project) {
        return project.id == projectId;
    });
    if (it != projects.end()) {
        projects.erase(it, projects.end());
        saveProjects();
        cout << "Project deleted successfully.\n";
    } else {
        cout << "Project not found.\n";
    }
}
```

IPB University

HASIL DAN PEMBAHASAN

02. **Manajemen Tugas dan Penugasan Tim**

```cpp
void addTask(Project& project) {
    Task task;
    cout << "Enter Task Description: ";
    cin.ignore();
    getline(cin, task.description);
    task.isComplete = false;
    task.userID = "";

    ofstream outfile("dataTask.txt", ios::app);
    if (outfile.is_open()) {
        outfile << project.id << " " << task.description << "|" << task.userID << "|" << task.isComplete << "\n";
        outfile.close();
    }

    cout << "Task added successfully.\n";
}
```

```cpp
void markTaskComplete(Project& project) {
    vector<Task> tasks;
    ifstream infileTasks("dataTask.txt");
    if (infileTasks.is_open()) {
        string projectID, description, userID, isCompleteStr;
        bool isComplete;
        while (getline(infileTasks, projectID, ' ') && getline(infileTasks, description, '|') &&
               getline(infileTasks, userID, '|') && getline(infileTasks, isCompleteStr)) {
            isComplete = (isCompleteStr == "1");
            if (projectID == project.id) {
                tasks.push_back({description, isComplete, userID});
            }
        }
        infileTasks.close();
    }

    cout << "List of Tasks:\n";
    for (size_t i = 0; i < tasks.size(); ++i) {
        cout << i + 1 << ". " << tasks[i].description << " [" << (tasks[i].isComplete ? "Complete" : "Incomplete") << "]\n";
    }

    cout << "Enter Task Number to Mark Complete: ";
    int taskNumber;
    cin >> taskNumber;
    if (taskNumber >= 1 && taskNumber <= tasks.size()) {
        cout << "Enter User ID who completed the task: ";
        string userID;
        cin >> userID;
        tasks[taskNumber - 1].isComplete = true;
        tasks[taskNumber - 1].userID = userID;
        ofstream outfile("dataTask.txt");
        if (outfile.is_open()) {
            for (const auto& task : tasks) {
                outfile << project.id << " " << task.description << "|" << task.userID << "|" << task.isComplete << "\n";
            }
            outfile.close();
        }
        cout << "Task marked as complete.\n";
    } else {
        cout << "Invalid task number.\n";
    }
}
```

HASIL DAN PEMBAHASAN

03. Komunikasi dan Kolaborasi Antar Anggota Tim

```
void sendMessage(Project& project, const string& sender) {
    Message message;
    message.sender = sender;
    cout << "Enter Message Content: ";
    cin.ignore();
    getline(cin, message.content);

    ofstream outfile("datamessage.txt", ios::app);
    if (outfile.is_open()) {
        outfile << project.id << "|" << message.sender << "|" << message.content << "\n";
        outfile.close();
    }

    cout << "Message sent successfully.\n";
}

void viewMessages(const Project& project) {
    cout << "Messages:\n";
    ifstream infile("datamessage.txt");
    if (infile.is_open()) {
        string projectID, sender, content;
        while (getline(infile, projectID, '|') && getline(infile, sender, '|') && getline(infile, content)) {
            if (projectID == project.id) {
                cout << sender << ": " << content << "\n";
            }
        }
        infile.close();
    }
}
```

IPB University

HASIL DAN PEMBAHASAN

04. Fungsi untuk analisis kinerja tim dan individu

```cpp
void analyzeOthers(const Project& project) {
    unordered_map<string, int> userTaskCount;
    unordered_map<string, string> userIdToUsername;
    int totalTasks = 0;

    ifstream infileUsers("datausers.txt");
    if (infileUsers.is_open()) {
        User user;
        while (infileUsers >> user.username >> user.passwordHash >> user.userID >> user.userProjectID >> user.isAdmin) {
            userIdToUsername[user.userID] = user.username;
        }
        infileUsers.close();
    }

    ifstream infile("dataTask.txt");
    if (infile.is_open()) {
        string projectID, description, userID, isCompleteStr;
        bool isComplete;
        while (getline(infile, projectID, ' ') && getline(infile, description, '|') &&
               getline(infile, userID, '|') && getline(infile, isCompleteStr)) {
            isComplete = (isCompleteStr == "1");
            if (projectID == project.id) {
                totalTasks++;
                if (isComplete) {
                    userTaskCount[userID]++;
                }
            }
        }
        infile.close();
    }

    cout << "Individual Performance Analysis:\n";
    cout << "Total team task = " << totalTasks << "\n";

    for (const auto& pair : userTaskCount) {
        const string& userID = pair.first;
        int completedTasks = pair.second;
        double completionPercentage = (static_cast<double>(completedTasks) / totalTasks) * 100.0;
        cout << "- [" << userID << "] " << userIdToUsername[userID] << " : " << completedTasks << " | " << completionPercentage << "%\n";
    }
}
```

```cpp
void analyzeTeam(const Project& project) {
    int completedTasks = 0;
    int totalTasks = 0;
    ifstream infileTasks("dataTask.txt");
    if (infileTasks.is_open()) {
        string projectID, description, userID, isCompleteStr;
        bool isComplete;
        while (getline(infileTasks, projectID, ' ') && getline(infileTasks, description, '|') &&
               getline(infileTasks, userID, '|') && getline(infileTasks, isCompleteStr)) {
            isComplete = (isCompleteStr == "1");
            if (projectID == project.id) {
                totalTasks++;
                if (isComplete) {
                    completedTasks++;
                }
            }
        }
        infileTasks.close();
    }

    double completionRate = (totalTasks == 0) ? 0.0 : static_cast<double>(completedTasks) / totalTasks * 100;
    cout << "Team Performance Analysis:\n";
    cout << "Total Tasks: " << totalTasks << "\n";
    cout << "Completed Tasks: " << completedTasks << "\n";
    cout << "Completion Rate: " << completionRate << "%\n";
}
```

SIMPULAN

Proyek manajemen proyek yang dikembangkan berhasil mengimplementasikan berbagai konsep struktur data seperti vector dan unordered_map untuk mengelola data proyek, pengguna, tugas, dan pesan secara efisien, dengan operasi dasar seperti insert, delete, dan save to file.

Sistem ini tidak hanya membantu dalam manajemen proyek dengan lebih terstruktur dan terorganisir, tetapi juga meningkatkan produktivitas dan kolaborasi tim. Dengan adanya alat manajemen yang efisien, tim dapat lebih fokus pada pencapaian tujuan proyek, mengurangi kesalahan manusia, dan meningkatkan transparansi dalam pelaksanaan proyek, sehingga proyek dapat diselesaikan tepat waktu.

IPB University

Oleh : P2-Kelompok 9

IT'S TIME FOR

DEMO

IPB University

Oleh : P2-Kelompok 9

SEKIAN

TERIMA KASIH

SELESAI