

LAPORAN PROYEK AKHIR
MATA KULIAH: ALGORITMA DAN STRUKTUR DATA



Oleh :

Shinta Putri Nirmala	1203230052
Felix Yohanes S.S	1203230068
Agnes Destiny Sava	1203230092

PROGRAM STUDI INFORMATIKA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM SURABAYA
TAHUN 2024

1. Nama Proyek

Implementasi Bahasa C dalam Database Film melalui Pohon Biner

2. Deskripsi Singkat

Proyek ini merupakan implementasi database sederhana dengan bahasa pemrograman C yang digunakan untuk mengelola daftar film menggunakan struktur data pohon biner, yang dimana pengguna dapat memasukkan detail film (ID, judul, tahun, dan rating) dan program akan menambahkannya ke dalam pohon biner. Pengguna juga dapat mencari film berdasarkan ID atau menampilkan semua film dalam urutan yang berbeda-beda.

3. Fitur Utama

a. Penambahan Film ke dalam Pohon Biner (Binary Search Tree)

Kode ini memungkinkan Anda untuk menambahkan film ke dalam pohon biner berdasarkan ID film. Pohon biner digunakan untuk menyimpan dan mengelola film dengan cara yang efisien, terutama untuk operasi pencarian. Pencarian Film: Mencari dan menampilkan detail film berdasarkan ID.

b. Pencarian Film berdasarkan ID

Anda dapat mencari film di dalam pohon biner berdasarkan ID. Ini memungkinkan Anda untuk dengan cepat menemukan film yang diinginkan.

c. Traversal dan Penampilan Film

Kode ini menyediakan tiga metode traversal untuk menampilkan film yang tersimpan di dalam pohon biner:

In-order Traversal: Menampilkan film dalam urutan ID yang meningkat.

Pre-order Traversal: Menampilkan film dalam urutan penyimpanan awal.

Post-order Traversal: Menampilkan film dalam urutan penyimpanan akhir.

d. Validasi Input ID

Fungsi ini memastikan bahwa ID yang dimasukkan adalah angka yang valid, sehingga mencegah kesalahan input.

e. Antarmuka Pengguna (User Interface) melalui Menu

Meskipun dalam versi yang dimodifikasi, menu ini secara otomatis menambahkan, mencari, dan menampilkan film, antarmuka ini biasanya menyediakan opsi bagi pengguna untuk: Menambahkan film, mencari film berdasarkan ID, menampilkan semua film dalam tiga urutan traversal yang berbeda, keluar dari program.

4. Struktur Data

Berikut adalah struktur data yang terdapat dalam kode tersebut:

a. Struck Movie:

Digunakan untuk menyimpan informasi tentang sebuah film.

Meliputi:

int id: ID film.

char title[100]: Judul film.

int year: Tahun rilis film.

float rating: Rating film.

b. Struck TreeNode:

Digunakan untuk membentuk node dari pohon biner.

Meliputi:

Movie data: Menyimpan data film dalam node.

TreeNode* left: Pointer ke anak kiri.

TreeNode* right: Pointer ke anak kanan.

c. Pohon Biner (Binary Tree) :

Struktur data utama yang digunakan untuk menyimpan dan mengelola node-node TreeNode.

Operasi yang dilakukan pada pohon biner meliputi:

insert: Menambahkan node baru ke dalam pohon biner.

search: Mencari node berdasarkan ID film.

inorder, preorder, postorder: Traversal pohon untuk menampilkan informasi film.

d. Struktur data lain yang juga digunakan dalam kode ini adalah:

String (char[]): Digunakan untuk menyimpan input dari pengguna seperti ID, judul, dan rating film.

Array (char[]): Digunakan untuk menyimpan string sementara seperti ID, judul, dan rating sebelum dikonversi menjadi tipe data lain.

Integer dan Float (int dan float): Digunakan untuk menyimpan data numerik seperti ID, tahun, dan rating film.

5. Fungsi Utama

a. createNode(Movie data): Membuat node baru dengan data film yang diberikan.

b. insert(TreeNode* root, Movie data): Menambahkan film baru ke dalam pohon biner.

- c. `search(TreeNode* root, int id)`: Mencari film berdasarkan ID dalam pohon biner.
- d. `inorder(TreeNode* root)`: Menampilkan semua film dalam urutan ID yang terurut menggunakan inorder traversal.
- e. `printMovie(Movie movie)`: Mencetak detail film.
- f. `preorder(TreeNode* root)`: mencetak detail film dari node akar terlebih dahulu, diikuti oleh subtree kirinya, dan kemudian subtree kanannya.
- g. `postorder(TreeNode* root)`: Menampilkan film dari subtree kiri, kemudian subtree kanan, dan terakhir node akar.
- h. `isValidNumber(char* str)`: untuk memeriksa apakah string yang diberikan (str) hanya berisi angka, yang digunakan untuk memvalidasi input pengguna untuk ID film.

6. Menu Program

Program menyediakan menu untuk pengguna berinteraksi dengan database, dengan pilihan sebagai berikut:

- a. Tambah Film
- b. Cari Film berdasarkan ID
- c. Tampilkan Semua Film (In-order): Pilihan ini menampilkan semua film dalam pohon biner menggunakan traversal in-order, yang menampilkan film dalam urutan ID yang meningkat.
- d. Tampilkan Semua Film (Pre-order): Pilihan ini menampilkan semua film dalam pohon biner menggunakan traversal pre-order, yang menampilkan film dalam urutan penyimpanan awal.
- e. Tampilkan Semua Film (Post-order): Pilihan ini menampilkan semua film dalam pohon biner menggunakan traversal post-order, yang menampilkan film dalam urutan penyimpanan akhir.
- f. Keluar

7. Struktur Program

- a. Header File dan Library: `#include <stdio.h>`, `#include <stdlib.h>`, dan `#include <string.h>`. digunakan untuk fungsi input/output, alokasi memori, dan manipulasi string.
- b. Definisi struct: `struct Movie` dan `struct TreeNode`. struktur-struktur ini menyediakan cara untuk mengatur data film dan membuat representasi menggunakan BST.

- c. Fungsi Utama untuk Pohon Biner: Implementasi fungsi untuk menambah, mencari, dan menampilkan data film.
- d. Main Program: Menyediakan antarmuka menu untuk pengguna dan memanggil fungsi-fungsi yang diperlukan berdasarkan pilihan pengguna.

8. Kode Program

- a. Mendefinisikan struct untuk Film

struct Movie digunakan untuk menyimpan informasi tentang setiap film. struct ini berisi empat kolom data: ID film, judul, tahun, dan rating.

```
5  typedef struct {
6      int id;          // ID Film
7      char title[100]; // Judul Film
8      int year;        // Tahun Film
9      float rating;    // Rating Film
10 } Movie;
```

- b. Mendefinisikan struct untuk Node Pohon

struct TreeNode digunakan untuk merepresentasikan node dalam pohon biner. struct ini berisi data film dan pointer ke anak kiri dan kanan dalam pohon.

```
12 typedef struct TreeNode {
13     Movie data;          // Data Film
14     struct TreeNode* left; // Pointer ke anak kiri
15     struct TreeNode* right; // Pointer ke anak kanan
16 } TreeNode;
```

- c. Prototipe Fungsi

Program ini mendeklarasikan beberapa fungsi yang digunakan untuk operasi pada pohon biner dan manajemen data film.

```
22 TreeNode* createNode(Movie data);
23 TreeNode* insert(TreeNode* root, Movie data);
24 TreeNode* search(TreeNode* root, int id);
25 void printMovie(Movie movie);
26 void inorder(TreeNode* root);
27 void preorder(TreeNode* root);
28 void postorder(TreeNode* root);
```

- d. Membuat Node Baru

Fungsi createNode digunakan untuk membuat node baru dengan data film yang diberikan dan mengembalikan pointer ke node tersebut.

```
31 TreeNode* createNode(Movie data) {
32     TreeNode* newNode = (TreeNode*)malloc(sizeof(TreeNode));
33     newNode->data = data;
34     newNode->left = newNode->right = NULL;
35     return newNode;
36 }
```

e. Menambahkan Film ke Pohon Biner

Fungsi insert menambahkan film ke pohon biner berdasarkan ID. Jika ID lebih kecil, data akan dimasukkan ke anak kiri; jika lebih besar, ke anak kanan.

```
39  TreeNode* insert(TreeNode* root, Movie data) {
40      if (root == NULL) {
41          return createNode(data);
42      }
43      if (data.id < root->data.id) {
44          root->left = insert(root->left, data);
45      } else {
46          root->right = insert(root->right, data);
47      }
48      return root;
49  }
```

f. Mencari Film Berdasarkan ID

Fungsi search mencari node dengan ID tertentu dalam pohon biner dan mengembalikan pointer ke node tersebut jika ditemukan.

```
52  TreeNode* search(TreeNode* root, int id) {
53      if (root == NULL || root->data.id == id) {
54          return root;
55      }
56      if (id < root->data.id) {
57          return search(root->left, id);
58      } else {
59          return search(root->right, id);
60      }
61  }
```

g. Mencetak Detail Film

Fungsi printMovie mencetak detail dari sebuah film.

```
64  void printMovie(Movie movie) {
65      printf("ID: %d\n", movie.id);
66      printf("Judul: %s\n", movie.title);
67      printf("Tahun: %d\n", movie.year);
68      printf("Rating: %.2f\n", movie.rating);
69  }
```

h. Traversal In-order

Fungsi inorder melakukan traversal in-order pada pohon biner, mencetak film dalam urutan berdasarkan ID dari terkecil ke terbesar.

```
72  void inorder(TreeNode* root) {
73      if (root != NULL) {
74          inorder(root->left);
75          printMovie(root->data);
76          inorder(root->right);
77      }
78  }
```

i. Traversal Pre-order

Fungsi preorder melakukan traversal pre-order pada pohon biner, mencetak film mulai dari node akar, kemudian anak kiri dan anak kanan.

```
81 void preorder(TreeNode* root) {
82     if (root != NULL) {
83         printMovie(root->data);
84         preorder(root->left);
85         preorder(root->right);
86     }
87 }
```

j. Traversal Post-order

Fungsi postorder melakukan traversal post-order pada pohon biner, mencetak film mulai dari anak kiri, anak kanan, dan terakhir node akar.

```
90 void postorder(TreeNode* root) {
91     if (root != NULL) {
92         postorder(root->left);
93         postorder(root->right);
94         printMovie(root->data);
95     }
96 }
```

k. Fungsi Utama dengan Menu

Program ini menyediakan menu yang memungkinkan pengguna untuk:

menambah film ke pohon biner, mencari film berdasarkan ID, menampilkan semua film dengan traversal in-order, menampilkan semua film dengan traversal pre-order, menampilkan semua film dengan traversal post-order, keluar dari program.

```

98 // Fungsi utama dengan menu untuk pengguna
99 int main() {
100     TreeNode* root = NULL;
101     Movie movies[] = {
102         {1, "The Shawshank Redemption", 1994, 9.3},
103         {2, "The Godfather", 1972, 9.2},
104         {3, "The Dark Knight", 2008, 9.0},
105         {4, "12 Angry Men", 1957, 9.0},
106         {5, "Schindler's List", 1993, 8.9}
107     };
108     int numMovies = sizeof(movies) / sizeof(movies[0]);
109
110     // Menambahkan film secara otomatis
111     for (int i = 0; i < numMovies; i++) {
112         root = insert(root, movies[i]);
113     }
114
115     // Mencari film berdasarkan ID secara otomatis
116     int searchIds[] = {1, 2, 6}; // ID film yang ingin dicari (termasuk ID yang t
117     for (int i = 0; i < sizeof(searchIds) / sizeof(searchIds[0]); i++) {
118         TreeNode* searchResult = search(root, searchIds[i]);
119         if (searchResult != NULL) {
120             printf("\nDetail Film dengan ID %d:\n", searchIds[i]);
121             printMovie(searchResult->data);
122         } else {
123             printf("\nFilm dengan ID %d tidak ditemukan!\n", searchIds[i]);
124         }
125     }
126
127     // Menampilkan semua film dalam urutan yang berbeda
128     printf("\nMenampilkan semua film (In-order):\n");
129     inorder(root);
130
131     printf("\nMenampilkan semua film (Pre-order):\n");
132     preorder(root);
133
134     printf("\nMenampilkan semua film (Post-order):\n");
135     postorder(root);
136
137     printf("Keluar dari program.\n");
138     return 0;
139 }
140

```

9. Run Program


```
PS C:\Users\User\Downloads\PUPUT\MATERI KULIAH\SEMESTER II\ALGORITMA DAN STRUKTUR DATA\VSCode
SCode\TUKIR ASD\" ; if ($?) { gcc code.c -o code } ; if ($?) { .\code }
```

Detail Film dengan ID 1:

ID: 1
Judul: The Shawshank Redemption
Tahun: 1994
Rating: 9.30

Detail Film dengan ID 2:

ID: 2
Judul: The Godfather
Tahun: 1972
Rating: 9.20

Film dengan ID 6 tidak ditemukan!

Menampilkan semua film (In-order):

ID: 1
Judul: The Shawshank Redemption
Tahun: 1994
Rating: 9.30
ID: 2
Judul: The Godfather
Tahun: 1972
Rating: 9.20
ID: 3
Judul: The Dark Knight
Tahun: 2008
Rating: 9.00
ID: 4
Judul: 12 Angry Men
Tahun: 1957
Rating: 9.00

```
ID: 5
Judul: Schindler's List
Tahun: 1993
Rating: 8.90

Menampilkan semua film (Pre-order):
ID: 1
Judul: The Shawshank Redemption
Tahun: 1994
Rating: 9.30
ID: 2
Judul: The Godfather
Tahun: 1972
Rating: 9.20
ID: 3
Judul: The Dark Knight
Tahun: 2008
Rating: 9.00
ID: 4
Judul: 12 Angry Men
Tahun: 1957
Rating: 9.00
ID: 5
Judul: Schindler's List
Tahun: 1993
Rating: 8.90

Menampilkan semua film (Post-order):
ID: 5
Judul: Schindler's List
Tahun: 1993
Rating: 8.90
ID: 4
Judul: 12 Angry Men
Tahun: 1957
Rating: 9.00
ID: 3
```

```
Judul: The Dark Knight
Tahun: 2008
Rating: 9.00
ID: 2
Judul: The Godfather
Tahun: 1972
Rating: 9.20
ID: 1
Judul: The Shawshank Redemption
Tahun: 1994
Rating: 9.30
Keluar dari program.
```

10. Kesimpulan

Dalam database film ini, struct digunakan untuk:

- Mengelompokkan Data Film: struct Movie menyimpan informasi yang terkait dengan setiap film, seperti ID, judul, tahun, dan rating.

- b. Merepresentasikan Node Pohon: struct TreeNode menyimpan data film dan pointer ke anak kiri dan kanan, memungkinkan pembentukan struktur pohon biner.
- c. Memudahkan Manipulasi Data: Fungsi-fungsi yang beroperasi pada struct memungkinkan penambahan, pencarian, dan penelusuran data film dengan cara yang efisien dan terorganisir.
- d. Penanganan kesalahan: program menyertakan validasi untuk input pengguna saat memasukkan id film menggunakan isValidNumber, jika id yang dimasukkan tidak valid maka pesan yang ditampilkan eror dan pengguna diminta untuk memasukkan id kembali.