

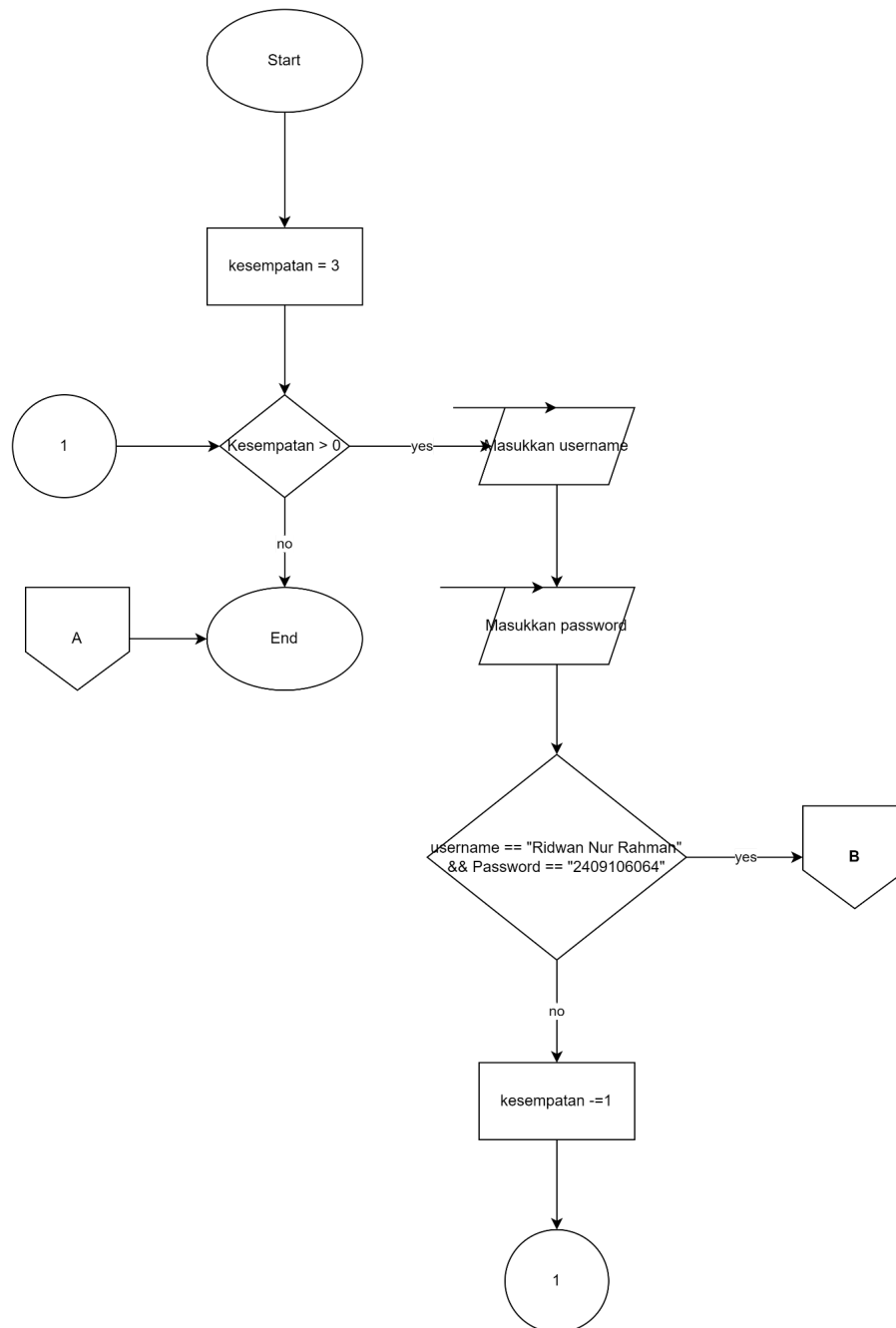
LAPORAN PRAKTIKUM
POSTTEST 6
ALGORITMA PEMROGRAMAN LANJUT



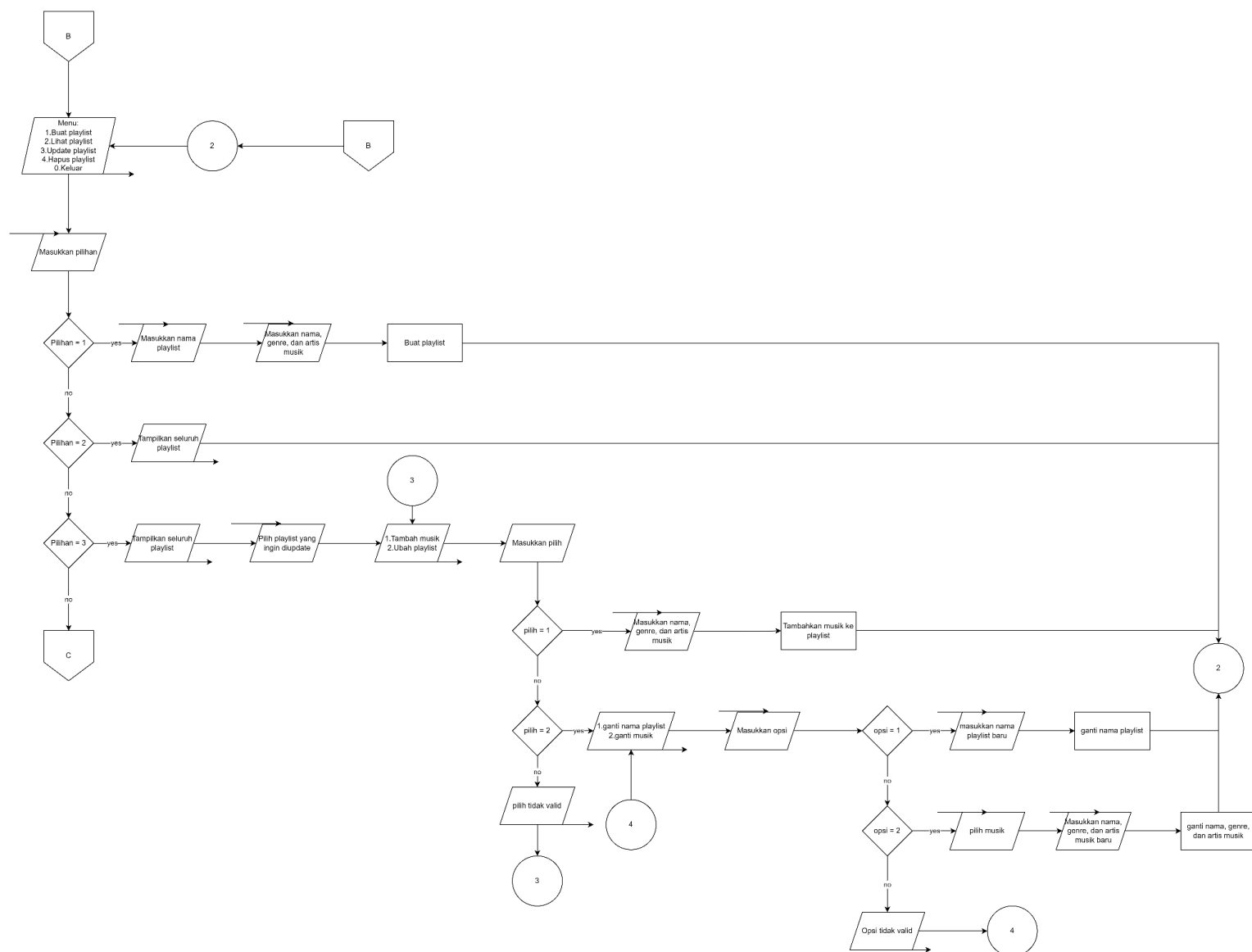
Disusun oleh:
Ridwan Nur Rahman (2409106064)
Kelas (B1 '2024)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

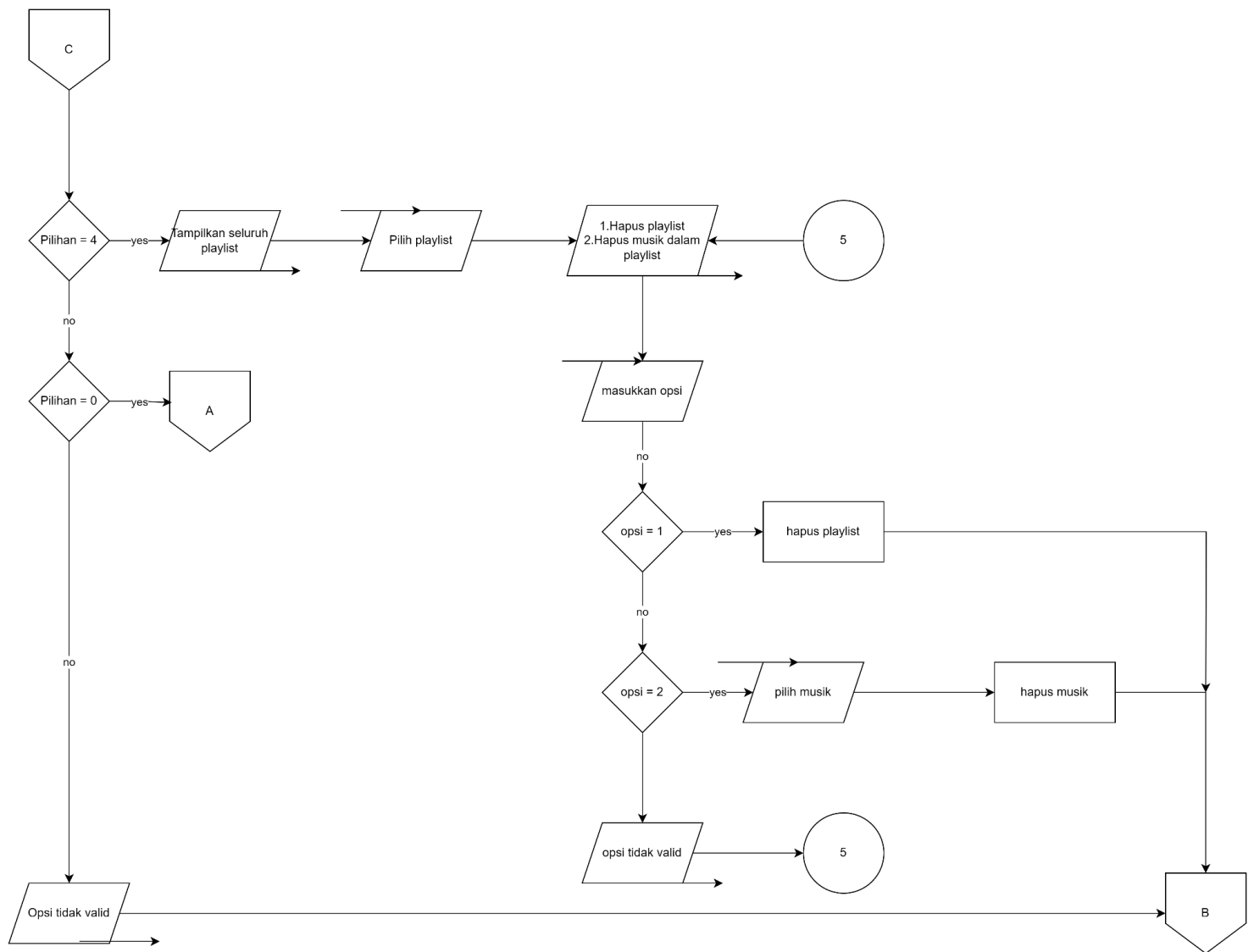
1. Flowchart



Gambar 1.1 Flowchart login



Gambar 1.2 Flowchart menu program



Gambar 1.3 Flowchart menu program

2. Analisis Program

2.1 Deskripsi Singkat Program

Program Manajemen Playlist Musik adalah program berbasis C++ yang memungkinkan pengguna untuk mengelola playlist musik dengan fitur CRUD (Create, Read, Update, Delete).

Program ini memiliki beberapa fitur utama, yaitu:

1. Login → Pengguna harus memasukkan username dan password sebelum mengakses sistem.
2. Buat Playlist → Membuat playlist baru dengan lagu pertama.
3. Lihat Playlist → Menampilkan daftar playlist dan lagu di dalamnya.
4. Update Playlist → Pengguna bisa menambah lagu atau mengubah playlist (ganti nama atau ubah lagu).
5. Hapus Playlist atau Lagu → Pengguna bisa menghapus seluruh playlist atau lagu tertentu dalam playlist.
6. Keluar dari Program → Program akan terus berjalan hingga pengguna memilih keluar.

Setiap fitur dalam program dipecah menjadi subprogram, yang terdiri dari prosedur dan fungsi. Fungsi menggunakan parameter, sedangkan prosedur digunakan untuk melakukan aksi tanpa mengembalikan nilai.

Program juga menerapkan beberapa metode sorting, yakni selection sort untuk nama playlist, merge sort untuk nama musik, dan quick sort untuk artis.

3. Source Code

A. Struct Musik

Struct yang berisi data apa saja yang dimiliki musik. Dalam hal ini yakni judul musik, genre musik, dan artis penyanyi.

Source code:

```
struct DataMusik {  
    string judul;  
    string genre;  
    string artis;  
};
```

B. Struct Playlist

Struct yang berisi nama playlist, banyaknya musik yang ada dalam playlist, termasuk tipe data dari “struct musik”.

Source code:

```
struct Playlist {  
    string nama;  
    DataMusik musik[10];  
    int jumlahMusik = 0;  
};
```

C. Fungsi Login

Berisi username dan password untuk nanti divalidasi dalam program. Fungsi berikut terdapat parameter dereference (string &user, string &pass)

Source code:

```
bool login(string &user, string &pass) {  
    return (user == "Ridwan Nur Rahman" && pass == "2409106064");  
}
```

D. Menu

Terdapat beberapa menu yang ditampilkan untuk memilih fitur pada program.

Source code:

```
while (true) {
    cout << "\n===== Menu Manajemen Playlist Musik =====" << endl;
    cout << "1. Buat playlist" << endl;
    cout << "2. Lihat playlist" << endl;
    cout << "3. Update playlist" << endl;
    cout << "4. Hapus playlist" << endl;
    cout << "0. Keluar program" << endl;
    cout << "Masukkan pilihan: ";
    int pilihan;
    cin >> pilihan;
    cin.ignore();
}
```

E. Fungsi Buat Playlist

Fitur untuk membuat playlist baru sekaligus menambahkan musik untuk yang pertama. Fungsi berikut terdapat parameter dereference (int &totalPlaylist)

```
void buatPlaylist(Playlist playlist[], int &totalPlaylist, int max_playlist)
{
    if (totalPlaylist < max_playlist) {
        cout << "Nama Playlist: ";
        getline(cin, playlist[totalPlaylist].nama);
        cout << "Judul Musik: ";
        getline(cin, playlist[totalPlaylist].musik[0].judul);
        cout << "Genre: ";
        getline(cin, playlist[totalPlaylist].musik[0].genre);
        cout << "Artis: ";
        getline(cin, playlist[totalPlaylist].musik[0].artis);

        playlist[totalPlaylist].jumlahMusik = 1;
        totalPlaylist++;

        sortPlaylistByNamaDesc(playlist, totalPlaylist);

        cout << "Playlist berhasil dibuat!\n" << endl;
    } else {
        cout << "Playlist sudah penuh!\n" << endl;
    }
}
```

F. Fungsi Lihat Playlist

Fitur untuk menampilkan seluruh playlist yang telah dibuat.

Source code:

```
void lihatPlaylist(Playlist playlist[], int totalPlaylist, bool sortByJumlah = false) {
    if (totalPlaylist == 0) {
        cout << "Belum ada playlist.\n" << endl;
    } else {
        if (sortByJumlah) {
            sortPlaylistByJumlahMusik(playlist, totalPlaylist);
            cout << "=== Daftar Playlist (Sort by Jumlah Lagu) ===\n";
        } else {
            sortPlaylistByNamaDesc(playlist, totalPlaylist);
            cout << "=== Daftar Playlist (Sort by Nama) ===\n";
        }

        for (int i = 0; i < totalPlaylist; i++) {
            mergeSort(playlist[i].musik, 0, playlist[i].jumlahMusik - 1);

            cout << i + 1 << ". " << playlist[i].nama << " (" <<
playlist[i].jumlahMusik << " lagu)\n";
            for (int j = 0; j < playlist[i].jumlahMusik; j++) {
                cout << "    - " << playlist[i].musik[j].judul << " | "
                    << playlist[i].musik[j].genre << " | "
                    << playlist[i].musik[j].artis << "\n";
            }
        }
    }
}
```

G. Fungsi Tambah Musik

Fitur untuk menambahkan musik ke dalam playlist. Fungsi berikut terdapat parameter dereference (Playlist &pl)

Source code:

```
void tambahMusik(Playlist &pl) {
    if (pl.jumlahMusik < 10) {
        int idx = pl.jumlahMusik;
        DataMusik *lagu = &pl.musik[idx];
        cout << "Judul Musik: ";
        getline(cin, lagu->judul);
        cout << "Genre: ";
        getline(cin, lagu->genre);
        cout << "Artis: ";
        getline(cin, lagu->artis);
        pl.jumlahMusik++;

        mergeSort(pl.musik, 0, pl.jumlahMusik - 1);
    }
}
```



```

        cout << "Musik berhasil ditambahkan!\n" << endl;
    } else {
        cout << "Playlist ini sudah penuh!\n" << endl;
    }
}

```

H. Fungsi Ubah Nama Playlist

Fitur untuk mengubah nama playlist. Fungsi berikut terdapat parameter dereference (Playlist &pl)

Source code:

```

void ubahNamaPlaylist(Playlist &pl, Playlist playlist[], int totalPlaylist)
{
    cout << "Masukkan nama baru: ";
    getline(cin, pl.nama);

    sortPlaylistByNamaDesc(playlist, totalPlaylist);

    cout << "Nama berhasil diubah!\n" << endl;
}

```

I. Fungsi Ubah Musik

Fitur untuk mengganti atau mengubah musik dalam playlist. Fungsi berikut terdapat parameter dereference (Playlist &pl)

Source code:

```

void ubahMusik(Playlist &pl) {
    for (int i = 0; i < pl.jumlahMusik; i++) {
        cout << i + 1 << ". " << pl.musik[i].judul << "\n";
    }

    int pilih;
    cout << "Pilih musik yang ingin diubah: ";
    cin >> pilih;
    cin.ignore();

    if (pilih > 0 && pilih <= pl.jumlahMusik) {
        cout << "Judul Baru: ";
        getline(cin, pl.musik[pilih - 1].judul);
        cout << "Genre Baru: ";
        getline(cin, pl.musik[pilih - 1].genre);
        cout << "Artis Baru: ";
        getline(cin, pl.musik[pilih - 1].artis);

        // Auto-sort musik setelah perubahan
        mergeSort(pl.musik, 0, pl.jumlahMusik - 1);
    }
}

```

```

        cout << "Musik berhasil diperbarui!\n" << endl;
    } else {
        cout << "Pilihan tidak valid!\n" << endl;
    }
}

```

J. Fungsi Hapus Playlist

Fitur untuk menghapus playlist. Fungsi berikut terdapat parameter address-of (int *totalPlaylist)

Source code:

```

void hapusPlaylist(Playlist playlist[], int &totalPlaylist, int index) {
    for (int i = index; i < totalPlaylist - 1; i++) {
        playlist[i] = playlist[i + 1];
    }
    totalPlaylist--;

    cout << "Playlist berhasil dihapus!\n" << endl;
}

```

K. Fungsi Hapus Musik di dalam Playlist

Fitur untuk menghapus musik yang ada di dalam playlist. Fungsi berikut terdapat parameter dereference (Playlist &pl)

Source code:

```

void hapusMusikDalamPlaylist(Playlist &pl) {
    for (int i = 0; i < pl.jumlahMusik; i++) {
        cout << i + 1 << ". " << pl.musik[i].judul << "\n";
    }

    int pilih;
    cout << "Pilih musik yang ingin dihapus: ";
    cin >> pilih;
    cin.ignore();

    if (pilih > 0 && pilih <= pl.jumlahMusik) {
        for (int i = pilih - 1; i < pl.jumlahMusik - 1; i++) {
            pl.musik[i] = pl.musik[i + 1];
        }
        pl.jumlahMusik--;

        mergeSort(pl.musik, 0, pl.jumlahMusik - 1);
    }
}

```

```

        cout << "Musik berhasil dihapus!\n" << endl;
    } else {
        cout << "Pilihan tidak valid!\n" << endl;
    }
}

```

L. Keluar Program

Fitur untuk keluar dari program.

Source code:

```

    } else if (pilihan == 0) {
        cout << "Terima kasih telah menggunakan program ini!\n";
        break;
    }
}

```

M. Fungsi Selection Sort

Fungsi untuk menyortir data nama playlist secara Descending.

Source code:

```

void sortPlaylistByNamaDesc(Playlist playlist[], int total) {
    for (int i = 0; i < total - 1; i++) {
        int maxIdx = i;
        for (int j = i + 1; j < total; j++) {
            if (playlist[j].nama > playlist[maxIdx].nama) {
                maxIdx = j;
            }
        }
        if (maxIdx != i) {
            swap(playlist[i], playlist[maxIdx]);
        }
    }
}

```

N. Fungsi Merge Sort

Fungsi untuk menyortir data nama musik secara Descending.

Source code:

```

void merge(DataMusik arr[], int left, int mid, int right) {
    int n1 = mid - left + 1;
    int n2 = right - mid;
    DataMusik L[10], R[10];

    for (int i = 0; i < n1; i++) L[i] = arr[left + i];
}

```

```

    for (int j = 0; j < n2; j++) R[j] = arr[mid + 1 + j];

    int i = 0, j = 0, k = left;
    while (i < n1 && j < n2) {
        if (L[i].judul > R[j].judul) {
            arr[k++] = L[i++];
        } else {
            arr[k++] = R[j++];
        }
    }

    while (i < n1) arr[k++] = L[i++];
    while (j < n2) arr[k++] = R[j++];
}

void mergeSort(DataMusik arr[], int left, int right) {
    if (left < right) {
        int mid = left + (right - left) / 2;
        mergeSort(arr, left, mid);
        mergeSort(arr, mid + 1, right);
        merge(arr, left, mid, right);
    }
}

```

O. Fungsi Insertion Sort

Fungsi untuk menyortir playlist berdasarkan jumlah musik di dalamnya.

Source code:

```

void sortPlaylistByJumlahMusik(Playlist playlist[], int total) {
    for (int i = 1; i < total; i++) {
        Playlist key = playlist[i];
        int j = i - 1;
        while (j >= 0 && playlist[j].jumlahMusik > key.jumlahMusik) {
            playlist[j + 1] = playlist[j];
            j--;
        }
        playlist[j + 1] = key;
    }
}

```

4. Screenshot Hasil Program

```
Masukkan Username: Ridwan Nur Rahman
Masukkan Password: 2409106064

===== Menu Manajemen Playlist Musik =====
1. Buat playlist
2. Lihat playlist (Sort by Nama)
3. Lihat playlist (Sort by Jumlah Musik)
4. Update playlist
5. Hapus playlist
0. Keluar
Pilihan: 
```

Gambar 4.1 berhasil login

```
Masukkan Username: Ucup
Masukkan Password: 25626811
Username atau password salah! Kesempatan tersisa: 2
Masukkan Username: 
```

Gambar 4.2 gagal login

```
Masukkan Username: Ucup
Masukkan Password: 25626811
Username atau password salah! Kesempatan tersisa: 2
Masukkan Username: ridwan
Masukkan Password: 2409208761
Username atau password salah! Kesempatan tersisa: 1
Masukkan Username: daffa
Masukkan Password: 575624
Username atau password salah! Kesempatan tersisa: 0
PS C:\Users\ADMIN\OneDrive\Desktop\praktikum-apl\post-test\post-test-a
```

Gambar 4.3 gagal login 3 kali

```
===== Menu Manajemen Playlist Musik =====
1. Buat playlist
2. Lihat playlist (Sort by Nama)
3. Lihat playlist (Sort by Jumlah Musik)
4. Update playlist
5. Hapus playlist
0. Keluar
Pilihan: 1
Nama Playlist: Mood
Judul Musik: sunflower
Genre: pop
Artis: post malone
Playlist berhasil dibuat!
```

Gambar 4.4 buat playlist

```
===== Menu Manajemen Playlist Musik =====
1. Buat playlist
2. Lihat playlist (Sort by Nama)
3. Lihat playlist (Sort by Jumlah Musik)
4. Update playlist
5. Hapus playlist
0. Keluar
Pilihan: 4
=== Daftar Playlist (Sort by Nama) ===
1. Mood (1 lagu)
   - sunflower | pop | post malone
Pilih Playlist: 1
1. Tambah Musik
2. Ubah Playlist
Pilihan: 1
Judul Musik: dynamite
Genre: dance pop
Artis: BTS
Musik berhasil ditambahkan!
```

Gambar 4.5 tambah musik

```

===== Menu Manajemen Playlist Musik =====
1. Buat playlist
2. Lihat playlist (Sort by Nama)
3. Lihat playlist (Sort by Jumlah Musik)
4. Update playlist
5. Hapus playlist
0. Keluar
Pilihan: 4
=== Daftar Playlist (Sort by Nama) ===
1. Mood (2 lagu)
   - sunflower | pop | post malone
   - dynamite | dance pop | BTS
Pilih Playlist: 1
1. Tambah Musik
2. Ubah Playlist
Pilihan: 2
1. Ubah Nama Playlist
2. Ubah Musik
Pilihan: 1
Masukkan nama baru: mood booster
Nama berhasil diubah!

```

Gambar 4.6 ganti nama playlist

```

===== Menu Manajemen Playlist Musik =====
1. Buat playlist
2. Lihat playlist (Sort by Nama)
3. Lihat playlist (Sort by Jumlah Musik)
4. Update playlist
5. Hapus playlist
0. Keluar
Pilihan: 4
=== Daftar Playlist (Sort by Nama) ===
1. mood booster (2 lagu)
   - sunflower | pop | post malone
   - dynamite | dance pop | BTS
Pilih Playlist: 1
1. Tambah Musik
2. Ubah Playlist
Pilihan: 2
1. Ubah Nama Playlist
2. Ubah Musik
Pilihan: 2
1. sunflower
2. dynamite
Pilih musik yang ingin diubah: 2
Judul Baru: uptown funk
Genre Baru: funk
Artis Baru: mark ronson
Musik berhasil diperbarui!

```

Gambar 4.7 ganti musik

```

===== Menu Manajemen Playlist Musik =====
1. Buat playlist
2. Lihat playlist (Sort by Nama)
3. Lihat playlist (Sort by Jumlah Musik)
4. Update playlist
5. Hapus playlist
0. Keluar
Pilihan: 5
=== Daftar Playlist (Sort by Nama) ===
1. mood booster (2 lagu)
   - uptown funk | funk | mark ronson
   - sunflower | pop | post malone
Pilih Playlist: 1
1. Hapus Playlist
2. Hapus Musik
Pilihan: 2
1. uptown funk
2. sunflower
Pilih musik yang ingin dihapus: 2
Musik berhasil dihapus!

```

Gambar 4.8 hapus musik

```

===== Menu Manajemen Playlist Musik =====
1. Buat playlist
2. Lihat playlist (Sort by Nama)
3. Lihat playlist (Sort by Jumlah Musik)
4. Update playlist
5. Hapus playlist
0. Keluar
Pilihan: 5
=== Daftar Playlist (Sort by Nama) ===
1. mood booster (1 lagu)
   - uptown funk | funk | mark ronson
Pilih Playlist: 1
1. Hapus Playlist
2. Hapus Musik
Pilihan: 1
Playlist berhasil dihapus!

```

Gambar 4.9 hapus playlist


```
===== Menu Manajemen Playlist Musik =====
1. Buat playlist
2. Lihat playlist (Sort by Nama)
3. Lihat playlist (Sort by Jumlah Musik)
4. Update playlist
5. Hapus playlist
0. Keluar
Pilihan: 0
Terima kasih telah menggunakan program ini!
```

Gambar 4.10 keluar program

```
===== Menu Manajemen Playlist Musik =====
1. Buat playlist
2. Lihat playlist (Sort by Nama)
3. Lihat playlist (Sort by Jumlah Musik)
4. Update playlist
5. Hapus playlist
0. Keluar
Pilihan: 2
=== Daftar Playlist (Sort by Nama) ===
1. rock legends (2 lagu)
   - sweet childe o'mine | hard rock | guns n rose
   - bohemian rhapsody | rock | queen
2. mood booster (3 lagu)
   - uptown funk | funk | mark robinson
   - sunflower | pop | post malone
   - dynamite | dance pop | BTS
3. late night drive (4 lagu)
   - nightcall | sythpop | kavinsky
   - midnight city | electronic | m83
   - digital love | disco | daft punk
   - binding lights | sythwave | the weeken
```

Gambar 4.11 sort playlist (nama)

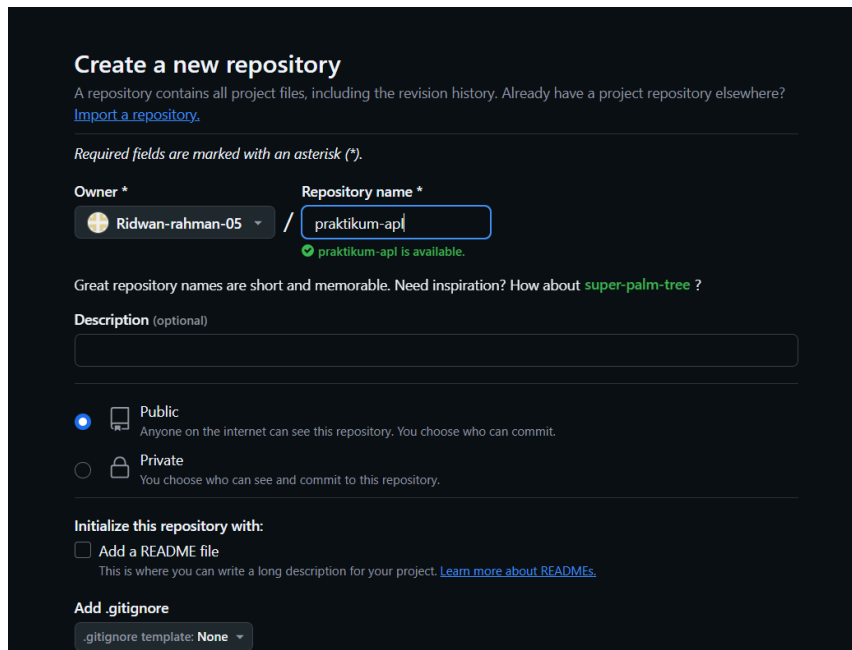
```
===== Menu Manajemen Playlist Musik =====
1. Buat playlist
2. Lihat playlist (Sort by Nama)
3. Lihat playlist (Sort by Jumlah Musik)
4. Update playlist
5. Hapus playlist
0. Keluar
Pilihan: 3
=== Daftar Playlist (Sort by Jumlah Lagu) ===
1. rock legends (2 lagu)
   - sweet childe o'mine | hard rock | guns n rose
   - bohemian rhapsody | rock | queen
2. mood booster (3 lagu)
   - uptown funk | funk | mark robinson
   - sunflower | pop | post malone
   - dynamite | dance pop | BTS
3. late night drive (4 lagu)
   - nightcall | sythpop | kavinsky
   - midnight city | electronic | m83
   - digital love | disco | daft punk
   - binding lights | sythwave | the weeken
```

Gambar 4.12 sort playlist (jumlah musik)

5. Langkah-langkah Git

1. Buat Repository di GitHub

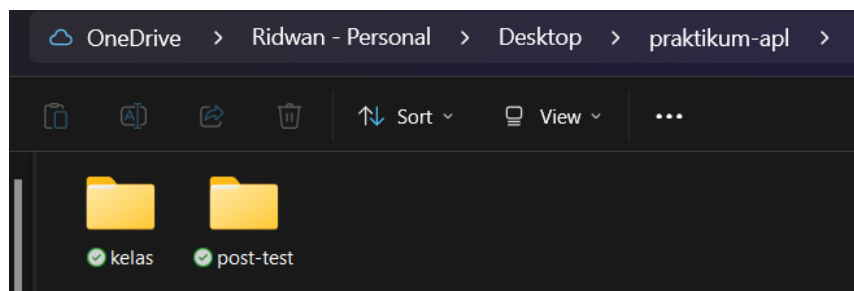
- Masuk ke akun GitHub.
- Klik tombol New Repository.
- Beri nama repository “praktikum-apl”.
- Pilih opsi Public agar bisa diakses oleh semua orang.
- Klik Create Repository.



Gambar 5.1 buat new repository

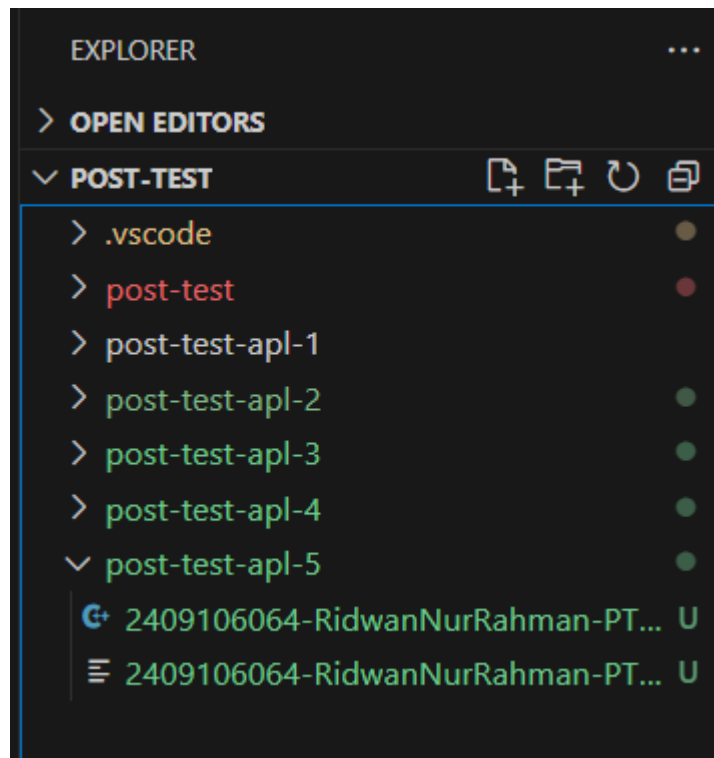
2. Buat Folder di Explorer

- Buka File Explorer dan buat folder baru dengan nama praktikum-apl.
- Di dalam folder praktikum-apl, buat dua subfolder: kelas dan post-test.



Gambar 5.2 buat folde kelas dan post-test dalam folder praktikum-apl

3. Buka Folder di VS Code
 - Buka Visual Studio Code.
 - Pilih File > Open Folder, lalu arahkan ke folder praktikum-apl.
4. Buat Folder Post-Test
 - Masuk ke folder post-test.
 - Buat folder baru post-test-6 untuk menyimpan hasil pengerjaan posttest.
 - Tambahkan file yang diperlukan di dalam folder ini.



Gambar 5.3 folder post-test di vs code

5. Buka Terminal
 - Tekan **Ctrl** + **~** untuk membuka terminal di VS Code.
 - Jika direktori tidak sesuai, gunakan perintah berikut untuk pindah ke folder utama: **cd ../../**

```
PS C:\Users\ADMIN\OneDrive\Desktop\praktikum-apl\post-test\post-test-apl-1> cd ../../
PS C:\Users\ADMIN\OneDrive\Desktop\praktikum-apl> █
```

Gambar 5.4 menyesuaikan direktori

6. Inisialisasi Repository Git
 - Jalankan perintah **git init** untuk menginisialisasi repository Git

```
PS C:\Users\ADMIN\OneDrive\Desktop\praktikum-apl\post-test\post-test> git init
Initialized empty Git repository in C:/Users/ADMIN/OneDrive/Desktop/praktikum-apl/post-test/post-test/.git/
```

Gambar 5.5 git init

7. Menambahkan File ke Staging Area

- Tambahkan file ke dalam staging area sebelum melakukan commit: **git add** <nama-file> atau untuk menambahkan semua file sekaligus: **git add**

```
PS C:\Users\ADMIN\OneDrive\Desktop\praktikum-apl\post-test\post-test> git add .
warning: in the working copy of 'post-test-apl-2/.$2409106064-RidwanNurRahman-PT-2.drawio.bkp', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'post-test-apl-2/2409106064-RidwanNurRahman-PT-2.drawio', LF will be replaced by CRLF the next time Git touches it
```

Gambar 5.6 git add

8. Melakukan commit

- Simpan perubahan ke dalam repository dengan memberikan pesan deskriptif, contoh : **git commit -m "Finish Post Test 6"**

```
PS C:\Users\ADMIN\OneDrive\Desktop\praktikum-apl\post-test> git commit -m "Finish Post Test 3"
[main f8d2cf5] Finish Post Test 3
16 files changed, 2229 insertions(+), 347 deletions(-)
create mode 100644 .vscode/settings.json
delete mode 100644 kelas/pertemuan-1/main.cpp
delete mode 100644 kelas/pertemuan-1/main.exe
create mode 100644 post-test-apl-2/.$2409106064-RidwanNurRahman-PT-2.drawio.bkp
create mode 100644 post-test-apl-2/2409106064-RidwanNurRahman-PT-2.cpp
create mode 100644 post-test-apl-2/2409106064-RidwanNurRahman-PT-2.drawio
create mode 100644 post-test-apl-2/2409106064-RidwanNurRahman-PT-2.exe
create mode 100644 post-test-apl-2/tempCodeRunnerFile.cpp
create mode 100644 post-test-apl-3/2409106064-RidwanNurRahman-PT-3.cpp
create mode 100644 post-test-apl-3/2409106064-RidwanNurRahman-PT-3.drawio
create mode 100644 post-test-apl-3/2409106064-RidwanNurRahman-PT-3.exe
```

Gambar 5.7 git commit

9. Menghubungkan Repository Lokal ke GitHub

- Salin URL repository dari GitHub, lalu tempel di terminal.

```
PS C:\Users\ADMIN\OneDrive\Desktop\praktikum-apl\post-test\post-test> git remote add origin https://github.com/Ridwan-rahman-05/praktikum-apl
```

Gambar 5.8 git remote

10. Mengunggah File ke GitHub

- Gunakan perintah **git push -u origin main** untuk mengunggah perubahan ke GitHub.

```

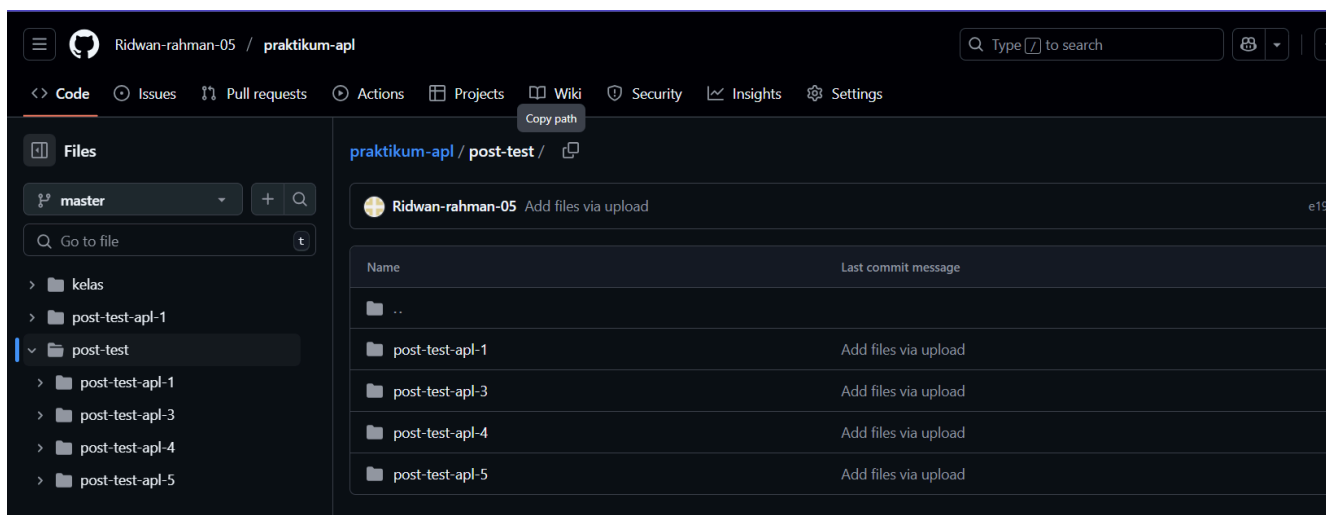
PS C:\Users\ADMIN\OneDrive\Desktop\praktikum-apl\post-test> git push -u origin main
Enumerating objects: 28, done.
Counting objects: 100% (28/28), done.
Delta compression using up to 20 threads
Compressing objects: 100% (22/22), done.
Writing objects: 100% (28/28), 1.58 MiB | 30.00 KiB/s, done.
Total 28 (delta 5), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (5/5), done.
remote:
remote: Create a pull request for 'main' on GitHub by visiting:
remote:   https://github.com/Ridwan-rahman-05/praktikum-apl/pull/new/main
remote:
To https://github.com/Ridwan-rahman-05/praktikum-apl.git

```

Gambar 5.9 git push -u origin main

11. Verifikasi di GitHub

- Buka kembali tab repository GitHub.
- Refresh halaman dan pastikan struktur folder serta file sudah muncul di GitHub.



Gambar 5.10 repository github