

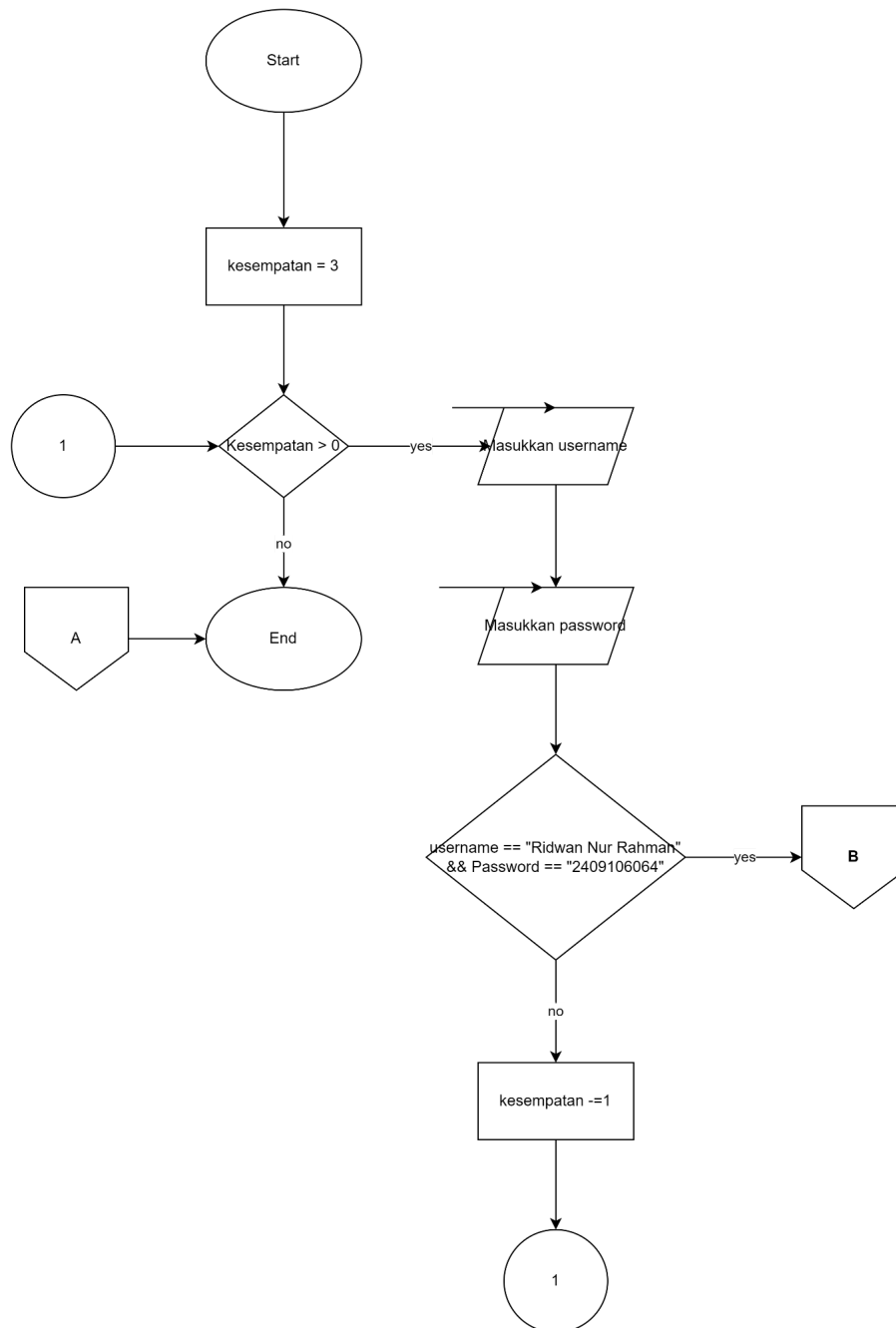
LAPORAN PRAKTIKUM
POSTTEST 3
ALGORITMA PEMROGRAMAN LANJUT



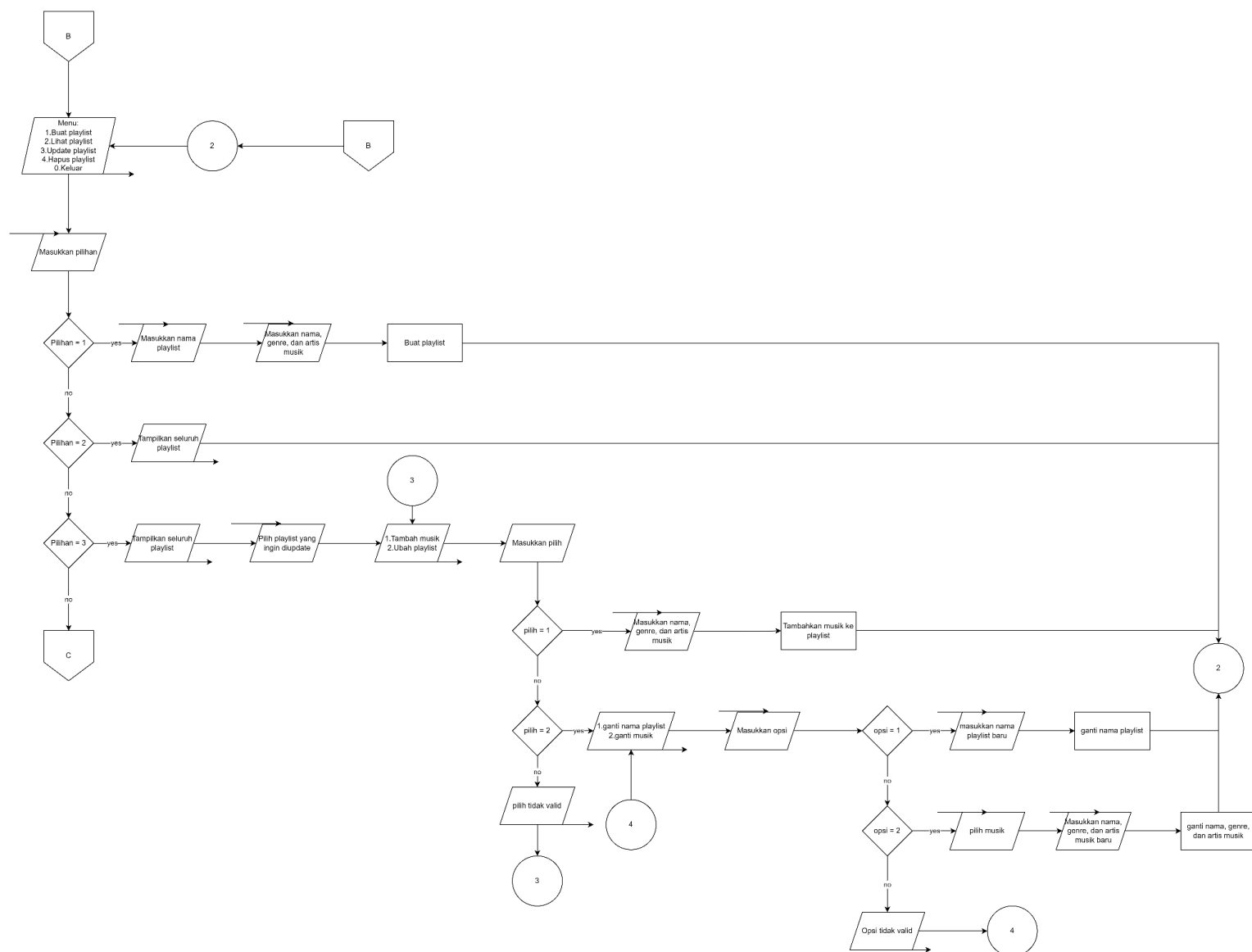
Disusun oleh:
Ridwan Nur Rahman (2409106064)
Kelas (B1 '2024)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

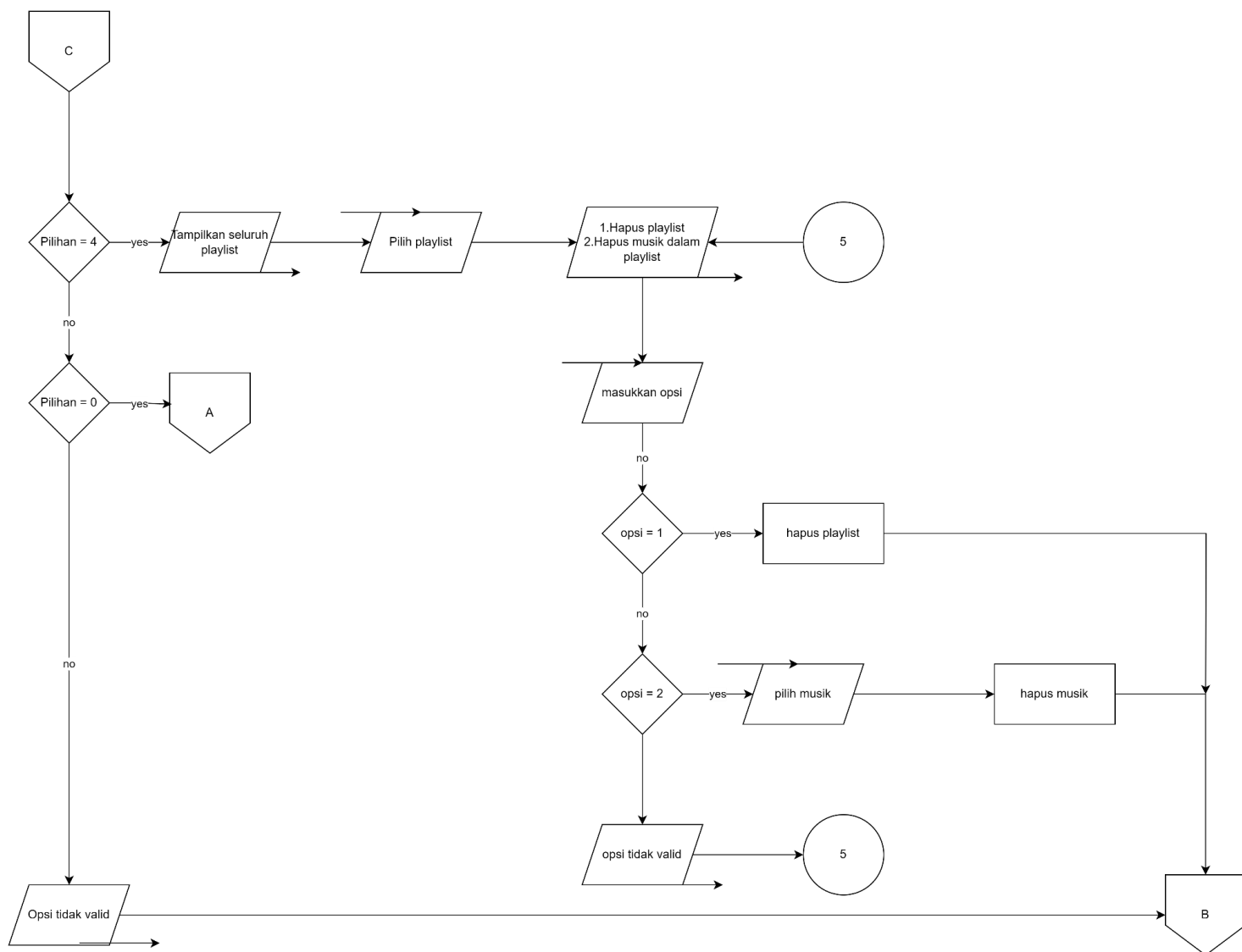
1. Flowchart



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.

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. Login

Fitur ini digunakan untuk memvalidasi user yang ingin menggunakan program.

Source code:

```
while (kesempatan > 0) {  
    cout << "Masukkan Username: " << endl;  
    getline(cin, username);  
    cout << "Masukkan Password: " << endl;  
    cin >> password;  
    cin.ignore();
```

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. Buat Playlist

Fitur untuk membuat playlist baru sekaligus menambahkan musik untuk yang pertama.

```
if (pilihan == 1) {
    if (totalPlaylist < max_playlist) {
        cout << "Nama Playlist: " << endl;
        getline(cin, playlist[totalPlaylist].nama);

        cout << "Musik: " << endl;
        getline(cin,
playlist[totalPlaylist].musik[0].judul);
        cout << "Genre: " << endl;
        getline(cin,
playlist[totalPlaylist].musik[0].genre);
        cout << "Artis: " << endl;
        getline(cin,
playlist[totalPlaylist].musik[0].artis);

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

        cout << "Playlist berhasil dibuat!\n" << endl;
    } else {
        cout << "Playlist sudah penuh! Hapus playlist untuk
menambah yang baru.\n" << endl;
    }
}
```

F. Lihat Playlist

Fitur untuk menampilkan seluruh playlist yang telah dibuat.

Source code:

```
else if (pilihan == 2) {
    if (totalPlaylist == 0) {
        cout << "Belum ada playlist yang dibuat.\n" << endl;
    } else {
        for (int i = 0; i < totalPlaylist; i++) {
            cout << i + 1 << ". " << playlist[i].nama << "\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. Update Playlist

Fitur untuk mengubah nama playlist, mengubah musik, dan menambah musik ke dalam playlist.

Source code:

```
else if (pilihan == 3) {
    if (totalPlaylist == 0) {
        cout << "Belum ada playlist yang bisa diubah.\n" <<
endl;
        continue;
    }

    for (int i = 0; i < totalPlaylist; i++) {
        cout << i + 1 << ". " << playlist[i].nama << "\n";
    }

    cout << "Pilih Playlist: " << endl;
    int pilih;
    cin >> pilih;
    cin.ignore();

    if (pilih > 0 && pilih <= totalPlaylist) {
        cout << "1. Tambah Musik\n" << endl;
        cout << "2. Ubah Playlist\n" << endl;
        cout << "Masukkan pilihan: " << endl;
        int opsi;
        cin >> opsi;
```



```

        cin.ignore();

        if (opsi == 1) {
            if (playlist[pilih - 1].jumlahMusik < 10) {
                int idx = playlist[pilih - 1].jumlahMusik;
                cout << "Musik: " << endl;
                getline(cin, playlist[pilih -
1].musik[idx].judul);

                cout << "Genre: " << endl;
                getline(cin, playlist[pilih -
1].musik[idx].genre);

                cout << "Artis: " << endl;
                getline(cin, playlist[pilih -
1].musik[idx].artis);

                playlist[pilih - 1].jumlahMusik++;
                cout << "Musik berhasil ditambahkan!\n" << endl;
            } else {
                cout << "Playlist ini sudah penuh!\n" << endl;
            }
        }

        else if (opsi == 2) {
            cout << "1. Ganti Nama Playlist\n" << endl;
            cout << "2. Ganti Musik\n" << endl;
            cout << "Masukkan pilihan: " << endl;
            int opsiubah;
            cin >> opsiubah;
            cin.ignore();

            if (opsiubah == 1) {
                cout << "Masukkan nama Playlist baru: " << endl;
                getline(cin, playlist[pilih - 1].nama);
                cout << "Nama Playlist berhasil diubah!\n" <<
endl;

            }

            else if (opsiubah == 2) {
                for (int j = 0; j < playlist[pilih -
1].jumlahMusik; j++) {
                    cout << j + 1 << ". " << playlist[pilih -
1].musik[j].judul << "\n";
                }

                cout << "Pilih musik yang ingin diubah: " <<
endl;

                int pilihMusik;
                cin >> pilihMusik;
                cin.ignore();

                if (pilihMusik > 0 && pilihMusik <=

```

```

playlist[pilih - 1].jumlahMusik) {
    cout << "Masukkan Nama Lagu Baru: " << endl;
    getline(cin, playlist[pilih -
1].musik[pilihMusik - 1].judul);
    cout << "Masukkan Genre Baru: " << endl;
    getline(cin, playlist[pilih -
1].musik[pilihMusik - 1].genre);
    cout << "Masukkan Artis Baru: " << endl;
    getline(cin, playlist[pilih -
1].musik[pilihMusik - 1].artis);

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

```

H. Hapus Playlist atau Musik

Fitur untuk menghapus playlist atau musik yang ada didalam playlist

Source code:

```

else if (pilihan == 4) {
    if (totalPlaylist == 0) {
        cout << "Belum ada playlist yang bisa dihapus.\n" <<
endl;
        continue;
    }

    for (int i = 0; i < totalPlaylist; i++) {
        cout << i + 1 << ". " << playlist[i].nama << "\n";
    }

    cout << "Pilih Playlist: " << endl;
    int pilih;
    cin >> pilih;
    cin.ignore();

    if (pilih > 0 && pilih <= totalPlaylist) {
        cout << "1. Hapus Playlist\n" << endl;
        cout << "2. Hapus Musik dalam Playlist\n" << endl;
        cout << "Masukkan pilihan: " << endl;
        int opsi;
    }
}

```

```

        cin >> opsi;
        cin.ignore();

        if (opsi == 1) {
            for (int i = pilih - 1; i < totalPlaylist - 1; i++)
            {
                playlist[i] = playlist[i + 1];
            }
            totalPlaylist--;
            cout << "Playlist berhasil dihapus!\n" << endl;
        }

        else if (opsi == 2) {
            for (int j = 0; j < playlist[pilih - 1].jumlahMusik;
j++) {
                cout << j + 1 << ". " << playlist[pilih -
1].musik[j].judul << "\n";
            }

            cout << "Pilih Musik yang ingin dihapus: " << endl;
            int pilihMusik;
            cin >> pilihMusik;
            cin.ignore();

            if (pilihMusik > 0 && pilihMusik <= playlist[pilih -
1].jumlahMusik) {
                for (int j = pilihMusik - 1; j < playlist[pilih
- 1].jumlahMusik - 1; j++) {
                    playlist[pilih - 1].musik[j] =
playlist[pilih - 1].musik[j + 1];
                }
                playlist[pilih - 1].jumlahMusik--;
                cout << "Musik berhasil dihapus!\n" << endl;
            } else {
                cout << "Pilihan tidak valid!\n" << endl;
            }
        } else {
            cout << "Pilihan tidak valid!\n" << endl;
        }
    } else {
        cout << "Playlist tidak ditemukan!\n" << endl;
    }
}

```

I. Keluar Program

Fitur untuk keluar dari program.

Source code:

```

else if (pilihan == 0) {

```

```
        cout << "Terima kasih telah menggunakan manajemen  
playlist!\n" << endl;  
        break;
```

4. Screenshot Hasil Program

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

===== Menu Manajemen Playlist Musik =====
1. Buat playlist
2. Lihat playlist
3. Update playlist
4. Hapus playlist
0. Keluar program
Masukkan pilihan: █
```

Gambar 4.1 berhasil login

```
Masukkan Username:
Ridwan Nur Rahman
Masukkan Password:
123456
Username atau Password salah. Kesempatan login tersisa: 2
```

Gambar 4.2 gagal login

```
Masukkan Username:
Ridwan Nur Rahman
Masukkan Password:
123456
Username atau Password salah. Kesempatan login tersisa: 2
Masukkan Username:
richo
Masukkan Password:
122435
Username atau Password salah. Kesempatan login tersisa: 1
Masukkan Username:
ridwan
Masukkan Password:
2403280
Username atau Password salah. Kesempatan login tersisa: 0
```

Gambar 4.3 gagal login 3 kali

```

===== Menu Manajemen Playlist Musik =====
1. Buat playlist
2. Lihat playlist
3. Update playlist
4. Hapus playlist
0. Keluar program
Masukkan pilihan: 1
Nama Playlist:
Galau
Musik:
blue
Genre:
indie
Artis:
yung kai
Playlist berhasil dibuat!

===== Menu Manajemen Playlist Musik =====
1. Buat playlist
2. Lihat playlist
3. Update playlist
4. Hapus playlist
0. Keluar program
Masukkan pilihan: 2
1. Galau
   - blue | indie | yung kai

```

Gambar 4.4 buat playlist

```

===== Menu Manajemen Playlist Musik =====
1. Buat playlist
2. Lihat playlist
3. Update playlist
4. Hapus playlist
0. Keluar program
Masukkan pilihan: 3
1. Galau
Pilih Playlist:
1
1. Tambah Musik

2. Ubah Playlist

Masukkan pilihan:
1
Musik:
That girl
Genre:
Dance pop
Artis:
Olly Murs
Musik berhasil ditambahkan!

===== Menu Manajemen Playlist Musik =====
1. Buat playlist
2. Lihat playlist
3. Update playlist
4. Hapus playlist
0. Keluar program
Masukkan pilihan: 2
1. Galau
   - blue | indie | yung kai
   - That girl | Dance pop | Olly Murs

```

Gambar 4.5 tambah musik

```

Masukkan pilihan:
2
1. Ganti Nama Playlist

2. Ganti Musik

Masukkan pilihan:
1
Masukkan nama Playlist baru:
sad
Nama Playlist berhasil diubah!

===== Menu Manajemen Playlist Musik =====
1. Buat playlist
2. Lihat playlist
3. Update playlist
4. Hapus playlist
0. Keluar program
Masukkan pilihan: 2
1. sad
   - blue | indie | yung kai
   - that girl | dance pop | olly murs

```

Gambar 4.6 ganti nama playlist

```

Masukkan pilihan:
2
1. Ganti Nama Playlist

2. Ganti Musik

Masukkan pilihan:
2
1. blue
2. that girl
Pilih musik yang ingin diubah:
1
Masukkan Nama Lagu Baru:
end of beginning
Masukkan Genre Baru:
dream pop
Masukkan Artis Baru:
joe kerry
Lagu berhasil diperbarui!

===== Menu Manajemen Playlist Musik =====
1. Buat playlist
2. Lihat playlist
3. Update playlist
4. Hapus playlist
0. Keluar program
Masukkan pilihan: 2
1. sad
   - end of beginning | dream pop | joe kerry
   - that girl | dance pop | olly murs

```

Gambar 4.7 ganti lagu

```

1. sad
Pilih Playlist:
1
1. Hapus Playlist

2. Hapus Musik dalam Playlist

Masukkan pilihan:
2
1. end of beginning
2. that girl
Pilih Musik yang ingin dihapus:
1
Musik berhasil dihapus!

===== Menu Manajemen Playlist Musik =====
1. Buat playlist
2. Lihat playlist
3. Update playlist
4. Hapus playlist
0. Keluar program
Masukkan pilihan: 2
1. sad
  - that girl | dance pop | olly murs

```

Gambar 4.8 hapus musik

```

1. sad
2. anime
Pilih Playlist:
1
1. Hapus Playlist

2. Hapus Musik dalam Playlist

Masukkan pilihan:
1
Playlist berhasil dihapus!

===== Menu Manajemen Playlist Musik =====
1. Buat playlist
2. Lihat playlist
3. Update playlist
4. Hapus playlist
0. Keluar program
Masukkan pilihan: 2
1. anime
  - Shinzo wo sasageyo | rock | linked horizon

```

Gambar 4.9 hapus playlist

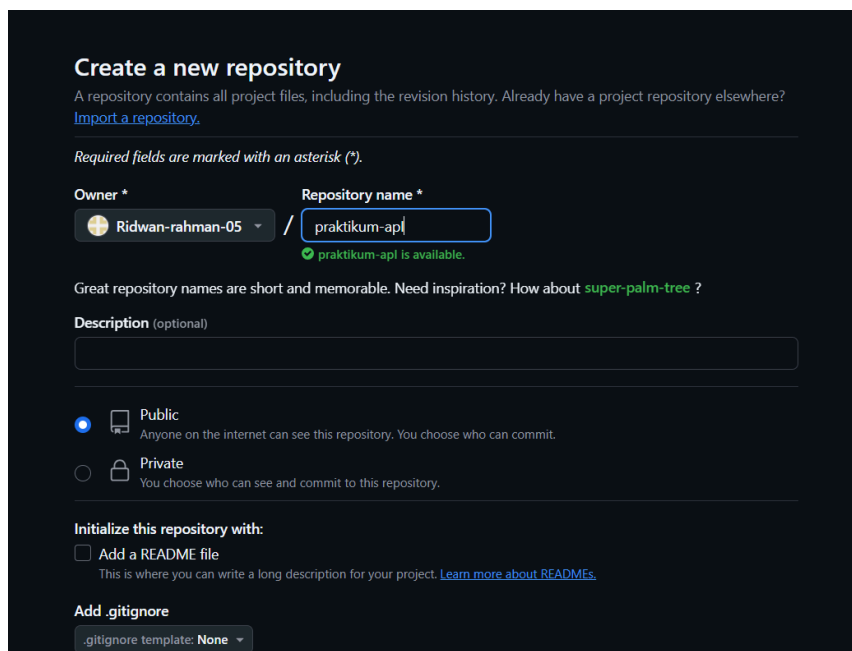

```
===== Menu Manajemen Playlist Musik =====  
1. Buat playlist  
2. Lihat playlist  
3. Update playlist  
4. Hapus playlist  
0. Keluar program  
Masukkan pilihan: 0  
Terima kasih telah menggunakan manajemen playlist!
```

Gambar 4.10 keluar program

5. Langkah-langkah Git

1. Buat Repository di GitHub

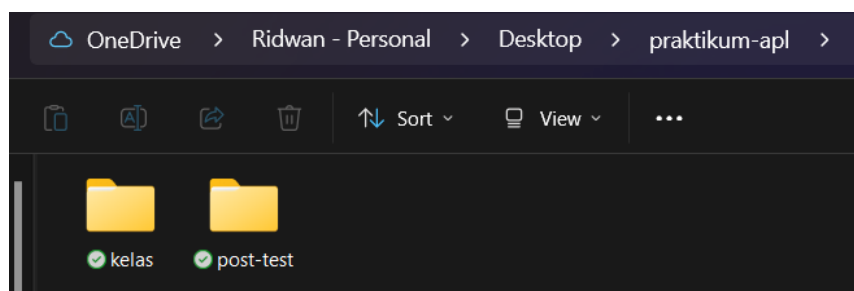
- Masuk ke akun GitHub.
- Klik tombol New Repository.
- Beri nama repository “praktikum-apl”.
- Pilih opsi Public agar bisa diakses oleh semua orang.
- Klik Create Repository.



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.



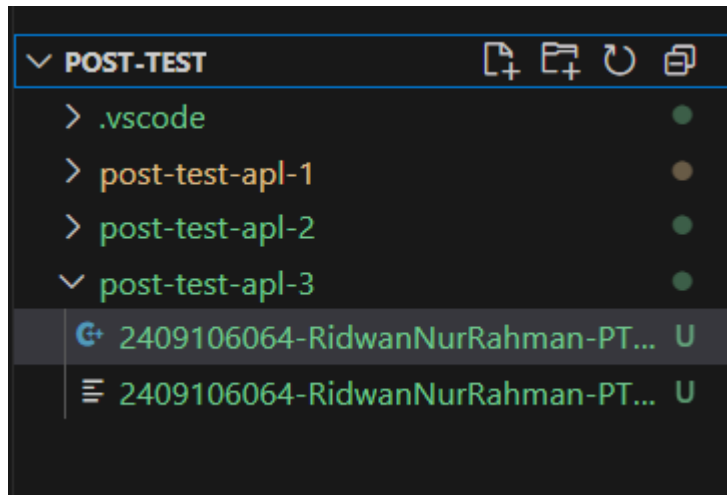
Gambar 5.2 buat folde kelas dan post-test dalam folder praktikum-apl

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-3 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 3"**

```
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

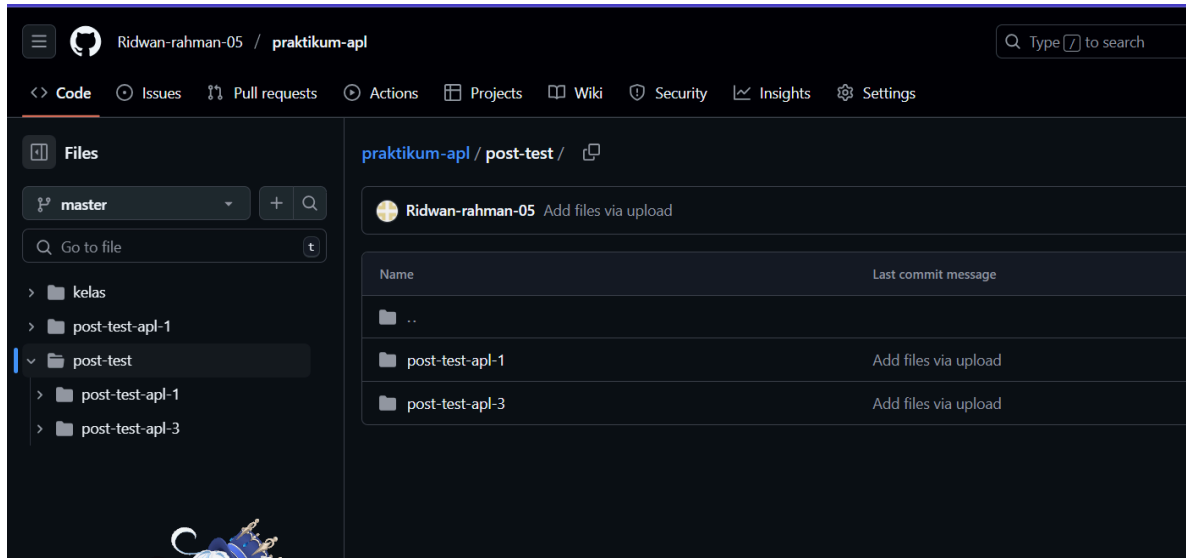
- Gunakan perintah **git push -u origin main** untuk mengunggah perubahan ke GitHub.

```
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