

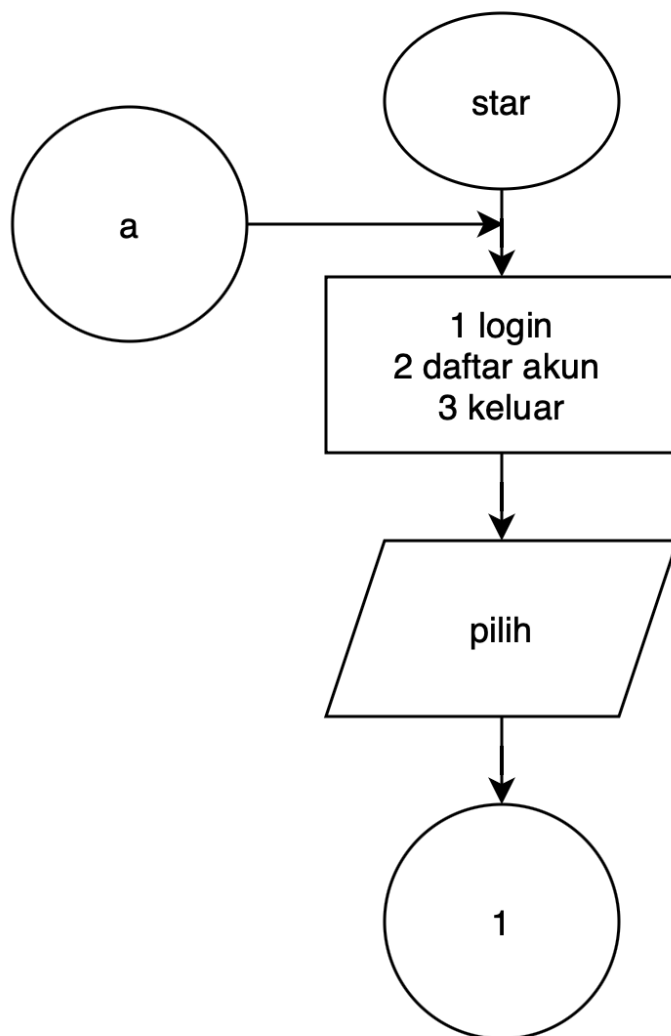
**LAPORAN PRAKTIKUM**  
**POSTTEST 4**  
**ALGORITMA PEMROGRAMAN LANJUT**



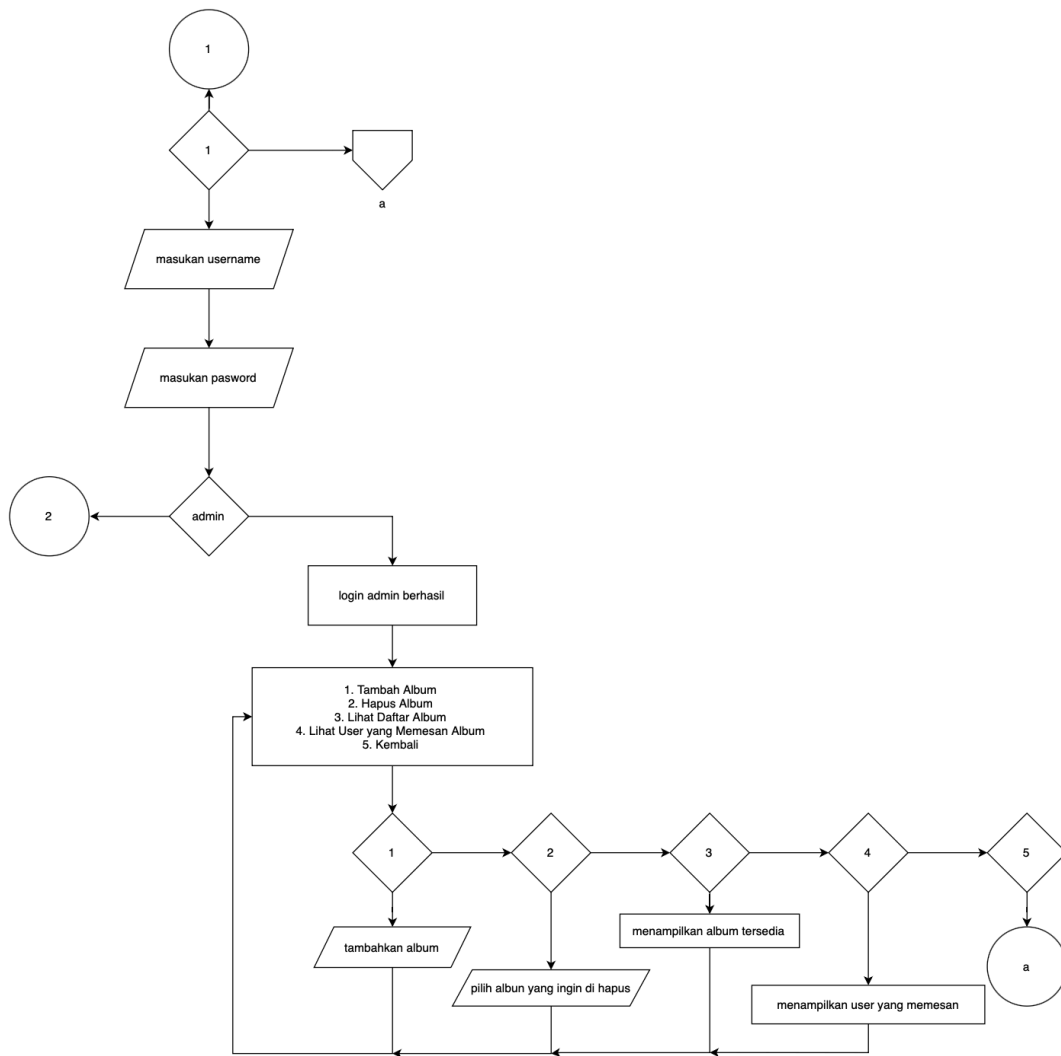
**Disusun oleh:**  
**Nama (2409106094)**  
**Kelas (c'1)**

**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2025**

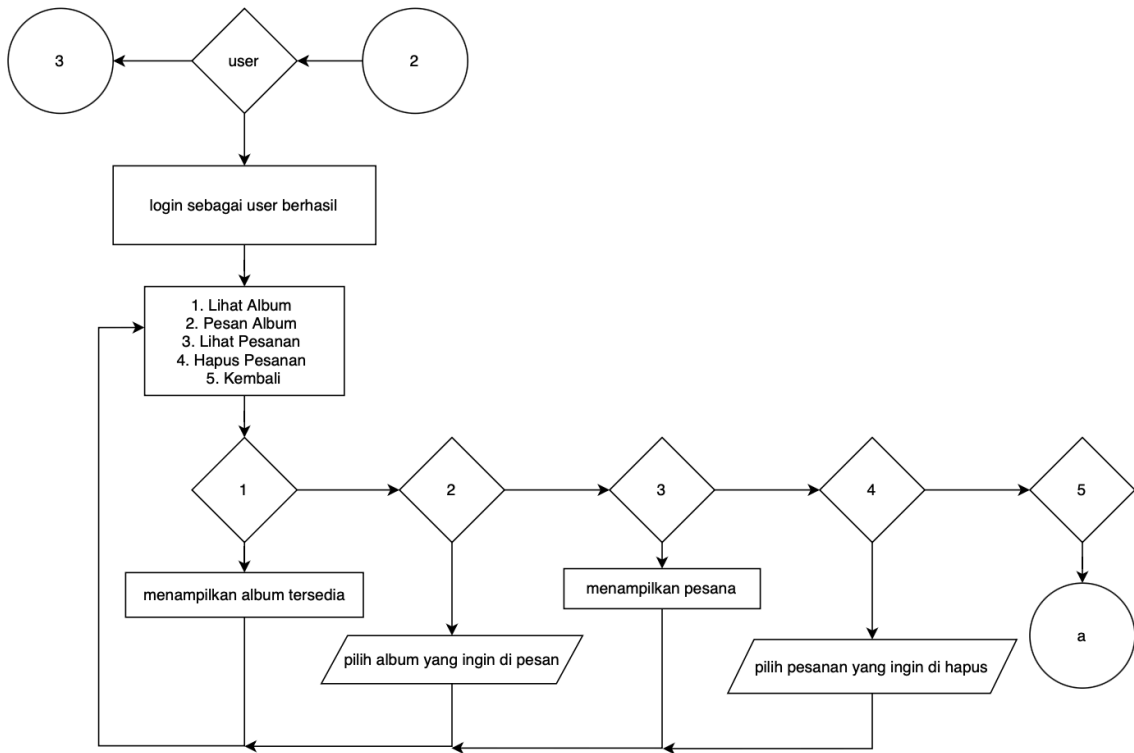
## 1. Flowchart



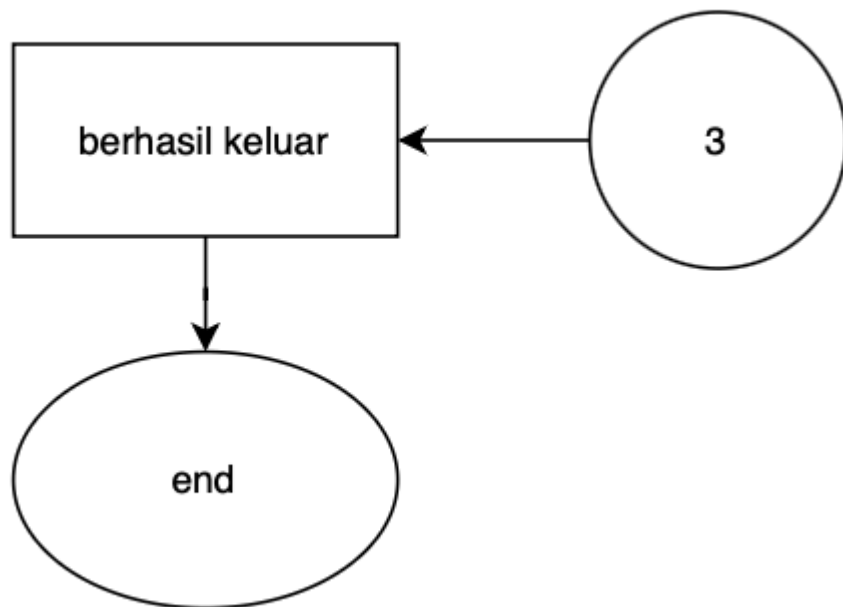
menu awal dari progream nya yang menampilkan opsi login, daftar, dan keluar yang bisa kalian pilih



dan ini adalah tampilan menu admin



kalo ini adalah menu user



ini menu end untuk keluar atau berhenti program

## 2. Analisis Program

### 2.1 Deskripsi Singkat Program

Program ini dibuat untuk gampang mempermudah admin untuk mendata orang atau user yang ingin pri order

## 3. Source Code

```
#include <iostream>
#include <string>
#include <limits>
using namespace std;
struct Album {
    string nama;
};
struct User {
    string username;
    string password;
    Album pesanan[10];
    int jumlahPesanan = 0;
};
const int MAKSIMUM_USER = 100;
const int MAKSIMUM_ALBUM = 10;
Album daftarAlbum[MAKSIMUM_ALBUM];
User daftarUser[MAKSIMUM_USER];
int totalAlbum = 5;
int totalUser = 0;
void tampilkanDaftarAlbum() {
    cout << "\nDaftar Album:\n";
    for (int i = 0; i < totalAlbum; i++) {
        cout << i + 1 << ". " << daftarAlbum[i].nama <<
endl;
    }
}
void tambahAlbum() {
    if (totalAlbum >= MAKSIMUM_ALBUM) {
        cout << "Kapasitas album penuh.\n";
        return;
    }
}
```

```

    }
    cout << "Masukkan nama album: ";
    cin.ignore();
    getline(cin, daftarAlbum[totalAlbum].nama);
    totalAlbum++;
    cout << "Album berhasil ditambahkan!\n";
}

void hapusAlbum() {
    int indeks;
    cout << "Masukkan nomor album yang ingin dihapus: ";
    cin >> indeks;
    if (indeks >= 1 && indeks <= totalAlbum) {
        for (int i = indeks - 1; i < totalAlbum - 1; i++) {
            daftarAlbum[i] = daftarAlbum[i + 1];
        }
        totalAlbum--;
        cout << "Album berhasil dihapus!\n";
    } else {
        cout << "Nomor album tidak valid.\n";
    }
}

void lihatUserDanPesannya() {
    cout << "\nUser yang Memesan Album:\n";
    for (int i = 0; i < totalUser; i++) {
        cout << "User: " << daftarUser[i].username << endl;
        for (int j = 0; j < daftarUser[i].jumlahPesanan;
j++) {
            cout << " - " << daftarUser[i].pesanan[j].nama
<< endl;
        }
    }
}

void menuAdmin() {
    int pilihanAdmin;
    do {
        cout << "\nMenu Admin:\n";
        cout << "1. Tambah Album\n2. Hapus Album\n3. Lihat

```

```

Daftar Album\n4. Lihat User yang Memesan Album\n5.
Kembali\nPilihan: ";
    cin >> pilihanAdmin;
    switch (pilihanAdmin) {
        case 1: tambahAlbum(); break;
        case 2: hapusAlbum(); break;
        case 3: tampilkanDaftarAlbum(); break;
        case 4: lihatUserDanPesannya(); break;
    }
} while (pilihanAdmin != 5);
}
void pesanAlbum(User &user) {
    int indeks;
    cout << "Masukkan nomor album yang ingin dipesan: ";
    cin >> indeks;
    if (indeks >= 1 && indeks <= totalAlbum) {
        user.pesanan[user.jumlahPesanan] =
daftarAlbum[indeks - 1];
        user.jumlahPesanan++;
        cout << "Album berhasil dipesan!\n";
    } else {
        cout << "Nomor album tidak valid.\n";
    }
}
void lihatPesanan(const User &user) {
    cout << "\nPesanan Anda:\n";
    for (int i = 0; i < user.jumlahPesanan; i++) {
        cout << i + 1 << ". " << user.pesanan[i].nama <<
endl;
    }
}
void hapusPesanan(User &user) {
    int indeks;
    cout << "Masukkan nomor pesanan yang ingin dihapus: ";
    cin >> indeks;
    if (indeks >= 1 && indeks <= user.jumlahPesanan) {
        for (int i = indeks - 1; i < user.jumlahPesanan -

```

```

1; i++) {
    user.pesanan[i] = user.pesanan[i + 1];
}
user.jumlahPesanan--;
cout << "Pesanan berhasil dihapus!\n";
} else {
    cout << "Nomor pesanan tidak valid.\n";
}
}

void menuUser(User &user) {
    int pilihanUser;
    do {
        cout << "\nMenu User:\n";
        cout << "1. Lihat Album\n2. Pesan Album\n3. Lihat
Pesanan\n4. Hapus Pesanan\n5. Kembali\nPilihan: ";
        cin >> pilihanUser;
        switch (pilihanUser) {
            case 1: tampilkanDaftarAlbum(); break;
            case 2: pesanAlbum(user); break;
            case 3: lihatPesanan(user); break;
            case 4: hapusPesanan(user); break;
        }
    } while (pilihanUser != 5);
}

void login(string username, string password) {
    if (username == "kelvin" && password == "2409106094") {
        cout << "Login sebagai admin berhasil!\n";
        menuAdmin();
    } else {
        bool loginBerhasil = false;
        for (int i = 0; i < totalUser; i++) {
            if (daftarUser[i].username == username &&
daftarUser[i].password == password) {
                loginBerhasil = true;
                cout << "Login sebagai user berhasil!\n";
                menuUser(daftarUser[i]);
                break;
            }
        }
    }
}

```



```

        }
    }
    if (!loginBerhasil) {
        cout << "Login gagal. Username atau password
salah.\n";
    }
}
}
void buatAkun() {
    if (totalUser >= MAKSIMUM_USER) {
        cout << "Kapasitas user penuh.\n";
        return;
    }
    cout << "Masukkan username baru: ";
    cin >> daftarUser[totalUser].username;
    cout << "Masukkan password baru: ";
    cin >> daftarUser[totalUser].password;
    totalUser++;
    cout << "Akun berhasil dibuat! Silakan login.\n";
}
int main() {
    daftarAlbum[0] = {"Pink Venom - Blackpink"};
    daftarAlbum[1] = {"Rosie - Rose"};
    daftarAlbum[2] = {"Ruby - Jennie"};
    daftarAlbum[3] = {"Alter Ego - Lisa"};
    daftarAlbum[4] = {"Amortage - Jiso"};
    int pilihan;
    string username, password;
    do {
        cout << "\nMenu Utama:\n1. Login\n2. Buat Akun\n3.
Keluar\nPilihan: ";
        cin >> pilihan;
        switch (pilihan) {
            case 1:
                cout << "Masukkan username: ";
                cin >> username;
                cout << "Masukkan password: ";

```

```
        cin >> password;
        login(username, password);
        break;
    case 2:
        buatAkun();
        break;
    }
} while (pilihan != 3);
return 0;
}
```

#### 4. Hasil Output

##### A. Pengujian Login

Menu Utama:

1. Login
2. Buat Akun
3. Keluar

Pilihan: 1

Masukkan username: tes

Masukkan password: tes

Login sebagai user berhasil!

## B. Pengujian Input Menu User

Menu User:

1. Lihat Album
2. Pesan Album
3. Lihat Pesanan
4. Hapus Pesanan
5. Kembali

Pilihan: 1

Daftar Album:

1. Pink Venom – Blackpink
2. Rosie – Rose
3. Ruby – Jennie
4. Alter Ego – Lisa
5. Amortage – Jiso

## C. Pengujian Input Menu admin

Menu Admin:

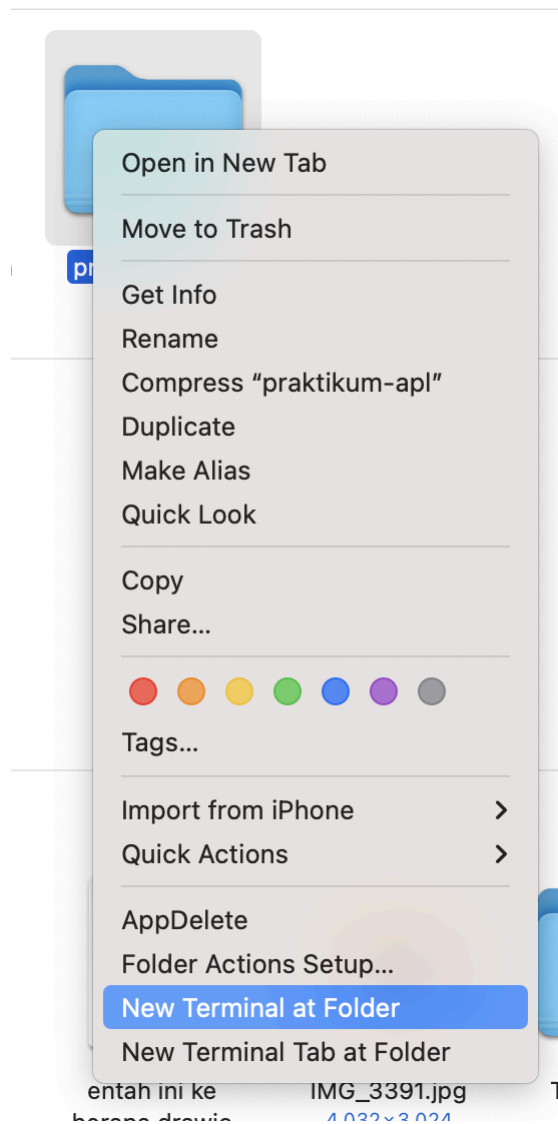
1. Tambah Album
2. Hapus Album
3. Lihat Daftar Album
4. Lihat User yang Memesan Album
5. Kembali

Pilihan: 1

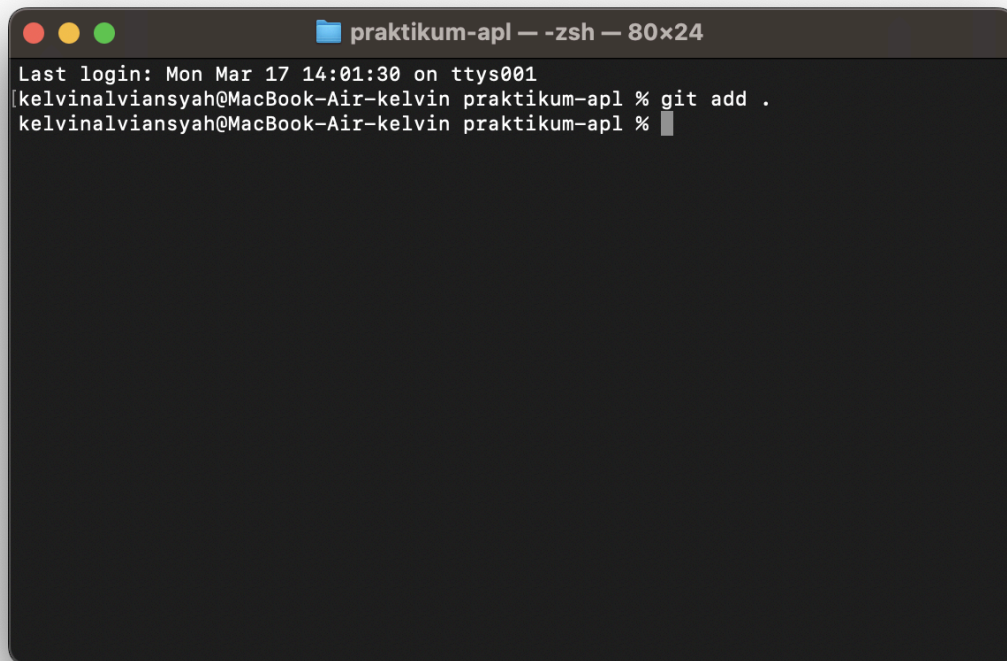
Masukkan nama album: Air – yeji

Album berhasil ditambahkan!

## 5. GIT



Petama klik kanan di folder lalu pilih seperti di gambar



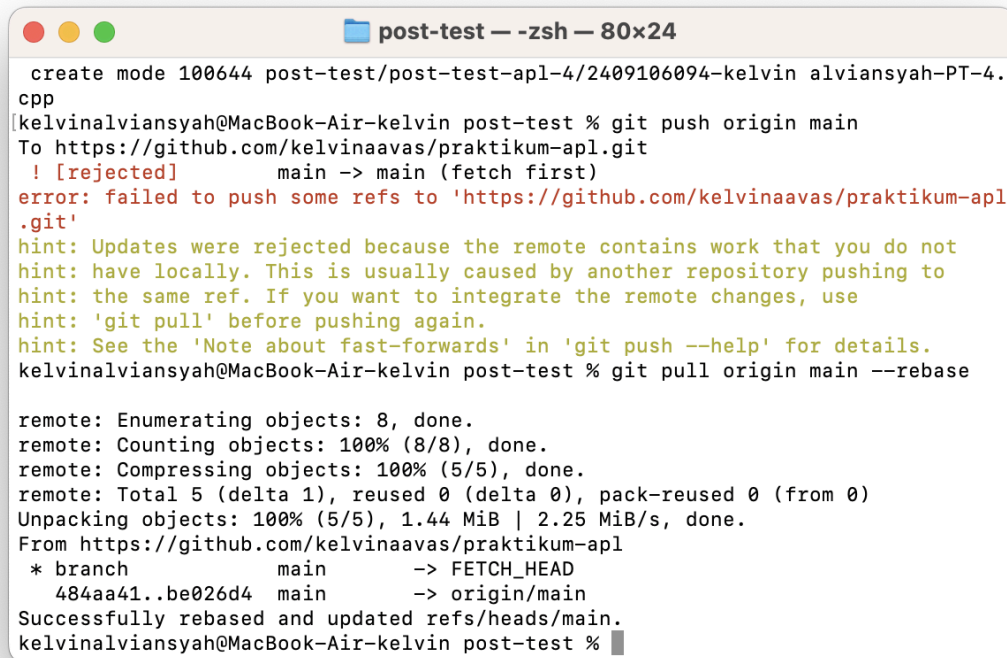
```
praktikum-apl — -zsh — 80x24
Last login: Mon Mar 17 14:01:30 on ttys001
[kelvinalviansyah@MacBook-Air-kelvin praktikum-apl % git add .
kelvinalviansyah@MacBook-Air-kelvin praktikum-apl %
```

lalu ketik `git add .`



```
post-test — -zsh — 80x24
Last login: Fri Apr 11 17:31:07 on ttys001
[kelvinalviansyah@MacBook-Air-kelvin post-test % git add .
[kelvinalviansyah@MacBook-Air-kelvin post-test % git commit -m "post-test-apl-4"
[main 78e5c66] "post-test-apl-4"
 2 files changed, 196 insertions(+)
 create mode 100644 post-test/post-test-apl-4/2409106094-kelvin alviansyah-PT-4.
  cpp
kelvinalviansyah@MacBook-Air-kelvin post-test %
```

lalu ketikan `git commit -m "nama file"`



```
post-test — -zsh — 80x24
create mode 100644 post-test/post-test-apl-4/2409106094-kelvin alviansyah-PT-4.
cpp
[kelvinalviansyah@MacBook-Air-kelvin post-test % git push origin main
To https://github.com/kelvinaavas/praktikum-apl.git
! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/kelvinaavas/praktikum-apl
.git'
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
kelvinalviansyah@MacBook-Air-kelvin post-test % git pull origin main --rebase

remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (5/5), 1.44 MiB | 2.25 MiB/s, done.
From https://github.com/kelvinaavas/praktikum-apl
* branch                main          -> FETCH_HEAD
   484aa41..be026d4      main          -> origin/main
Successfully rebased and updated refs/heads/main.
kelvinalviansyah@MacBook-Air-kelvin post-test %
```

lalu ketik `git push origin main` lalu file sudah masuk di git hub

```
post-test — -zsh — 80x24
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
kelvinalviansyah@MacBook-Air-kelvin post-test % git pull origin main --rebase

remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (5/5), 1.44 MiB | 2.25 MiB/s, done.
From https://github.com/kelvinaavas/praktikum-apl
* branch      main      -> FETCH_HEAD
   484aa41..be026d4  main    -> origin/main
Successfully rebased and updated refs/heads/main.
kelvinalviansyah@MacBook-Air-kelvin post-test % git push origin main

Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 2.93 KiB | 2.93 MiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/kelvinaavas/praktikum-apl.git
   be026d4..6515726  main -> main
kelvinalviansyah@MacBook-Air-kelvin post-test % █
```

di sini karna terjadi eror kita ketikan `git pull origin main --rebase` lalu mengetikan kembali `git push origin main` lalu file masuk ke github