LAPORAN PRAKTIKUM
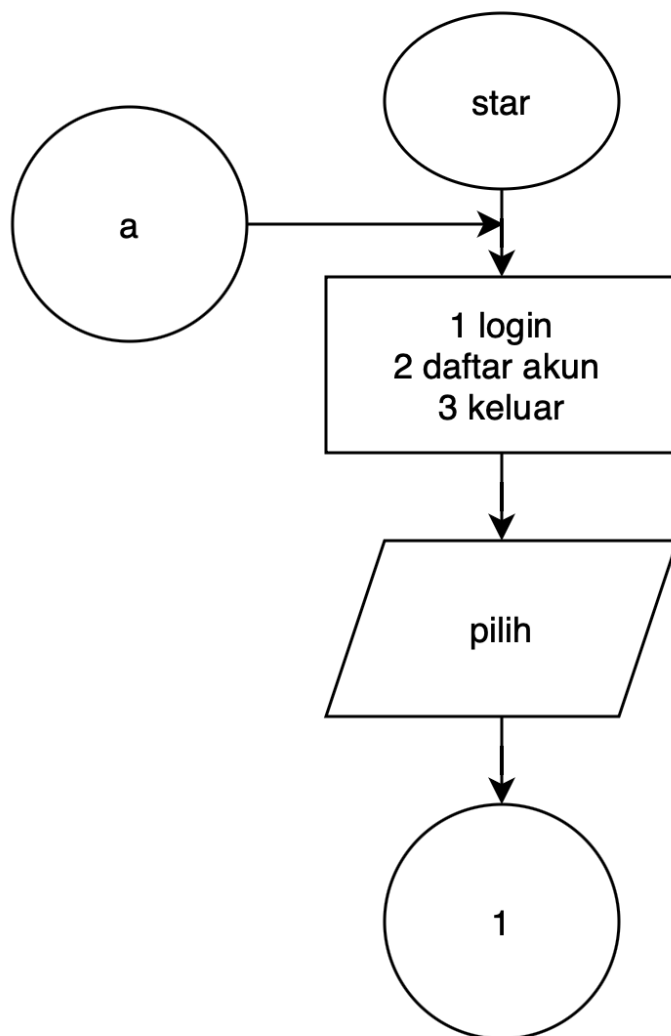
POSTTEST 5
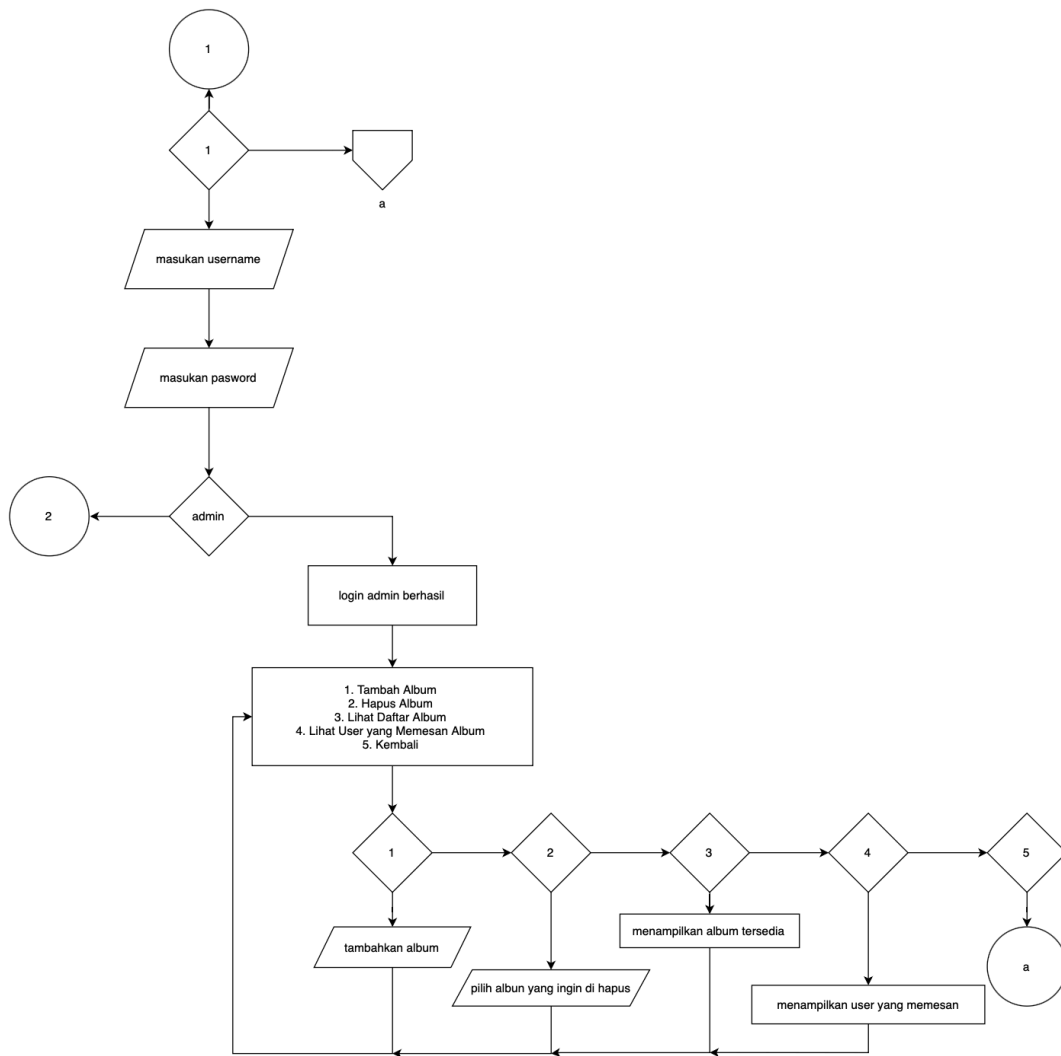
ALGORITMA PEMROGRAMAN LANJUT

Disusun oleh:

Nama (2409106094)

Kelas (c'1)

PROGRAM STUDI INFORMATIKA
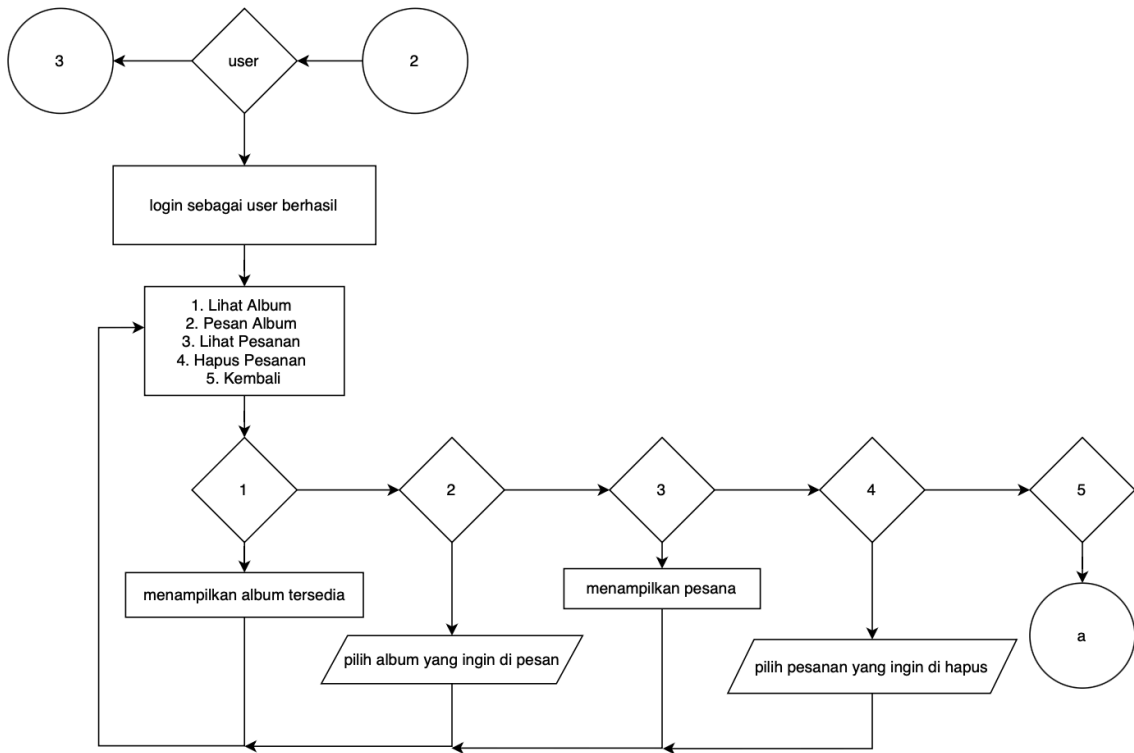
UNIVERSITAS MULAWARMAN
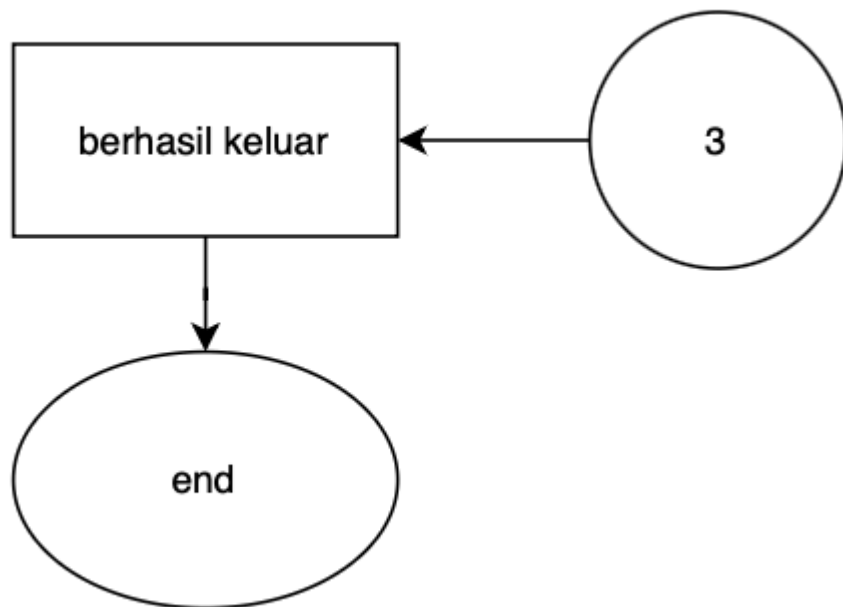
SAMARINDA

2025

1. Flowchart

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

```
            ( 1 )
             |
             ↑
        ◇ 1 ──────→ ⬠
             |         a
             ↓
     ╱ masukan username ╱
             |
             ↓
     ╱ masukan pasword ╱
             |
             ↓
( 2 ) ←── ◇ admin
                  |
                  ↓
        ┌──────────────┐
        │ login admin berhasil │
        └──────────────┘
                  |
                  ↓
   ┌──────────────────────────────┐
   │  1. Tambah Album              │
   │  2. Hapus Album               │
   │  3. Lihat Daftar Album        │
   │  4. Lihat User yang Memesan Album │
   │  5. Kembali                   │
   └──────────────────────────────┘
        |
        ↓
     ◇ 1 ──→ ◇ 2 ──→ ◇ 3 ──→ ◇ 4 ──→ ◇ 5
        |        |        |        |        |
        ↓        ↓        ↓        ↓        ↓
  ╱ tambahkan ╱   ╱ pilih albun ╱  ┌─────────────┐  ┌──────────────┐  ( a )
  ╱ album    ╱   ╱ yang ingin ╱   │ menampilkan │  │ menampilkan  │
                ╱ di hapus   ╱    │ album tersedia│ │ user yang memesan│
                                  └─────────────┘  └──────────────┘
```

dan ini adalah tampilan menu admin

```
                3  ←  user  →  2
                        │
                        ▼
              login sebagai user berhasil
                        │
                        ▼
                  1. Lihat Album
                  2. Pesan Album
                  3. Lihat Pesanan
                  4. Hapus Pesanan
                  5. Kembali
                        │
                        ▼
        1  →  2  →  3  →  4  →  5
        │           │           │        │
        ▼           ▼           ▼        ▼
  menampilkan   pilih album   menampilkan   pilih pesanan    a
  album         yang ingin di  pesana       yang ingin
  tersedia      pesan                       di hapus
```

kalo ini adalah menu user

```
  berhasil keluar  ←  3

        │
        ▼
       end
```

ini menu end untuk keluar atau berhenti program

2. Analisis Program

2.1 Deskripsi Singkat Program

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

3. Source Code

```cpp
#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;
// Fungsi menggunakan pointer (address-of & dereference)
void tambahAlbumPointer(Album* ptrAlbum, int* total) {
    if (*total >= MAKSIMUM_ALBUM) {
        cout << "Kapasitas album penuh.\n";
        return;
    }
    cout << "Masukkan nama album: ";
    cin.ignore();
    getline(cin, (ptrAlbum + *total)->nama);
    (*total)++;
    cout << "Album berhasil ditambahkan melalui pointer!\n";
}
void tampilkanDaftarAlbum() {
    cout << "\nDaftar Album:\n";
    for (int i = 0; i < totalAlbum; i++) {
        cout << i + 1 << ". " << daftarAlbum[i].nama << endl;
    }
}
```

```cpp
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 lihatUserDanPesanannya() {
    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: tambahAlbumPointer(daftarAlbum, &totalAlbum); break;
// penggunaan pointer
            case 2: hapusAlbum(); break;
            case 3: tampilkanDaftarAlbum(); break;
            case 4: lihatUserDanPesanannya(); 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++;
```

```cpp
            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;
```

```cpp
        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!
```
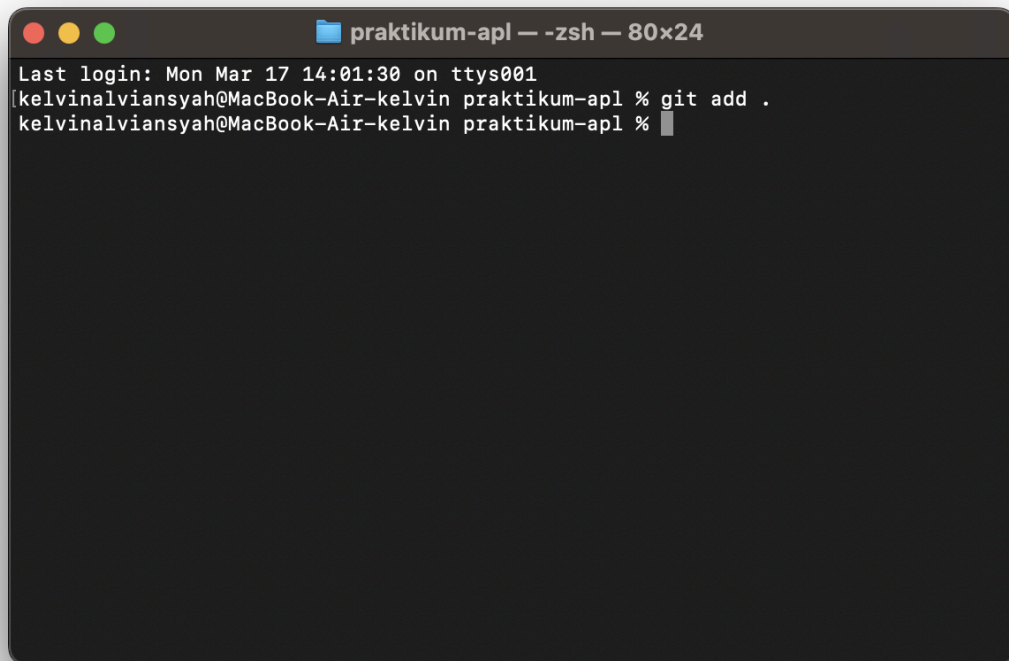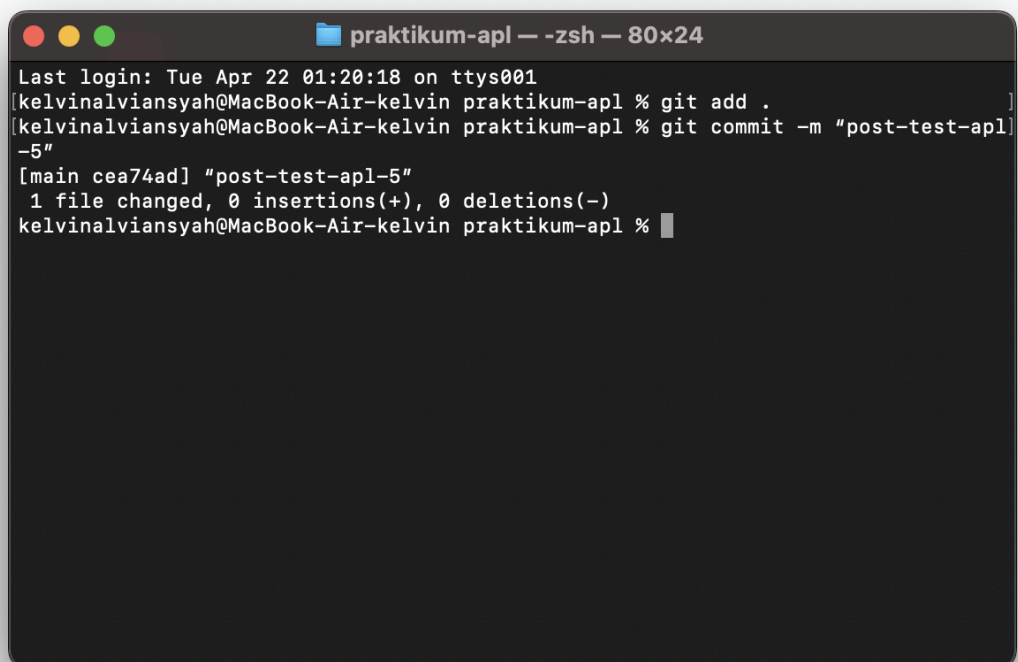
5. GIT
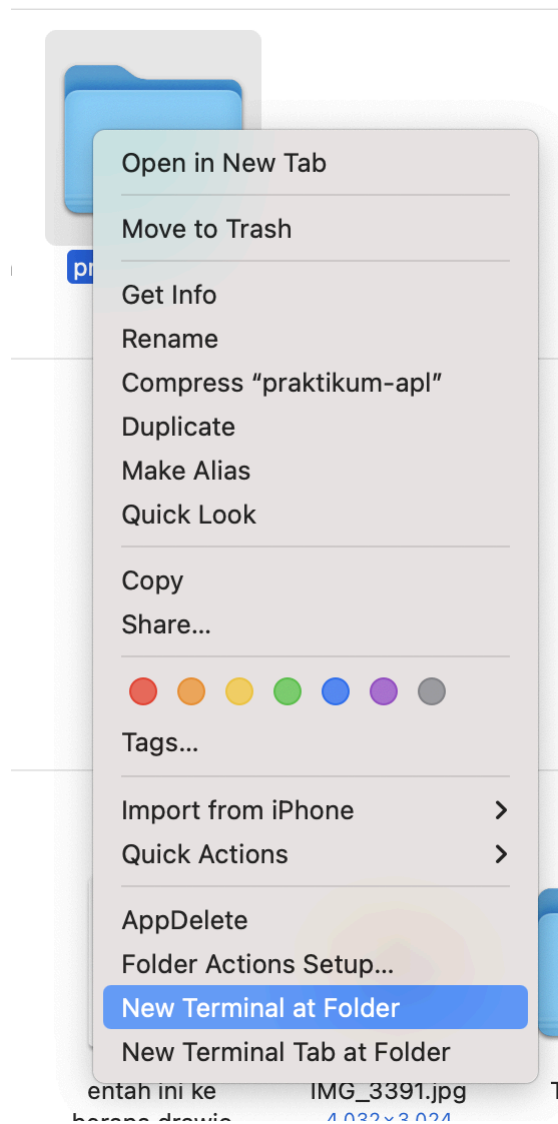
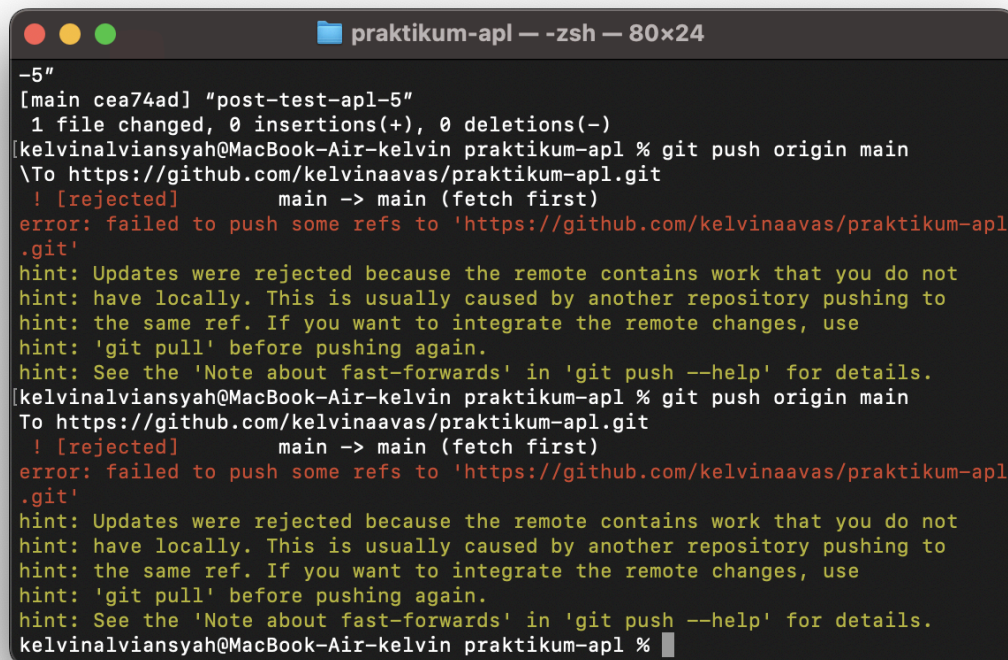Petama klik kanan di folder lalu pilih seperti di gambar

```
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 .`


```
Last login: Tue Apr 22 01:20:18 on ttys001
[kelvinalviansyah@MacBook-Air-kelvin praktikum-apl % git add .
[kelvinalviansyah@MacBook-Air-kelvin praktikum-apl % git commit -m "post-test-apl]
-5"
[main cea74ad] "post-test-apl-5"
 1 file changed, 0 insertions(+), 0 deletions(-)
kelvinalviansyah@MacBook-Air-kelvin praktikum-apl %
```

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

```
-5"
[main cea74ad] "post-test-apl-5"
 1 file changed, 0 insertions(+), 0 deletions(-)
[kelvinalviansyah@MacBook-Air-kelvin praktikum-apl % 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 praktikum-apl % 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 praktikum-apl %
```

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

```
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
kelvinalviansyah@MacBook-Air-kelvin praktikum-apl % git pull origin main --rebas
e
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), 1.65 MiB | 2.37 MiB/s, done.
From https://github.com/kelvinaavas/praktikum-apl
 * branch            main       -> FETCH_HEAD
   6515726..bc24349  main       -> origin/main
Successfully rebased and updated refs/heads/main.
[kelvinalviansyah@MacBook-Air-kelvin praktikum-apl % git push origin main        ]
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 8 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 2.33 KiB | 2.33 MiB/s, done.
Total 9 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To https://github.com/kelvinaavas/praktikum-apl.git
   bc24349..113c8b9  main -> main
kelvinalviansyah@MacBook-Air-kelvin praktikum-apl %
```

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