LAPORAN PRAKTIKUM

POSTTEST 2

ALGORITMA PEMROGRAMAN LANJUT

Disusun oleh:

Dewi Astuti - 2409106007

Kelas A1’24

PROGRAM STUDI INFORMATIKA

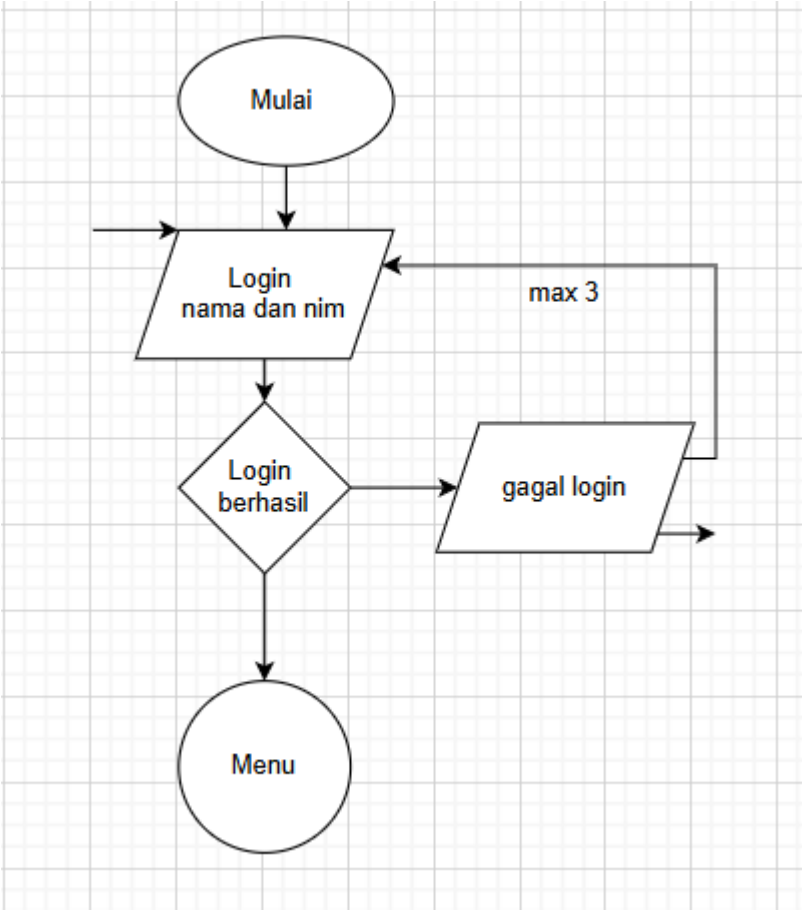
UNIVERSITAS MULAWARMAN

SAMARINDA

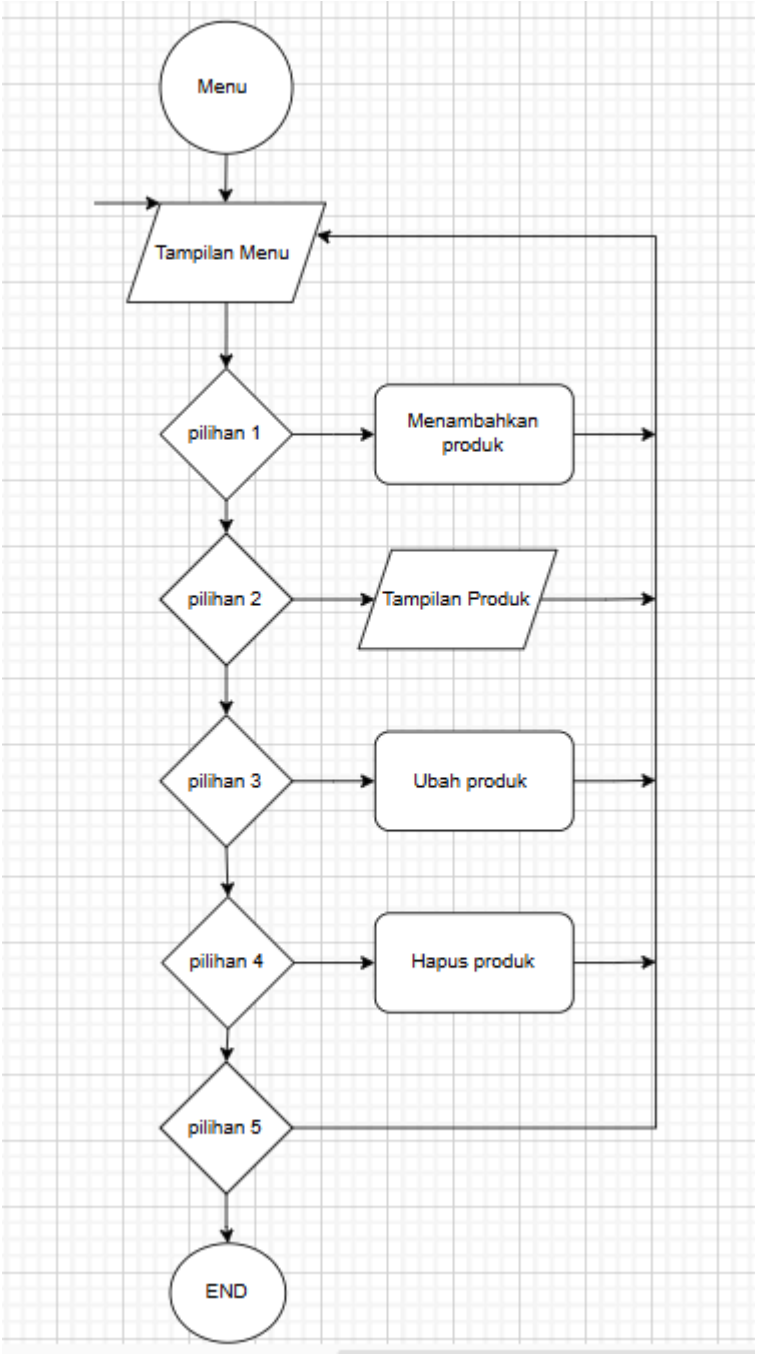
2025

1. Flowchart

A. Login



B. Menu



2. Analisis Program

2.1 Deskripsi Singkat Program

Program CRUD Manajemen Produk Perangkat Jaringan dirancang untuk membantu pengguna dalam mengelola data produk dengan lebih efisien. Program ini menyediakan fitur CRUD (Create, Read, Update, Delete) yang memungkinkan pengguna untuk menambahkan, menampilkan, memperbarui, dan menghapus data produk, termasuk nama, tipe, harga, dan

stok. Selain itu, program dilengkapi dengan sistem login untuk memastikan keamanan akses data.

Dengan adanya program ini, pengguna dapat mencatat dan melihat daftar produk dengan format yang rapi serta melakukan perubahan data sesuai kebutuhan. Program ini juga membantu dalam manajemen stok dan harga, sehingga mempermudah pengelolaan inventaris perangkat jaringan. Cocok digunakan oleh penjual, distributor, atau administrator inventaris yang membutuhkan sistem sederhana, praktis, dan efektif dalam mengatur data produk mereka.

2.2 Penjelasan Alur & Algoritma

1. Program Dimulai

    - Program meminta pengguna untuk login dengan memasukkan username dan password.
    - Jika login benar, pengguna akan masuk ke menu utama.
    - Jika salah, pengguna diberi kesempatan hingga 3 kali percobaan, jika gagal program akan berhenti.
2. Menampilkan Menu Utama
    - Setelah berhasil login, program menampilkan 5 pilihan menu utama:
        1. Tambah Produk
        2. Tampilkan Produk
        3. Update Produk
        4. Hapus Produk
        5. Keluar
    - Program akan terus menampilkan menu ini berulang kali hingga pengguna memilih opsi Keluar.
3. Menampilkan Produk Awal (Data Default)
    - Program menyimpan 3 perangkat jaringan bawaan menggunakan array multidimensi.
    - Saat pengguna memilih menu "Tampilkan Produk", program akan menampilkan data ini dalam format tabel.
4. Menambah Produk (Create)
    - Pengguna memilih menu "Tambah Produk".

- Program meminta input Nama Produk, Tipe, Harga, dan Stok.
- Data yang dimasukkan akan disimpan dalam array satu dimensi (vector).
- Program menampilkan pesan bahwa produk berhasil ditambahkan.

5. Menampilkan Semua Produk (Read)
    - Pengguna memilih menu "Tampilkan Produk".
    - Program menampilkan produk bawaan dari array multidimensi dan produk yang sudah ditambahkan dalam format tabel.
6. Memperbarui Data Produk (Update)
    - Pengguna memilih menu "Update Produk".
    - Program meminta pengguna untuk memasukkan index produk yang ingin diperbarui.
    - Jika index valid, program meminta input data baru untuk mengganti Nama, Tipe, Harga, dan Stok.
    - Jika index tidak valid, program menampilkan pesan error.
7. Menghapus Produk (Delete)
    - Pengguna memilih menu "Hapus Produk".
    - Program meminta pengguna untuk memasukkan index produk yang ingin dihapus.
    - Jika index valid, produk akan dihapus dari daftar.
    - Jika index tidak valid, program menampilkan pesan error.
8. Keluar dari Program (Exit)
    - Jika pengguna memilih "Keluar", program menampilkan pesan perpisahan.
    - Program berhenti berjalan.

🔹 Algoritma Program

1. Mulai
2. Inisialisasi daftar produk bawaan menggunakan array multidimensi.
3. Login
    - Looping maksimal 3 kali untuk meminta username & password.
    - Jika benar, lanjut ke menu utama.
    - Jika salah 3 kali, program berhenti.
4. Menampilkan Menu Utama

- Menampilkan opsi Tambah, Tampilkan, Update, Hapus, Keluar.
- Menggunakan perulangan (looping) agar menu terus muncul hingga pengguna memilih Keluar.

5. Memproses Pilihan Menu
    - Tambah Produk
        - Meminta input Nama, Tipe, Harga, Stok.
        - Menyimpan ke array satu dimensi.
    - Tampilkan Produk
        - Menampilkan produk dari array multidimensi + array satu dimensi.
    - Update Produk
        - Meminta index produk, jika valid minta data baru.
    - Hapus Produk
        - Meminta index produk, jika valid hapus dari daftar.
    - Keluar
        - Menghentikan program.
6. Kembali ke Menu Utama (Jika Belum Keluar)
7. Selesai

3. Source Code

1. Struktur Data Produk

- Produk adalah struktur data yang menyimpan informasi tentang produk.
- Berisi nama produk, tipe produk, harga produk, dan stok produk.

```
struct Produk {
    string nama;
    string tipe;
    double harga;
    int stok;
};
```

2. Data Produk Bawaan (Vector)

- perangkatJaringan adalah vector yang menyimpan produk bawaan.
- Setiap elemen menyimpan informasi produk dalam bentuk struct Produk.
- Data bawaan ini bisa diubah atau dihapus oleh pengguna.

```
vector<Produk> perangkatJaringan = {
    {"TP-LINK 2340", "Router", 450000, 2},
    {"Cisco 2901", "Router", 3500000, 5},
    {"MikroTik hAP AC2", "Access Point", 850000, 10}
};
```

3. Sistem Login

- Meminta pengguna memasukkan username dan password.
- Maksimal 3 kali percobaan sebelum program ditutup.
- Jika login berhasil, program lanjut ke menu utama.

```
string username, nim;
int attempts = 0;
bool loginSukses = false;
```

```
while (attempts < 3) {
    cout << "Username: "; getline(cin, username);
    cout << "Password: "; cin >> nim; cin.ignore();
    if (username == "dewi astuti" && nim == "2409106007") {
        cout << "Login berhasil!\n"; loginSukses = true; break;
    } else {
        cout << "Login gagal! Coba lagi.\n"; attempts++;
    }
}

if (!loginSukses) {
    cout << "Gagal login 3 kali. Program dihentikan.\n"; return 0;
}
```

4. Menu Utama

- Loop tak terbatas (while (true)) agar program terus berjalan sampai pengguna memilih keluar.
- Menampilkan menu utama dengan opsi:
    1. Tambah Produk
    2. Tampilkan Produk
    3. Update Produk
    4. Hapus Produk
    5. Keluar

```
while (true) {
    int pilihan;
    cout << "\nMenu:\n1. Tambah Produk\n2. Tampilkan Produk\n3. Update Produk\n4. Hapus Produk\n5. Keluar\nPilih: ";
    cin >> pilihan;
```

5. Tambah Produk

- Meminta pengguna memasukkan nama, tipe, harga, dan stok produk.
- Data produk baru disimpan dalam produkList.
- Menggunakan getline(cin, produkBaru.nama) agar dapat membaca input dengan spasi.
- push_back(produkBaru) menambahkan produk ke dalam vector.

```
case 1: {
    Produk produkBaru;
    cout << "Nama Produk: "; cin.ignore(); getline(cin, produkBaru.nama);
    cout << "Tipe Produk: "; getline(cin, produkBaru.tipe);
    cout << "Harga Produk (Rp): "; cin >> produkBaru.harga;
    cout << "Stok Produk: "; cin >> produkBaru.stok;
    produkList.push_back(produkBaru);
    cout << "Produk berhasil ditambahkan!\n";
    break;
}
```

6. Tampilkan Produk

- Menampilkan daftar produk dari produk bawaan dan produk yang ditambahkan oleh user.
- Menggunakan setw() untuk format tampilan yang rapi.
- fixed << setprecision(0) memastikan harga tampil tanpa angka desimal.

```
case 2: {
    cout << "\nDaftar Produk:\n";
    cout << left << setw(5) << "No" << setw(20) << "Nama Produk" << setw(20) << "Tipe"
         << setw(15) << "Harga (Rp)" << setw(10) << "Stok\n" << string(70, '-') << endl;

    int index = 0;
    for (const auto& p : perangkatJaringan)
        cout << left << setw(5) << index++ << setw(20) << p.nama << setw(20) << p.tipe
             << setw(15) << fixed << setprecision(0) << p.harga << setw(10) << p.stok <<
endl;
```

```
}
```

7. Update Produk

- Menampilkan daftar produk dengan nomor indeks untuk dipilih.
- Meminta input untuk mengganti nama, tipe, harga, dan stok produk.
- Mengecek validitas indeks sebelum update dilakukan.

```
case 3: {
    cout << "\nDaftar Produk untuk Diperbarui:\n";
    int index = 0;
    for (const auto& p : perangkatJaringan)
        cout << index++ << ". " << p.nama << " (" << p.tipe << ")\n";

    cout << "Masukkan nomor produk yang ingin diupdate: ";
    int pilihIndex;
    cin >> pilihIndex;
    cin.ignore();

    if (pilihIndex >= 0 && pilihIndex < perangkatJaringan.size()) {
        cout << "Masukkan Nama Produk baru: ";
        getline(cin, perangkatJaringan[pilihIndex].nama);
        cout << "Masukkan Tipe Produk baru: ";
        getline(cin, perangkatJaringan[pilihIndex].tipe);
        cout << "Masukkan Harga Produk baru (Rp): ";
        cin >> perangkatJaringan[pilihIndex].harga;
        cout << "Masukkan Stok Produk baru: ";
        cin >> perangkatJaringan[pilihIndex].stok;
        cout << "Produk berhasil diupdate!\n";
    } else {
        cout << "Index tidak valid!\n";
    }
    break;
```

```
}
```

---

8. Hapus Produk

- Meminta nomor produk untuk dihapus.
- Menghapus produk dari daftar menggunakan erase().
- Memastikan indeks valid sebelum menghapus.

```
case 4: {
    cout << "Masukkan nomor produk yang ingin dihapus: ";
    int index;
    cin >> index;

    if (index >= 0 && index < perangkatJaringan.size()) {
        perangkatJaringan.erase(perangkatJaringan.begin() + index);
        cout << "Produk bawaan berhasil dihapus!\n";
    } else {
        cout << "Index tidak valid!\n";
    }
    break;
}
```

9. Keluar dari Program

- Meminta konfirmasi sebelum keluar.
- Menghentikan program jika pengguna memilih y atau Y.

```
case 5: {
    char konfirmasi;
    cout << "Apakah Anda yakin ingin keluar? (y/n): ";
    cin >> konfirmasi;
```

```
    if (konfirmasi == 'y' || konfirmasi == 'Y') {
        cout << "Terima kasih! Program selesai.\n";
        return 0;
    }
    break;
}
```

4. Uji Coba dan Hasil Output

4.1 Uji Coba

1. Skenario 1

   ● Percobaan login pertama input nama dan nim yang salah hinggah 3 kali

   ● Percobaan login benar

   ● Memilih menu 2

   ● pilih menu 5 , pilih Y untuk kelur

2. Skenario 2

   - masukkan inputan login yang benar
   - pilih menu 2
   - lalu pilih menu 1 untuk tambah
   - pilih menu 3 untuk update
   - pilih menu 4 untuk hapus
   - dan pilih menu 5, dan pilih y jika ingin program berhenti dan keluar

4.2 Hasil Output

- Skenario

  ● Percobaan login pertama input nama dan nim yang salah hinggah 3 kali

```
PS D:\Praktikum-apl\post-test\post-test2> cd "d:\Prak
p -o 2409106007_DewiAstuti_Posttest2_apl } ; if ($?)
Username: Dewi astuti
Password: 12
Login gagal! Coba lagi.
Username: dewi as
Password: 12
Login gagal! Coba lagi.
Username: dewi
Password: 1
Login gagal! Coba lagi.
Anda telah gagal login 3 kali. Program dihentikan.
PS D:\Praktikum-apl\post-test\post-test2>
```

● Percobaan login benar

```
PS D:\Praktikum-apl\post-test\pos
p -o 2409106007_DewiAstuti_Postte
Username: Dewi Astuti
Password: 2409106007
Login berhasil!

Menu:
1. Tambah Produk
2. Tampilkan Produk
3. Update Produk
4. Hapus Produk
5. Keluar
Pilih menu: 2
```

● Memilih menu 2

```
Pilih menu: 2

Daftar Produk:
No   Nama Produk         Tipe               Harga (Rp)     Stok
-------------------------------------------------------------------------
0    TP-LINK 2340        Router             450000         2
1    Cisco 2901          Router             3500000        5
2    MikroTik hAP AC2    Access Point       850000         10
```

● pilih menu 5 dan pilih Y untuk kelur

```
Pilih menu: 5
Apakah Anda yakin ingin keluar? (y/n): y
Terima kasih! Program selesai.
PS D:\Praktikum-apl\post-test\post-test2>
```

- Skenario 2
    - masukkan inputan login yang benar

```
PS D:\Praktikum-apl\post-tes
p -o 2409106007_DewiAstuti_P
Username: dewi astuti
Password: 2409106007
Login berhasil!

Menu:
1. Tambah Produk
2. Tampilkan Produk
3. Update Produk
4. Hapus Produk
5. Keluar
Pilih menu:
```

- pilih menu 2

```
Pilih menu: 2

Daftar Produk:
No   Nama Produk         Tipe               Harga (Rp)     Stok
-------------------------------------------------------------------------
0    TP-LINK 2340        Router             450000         2
1    Cisco 2901          Router             3500000        5
2    MikroTik hAP AC2    Access Point       850000         10
```

- lalu pilih menu 1 untuk tambah

```
Pilih menu: 1
Masukkan Nama Produk: Mi wifi4
Masukkan Tipe Produk: router
Masukkan Harga Produk (Rp): 375000
Masukkan Stok Produk: 2
Produk berhasil ditambahkan!
```

- pilih menu 3 untuk update

```
Pilih menu: 3

Daftar Produk untuk Diperbarui:
0. TP-LINK 2340 (Router)
1. Cisco 2901 (Router)
2. MikroTik hAP AC2 (Access Point)
3. Mi wifi4 (router)
Masukkan nomor produk yang ingin diupdate: 3
Masukkan Nama Produk baru: MI WIFI4
Masukkan Tipe Produk baru: router
Masukkan Harga Produk baru (Rp): 375000
Masukkan Stok Produk baru: 3
Produk tambahan berhasil diupdate!
```

- pilih menu 4 untuk hapus

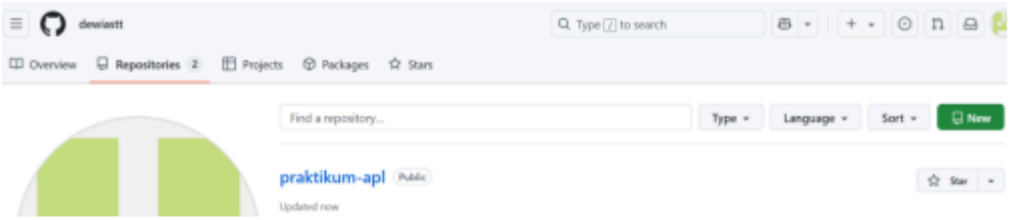
```
Pilih menu: 4
Masukkan nomor produk yang ingin dihapus: 3
Produk tambahan berhasil dihapus!

Menu:
1. Tambah Produk
2. Tampilkan Produk
3. Update Produk
4. Hapus Produk
5. Keluar
Pilih menu: 2

Daftar Produk:
No   Nama Produk         Tipe               Harga (Rp)     Stok
--------------------------------------------------------------------------------
0    TP-LINK 2340        Router             450000         2
1    Cisco 2901          Router             3500000        5
2    MikroTik hAP AC2    Access Point       850000         10
```

- dan pilih menu 5, dan pilih y jika ingin program berhenti dan keluar

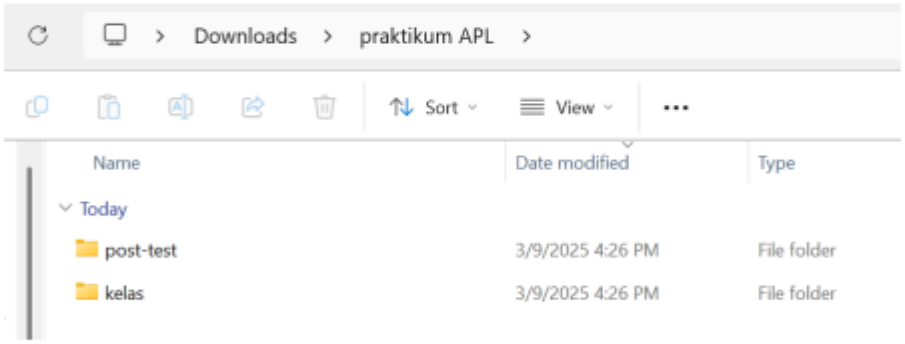
```
Pilih menu: 5
Apakah Anda yakin ingin keluar? (y/n): n

Menu:
1. Tambah Produk
2. Tampilkan Produk
3. Update Produk
4. Hapus Produk
5. Keluar
Pilih menu: 5
Apakah Anda yakin ingin keluar? (y/n): y
Terima kasih! Program selesai.
PS D:\Praktikum-apl\post-test\post-test2>
```

5. Langkah-langkah GIT

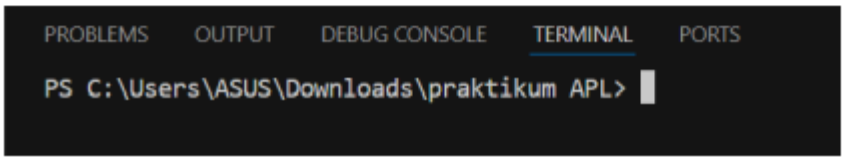
1. Membuat Repository Publik di GitHub



2. Membuat Folder Praktikum di File Explorer



3. Penyesuaian Path



4. Git Init (Inisiasi Repository Git)

```
PS D:\Praktikum-apl> git init
Initialized empty Git repository in D:/Praktikum-apl/.git/
```

5. Git Add (Menambah File yang Ingin Dicommit)

```
PS D:\Praktikum-apl> git add .
```

6. Git Commit (CheckPoint)

```
PS D:\Praktikum-apl> git commit -m "Finish Commit"
[main (root-commit) ff84989] Finish Commit
 8 files changed, 416 insertions(+)
 create mode 100644 kelas/Pertemuan 2/pertemuan2.cpp
 create mode 100644 kelas/Pertemuan-1/Modul-1-APL.pdf
 create mode 100644 post-test/post-test1/2409106007-DewiAstuti-PT-1.cpp
 create mode 100644 post-test/post-test1/2409106007-DewiAstuti-PT-1.exe
 create mode 100644 post-test/post-test1/2409106007-DewiAstuti-PT-1.pdf
 create mode 100644 post-test/post-test2/2409106007-DewiAstuti_Posttest2_apl.pdf
 create mode 100644 post-test/post-test2/2409106007_DewiAstuti_Posttest2_apl.cpp
 create mode 100644 post-test/post-test2/2409106007_DewiAstuti_Posttest2_apl.exe
```

7. Git Remote (Menghubungkan Repository Lokal dengan GitHub)

```
PS D:\Praktikum-apl> git remote add origion https://github.com/dewiastt/praktikum-apl.git
```

8. Git pull origion main –rebase (Mengupdate Semua yang Ada di Repository Lokal)

```
PS D:\Praktikum-apl> git pull origin main --rebase
>>
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (14/14), done.
Unpacking objects:  66% (14/21), 1.71 MiB | 3.41 MiBremote: Total 21
Unpacking objects:  71% (15/21), 1.71 MiB | 3.41 MiBUnpacking object
1 MiB | 3.41 MiBUnpacking objects:  85% (18/21), 1.71 MiB | 3.41 MiB
95% (20/21), 1.71 MiB | 3.41 MiBUnpacking objects: 100% (21/21), 1.7
done.
From https://github.com/dewiastt/praktikum-apl
 * branch            main       -> FETCH_HEAD
 * [new branch]      main       -> origin/main
Successfully rebased and updated refs/heads/main.
Counting objects: 100% (13/13), done.
Delta compression using up to 12 threads
Compressing objects: 100% (9/9), done.
Writing objects:  90% (9/10), 984.00 KiB | 867.00 KiWriting objects:
MiB | 996.00 KiB/s, done.
Total 10 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/dewiastt/praktikum-apl.git
   744275c..2e735d6  main -> main
```