LAPORAN PRAKTIKUM

POSTTEST 2

ALGORITMA PEMROGRAMAN LANJUT

Disusun oleh:

Muhammad Dzaki Rifa'I (2409106056)

Kelas (B1'24)

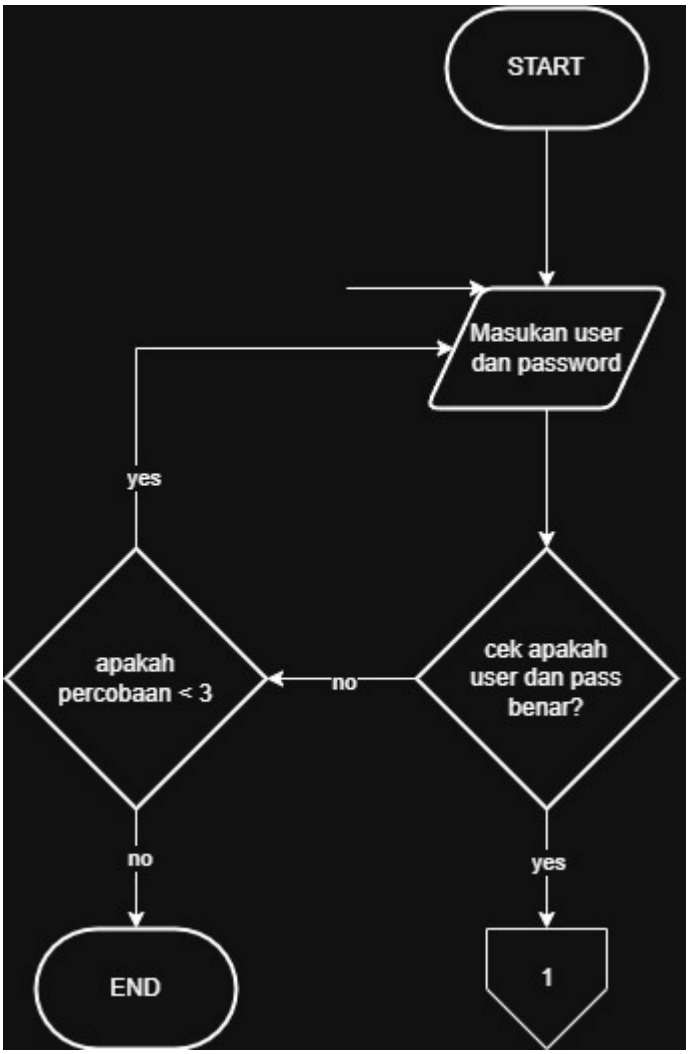
PROGRAM STUDI INFORMATIKA

UNIVERSITAS MULAWARMAN

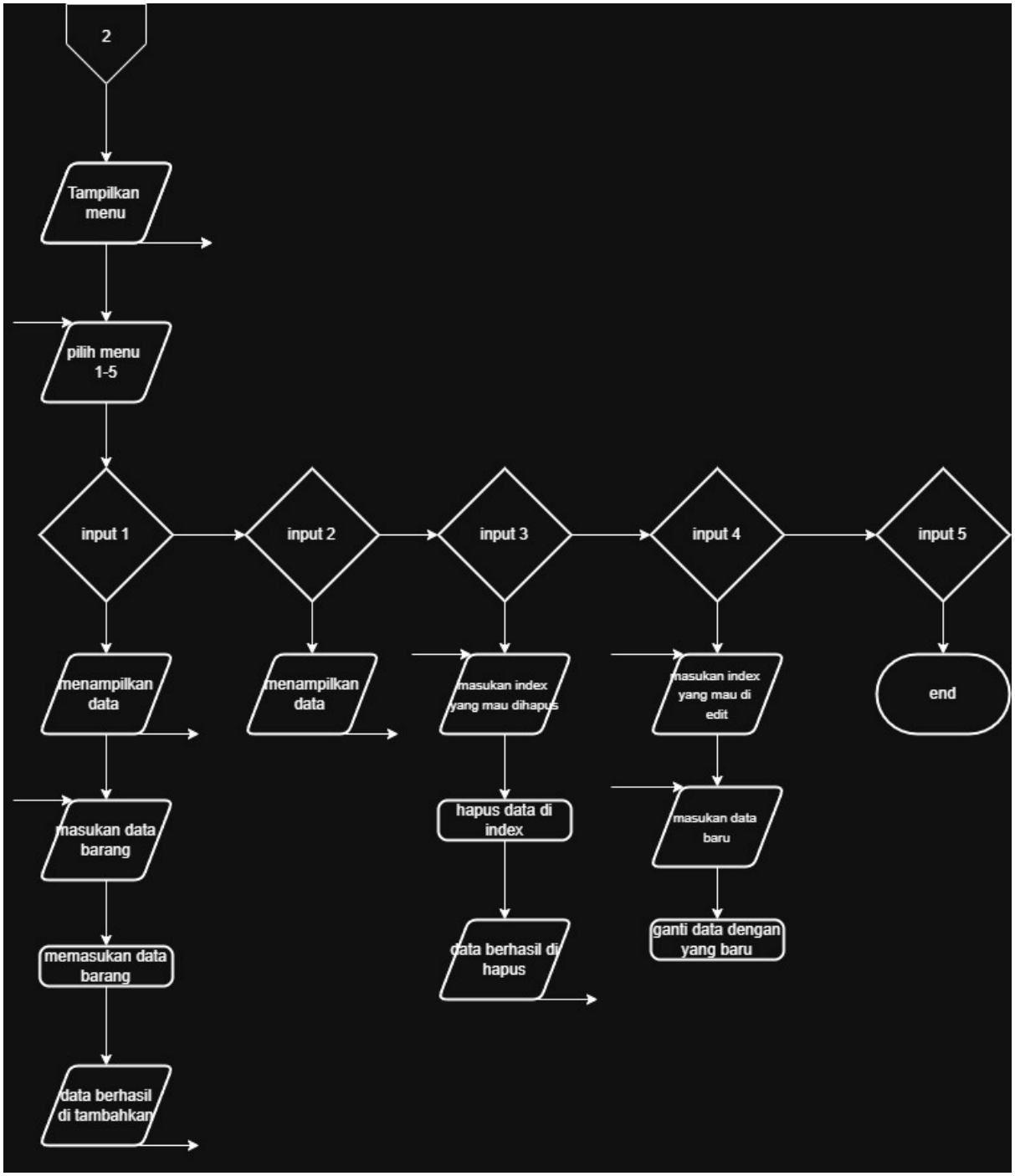
SAMARINDA

2025

1. Flowchart



Gambar 1.1 Flowchart program login



Gambar 1.2 Flowchart menu

2. Analisis Program

2.1. Deklarasi dan Inisialisasi Variabel

- username, password → Menyimpan kredensial login.
- inputUser, inputPass → Menyimpan input login pengguna.
- dataBarang[MAX_DATA][3] → Array 2D untuk menyimpan kode, nama, dan harga barang.
- jumlahData → Menyimpan jumlah barang yang tersedia.
- pilihan → Menyimpan pilihan menu utama.
- index → Digunakan saat memilih data dalam fitur edit atau hapus.
- harga → Menyimpan input harga barang dengan validasi angka.

2.2. Alur Kerja Program

1. Inisialisasi:

- **Program mulai dengan mendeklarasikan array data barang, variabel login, dan jumlah data.**

2. Verifikasi Login:

- **Pengguna diberikan 3 kesempatan untuk memasukkan username dan password.**
- **Jika login berhasil, pengguna masuk ke menu utama.**
- **Jika gagal dalam 3 percobaan, program menampilkan "Terlalu banyak percobaan gagal. Program berhenti." lalu berhenti.**

2.3. Menu Utama

Program menggunakan loop do-while untuk menjaga interaksi hingga pengguna memilih keluar.

- 1. Tambah Barang:
    - Pengguna memasukkan Kode Barang, Nama Barang, dan Harga Barang.
    - Harga harus berupa angka (dilakukan validasi).
    - Data disimpan dalam array dan jumlahData bertambah.
- 2. Tampilkan Barang:
    - Menampilkan tabel barang dengan format rapi.
    - Jika belum ada data, program menampilkan "Data masih kosong!".
- 3. Hapus Barang:
    - Menampilkan tabel barang agar pengguna dapat memilih yang ingin dihapus.
    - Jika nomor barang valid, data digeser untuk menghapusnya tanpa meninggalkan celah kosong.
- 4. Edit Barang:
    - Pengguna memilih barang berdasarkan nomor.
    - Data bisa diperbarui (Kode, Nama, dan Harga) dengan validasi harga tetap dilakukan.
- 5. Keluar:
    - Program berhenti dan menampilkan pesan perpisahan.

2.4. Validasi Input

- Login: Memastikan username dan password benar, serta memberikan batas 3 kali percobaan.
- Harga Barang: Memastikan input hanya angka dengan perulangan do-while.
- Nomor Barang dalam Edit/Hapus: Memastikan nomor yang dipilih valid berdasarkan jumlah data.

2.5. Proses CRUD (Transaksi Data Barang)

- Program memproses input pengguna dalam fitur CRUD sesuai pilihan menu.
- Data dalam array diperbarui sesuai aksi pengguna (tambah, edit, hapus).
- Looping menu terus berjalan hingga pengguna memilih opsi Keluar (5).

2.6. Keluar dari Program

- Jika pengguna memilih 5. Keluar, program menampilkan pesan perpisahan lalu berhenti.

Penjelasan Detail Blok Kode:

- Loop while (Login): Memberi 3 kali percobaan sebelum menutup program.
- Loop do-while (Menu Utama): Memastikan menu terus muncul hingga pengguna memilih keluar.
- Loop for (Tampilkan Data): Menampilkan daftar barang dalam tabel rapi.
- Loop for (Hapus Data): Menggeser data setelah penghapusan agar tidak ada celah kosong.
- if-else (Validasi Input): Memeriksa apakah input harga adalah angka dan apakah nomor barang valid saat edit/hapus.

3. Source Code

A. **Verifikasi user**

Fitur ini digunakan untuk memverifikasi input user dan password yang dimasukan oleh user, jika user salah memasukan password atau user sampai 3 kali maka program akan berhenti.

```
int main() {
    string username = "dzaki";
    string password = "2409106056";
    string inputUser, inputPass;
    int attempts = 0;

    // Proses Login
    while (attempts < 3) {
        system("cls");
        cout << "\n====================================\n";
        cout << "|          LOGIN SISTEM          |\n";
        cout << "====================================\n";
        cout << "| Username: ";
        cin >> inputUser;
        cout << "| Password: ";
        cin >> inputPass;
        cout << "====================================\n";

        if (inputUser == username && inputPass == password) {
            cout << "Login berhasil!\n";
            cin.ignore();
            cin.get();
            break;
        } else {
            cout << "Login gagal, coba lagi!\n";
            attempts++;
            cin.ignore();
            cin.get();
```

```
        }
    }

    if (attempts == 3) {
        cout << "\nTerlalu banyak percobaan gagal. Program berhenti.\n";
        return 0;
    }
```

B. Menu Utama

Menu utama terdiri dari 5 pilihan

1. Tambah barang

```
if (pilihan == 1) {
            system("cls");
            if (jumlahData >= MAX_DATA) {
                cout << "\nData penuh!\n";
                cin.ignore();
                cin.get();
                continue;
            }
            cout << "\nMasukkan Kode Barang: ";
            cin >> dataBarang[jumlahData][0];
            cout << "Masukkan Nama Barang: ";
            cin.ignore();
            getline(cin, dataBarang[jumlahData][1]);

            string harga;
            bool valid;
            do {
                valid = true;
                cout << "Masukkan Harga Barang: ";
                cin >> harga;
                for (char c : harga) {
                    if (c < '0' || c > '9') {
```

```
                    valid = false;
                    break;
                }
            }
            if (!valid) {
                cout << "Harga harus berupa angka! Coba lagi.\n";
            }
        } while (!valid);

        dataBarang[jumlahData][2] = harga;
        jumlahData++;
        cout << "Data berhasil ditambahkan!\n";
        cin.ignore();
        cin.get();
    }
```

2. lihat barang

```
 else if (pilihan == 2) {
        system("cls");
        if (jumlahData == 0) {
            cout << "\nData masih kosong!\n";
        } else {
            cout << "\n-------------------------------------------------------
---------------\n";
            cout << "| No |   Kode   |       Nama
Barang       |      Harga      |\n";
            cout << "-------------------------------------------------------
--------------\n";
            for (int i = 0; i < jumlahData; i++) {
                cout << "| " << setw(2) << i + 1 << " | " << setw(8) <<
dataBarang[i][0] << " | "
                     << setw(20) << dataBarang[i][1] << " | Rp. " << setw(12)
<< dataBarang[i][2] << " |\n";
            }
```

```
                cout << "-------------------------------------------------------
--------------\n";
            }
            cin.ignore();
            cin.get();
        }
```

3. hapus barang dan 4 edit barang berada dalam 1 loop agar menampilkan data tampa menulis ulang kode

```
 else if (pilihan == 3 || pilihan == 4) {
            system("cls");
            if (jumlahData == 0) {
                cout << "\nData masih kosong!\n";
                cin.ignore();
                cin.get();
                continue;
            }

            cout << "\nDaftar Barang:\n";
            cout << "-------------------------------------------------------------
----------\n";
            cout << "| No |   Kode   |       Nama
Barang       |      Harga      |\n";
            cout << "-------------------------------------------------------------
----------\n";
            for (int i = 0; i < jumlahData; i++) {
                cout << "| " << setw(2) << i + 1 << " | " << setw(8) <<
dataBarang[i][0] << " | "
                     << setw(20) << dataBarang[i][1] << " | Rp. " << setw(12) <<
dataBarang[i][2] << " |\n";
            }
            cout << "-------------------------------------------------------------
----------\n";
```

```
            int index;
            cout << "Masukkan nomor data yang ingin diubah: ";
            cin >> index;
            if (index < 1 || index > jumlahData) {
                cout << "Nomor tidak valid!\n";
                cin.ignore();
                cin.get();
                continue;
            }

            if (pilihan == 3) {
                for (int i = index - 1; i < jumlahData - 1; i++) {
                    dataBarang[i][0] = dataBarang[i + 1][0];
                    dataBarang[i][1] = dataBarang[i + 1][1];
                    dataBarang[i][2] = dataBarang[i + 1][2];
                }
                jumlahData--;
                cout << "Data berhasil dihapus!\n";
            } else {
                cout << "Masukkan Kode Barang baru: ";
                cin >> dataBarang[index - 1][0];
                cout << "Masukkan Nama Barang baru: ";
                cin.ignore();
                getline(cin, dataBarang[index - 1][1]);

                string harga;
                bool valid;
                do {
                    valid = true;
                    cout << "Masukkan Harga Barang baru: ";
                    cin >> harga;
                    for (char c : harga) {
                        if (c < '0' || c > '9') {
                            valid = false;
```

```
                        break;
                    }
                }
                if (!valid) {
                    cout << "Harga harus berupa angka! Coba lagi.\n";
                }
            } while (!valid);

            dataBarang[index - 1][2] = harga;
            cout << "Data berhasil diedit!\n";
        }
        cin.ignore();
        cin.get();
    }
```

5.keluar

4. Uji Coba dan Hasil Output

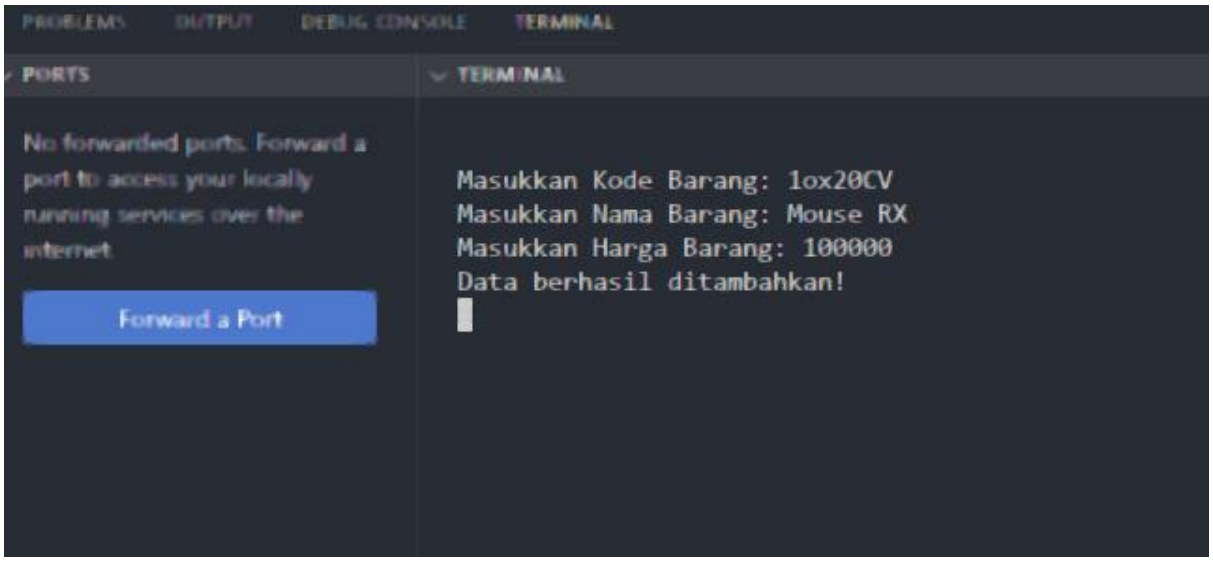
4.1 Uji Coba

1. Skenario 1

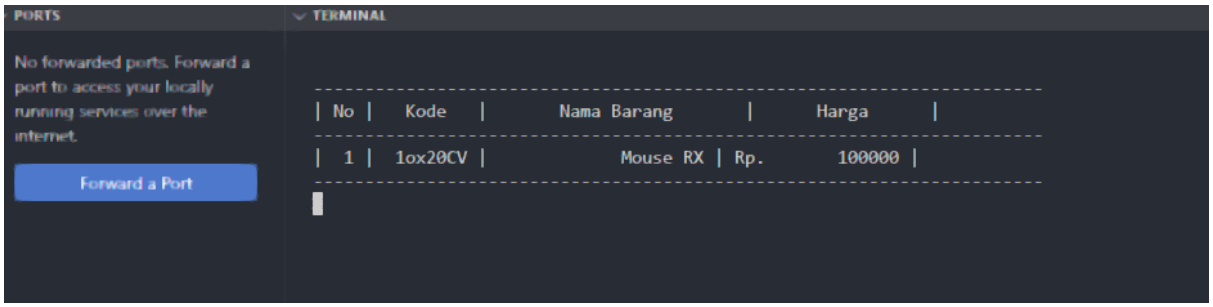
   Uji coba tambah data apakah data berhasil tertambah?
2. Skenario 2

   Uji coba hapus data apakah data berhasil terhapus?

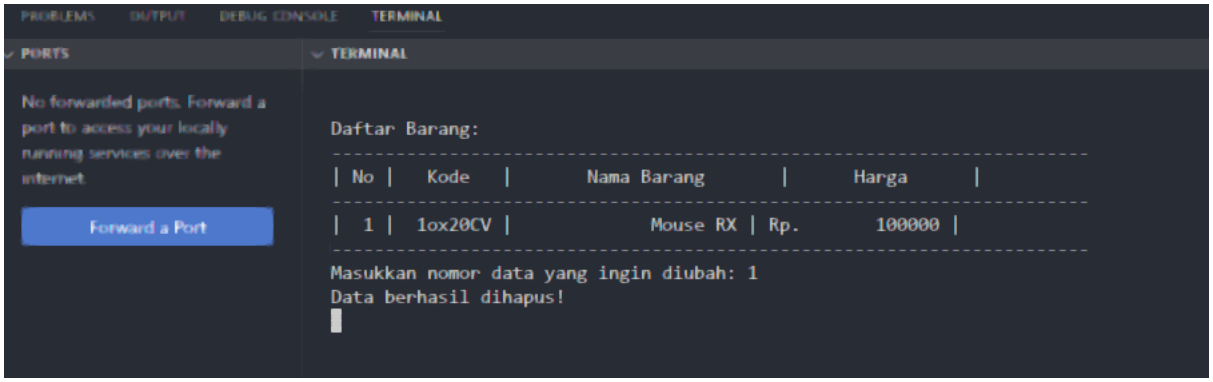
4.2 Hasil Output



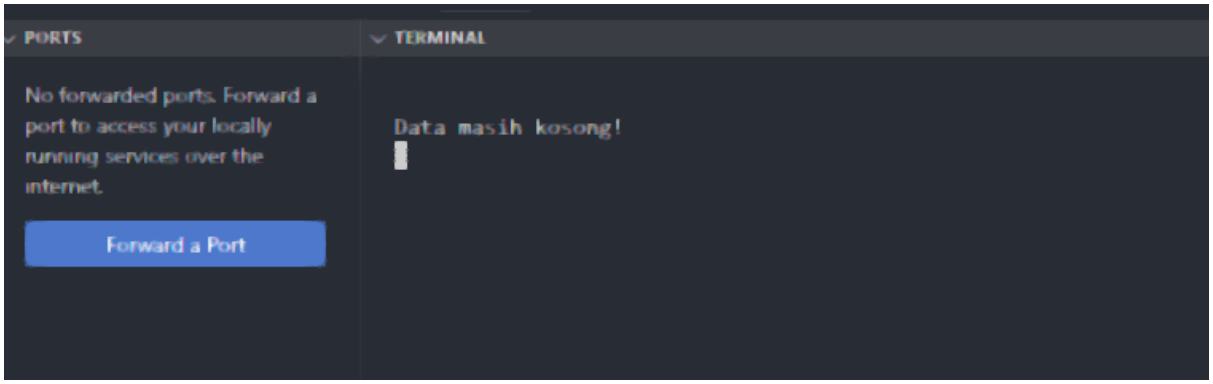
Gambar 4.1 contoh input



Gambar 4.2 tampilan data



Gambar 4.3 mencoba menghapus data



Gambar 4.4 data berhasil dihapus

5. Git

5.1 Git Add

```
ASUS@Dzakzak MINGW64 ~/Praktikum-APL/Post-test/Post-test-APL-1 (main)
$ git add Posttest1.cpp
```

Git add berfungsi untuk Memasukkan file ke staging area agar siap untuk commit.

5.2 Git Commit

```
ASUS@Dzakzak MINGW64 ~/Praktikum-APL/Post-test/Post-test-APL-1 (main)
$ git commit -m "login done"
[main b75faf2] login done
 1 file changed, 29 insertions(+)
 create mode 100644 Post-test/Post-test-APL-1/Posttest1.cpp
```

Git commit berfungsi menyimpan perubahan di repository lokal.
Pesan di -m "..." menjelaskan perubahan yang dilakukan.

5.3 Git push origin main

```
ASUS@Dzakzak MINGW64 ~/Praktikum-APL (main)
$ git push origin main
info: please complete authentication in your browser...
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 20 threads
Compressing objects: 100% (14/14), done.
Writing objects: 100% (23/23), 1.01 MiB | 448.00 KiB/s, done.
Total 23 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/dzak2323/Praktikum-APL.git
 * [new branch]      main -> main
```

Git push origin main Mengunggah commit yang ada di branch main ke repository GitHub.
Jika branch belum ada di GitHub, jalankan git push -u origin main agar branch main menjadi default untuk push berikutnya.