

LAPORAN PRAKTIKUM
POSTTEST 6
ALGORITMA PEMROGRAMAN LANJUT



Disusun oleh:

Nama: Riduan Ali

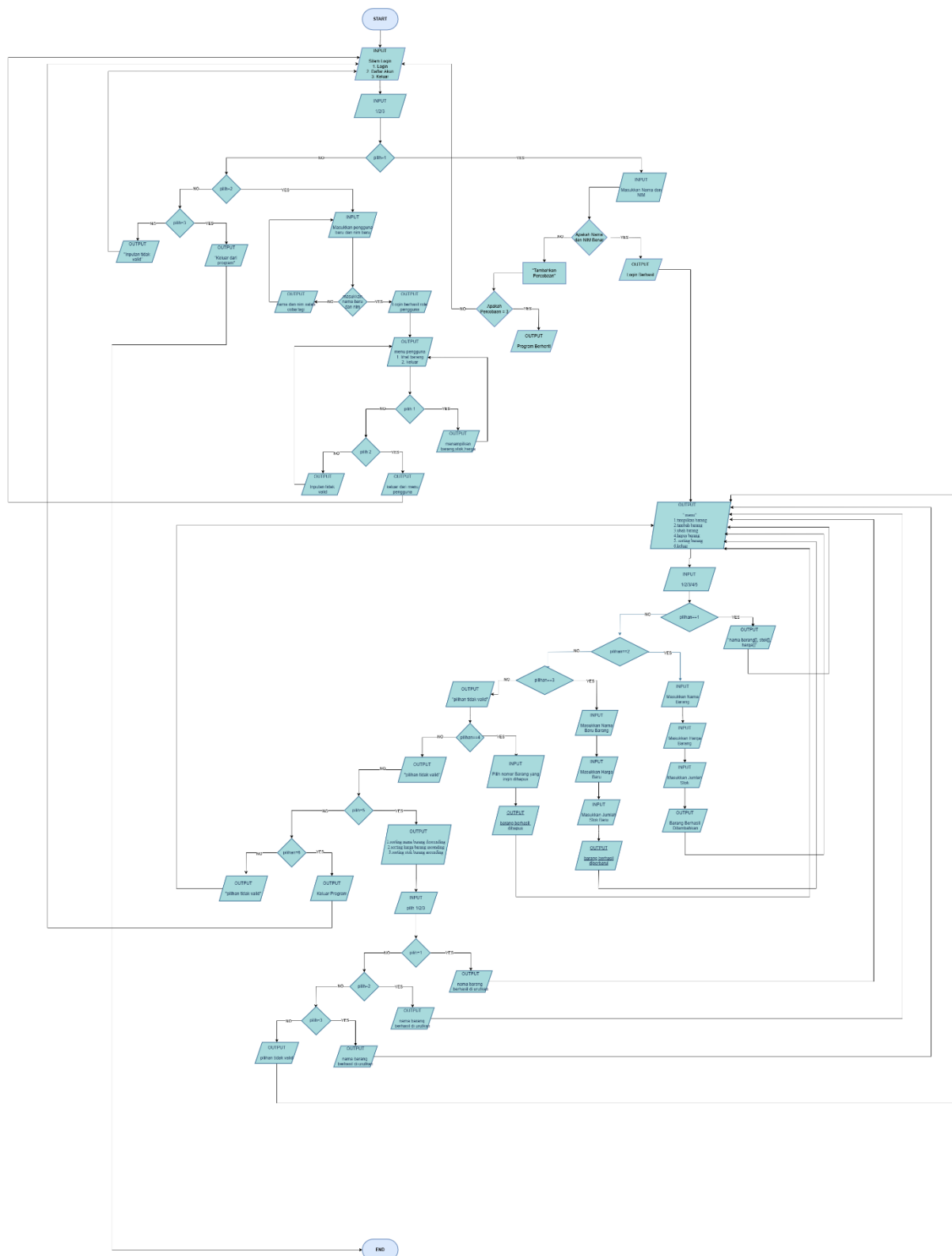
Nim: 2409106114

Kelas C2 '24

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA

2025

1. Flowchart



Gambar 1.1

2. Analisis Program

Tujuan Program

Program ini dibuat untuk mensimulasikan sistem manajemen barang dengan fitur login multiuser, di mana terdapat dua jenis pengguna:

1. Admin (Riduan Ali) → Memiliki akses penuh untuk mengelola barang .
2. Pengguna Biasa → Hanya dapat melihat daftar barang yang tersedia.

Program Utama

1. Keamanan Data
 - a. Sistem login dengan validasi NIM untuk membedakan admin dan pengguna biasa.
 - b. Jika gagal login 3 kali, program akan berhenti untuk menghindari penyalahgunaan.
2. Manajemen Barang
 - a. Admin dapat menambahkan, mengubah, dan menghapus barang, sehingga mempermudah pengelolaan data barang.
 - b. Barang tersimpan dalam array, yang memungkinkan manipulasi data secara efisien.
3. Multiuser dengan Pendaftaran Akun
 - a. Pengguna baru bisa mendaftarkan akun sebelum login.
 - b. Setelah login, pengguna hanya bisa melihat daftar barang dan tidak bisa melakukan perubahan.
4. Looping hingga Pengguna Memilih Keluar
 - a. Program berjalan terus hingga pengguna memilih keluar dari menu utama atau menu admin/pengguna.
5. Penggunaan Pointer
 1. Program telah menerapkan prinsip pointer sesuai modul:
 - a. Fungsi login() dan daftarAkun() menggunakan parameter pointer.
 - b. Operasi dengan address-of operator (&) dan dereference operator (*) digunakan untuk memodifikasi data asli.
 2. Hal ini memungkinkan efisiensi memori dan mendukung prinsip pass by reference.

3. Source Code

1. Memvalidasi login untuk membedakan admin dan pengguna biasa.

```
while (percobaan < 3) {
    cout << "Masukkan nama: ";
    cin.ignore();
    getline(cin, nama);
    cout << "Masukkan NIM: ";
    getline(cin, nim);

    for (int i = 0; i < jumlahPengguna; i++) {
        if (nama == namaPengguna[i] && nim == nimPengguna[i]) {
            cout << "Login berhasil!" << endl;
            *loginBerhasil = 1;
            if (nim.back() % 2 == 0) {
                *isAdmin = 1;
            } else {
                *isAdmin = 0;
            }
            return;
        }
    }
    percobaan++;
    cout << "Login gagal! Sisa percobaan: " << (3 - percobaan) << endl;
}
```

Gambar 3.1

2. Ini adalah daftar menu atau menu admin di program saya yaitu pembelian lampu projie

```
void menuAdmin(string Barang[][2], int stok[], int &jumlahBarang, int isAdmin) {
    int pilihan;
    do {
        cout << "\n===== MENU ADMIN =====" << endl;
        cout << "1. Lihat Barang" << endl;
        cout << "2. Tambah Barang" << endl;
        cout << "3. Ubah Barang" << endl;
        cout << "4. Hapus Barang" << endl;
        cout << "5. Sorting Barang" << endl;
        cout << "6. Keluar" << endl;
        cout << "Pilih menu: ";
        cin >> pilihan;
        cin.ignore();
    }
```

Gambar 3.2

3. Ini adalah program cara menampilkan daftar barang yang tersedia.

```
void lihatBarang(string Barang[][2], int stok[], int jumlahBarang) {
    cout << "\n===== DAFTAR BARANG =====" << endl;
    for (int i = 0; i < jumlahBarang; i++) {
        cout << i + 1 << ". " << Barang[i][0] << " | Stok: " << stok[i] << " | Harga: " << Barang[i][1] << endl;
    }
}
```

Gambar 3.3

4. Ini adalah program yang memungkinkan pengguna menambahkan barang.

```
} else if (pilihan == 2) {  
    cout << "Masukkan nama Barang: ";  
    getline(cin, Barang[jumlahBarang][0]);  
    cout << "Masukkan harga Barang: ";  
    getline(cin, Barang[jumlahBarang][1]);  
    cout << "Masukkan jumlah stok: ";  
    cin >> stok[jumlahBarang];  
    cin.ignore();  
    jumlahBarang++;  
    cout << "Barang berhasil ditambahkan!" << endl;
```

Gambar 3.4

5. Ini adalah program yang memungkinkan pengguna memperbarui data atau memperbarui barang.

```
} else if (pilihan == 3) {  
    cout << "Pilih nomor Barang yang ingin diubah: ";  
    int index;  
    cin >> index;  
    cin.ignore();  
    if (index > 0 && index <= jumlahBarang) {  
        cout << "Masukkan nama baru Barang: ";  
        getline(cin, Barang[index - 1][0]);  
        cout << "Masukkan harga baru: ";  
        getline(cin, Barang[index - 1][1]);  
        cout << "Masukkan jumlah stok baru: ";  
        cin >> stok[index - 1];  
        cin.ignore();  
        cout << "Barang berhasil diperbarui!" << endl;  
    } else {  
        cout << "Nomor Barang tidak valid!" << endl;  
    }  
}
```

Gambar 3.5

6. Ini adalah program yang memungkinkan pengguna menghapus barang dari daftar.

```
    } else if (pilihan == 4) {
        cout << "Pilih nomor Barang yang ingin dihapus: ";
        int index;
        cin >> index;
        cin.ignore();
        if (index > 0 && index <= jumlahBarang) {
            for (int i = index - 1; i < jumlahBarang - 1; i++) {
                Barang[i][0] = Barang[i + 1][0];
                Barang[i][1] = Barang[i + 1][1];
                stok[i] = stok[i + 1];
            }
            jumlahBarang--;
            cout << "Barang berhasil dihapus!" << endl;
        } else {
            cout << "Nomor Barang tidak valid!" << endl;
        }
    }
```

Gambar 3.6

7. Ini adalah program yang memungkinkan pengguna baru bisa mendaftar sebelum login.

```
void daftarAkun(string* namaPengguna, string* nimPengguna, int* jumlahPengguna) {
    cin.ignore();
    cout << "Masukkan Nama: ";
    getline(cin, namaPengguna[*jumlahPengguna]);
    cout << "Masukkan NIM: ";
    getline(cin, nimPengguna[*jumlahPengguna]);
    (*jumlahPengguna)++;
    cout << "Akun berhasil didaftarkan! Silakan login." << endl;
}
```

Gambar 3.7

8. Ini adalah daftar menu sorting yang mau ingin di urutkan

```
    else if (pilihan == 5) {
        if (isAdmin) {
            int sortingChoice;
            cout << "\n===== MENU SORTING =====" << endl;
            cout << "1. Sorting Nama Barang (Descending)" << endl;
            cout << "2. Sorting Harga Barang (Ascending)" << endl;
            cout << "3. Sorting Stok Barang (Ascending)" << endl;
            cout << "Pilih metode sorting: ";
            cin >> sortingChoice;
```

Gambar 3.8

9. Fungsi ini akan menjalankan proses pengurutan sorting

```
if (sortingChoice == 1) {
    sortNamaBarangDesc(Barang, stok, jumlahBarang);
    cout << "Nama barang berhasil diurutkan (Descending)." << endl;
} else if (sortingChoice == 2) {
    sortHargaBarangAsc(Barang, stok, jumlahBarang);
    cout << "Harga barang berhasil diurutkan (Ascending)." << endl;
} else if (sortingChoice == 3) {
    sortStokAsc(Barang, stok, jumlahBarang);
    cout << "Stok barang berhasil diurutkan (Ascending)." << endl;
} else {
    cout << "Pilihan tidak valid!" << endl;
}
```

Gambar 3.9

10. Fungsi ini mengurutkan daftar nama barang secara menurun dari Z ke A dan ikut menukar harga serta stok agar tetap sesuai dengan nama barangnya

```
void sortNamaBarangDesc(string Barang[][2], int stok[], int jumlahBarang) {
    for (int i = 0; i < jumlahBarang - 1; i++) {
        for (int j = i + 1; j < jumlahBarang; j++) {
            if (Barang[i][0] < Barang[j][0]) {
                tukar(Barang[i][0], Barang[j][0]);
                tukar(Barang[i][1], Barang[j][1]);
                tukar(stok[i], stok[j]);
            }
        }
    }
}
```

11. Fungsi ini mengurutkan harga barang secara menaik dari harga paling murah ke mahal dan stok serta nama barang ikut disesuaikan dengan urutan harga.

```
void sortHargaBarangAsc(string Barang[][2], int stok[], int jumlahBarang) {
    for (int i = 0; i < jumlahBarang - 1; i++) {
        for (int j = i + 1; j < jumlahBarang; j++) {
            if (stoi(Barang[i][1]) > stoi(Barang[j][1])) {
                tukar(Barang[i][0], Barang[j][0]);
                tukar(Barang[i][1], Barang[j][1]);
                tukar(stok[i], stok[j]);
            }
        }
    }
}
```

12. Fungsi ini mengurutkan barang berdasarkan jumlah stok secara menaik stok sedikit ke banyak dengan tetap menjaga urutan nama dan harga agar sesuai.

```
void sortStokAsc(string Barang[][2], int stok[], int jumlahBarang) {  
    for (int i = 0; i < jumlahBarang - 1; i++) {  
        for (int j = i + 1; j < jumlahBarang; j++) {  
            if (stok[i] > stok[j]) {  
                tukar(Barang[i][0], Barang[j][0]);  
                tukar(Barang[i][1], Barang[j][1]);  
                tukar(stok[i], stok[j]);  
            }  
        }  
    }  
}
```


4. Uji Coba dan Hasil Output

4.1 Uji Coba

1. Uji coba login: Jika saya memasukkan nama dan nim yang salah dalam 3 kali percobaan, misal yaitu:

- a. Nama: IlhamGod
- b. Nim: 11114

Ini nama dan nim yang benar

- a. Username: Riduan Ali
- b. Password: 114

- 2. **Mencoba Lihat Barang:**
- 3. **Mencoba Tambah Barang:**
- 4. **Mencobah Ubah Barang:**
- 5. **Mencoba Hapus Barang:**
- 6. **Mencoba Sorting Baranng:**
- 7. **Daftar akun**
- 8. **Mencoba masuk di menu pengguna**

4.2 Hasil Output

Jika nama dan nim salah.

```
Masukkan Nama: IlhamGod
Masukkan NIM: 11114
Nama yang dimasukkan: 'IlhamGod'
NIM yang dimasukkan: '11114'
Nama atau NIM salah! Percobaan ke-1 dari 3 percobaan.
```

Gambar 4.1

Jika nama dan nim salah dan sudah 3 kali percobaan.

```
Masukkan Nama: IlhamGod
Masukkan NIM: 11114
Nama yang dimasukkan: 'IlhamGod'
NIM yang dimasukkan: '11114'
Nama atau NIM salah! Percobaan ke-3 dari 3 percobaan.
Anda sudah mencoba 3 kali. Program berhenti.
PS C:\Users\Pongo\OneDrive\Documents\Praktikum APL C++\posttest-2>
```

Gambar 4.2

Jika pengguna input opsi 1 atau pilih 1 yaitu Lihat Barang.

```

===== MENU ADMIN =====
1. Lihat Barang
2. Tambah Barang
3. Ubah Barang
4. Hapus Barang
5. Keluar
Pilih menu: 1

===== DAFTAR BARANG =====
1. Lampu Projie Biru | Stok: 25 | Harga: 15000000
2. Lampu Projie Ungu | Stok: 20 | Harga: 25000000
3. Lampu Projie Putih | Stok: 15 | Harga: 20000000
4. Lampu LED Kuning | Stok: 30 | Harga: 10000000
5. Lampu LED RGB | Stok: 10 | Harga: 18000000

```

Gambar 4.3

Jika pengguna memilih opsi 2 yaitu Tambah Barang.

```

===== MENU ADMIN =====
1. Lihat Barang
2. Tambah Barang
3. Ubah Barang
4. Hapus Barang
5. Keluar
Pilih menu: 2
Masukkan nama Barang: Lampu Projie Neon
Masukkan harga Barang: 25000000
Masukkan jumlah stok: 25
Barang berhasil ditambahkan!

```

Gambar 4.4

Jika pengguna sudah menambahkan barang, dan opsi 1 nya akan seperti ini.

```

===== DAFTAR BARANG =====
1. Lampu Projie Biru | Stok: 25 | Harga: 15000000
2. Lampu Projie Ungu | Stok: 20 | Harga: 25000000
3. Lampu Projie Putih | Stok: 15 | Harga: 20000000
4. Lampu LED Kuning | Stok: 30 | Harga: 10000000
5. Lampu LED RGB | Stok: 10 | Harga: 18000000
6. Lampu Projie Neon | Stok: 25 | Harga: 25000000

```

Gambar 4.5

Jika pengguna memilih opsi 3 yaitu Ubah Barang.

```

===== MENU ADMIN =====
1. Lihat Barang
2. Tambah Barang
3. Ubah Barang
4. Hapus Barang
5. Keluar
Pilih menu: 3
Pilih nomor Barang yang ingin diubah: 6
Masukkan nama baru Barang: Lampu Projie Merah
Masukkan harga baru: 25000000
Masukkan jumlah stok baru: 25
Barang berhasil diperbarui!

```

Gambar 4.6

Jika pengguna sudah mengubah barang, dan opsi 1 nya akan seperti ini.

```
===== DAFTAR BARANG =====
1. Lampu Projie Biru | Stok: 25 | Harga: 15000000
2. Lampu Projie Ungu | Stok: 20 | Harga: 25000000
3. Lampu Projie Putih | Stok: 15 | Harga: 20000000
4. Lampu LED Kuning | Stok: 30 | Harga: 10000000
5. Lampu LED RGB | Stok: 10 | Harga: 18000000
6. Lampu Projie Merah | Stok: 25 | Harga: 25000000
```

Gambar 4.7

Jika pengguna memilih opsi 4 yaitu Hapus Barang.

```
===== MENU ADMIN =====
1. Lihat Barang
2. Tambah Barang
3. Ubah Barang
4. Hapus Barang
5. Keluar
Pilih menu: 4
Pilih nomor Barang yang ingin dihapus: 6
Barang berhasil dihapus!
```

Gambar 4.8

Jika pengguna sudah menghapus barang, dan opsi 1 nya akan seperti ini.

```
===== DAFTAR BARANG =====
1. Lampu Projie Biru | Stok: 25 | Harga: 15000000
2. Lampu Projie Ungu | Stok: 20 | Harga: 25000000
3. Lampu Projie Putih | Stok: 15 | Harga: 20000000
4. Lampu LED Kuning | Stok: 30 | Harga: 10000000
5. Lampu LED RGB | Stok: 10 | Harga: 18000000
```

Gambar 4.9

Jika pengguna memilih opsi 5 yaitu Sorting Barang

```
Pilih menu: 5

===== MENU SORTING =====
1. Sorting Nama Barang (Descending)
2. Sorting Harga Barang (Ascending)
3. Sorting Stok Barang (Ascending)
Pilih metode sorting: █
```

Jika opsi 1 dan lihat barang

```
===== MENU SORTING =====
1. Sorting Nama Barang (Descending)
2. Sorting Harga Barang (Ascending)
3. Sorting Stok Barang (Ascending)
Pilih metode sorting: 1
Nama barang berhasil diurutkan (Descending).

===== MENU ADMIN =====
1. Lihat Barang
2. Tambah Barang
3. Ubah Barang
4. Hapus Barang
5. Sorting Barang
6. Keluar
Pilih menu: 1

===== DAFTAR BARANG =====
1. Lampu Projie Ungu | Stok: 20 | Harga: 25000000
2. Lampu Projie Putih | Stok: 15 | Harga: 20000000
3. Lampu Projie Biru | Stok: 25 | Harga: 15000000
4. Lampu LED RGB | Stok: 10 | Harga: 18000000
5. Lampu LED Kuning | Stok: 30 | Harga: 10000000
```

Jika opsi 2 dan lihat barang

```
===== MENU SORTING =====
1. Sorting Nama Barang (Descending)
2. Sorting Harga Barang (Ascending)
3. Sorting Stok Barang (Ascending)
Pilih metode sorting: 2
Harga barang berhasil diurutkan (Ascending).

===== MENU ADMIN =====
1. Lihat Barang
2. Tambah Barang
3. Ubah Barang
4. Hapus Barang
5. Sorting Barang
6. Keluar
Pilih menu: 1

===== DAFTAR BARANG =====
1. Lampu LED Kuning | Stok: 30 | Harga: 10000000
2. Lampu Projie Biru | Stok: 25 | Harga: 15000000
3. Lampu LED RGB | Stok: 10 | Harga: 18000000
4. Lampu Projie Putih | Stok: 15 | Harga: 20000000
5. Lampu Projie Ungu | Stok: 20 | Harga: 25000000
```

Jika opsi 3 dan lihat barang

```
===== MENU SORTING =====
1. Sorting Nama Barang (Descending)
2. Sorting Harga Barang (Ascending)
3. Sorting Stok Barang (Ascending)
Pilih metode sorting: 3
Stok barang berhasil diurutkan (Ascending).

===== MENU ADMIN =====
1. Lihat Barang
2. Tambah Barang
3. Ubah Barang
4. Hapus Barang
5. Sorting Barang
6. Keluar
Pilih menu: 1

===== DAFTAR BARANG =====
1. Lampu LED RGB | Stok: 10 | Harga: 18000000
2. Lampu Projie Putih | Stok: 15 | Harga: 20000000
3. Lampu Projie Ungu | Stok: 20 | Harga: 25000000
4. Lampu Projie Biru | Stok: 25 | Harga: 15000000
5. Lampu LED Kuning | Stok: 30 | Harga: 10000000
```

Menu utama dan menu daftar akun

```
===== MENU UTAMA =====
1. Login
2. Daftar Akun
3. Keluar
Pilih menu: 2
Masukkan Nama: IlhamGod
Masukkan NIM: 123
Akun berhasil didaftarkan! Silakan login.
```

Terus login ke menu pengguna lagi menggunakan nama yang sudah di daftarkan

```
===== MENU UTAMA =====
1. Login
2. Daftar Akun
3. Keluar
Pilih menu: 1
Masukkan Nama: IlhamGod
Masukkan NIM: 123

===== MENU PENGGUNA =====
1. Lihat Barang
2. Keluar
```

Lihat barang di menu pengguna

```
===== MENU PENGGUNA =====  
1. Lihat Barang  
2. Keluar  
Pilih menu: 1  
  
===== DAFTAR BARANG =====  
1. Lampu Projie Biru | Stok: 25 | Harga: 15000000  
2. Lampu Projie Ungu | Stok: 20 | Harga: 25000000  
3. Lampu Projie Putih | Stok: 15 | Harga: 20000000  
4. Lampu LED Kuning | Stok: 30 | Harga: 10000000  
5. Lampu LED RGB | Stok: 10 | Harga: 18000000
```

5. Sistem Git

1. Langkah-Langkah Git

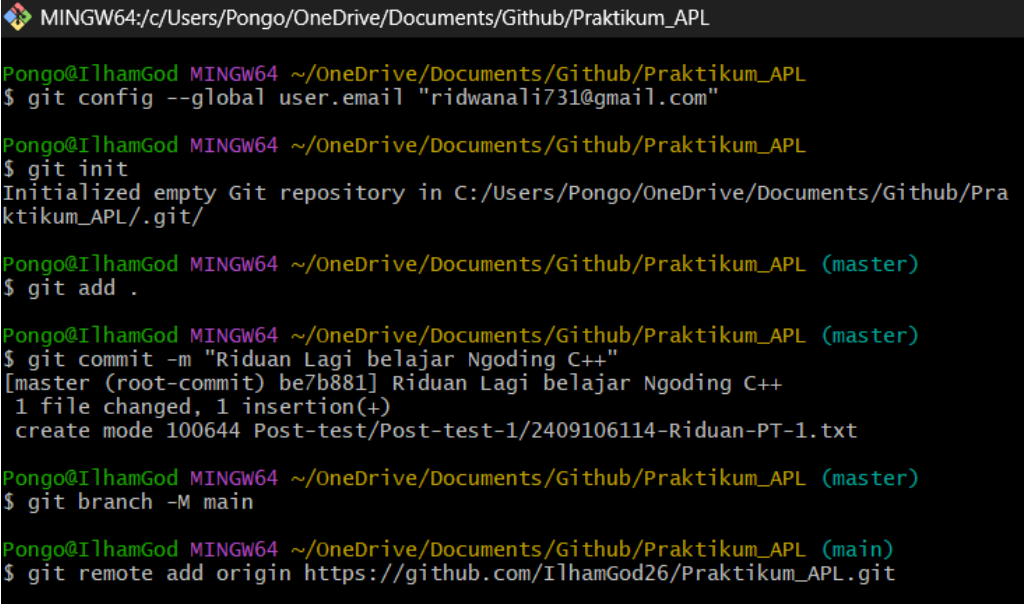
Langkah 1: Download Git nya dulu di website resminya Git.

Langkah 2: Kalian installasi Git nya.

Langkah 3: Kalian buka Git nya dan kalian ketik ini “git config –global user.email “ini email kalian yah yang terhubung di GitHub”” trus enter.

Langkah 4: ketik ini Git add untuk menambah file yang akan di komit dan Git comit –m “Update” untuk membuat check point.

Langkah 5: Terus kalian buka GitHub biasanya kalau sudah manambahkan file atau folder yang ada di langkah 4 itu jika sudah terhubung biasanya ada bacaanya tuh jika sudah nemu yang bacaan “git remote add origin gitu lah” seperti ini.



```
MINGW64:/c/Users/Pongo/OneDrive/Documents/Github/Praktikum_APL
Pongo@IlhamGod MINGW64 ~/OneDrive/Documents/Github/Praktikum_APL
$ git config --global user.email "ridwanali731@gmail.com"

Pongo@IlhamGod MINGW64 ~/OneDrive/Documents/Github/Praktikum_APL
$ git init
Initialized empty Git repository in C:/Users/Pongo/OneDrive/Documents/Github/Praktikum_APL/.git/

Pongo@IlhamGod MINGW64 ~/OneDrive/Documents/Github/Praktikum_APL (master)
$ git add .

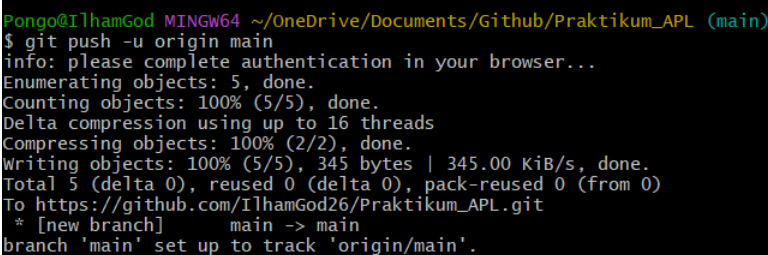
Pongo@IlhamGod MINGW64 ~/OneDrive/Documents/Github/Praktikum_APL (master)
$ git commit -m "Riduan Lagi belajar Ngoding C++"
[master (root-commit) be7b881] Riduan Lagi belajar Ngoding C++
1 file changed, 1 insertion(+)
create mode 100644 Post-test/Post-test-1/2409106114-Riduan-PT-1.txt

Pongo@IlhamGod MINGW64 ~/OneDrive/Documents/Github/Praktikum_APL (master)
$ git branch -M main

Pongo@IlhamGod MINGW64 ~/OneDrive/Documents/Github/Praktikum_APL (main)
$ git remote add origin https://github.com/IlhamGod26/Praktikum_APL.git
```

Gambar 5.5

Langkah 6: terus ketik “git push -u origin main” seperti ini



```
Pongo@IlhamGod MINGW64 ~/OneDrive/Documents/Github/Praktikum_APL (main)
$ git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (5/5), 345 bytes | 345.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/IlhamGod26/Praktikum_APL.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Gambar 5.6