

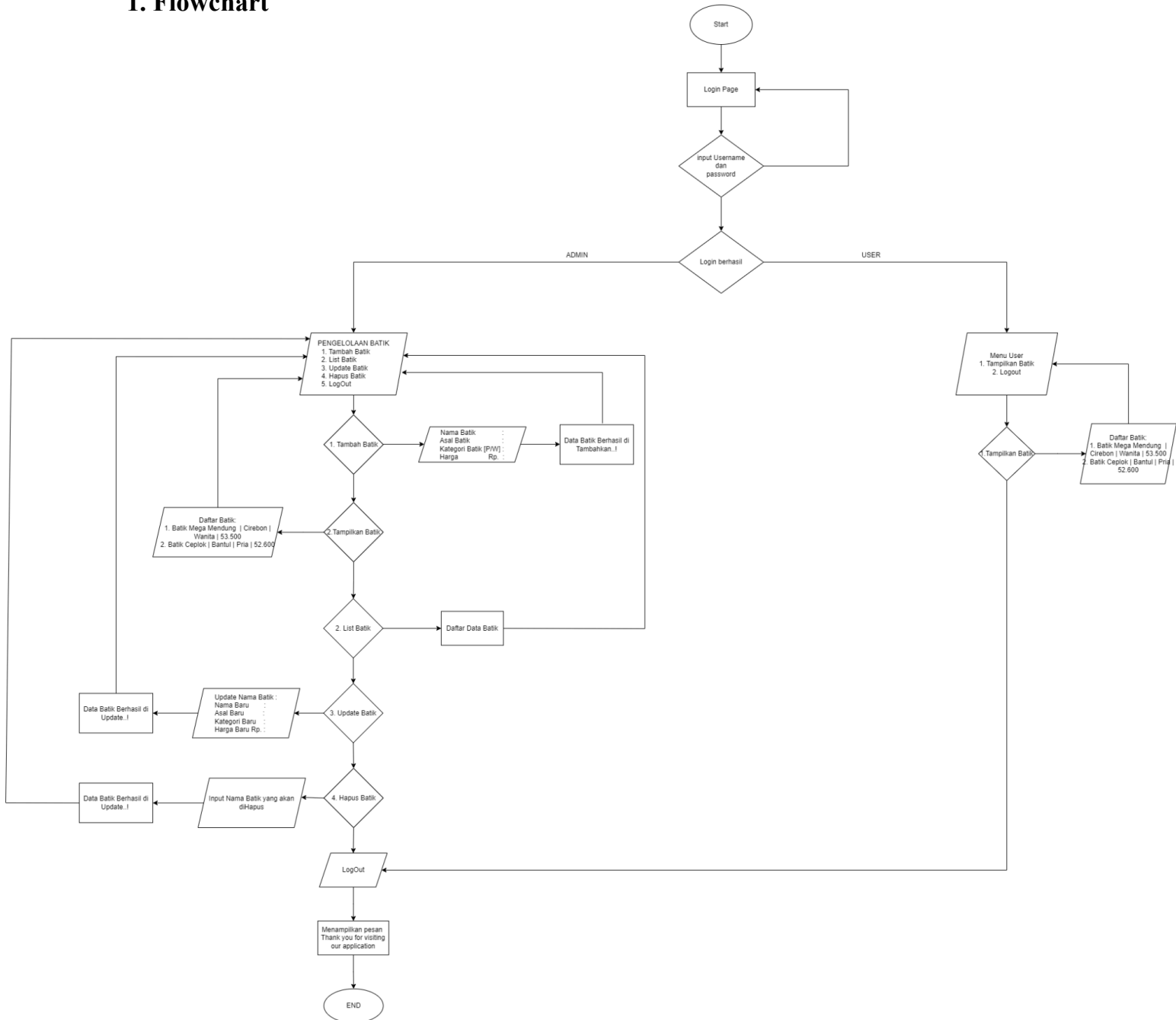
LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN LANJUT



Disusun oleh:
Wina Okta Ramadhani (2409106100)
Kelas C'24

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



2. Analisis Program

2.1 Deskripsi Singkat Program

Program ini dibuat menggunakan bahasa pemrograman C++ tujuan dari Program ini untuk mengelola dan menampilkan data batik dalam sebuah toko yang saya beri nama “Toko Batik Nusantara”. di program ini memiliki dua User yaitu Admin (Wina Okta Ramadhani) dan Pengguna sebagai sample (Syalomitha Novindriani Depe).

2.2 Penjelasan Alur & Algoritma

Alur Program

```
#include <iostream>
#include <string>
#include <iomanip>
#include <limits>

using namespace std;
```

#include <iostream> → Digunakan untuk input-output standar (**cin**, **cout**).

#include <string> → Memungkinkan penggunaan tipe data string.

#include <iomanip> → Digunakan untuk manipulasi output (tidak digunakan dalam program ini).

#include <limits> → Berguna untuk validasi input.

```
#define MAX_BATIK 100
#define MAX_FAVORITE 50
```

#define MAX_BATIK 100 → Menentukan batas maksimal jumlah batik yang bisa disimpan.

#define MAX_FAVORITE 50 → Menentukan batas maksimal jumlah batik favorit (belum digunakan dalam program ini).

```
string batikNames[MAX_BATIK];
string batikOrigins[MAX_BATIK];
string batikCategories[MAX_BATIK];
string batikPrices[MAX_BATIK];
int batikCount = 0;
```

batikNames[] → Nama batik.

batikOrigins[] → Asal batik.

batikCategories[] → Kategori (Pria/Wanita).

batikPrices[] → Harga batik.

batikCount → Menghitung jumlah batik yang telah dimasukkan.

```
string adminUsername = "Wina Okta Ramadhani";
string adminPassword = "2409106100";
string userUsername = "SyaLomitha Novindriani Depe";
string userPassword = "2409106098";
```

Variabel untuk menyimpan username dan password admin dan User

```
while (true) {
    cout << "|=====|\n";
    cout << "|          Toko Batik Nusantara          |\n";
    cout << "|          ~Silahkan Login~              |\n";
    cout << "|=====|\n";
    cout << "|    1. Login                            |\n";
    cout << "|    2. Exit                             |\n";
    cout << "|=====|\n";
    cout << "Pilih opsi: ";

    int opsi;
    cin >> opsi;
    cin.ignore();
}
```

Loop while (true) → Program berjalan terus sampai pengguna memilih **Exit**.

Menampilkan menu login.

Menerima input pilihan pengguna (**opsi**):

Jika **1** → Melanjutkan ke proses login.

Jika **2** → Keluar dari program.

```

    if (opsi == 1) {
        string username, password;
        int attempts = 0;
        bool loginSuccess = false;

        while (attempts < 3) {
            cout << "Username: ";
            getline(cin, username);
            cout << "Password: ";
            getline(cin, password);

            if ((username == adminUsername && password == adminPassword) ||
                (username == userUsername && password == userPassword)) {
                loginSuccess = true;
                cout << "\nLogin successful, Welcome! " << username << "!\n";
                break;
            } else {
                attempts++;
                cout << "\nLogin failed! Remaining login attempts: " << (3 - attempts) <<
                "\n";
            }
        }

        if (!loginSuccess) {
            cout << "You failed to login! Your access is blocked.\n";
            continue;
        }
    }
}

```

Loop **while** (**attempts < 3**):

Meminta **username** dan **password**, Jika cocok dengan data Admin/User, login berhasil. Jika salah, kesempatan login berkurang **maksimal 3 kali**. Jika gagal setelah 3 kali, akses diblokir (**continue** → kembali ke menu utama).

```

if (username == adminUsername && password == adminPassword) {
    int adminOption;
    do {
        cout << "|=====|\n";
        cout << "|          Pengelolaan Batik          |\n";
        cout << "|=====|\n";
        cout << "|      1. Tambah Batik                    |\n";
        cout << "|      2. Tampilkan Batik                 |\n";
        cout << "|      3. Update Batik                   |\n";
        cout << "|      4. Hapus Batik                    |\n";
        cout << "|      5. Logout                         |\n";
        cout << "|=====|\n";
        cout << "Pilih opsi: ";
        cin >> adminOption;
    } while (adminOption < 1 || adminOption > 5);
}

```

```
cin.ignore();
```

Menu untuk Admin dengan opsi CRUD:

1 → Tambah Batik.

2 → Tampilkan Batik.

3 → Update Batik.

4 → Hapus Batik.

5 → Logout.

```
if (adminOption == 1 && batikCount < MAX_BATIK) {
    cout << "Nama Batik: ";
    getline(cin, batikNames[batikCount]);
    cout << "Asal Batik: ";
    getline(cin, batikOrigins[batikCount]);
    cout << "Kategori (Pria/Wanita): ";
    getline(cin, batikCategories[batikCount]);
    cout << "Harga: ";
    getline(cin, batikPrices[batikCount]);
    batikCount++;
    cout << "Batik berhasil ditambahkan!\n";
}
```

Memasukkan data batik ke dalam array jika belum mencapai batas **MAX_BATIK**.

```
else if (adminOption == 2) {
    cout << "Daftar Batik:\n";
    for (int i = 0; i < batikCount; i++) {
        cout << i + 1 << ". " << batikNames[i] << " | " <<
        batikOrigins[i] << " | " << batikCategories[i] << " | " << batikPrices[i] <<
        "\n";
    }
}
```

Menampilkan semua batik yang telah di input

```
else if (adminOption == 3) {
    cout << "Masukkan nama batik yang ingin diupdate: ";
    string namaBatik;
    getline(cin, namaBatik);
    bool found = false;
    for (int i = 0; i < batikCount; i++) {
        if (batikNames[i] == namaBatik) {
            cout << "Nama Baru: ";
            getline(cin, batikNames[i]);
        }
    }
}
```

```

        cout << "Asal Baru: ";
        getline(cin, batikOrigins[i]);
        cout << "Kategori Baru (Pria/Wanita): ";
        getline(cin, batikCategories[i]);
        cout << "Harga Baru: ";
        getline(cin, batikPrices[i]);
        cout << "\nBatik berhasil diupdate!\n";
        found = true;
        break;
    }
}
if (!found) {
    cout << "Batik tidak ditemukan.\n";
}
}

```

Mencari batik berdasarkan nama dan memperbarui data jika ditemukan.

```

else if (adminOption == 4) {
    cout << "Masukkan nama batik yang ingin dihapus: ";
    string namaBatik;
    getline(cin, namaBatik);
    bool found = false;

```

Menghapus batik dengan menggeser elemen array.

```

else if (opsi == 2) {
    cout << "Thank you for visiting our application.\n";
    break;
}

```

Program berhenti jika memilih opsi logout

Algoritma Program

Pengguna memilih Login.

Pengguna memasukkan username dan password yang benar (sebagai admin).

Admin memilih Tambah Batik dan memasukkan data:

- *Nama: "Mega Mendung"*
- *Asal: "Cirebon"*
- *Kategori: "Wanita"*
- *Harga: "Rp 500.000"*

Admin memilih Tampilkan Batik untuk melihat data yang telah ditambahkan.

Admin memilih Logout dan kembali ke menu utama.

Pengguna memilih Exit untuk mengakhiri program.

3. Source Code

```
#include <iostream>
#include <string>
#include <iomanip>
#include <limits>

using namespace std;

#define MAX_BATIK 100
#define MAX_FAVORITE 50

int main() {
    string batikNames[MAX_BATIK];
    string batikOrigins[MAX_BATIK];
    string batikCategories[MAX_BATIK];
    string batikPrices[MAX_BATIK];
    int batikCount = 0;

    string adminUsername = "Wina Okta Ramadhani";
    string adminPassword = "2409106100";
    string userUsername = "Syalomitha Novindriani Depe";
    string userPassword = "2409106098";

    while (true) {
        cout << " |=====|\n";
        cout << " |          Toko Batik Nusantara          |\n";
        cout << " |          ~Silahkan Login~          |\n";
        cout << " |=====|\n";
        cout << " |      1. Login      |\n";
        cout << " |      2. Exit       |\n";
        cout << " |=====|\n";
        cout << "Pilih opsi: ";

        int opsi;
        cin >> opsi;
        cin.ignore();

        if (opsi == 1) {
            string username, password;
            int attempts = 0;
            bool loginSuccess = false;

            while (attempts < 3) {
                cout << "Username: ";
                getline(cin, username);
                cout << "Password: ";
                getline(cin, password);

                if ((username == adminUsername && password == adminPassword) ||
```

```

        (username == userUsername && password == userPassword)) {
            loginSuccess = true;
            cout << "\nLogin successful, Welcome! " << username << "!\n";
            break;
        } else {
            attempts++;
            cout << "\nLogin failed! Remaining login attempts: " << (3 - attempts) <<
            "\n";
        }

    }

    if (!loginSuccess) {
        cout << "You failed to login! Your access is blocked.\n";
        continue;
    }

    if (username == adminUsername && password == adminPassword) {
        int adminOption;
        do {
            cout << "|=====|\n";
            cout << "|                Pengelolaan Batik                |\n";
            cout << "|=====|\n";
            cout << "|    1. Tambah Batik                                |\n";
            cout << "|    2. Tampilkan Batik                             |\n";
            cout << "|    3. Update Batik                                |\n";
            cout << "|    4. Hapus Batik                                 |\n";
            cout << "|    5. Logout                                       |\n";
            cout << "|=====|\n";
            cout << "Pilih opsi: ";
            cin >> adminOption;
            cin.ignore();

            if (adminOption == 1 && batikCount < MAX_BATIK) {
                cout << "Nama Batik: ";
                getline(cin, batikNames[batikCount]);
                cout << "Asal Batik: ";
                getline(cin, batikOrigins[batikCount]);
                cout << "Kategori (Pria/Wanita): ";
                getline(cin, batikCategories[batikCount]);
                cout << "Harga: ";
                getline(cin, batikPrices[batikCount]);
                batikCount++;
                cout << "Batik berhasil ditambahkan!\n";
            }
            else if (adminOption == 2) {
                cout << "Daftar Batik:\n";
                for (int i = 0; i < batikCount; i++) {
                    cout << i + 1 << ". " << batikNames[i] << " | " <<
                    batikOrigins[i] << " | " << batikCategories[i] << " | " << batikPrices[i] <<
                    "\n";
                }
            }
        } while (adminOption != 5);
    }
}

```

```

    }
    else if (adminOption == 3) {
        cout << "Masukkan nama batik yang ingin diupdate: ";
        string namaBatik;
        getline(cin, namaBatik);
        bool found = false;
        for (int i = 0; i < batikCount; i++) {
            if (batikNames[i] == namaBatik) {
                cout << "Nama Baru: ";
                getline(cin, batikNames[i]);
                cout << "Asal Baru: ";
                getline(cin, batikOrigins[i]);
                cout << "Kategori Baru (Pria/Wanita): ";
                getline(cin, batikCategories[i]);
                cout << "Harga Baru: ";
                getline(cin, batikPrices[i]);
                cout << "\nBatik berhasil diupdate!\n";
                found = true;
                break;
            }
        }
        if (!found) {
            cout << "Batik tidak ditemukan.\n";
        }
    }
    else if (adminOption == 4) {
        cout << "Masukkan nama batik yang ingin dihapus: ";
        string namaBatik;
        getline(cin, namaBatik);
        bool found = false;
        for (int i = 0; i < batikCount; i++) {
            if (batikNames[i] == namaBatik) {
                for (int j = i; j < batikCount - 1; j++) {
                    batikNames[j] = batikNames[j + 1];
                    batikOrigins[j] = batikOrigins[j + 1];
                    batikCategories[j] = batikCategories[j + 1];
                    batikPrices[j] = batikPrices[j + 1];
                }
                batikCount--;
                cout << "\nBatik berhasil dihapus!\n";
                found = true;
                break;
            }
        }
        if (!found) {
            cout << "Batik tidak ditemukan.\n";
        }
    }
} while (adminOption != 5);
}
else if (username == userUsername && password == userPassword) {

```

```

        cout << "\nLogin berhasil sebagai User!\n";
        int userOption;
        do {
            cout << "|=====|\n";
            cout << "|          Menu User          |\n";
            cout << "|=====|\n";
            cout << "|    1. Tampilkan Batik    |\n";
            cout << "|    2. Logout             |\n";
            cout << "|=====|\n";
            cout << "Pilih opsi: ";
            cin >> userOption;
            cin.ignore();

            if (userOption == 1) {
                for (int i = 0; i < batikCount; i++) {
                    cout << i + 1 << ". " << batikNames[i] << " | " <<
batikOrigins[i] << " | " << batikCategories[i] << " | " << batikPrices[i] <<
"\n";
                }
            }
        } while (userOption != 2);
    }
}
}
}
}

```

4. Uji Coba dan Hasil Output

Hasil Uji coba dan Output

1. Login Page

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ppela\Documents\2409106100-WinaOktaRamadhani-PT-2>
; if ($?) { g++ 2409106100-WinaOktaRamadhani-PT-2.cpp -o 240910
madhani-PT-2 }
=====
          Toko Batik Nusantara
          ~Silahkan Login~
=====
      1. Login
      2. Exit
=====
Pilih opsi: 1
Username: winacantikbagetgaiss
Password: 12345678910

Login failed! Remaining login attempts: 2
Username: wowternyatasalah
Password: 12345678910

Login failed! Remaining login attempts: 1
Username: kacaw sala lagi ternyata
Password: 12345678910

Login failed! Remaining login attempts: 0
You failed to login! Your access is blocked.
=====
          Toko Batik Nusantara
          ~Silahkan Login~
=====
      1. Login
      2. Exit
=====
Pilih opsi: |
```

di dalam login ini user maupun admin di berikan 3 kali kesempatan login, apabila dalam 3 kali percobaan login selalu gagal program akan memblokir akses dan kembali ke menu utama.

2. Menu Tambah Batik dan Tampilkan Batik

```
=====
          Pengelolaan Batik
          =====
      1. Tambah Batik
      2. Tampilkan Batik
      3. Update Batik
      4. Hapus Batik
      5. Logout
=====
Pilih opsi: 1
Nama Batik: Batik Ceplok
Asal Batik: Bantul
Kategori (Pria/Wanita): Pria
Harga: 52.600
Batik berhasil ditambahkan!
=====
          Pengelolaan Batik
          =====
      1. Tambah Batik
      2. Tampilkan Batik
      3. Update Batik
      4. Hapus Batik
      5. Logout
=====
Pilih opsi: 2
Daftar Batik:
1. Batik Mega Mendung | Cirebon | Wanita | 53.500
2. Batik Ceplok | Bantul | Pria | 52.600
=====
          Pengelolaan Batik
          =====
      1. Tambah Batik
      2. Tampilkan Batik
      3. Update Batik
      4. Hapus Batik
      5. Logout
=====
Pilih opsi: |
```

di sini saya login sebagai ADMIN, karena saya tidak menggunakan data base jadi harus menambahkan batik nya terlebih dahulu agar tampilan menu batik nya bisa muncul

3. Menu Update Batik

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

=====
1. Tambah Batik
2. Tampilkan Batik
3. Update Batik
4. Hapus Batik
5. Logout
=====
Pilih opsi: 3
Masukkan nama batik yang ingin diupdate: Batik Ceplok
Nama Baru: Ceplok Batik
Asal Baru: Bantul
Kategori Baru (Pria/Wanita): Pria
Harga Baru: 100.000

Batik berhasil diupdate!

=====
                Pengelolaan Batik
=====
1. Tambah Batik
2. Tampilkan Batik
3. Update Batik
4. Hapus Batik
5. Logout
=====
Pilih opsi: 2
Daftar Batik:
1. Batik Mega Mendung | Cirebon | Wanita | 53.500
2. Ceplok Batik | Bantul | Pria | 100.000

=====
                Pengelolaan Batik
=====
1. Tambah Batik
2. Tampilkan Batik
3. Update Batik
4. Hapus Batik
5. Logout
=====
Pilih opsi: 
```

Setelah memilih opsi update, akan muncul list untuk pembaharuan data (nama batik, asal, kategori, dan harga) setelah sudah di ubah tampilan akan kembali ke menu pengelolaan batik.

4. Menu Hapus Batik

```
=====
                Pengelolaan Batik
=====
1. Tambah Batik
2. Tampilkan Batik
3. Update Batik
4. Hapus Batik
5. Logout
=====
Pilih opsi: 4
Masukkan nama batik yang ingin dihapus: Batik Mega Mendung

Batik berhasil dihapus!

=====
                Pengelolaan Batik
=====
1. Tambah Batik
2. Tampilkan Batik
3. Update Batik
4. Hapus Batik
5. Logout
=====
Pilih opsi: 2
Daftar Batik:
1. Ceplok Batik | Bantul | Pria | 100.000

=====
                Pengelolaan Batik
=====
1. Tambah Batik
2. Tampilkan Batik
3. Update Batik
4. Hapus Batik
5. Logout
=====
Pilih opsi: 
```

Di opsi Hapus batik hanya meminta input nama batik, setelah itu data batik yang di input akan hilang dari daftar data Batik.

5. Logout

```
=====
|                                     |
|           Pengelolaan Batik        |
|                                     |
|      1. Tambah Batik              |
|      2. Tampilkan Batik           |
|      3. Update Batik              |
|      4. Hapus Batik               |
|      5. Logout                    |
|                                     |
| Pilih opsi: 5                     |
|                                     |
|           Toko Batik Nusantara     |
|           ~Silahkan Login~         |
|                                     |
|      1. Login                     |
|      2. Exit                      |
|                                     |
| Pilih opsi: |
```

Setelah memilih opsi Logout program akan kembali ke Login Page

6. Login sebagai User

```
=====
|           Toko Batik Nusantara     |
|           ~Silahkan Login~         |
|                                     |
|      1. Login                     |
|      2. Exit                      |
|                                     |
| Pilih opsi: 1                     |
| Username: Syalomitha Novindriani Depe |
| Password: 2409106098              |
|                                     |
| Login successful, Welcome! Syalomitha Novindriani Depe! |
|                                     |
| Login berhasil sebagai User!        |
|                                     |
|           Menu User                 |
|                                     |
|      1. Tampilkan Batik             |
|      2. Logout                     |
|                                     |
| Pilih opsi: 1                     |
| 1. Ceplok Batik | Bantul | Pria | 100.000 |
```

Di bagian User ini hanya Bisa menampilkan Menu Batik saja.

7. Logout

```
=====
|      1. Tampilkan Batik            |
|      2. Logout                    |
|=====|
| Pilih opsi: 2                     |
|=====|
|           Toko Batik Nusantara     |
|           ~Silahkan Login~         |
|=====|
|      1. Login                     |
|      2. Exit                      |
|=====|
```

Setelah memilih opsi Logout program akan kembali ke Login Page.

5. Git

```
ppela@LAPTOP-6HEHUF19 MINGW64 ~/Documents/github/praktikum-apl (master)
$ git add .

ppela@LAPTOP-6HEHUF19 MINGW64 ~/Documents/github/praktikum-apl (master)
$ git commit -m "Post-test-apl-2"
On branch master
nothing to commit, working tree clean

ppela@LAPTOP-6HEHUF19 MINGW64 ~/Documents/github/praktikum-apl (master)
$ git push --force origin main
error: src refspec main does not match any
error: failed to push some refs to 'https://github.com/winaoktaar/praktikum-apl.git'

ppela@LAPTOP-6HEHUF19 MINGW64 ~/Documents/github/praktikum-apl (master)
$ git push --force origin master
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 12 threads
Compressing objects: 100% (15/15), done.
Writing objects: 100% (18/18), 1.64 MiB | 769.00 KiB/s, done.
Total 18 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 2 local objects.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/winaoktaar/praktikum-apl/pull/new/master
remote:
To https://github.com/winaoktaar/praktikum-apl.git
 * [new branch]      master -> master

ppela@LAPTOP-6HEHUF19 MINGW64 ~/Documents/github/praktikum-apl (master)
$ |
```

\$ git add .

- Perintah ini menambahkan semua file yang ada di dalam folder ke staging area.
- Staging area adalah tempat sementara sebelum file dikomit ke dalam repository.

\$ git commit -m "post-test-apl-2"

- Perintah ini menyimpan perubahan dalam repository dengan commit dan pesan "update".
- File yang dikomit:
 - Post-test/Post-test-apl-2/2409106100-WinaOktaRamadhani-PT-2.cpp
 - Post-test/Post-test-apl-2/2409106100-WinaOktaRamadhani-PT-2.exe

git push --force origin main/master

digunakan untuk memaksa mengirim perubahan dari komputer ke repository di GitHub (atau remote lainnya), meskipun ada perubahan di remote yang berbeda dengan yang ada di lokal.