

LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN LANJUT



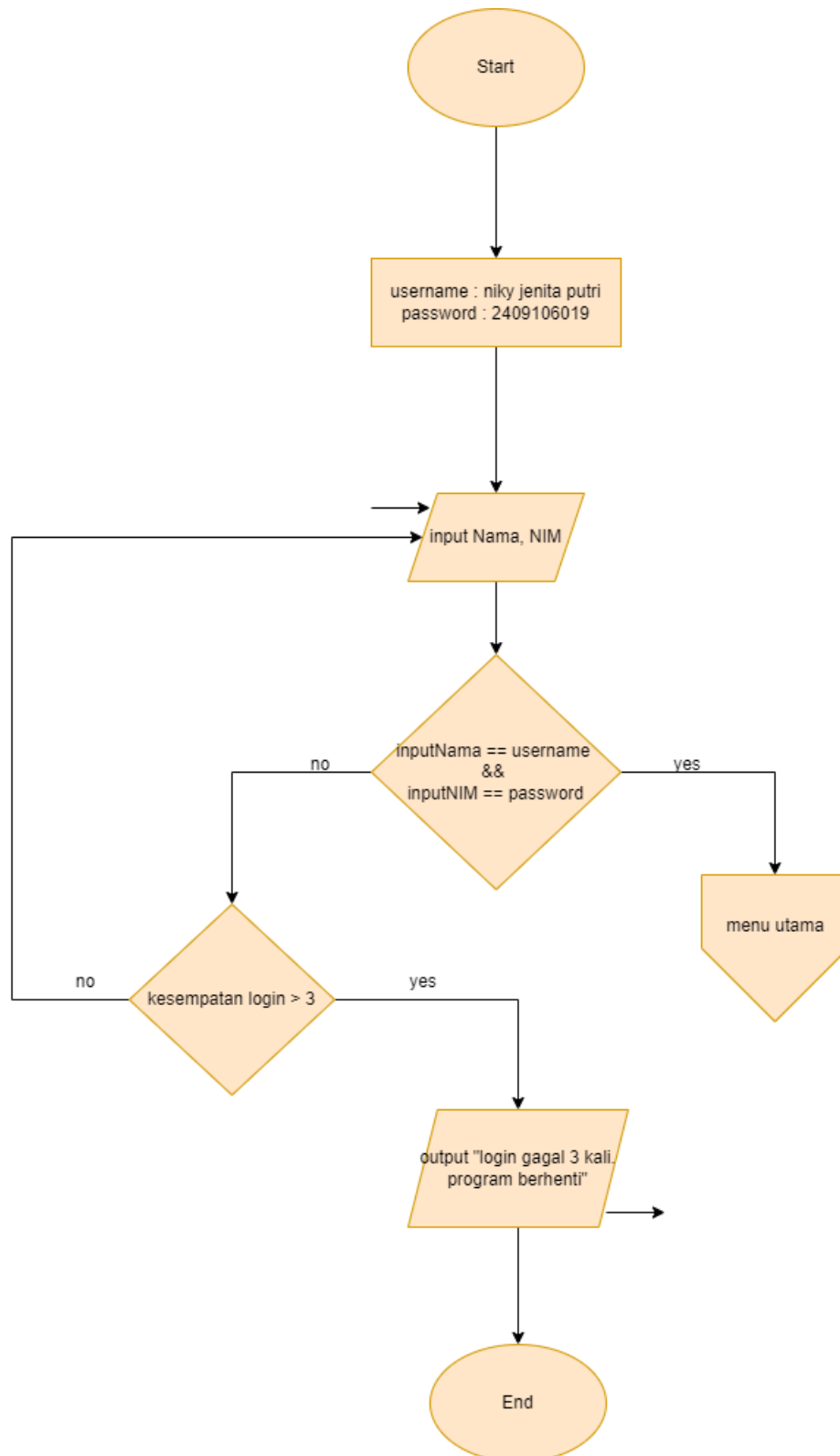
Disusun oleh:
Niky Jenita Putri (2409106019)
Kelas (A1 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA

2025

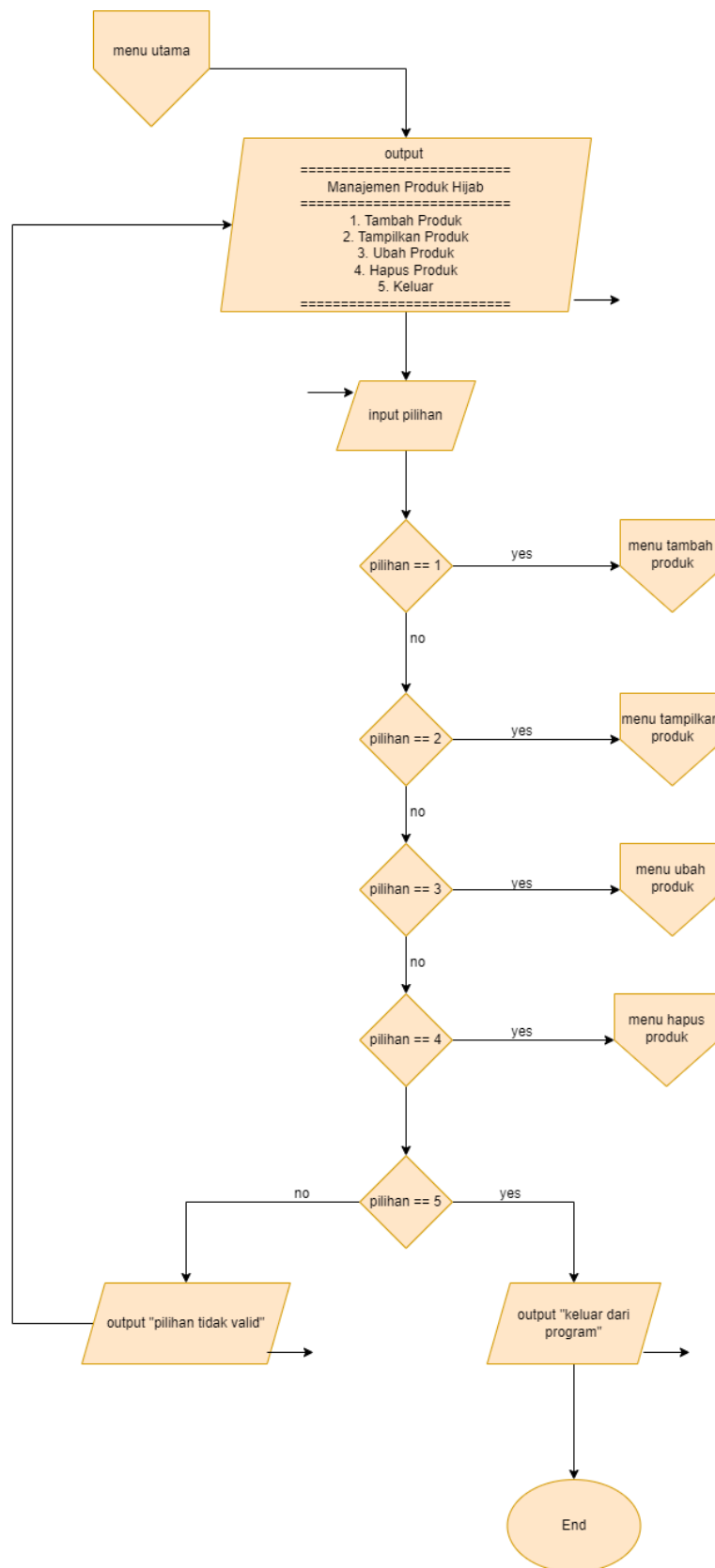
1. Flowchart

1.1 Menu Awal



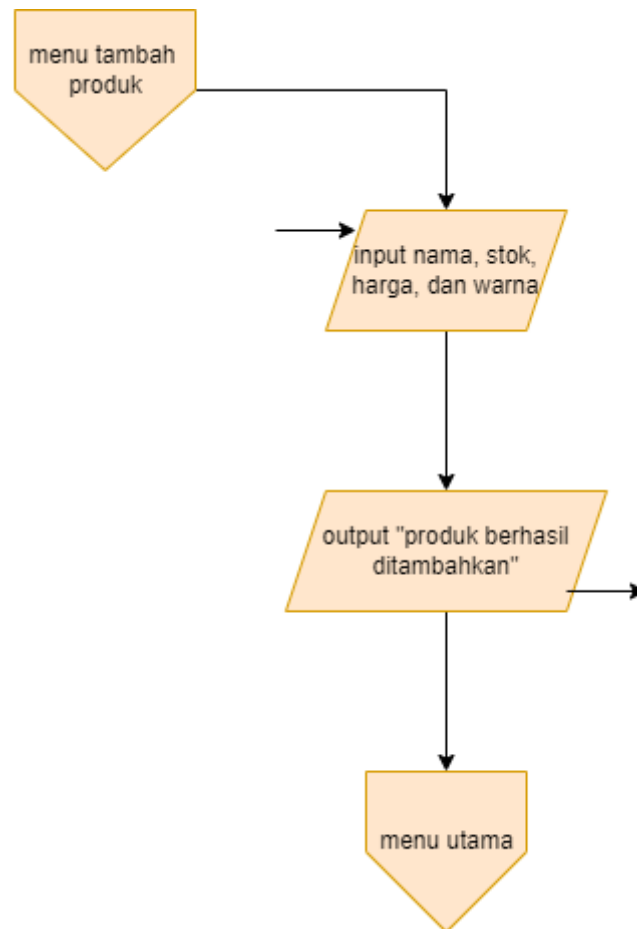
Gambar 1.1 Menu Awal

1.2 Menu Utama



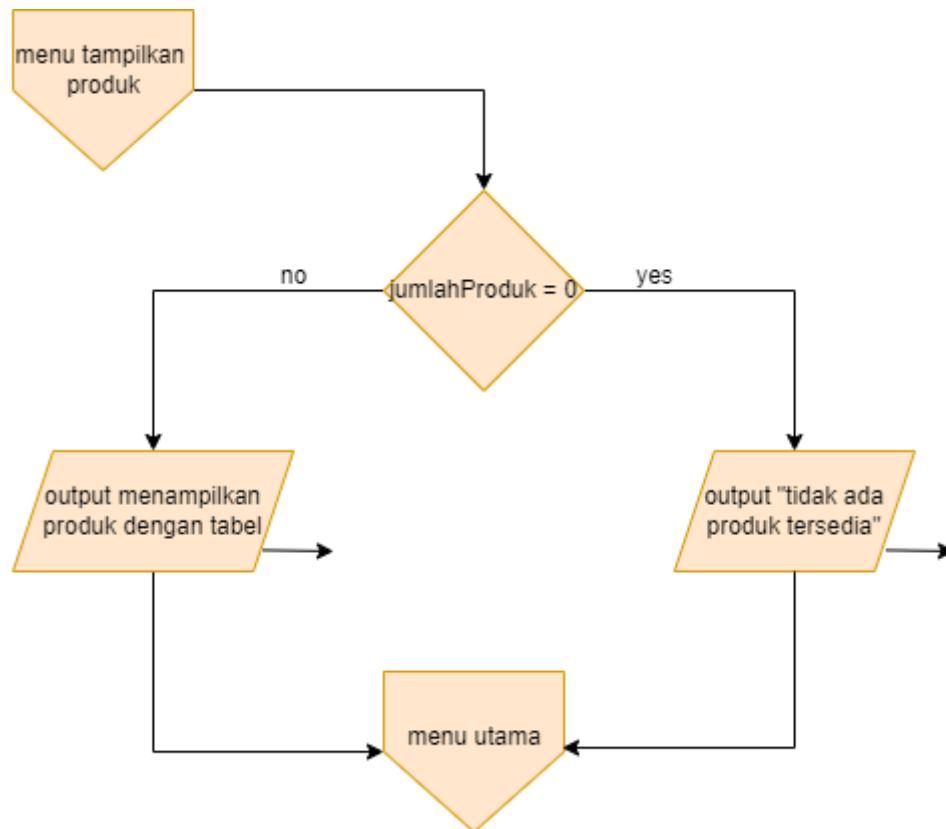
Gambar 1.2 Menu Utama

1.3 Menu Tambah Produk



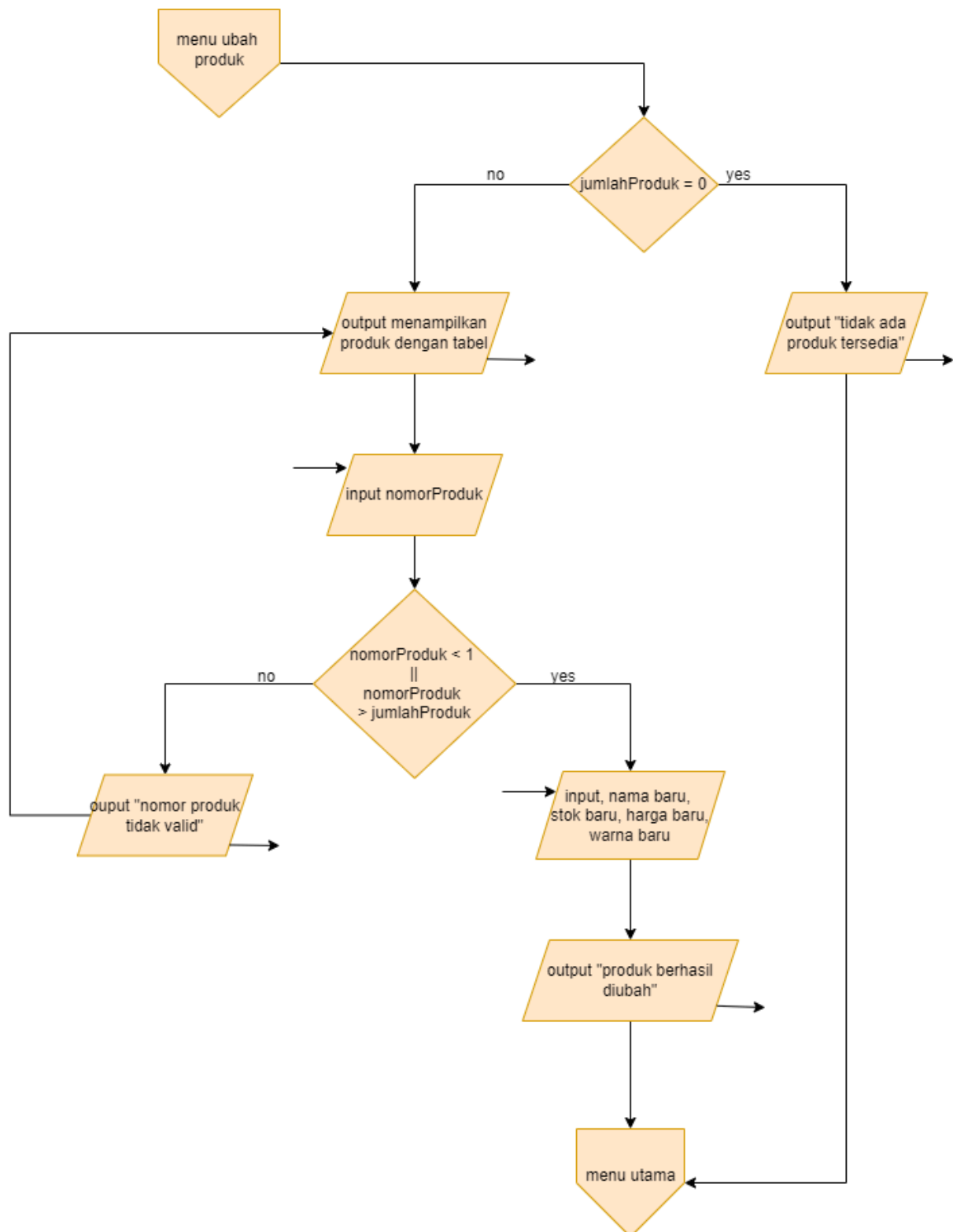
Gambar 1.3 Menu Tambah Produk

1.4 Menu Tampilkan Produk



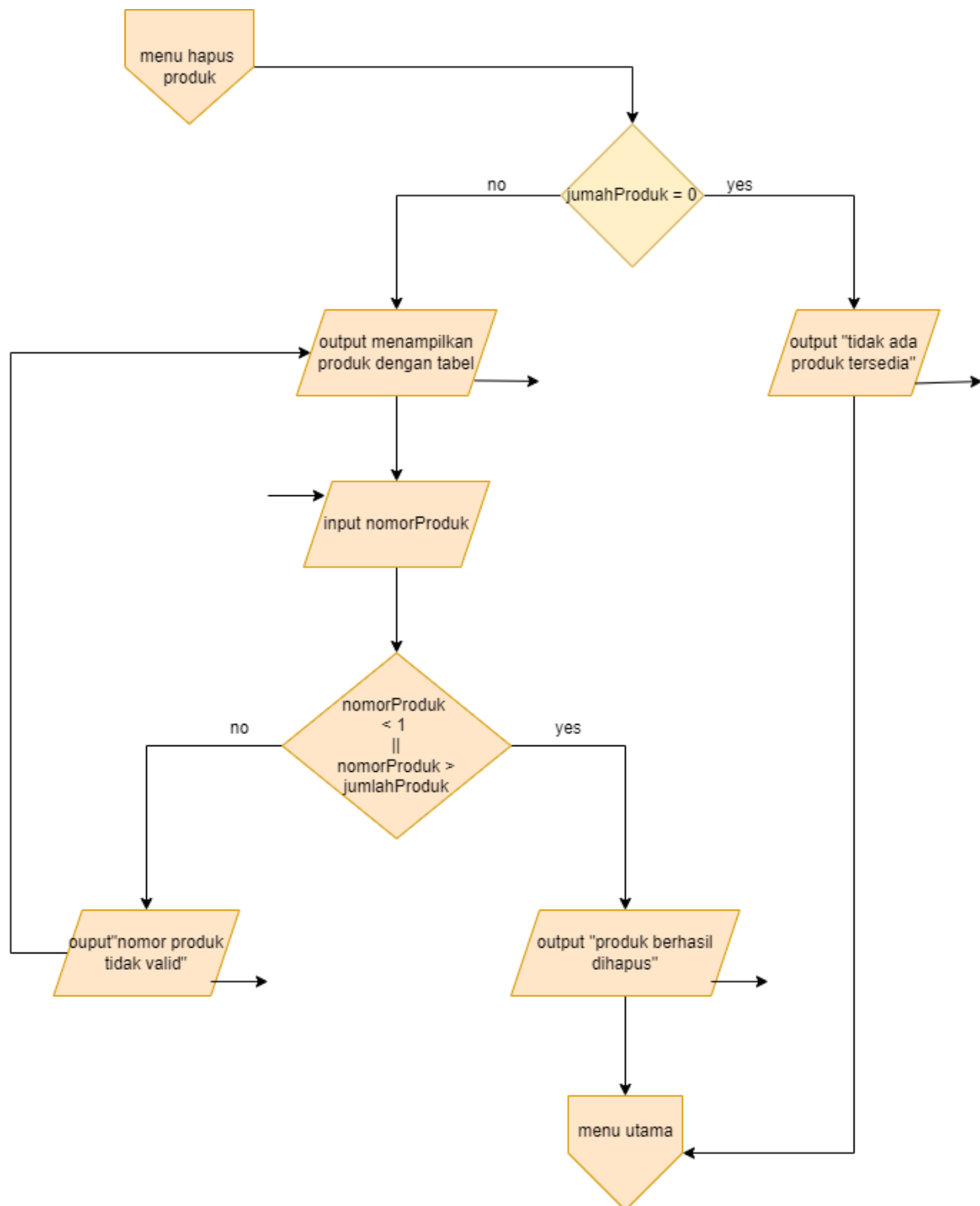
Gambar 1.4 Menu Tampilkan Produk

1.5 Menu Ubah Produk



Gambar 1.5 Menu Ubah Produk

1.6 Menu Hapus Produk



Gambar 1.6 Menu Hapus Produk

2. Analisis Program

2.1 Deskripsi Singkat Program

Program ini dirancang untuk mengelola data produk hijab dengan fitur utama seperti login keamanan menggunakan Nama dan NIM, penambahan produk baru dengan informasi lengkap, serta pengeditan, penghapusan, dan tampilan daftar produk dalam format tabel. Dengan adanya sistem ini, pengelolaan stok hijab menjadi lebih efisien, pencatatan produk lebih sistematis, serta meminimalkan risiko kehilangan data, sehingga membantu pengguna dalam mengatur inventaris dengan lebih mudah dan terstruktur.

2.2 Penjelasan Alur & Algoritma

A. Login

Program dimulai dengan meminta pengguna untuk melakukan login dengan memasukkan Nama dan NIM.

1. Username default: "niky jenita putri"
2. Password default: "2409106019"
3. Maksimal percobaan login adalah 3 kali (max_login_attempts).
4. Jika input cocok dengan username dan password, maka login berhasil.
5. Jika salah selama 3 kali berturut-turut, program berhenti dengan pesan:
"Login gagal 3 kali. Program berhenti."

B. Menampilkan Menu Utama

Setelah login berhasil, program menampilkan menu utama Manajemen Produk Hijab, dengan pilihan:

1. Tambah Produk
2. Tampilkan Produk
3. Ubah Produk
4. Hapus Produk
5. Keluar

Pengguna memilih menu dengan memasukkan angka 1-5.

C. Fitur Manajemen Produk

Program menyimpan data produk dalam array dengan batas maksimal 100 produk (max_produk).

Setiap produk memiliki atribut:

1. Nama Produk
2. Stok
3. Harga
4. Warna dalam 2 kategori:
Standar (3 warna)
Premium (3 warna)

D. Tambah Produk (Pilihan 1)

1. Pengguna memasukkan Nama Produk, Stok, dan Harga.
2. Pengguna memasukkan 3 warna untuk kategori Standar.
3. Pengguna memasukkan 3 warna untuk kategori Premium.
4. Jika produk sudah mencapai batas (max_produk), program menampilkan pesan:
"Data penuh!!"

E. Tampilkan Produk (Pilihan 2)

1. Jika tidak ada produk, program menampilkan:
"Tidak ada produk tersedia."
2. Jika ada produk, program menampilkan daftar dalam bentuk tabel yang mencakup:
 - Nomor
 - Nama Produk
 - Stok
 - Harga
 - Warna Standar
 - Warna Premium

F. Ubah Produk (Pilihan 3)

1. Program menampilkan daftar produk dengan nomor urut.
2. Pengguna memasukkan nomor produk yang ingin diubah.

3. Jika nomor produk tidak valid, program meminta input ulang.
4. Pengguna memasukkan data baru:
 - Nama baru
 - Stok baru
 - Harga baru
 - 3 warna baru untuk Standar
 - 3 warna baru untuk Premium

G. Hapus Produk (Pilihan 4)

1. Program menampilkan daftar produk.
2. Pengguna memilih nomor produk yang ingin dihapus.
3. Jika nomor produk tidak valid, program meminta input ulang.
4. Setelah memilih produk, data produk tersebut digeser ke atas dalam array untuk menghapusnya.
5. Jumlah produk dikurangi (jumlahProduk--).
6. Program menampilkan pesan:
"Produk berhasil dihapus."

H. Keluar dari Program (Pilihan 5)

1. Jika pengguna memilih 5, program menampilkan:
"Keluar dari program."
2. Program berhenti

3. Source Code

3.1 Login

Fitur ini digunakan untuk memastikan hanya pengguna yang memiliki kredensial yang benar yang dapat mengakses program. Pengguna diminta memasukkan Nama dan NIM. Jika kredensial benar, pengguna dapat melanjutkan ke menu utama. Jika salah, pengguna diberikan 3 kesempatan untuk mencoba kembali. Jika tetap gagal, program akan berhenti.

```
while (loginAttempts < max_login_attempts) {
    cout << "Masukkan Nama : ";
    getline(cin, inputNama);
    cout << "Masukkan NIM : ";
    getline(cin, inputNIM);

    if (inputNama == username && inputNIM == password){
        cout << "Login berhasil!!\n";
        break;
    }else {
        cout << "Login gagal. Coba lagi.\n";
        loginAttempts++;
    }
}

if (loginAttempts == max_login_attempts) {
    cout << "Login gagal 3 kali. Program berhenti.\n";
    return 0;
}
```

3.2 Menu Utama

Menu utama menampilkan menu manajemen produk hijab. Program ini memungkinkan pengguna untuk memilih berbagai opsi, seperti menambah, menampilkan, mengubah, dan menghapus produk, serta keluar dari program. Setiap iterasi dalam loop akan menampilkan menu tersebut dan meminta pengguna untuk memasukkan pilihan mereka melalui variabel pilihan.

```
int pilihan;
do{
    cout << "\n=====\\n";
    cout << " |          MANAJEMEN PRODUK HIJAB          |\\n";
    cout << "\n=====\\n";
    cout << " |          1. Tambah Produk          |\\n";
    cout << " |          2. Tampilkan Produk       |\\n";
    cout << " |          3. Ubah Produk            |\\n";
    cout << " |          4. Hapus Produk           |\\n";
    cout << " |          5. Keluar                 |\\n";
```

```

cout << "=====\n";
cout << "Masukkan pilihan : ";
cin >> pilihan;
cin.ignore();

```

3.3 Tambah Produk

Menu tambah produk menangani penambahan produk hijab dalam sistem manajemen toko. Saat pengguna memilih opsi Tambah Produk (case 1), program akan meminta beberapa data terkait produk yang akan dimasukkan ke dalam array.

```

case 1:
    if (jumlahProduk < max_produk) {
        cout << "Masukkan Nama Produk : ";
        getline(cin,namaProduk[jumlahProduk]);
        cout << "Masukkan Stok : ";
        cin >> stokProduk[jumlahProduk];
        cout << "Masukkan Harga : ";
        cin >> hargaProduk[jumlahProduk];
        cin.ignore();

        for (int k = 0; k < 2; k++) {
            string kategori = (k == 0) ? "Standar" : "Premium";
            cout << "\nMasukkan Warna untuk Kategori " << kategori << ":\n";
            for (int i = 0; i < 3; i++) {
                cout << "Warna ke-" << i+1 << " : ";
                getline(cin, warnaProduk[jumlahProduk][k][i]);
            }
        }
        jumlahProduk++;
        cout << "Produk berhasil ditambahkan";
    } else {
        cout << "Data penuh!!\n";
    }
    break;

```

3.4 Tampilkan Produk

Menu tampilkan produk yang menangani tampilan daftar produk hijab dalam sistem manajemen toko. Saat pengguna memilih Tampilkan Produk (case 2), program akan menampilkan semua produk yang telah disimpan dalam bentuk tabel. Jika tidak ada produk maka akan menampilkan pesan kesalahan.

```

case 2:
    if (jumlahProduk == 0) {
        cout << "\nTidak ada produk tersedia.\n";
    } else {
        cout <<

```

```

"\n=====
===\n";
    cout << " | No | Nama          | Stok | Harga |      Warna Standar      |
Warna Premium      |\n";
    cout <<
"\n=====
===\n";
    for (int i = 0; i < jumlahProduk; i++) {
        cout << " | " << setw(2) << i+1 << " | "
            << setw(12) << namaProduk[i] << " | "
            << setw(6) << stokProduk[i] << " | "
            << setw(7) << hargaProduk[i] << " | "
            << setw(20) << warnaProduk[i][0][0] << ", " << warnaProduk[i][0][1]
+ ", " << warnaProduk[i][0][2] << " | "
            << setw(25) << warnaProduk[i][1][0] << ", " << warnaProduk[i][1][1]
+ ", " << warnaProduk[i][1][2] << " |\n";
    }

    cout <<
"\n=====
===\n";
}
break;

```

3.5 Ubah Produk

Case 3 memungkinkan pengguna mengubah data produk. Jika tidak ada produk, program menampilkan pesan dan keluar. Jika ada, daftar produk ditampilkan, lalu pengguna memilih nomor produk dengan validasi input. Setelah itu, pengguna memasukkan nama, stok, harga, dan warna baru untuk kategori Standar dan Premium. Perubahan disimpan, dan program menampilkan pesan "Produk berhasil diubah".

```

case 3:
    if (jumlahProduk == 0) {
        cout << "\nTidak ada produk tersedia.\n";
        break;
    }

    int nomorProduk;
    do {
        cout << "Daftar Produk : \n";
        cout <<
"\n=====
=====
\n";
        cout << " | No | Nama          | Stok | Harga |      Warna Standar      |
Warna Premium      |\n";
        cout <<
"\n=====
=====
\n";
    }

```

```

        for (int i = 0; i < jumlahProduk; i++) {
            cout << "| " << setw(2) << i+1 << " |"
                << setw(12) << namaProduk[i] << " |"
                << setw(6) << stokProduk[i] << " |"
                << setw(7) << hargaProduk[i] << " |"
                << setw(20) << warnaProduk[i][0][0] << ", " <<
warnaProduk[i][0][1] << ", " << warnaProduk[i][0][2] << " |"
                << setw(25) << warnaProduk[i][1][0] << ", " <<
warnaProduk[i][1][1] << ", " << warnaProduk[i][1][2] << " |\n";
        }
        cout <<
"\n=====
=====\\n";
        cout << "\\nMasukkan nomor produk yang ingin diubah : ";
        cin >> nomorProduk;
        if (nomorProduk < 1 || nomorProduk > jumlahProduk) {
            cout << "Nomor produk tidak valid!!\\n";
        }
    } while (nomorProduk < 1 || nomorProduk > jumlahProduk);

    nomorProduk--;

    cout << "Masukkan Nama Baru : ";
    cin.ignore();
    getline(cin, namaProduk[nomorProduk]);
    cout << "Masukkan Stok Baru : ";
    cin >> stokProduk[nomorProduk];
    cout << "Masukkan Harga Baru : ";
    cin >> hargaProduk[nomorProduk];
    cin.ignore();

    for (int k = 0; k < 2; k++) {
        string kategori = (k == 0) ? "Standar" : "Premium";
        cout << "\\nMasukkan Warna Baru untuk Kategori " << kategori << " :\\n";
        for (int i = 0; i < 3; i++) {
            cout << "Warna ke-" << i+1 << " : ";
            getline(cin, warnaProduk[nomorProduk][k][i]);
        }
    }
    cout << "produk berhasil diubah";
    break;

```

3.6 Hapus Produk

Case 4 digunakan untuk menghapus produk dari daftar. Jika tidak ada produk, program menampilkan pesan dan keluar. Jika ada, daftar produk ditampilkan, lalu pengguna memilih nomor produk yang ingin dihapus dengan validasi input.

```

case 4:
    if (jumlahProduk == 0) {
        cout << "\nTidak ada produk tersedia.\n";
        break;
    }

    do {
        cout <<
        "\n=====
        =====\n";
        cout << " | No | Nama          | Stok | Harga |      Warna Standar      |
        Warna Premium      |\n";
        cout <<
        "\n=====
        =====\n";
        for (int i = 0; i < jumlahProduk; i++) {
            cout << " | " << setw(2) << i+1 << " | "
                    << setw(12) << namaProduk[i] << " | "
                    << setw(6) << stokProduk[i] << " | "
                    << setw(7) << hargaProduk[i] << " | "
                    << setw(20) << warnaProduk[i][0][0] << ", " <<
                    warnaProduk[i][0][1] << ", " << warnaProduk[i][0][2] << " | "
                    << setw(25) << warnaProduk[i][1][0] << ", " <<
                    warnaProduk[i][1][1] << ", " << warnaProduk[i][1][2] << " |\n";
        }

        cout <<
        "\n=====
        =====\n";
        cout << "\nMasukkan nomor produk yang ingin diubah : ";
        cin >> nomorProduk;
        if (nomorProduk < 1 || nomorProduk > jumlahProduk) {
            cout << "Nomor produk tidak valid!!\n";
        }
    } while (nomorProduk < 1 || nomorProduk > jumlahProduk);

    nomorProduk--;

    for (int i = nomorProduk; i < jumlahProduk - 1; i++) {
        namaProduk[i] = namaProduk[i + 1];
        stokProduk[i] = stokProduk[i + 1];
        hargaProduk[i] = hargaProduk[i + 1];
    }
    jumlahProduk--;
    cout << "Produk berhasil dihapus.\n";
    break;

```

4. Uji Coba dan Hasil Output

4.1 Uji Coba

Di sebuah kota kecil, seorang pemilik toko hijab bernama Niky ingin mengelola produknya dengan lebih baik. Ia telah membuat sebuah sistem manajemen produk hijab yang membantunya mencatat stok, harga, dan warna hijab yang tersedia. Hari ini, ia akan menguji program tersebut untuk memastikan semuanya berjalan dengan lancar.

- A. Niky menyalakan komputer dan membuka programnya. Dia memasukkan nama dan NIM yang telah ia daftarkan sebelumnya.
- B. Setelah masuk ke sistem, Niky ingin menambahkan produk baru, yaitu hijab Pashmina dengan stok 123 dan harga sebesar Rp40.000 yang terdiri dari warna standar yaitu hitam, navy, putih dan warna premium yaitu coklat, maroon, pink.
- C. Setelah menambahkan produk, Niky ingin memastikan produknya tersimpan dengan benar. Dia memilih opsi Tampilkan Produk.
- D. Beberapa hari kemudian, Niky menyadari ada kesalahan pada harga Pashmina, seharusnya Rp45.000. Dia pun memilih opsi Ubah Produk.
- E. Beberapa minggu kemudian, Niky kehabisan stok Pashmina dan ingin menghapusnya dari daftar produk.
- F. Setelah semua data diperbarui, Niky memutuskan untuk keluar dari sistem. Dia memilih opsi 5 di menu utama.
- G. Suatu hari, asistennya, Dina, mencoba masuk ke sistem menggunakan nama dan NIM yang salah. Dina mencoba tiga kali dan tetap salah. Akhirnya, sistem menutup program.

4.2 Hasil Output

A. Skenario 1

```
=====
|               MANAJEMEN PRODUK HIJAB               |
=====
Masukkan Nama : niky jenita putri
Masukkan NIM : 2409106019
Login berhasil!!

=====
|               MANAJEMEN PRODUK HIJAB               |
=====
|               1. Tambah Produk                     |
|               2. Tampilkan Produk                   |
|               3. Ubah Produk                       |
|               4. Hapus Produk                      |
|               5. Keluar                             |
=====
Masukkan pilihan : █
```

Gambar 4.1 Skenario 1

B. Skenario 2

```
=====
|               MANAJEMEN PRODUK HIJAB               |
=====
|               1. Tambah Produk                     |
|               2. Tampilkan Produk                   |
|               3. Ubah Produk                       |
|               4. Hapus Produk                      |
|               5. Keluar                             |
=====
Masukkan pilihan : 1
Masukkan Nama Produk : pashmina
Masukkan Stok : 123
Masukkan Harga : 40000

Masukkan Warna untuk Kategori Standar:
Warna ke-1 : hitam
Warna ke-2 : navy
Warna ke-3 : putih

Masukkan Warna untuk Kategori Premium:
Warna ke-1 : coklat
Warna ke-2 : maroon
Warna ke-3 : pink
Produk berhasil ditambahkan
```

Gambar 4.2 Skenario 2

C. Skenario 3

```
=====
|          MANAJEMEN PRODUK HIJAB          |
=====
|          1. Tambah Produk                |
|          2. Tampilkan Produk              |
|          3. Ubah Produk                  |
|          4. Hapus Produk                 |
|          5. Keluar                      |
=====
Masukkan pilihan : 2

=====
| No | Nama      | Stok | Harga | Warna Standar | Warna Premium |
=====
|  1 | pashmina | 123  | 40000 | hitam, navy, putih | coklat, maroon, pink |
=====
```

Gambar 4.3 Skenario 3

D. Skenario 4

```
=====
|          MANAJEMEN PRODUK HIJAB          |
=====
|          1. Tambah Produk                |
|          2. Tampilkan Produk              |
|          3. Ubah Produk                  |
|          4. Hapus Produk                 |
|          5. Keluar                      |
=====
Masukkan pilihan : 3
Daftar Produk :

=====
| No | Nama      | Stok | Harga | Warna Standar | Warna Premium |
=====
|  1 | pashmina | 123  | 40000 | hitam, navy, putih | coklat, maroon, pink |
=====

Masukkan nomor produk yang ingin diubah : 1
Masukkan Nama Baru : pashmina
Masukkan Stok Baru : 123
Masukkan Harga Baru : 45000

Masukkan Warna Baru untuk Kategori Standar:
Warna ke-1 : hitam
Warna ke-2 : navy
Warna ke-3 : putih

Masukkan Warna Baru untuk Kategori Premium:
Warna ke-1 : coklat
Warna ke-2 : maroon
Warna ke-3 : pink
produk berhasil diubah
```

Gambar 4.4 Skenario 4

E. Skenario 5

```
=====
|                                     |
|          MANAJEMEN PRODUK HIJAB    |
|                                     |
|=====|
|          1. Tambah Produk          |
|          2. Tampilkan Produk       |
|          3. Ubah Produk            |
|          4. Hapus Produk           |
|          5. Keluar                 |
|=====|
Masukkan pilihan : 4

=====
| No | Nama      | Stok | Harga | Warna Standar | Warna Premium |
|=====|
|  1 | pashmina |  123 | 45000 | hitam, navy, putih | coklat, maroon, pink |
|=====|

Masukkan nomor produk yang ingin diubah : 1
Produk berhasil dihapus.
```

Gambar 4.5 Skenario 5

F. Skenario 6

```
=====
|                                     |
|          MANAJEMEN PRODUK HIJAB    |
|                                     |
|=====|
|          1. Tambah Produk          |
|          2. Tampilkan Produk       |
|          3. Ubah Produk            |
|          4. Hapus Produk           |
|          5. Keluar                 |
|=====|
Masukkan pilihan : 5
Keluar dari program.
PS D:\praktikum-apl\post-test\post-test-apl-2>
```

Gambar 4.6 Skenario 6

G. Skenario 7

```
=====
|          MANAJEMEN PRODUK HIJAB          |
=====
Masukkan Nama : niky jenita
Masukkan NIM : 2409106019
Login gagal. Coba lagi.
Masukkan Nama : niky jenitaa putri
Masukkan NIM : 2409016019
Login gagal. Coba lagi.
Masukkan Nama : niky putri
Masukkan NIM : 234567
Login gagal. Coba lagi.
Login gagal 3 kali. Program berhenti.
PS D:\praktikum-apl\post-test\post-test-apl-2>
```

Gambar 4.7 Skenario 7

5. Langkah-Langkah Git

5.1 Git Init

Git init adalah perintah Git yang digunakan untuk menginisialisasi repository Git baru dalam sebuah direktori. Perintah ini akan membuat subdirektori tersembunyi bernama .git yang berisi semua file dan metadata yang diperlukan untuk melacak perubahan dalam proyek tersebut.

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

Gambar 5.1 Git Init

5.2 Git Add

Git add . adalah perintah Git yang digunakan untuk menambahkan semua perubahan dalam direktori kerja (working directory) ke dalam staging area. Dengan kata lain, perintah ini akan menandai semua file yang telah dibuat, diubah, atau dihapus agar siap untuk di-commit.

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

Gambar 5.2 Git Add

5.3 Git Commit -m

Git commit -m adalah perintah Git yang digunakan untuk menyimpan perubahan yang telah ditambahkan ke staging area ke dalam repository dengan pesan commit yang diberikan.

```
PS D:\praktikum-apl> git commit -m "selesai"
[main 612e7ec] selesai
3 files changed, 204 insertions(+)
create mode 100644 post-test/post-test-apl-2/2409106019-NikyJenitaPutri-PT-2.cpp
create mode 100644 post-test/post-test-apl-2/2409106019-NikyJenitaPutri-PT-2.exe
```

Gambar 5.3 Git Commit

5.4 Git Remote

Git remote add origin digunakan untuk menghubungkan repository lokal dengan repository di GitHub.

```
PS D:\praktikum-apl> git remote add origin https://github.com/NikyJenitaPutri/praktikum-apl.git
PS D:\praktikum-apl>
```

Gambar 5.4 Git Remote

5.5 Git Push

Git push adalah perintah Git yang digunakan untuk mengunggah (upload) perubahan dari repository lokal ke repository di GitHub. -u origin main berarti mengatur branch main sebagai default untuk push ke origin

```
PS D:\praktikum-apl> git push -u origin main
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 2 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 679.87 KiB | 3.28 MiB/s, done.
Total 8 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/NikyJenitaPutri/praktikum-apl.git
 71098ba..612e7ec main -> main
branch 'main' set up to track 'origin/main'.
PS D:\praktikum-apl> 
```

Gambar 5.5 Git Push