

**LAPORAN PRAKTIKUM**  
**POSTTEST (II)**  
**ALGORITMA PEMROGRAMAN LANJUT**



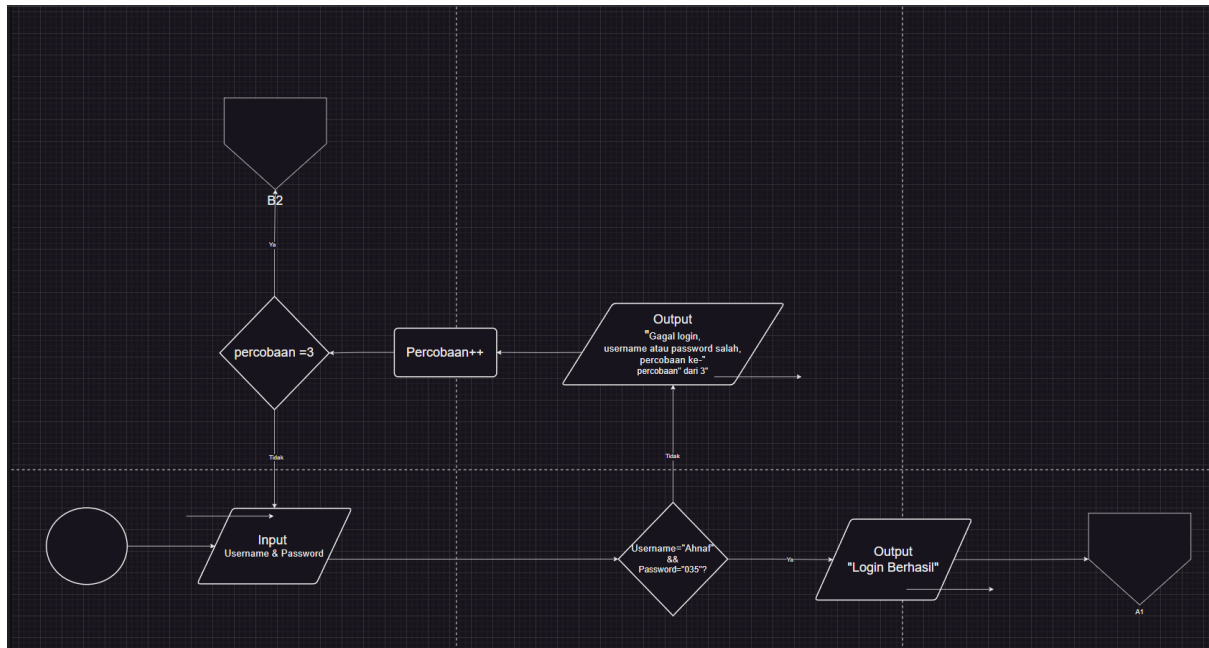
**Disusun oleh:**

**Nama (2409106035)**

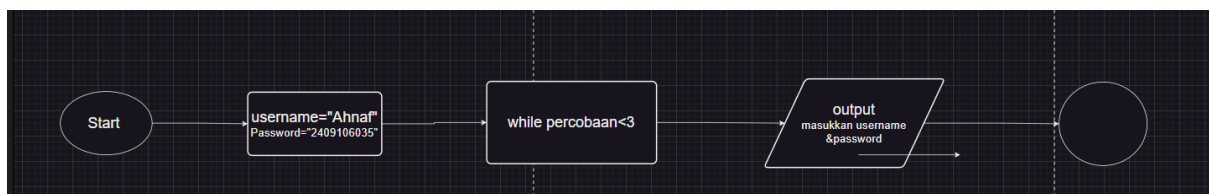
**Kelas (A2 '24)**

**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2025**

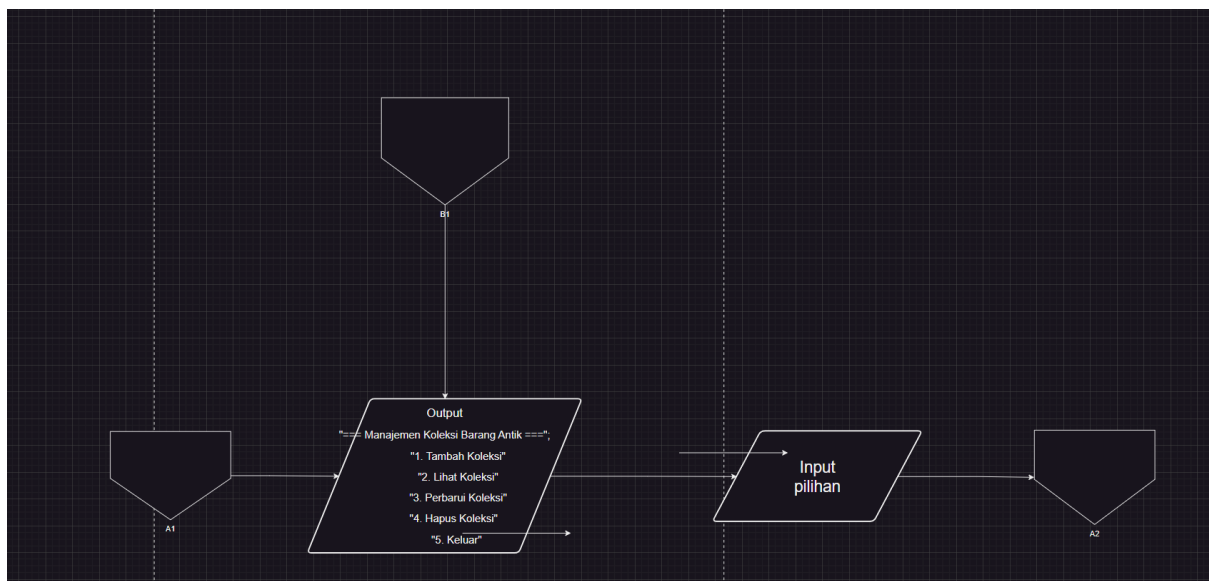
# 1. Flowchart



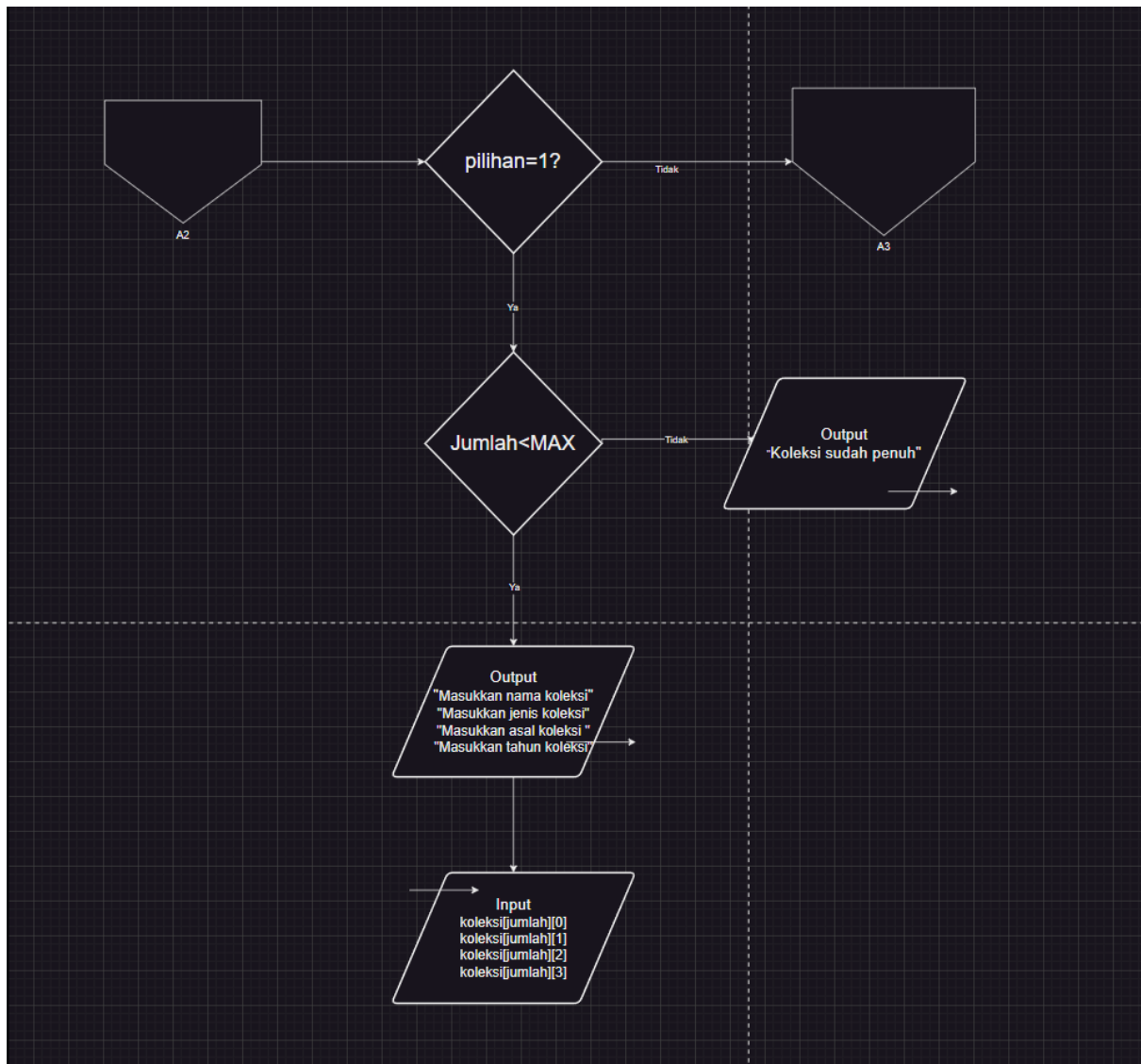
Gambar 1.1 flowchart login 1



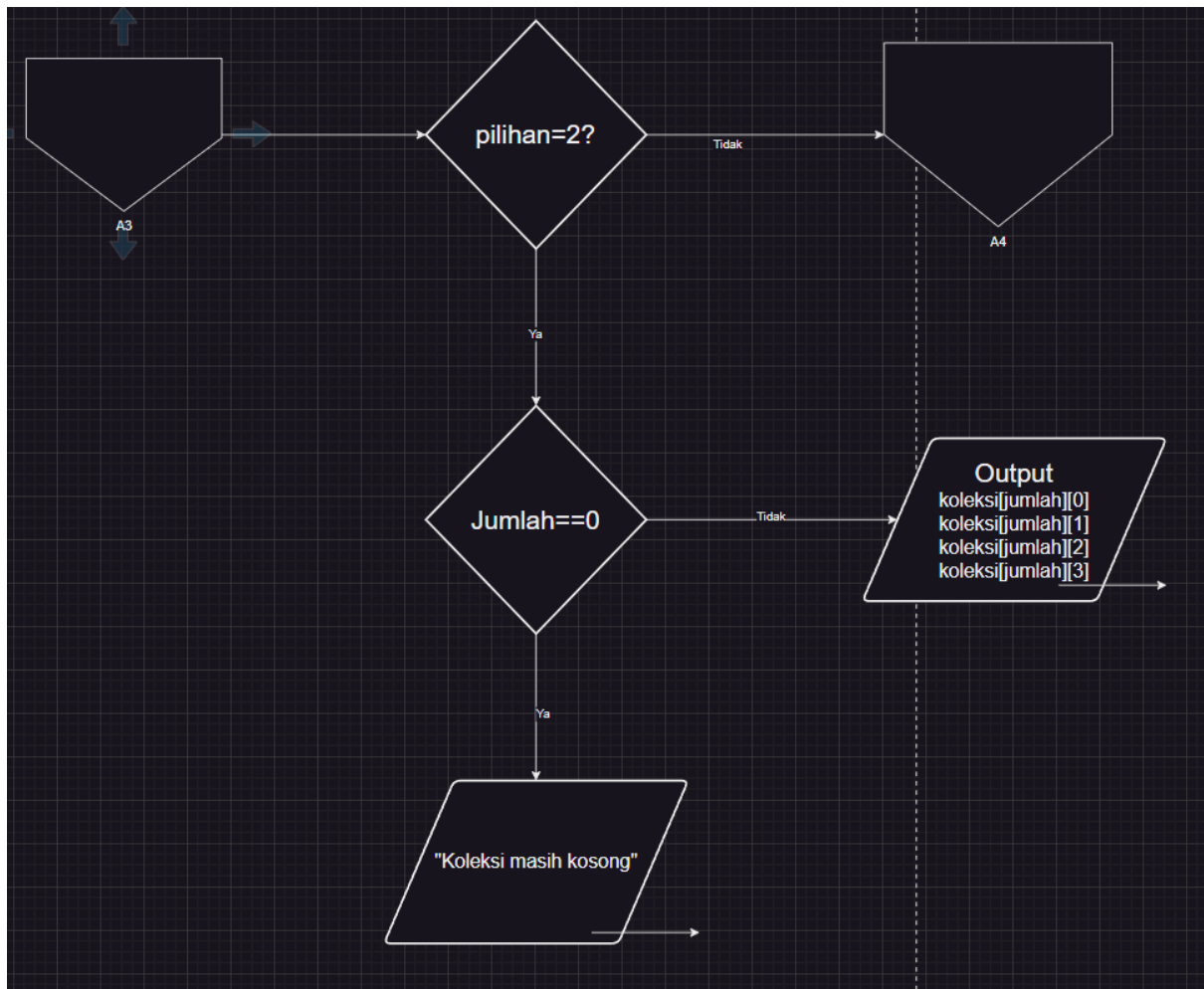
Gambar 1.2 flowchart login 2



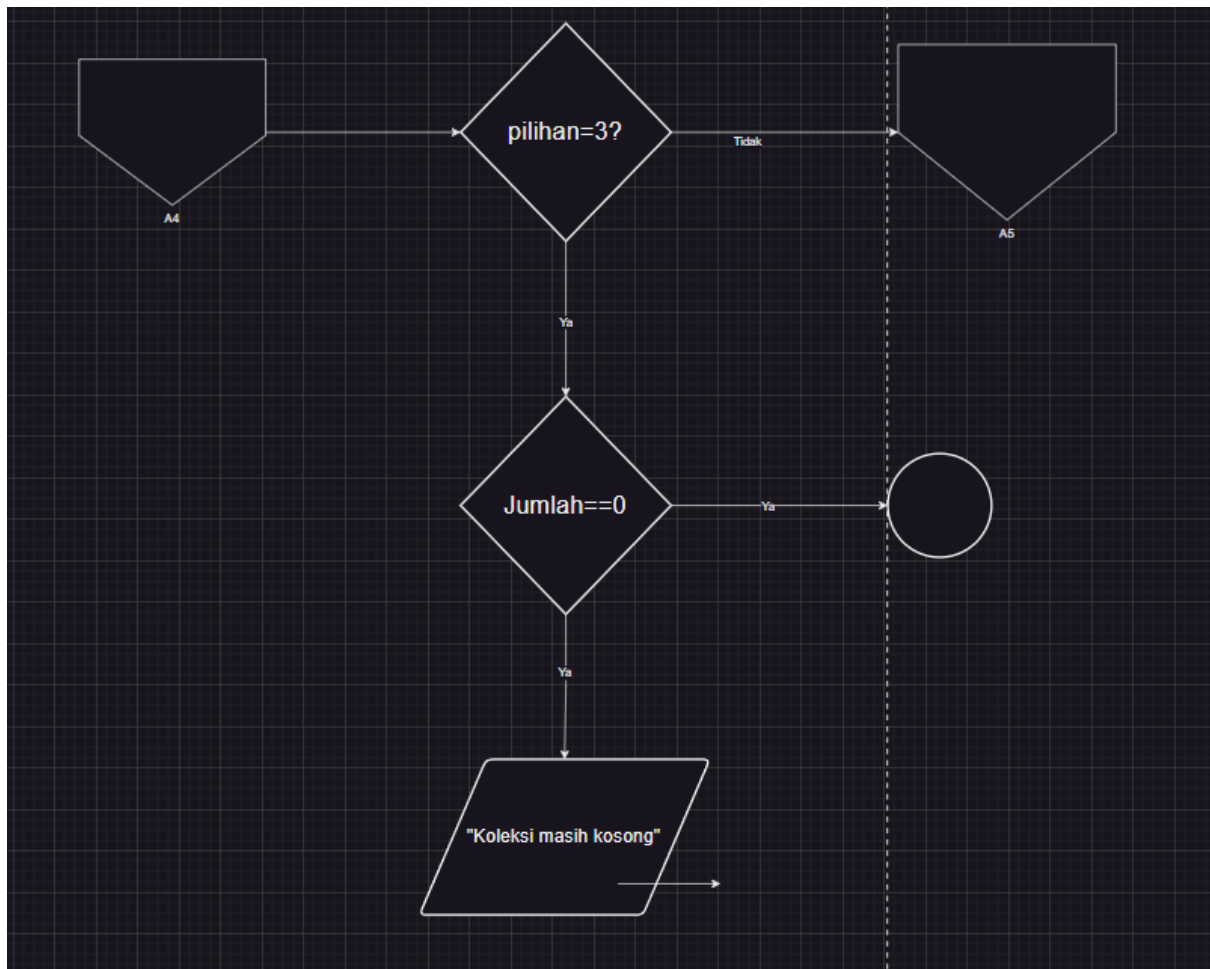
Gambar 1.3 pilihan menu



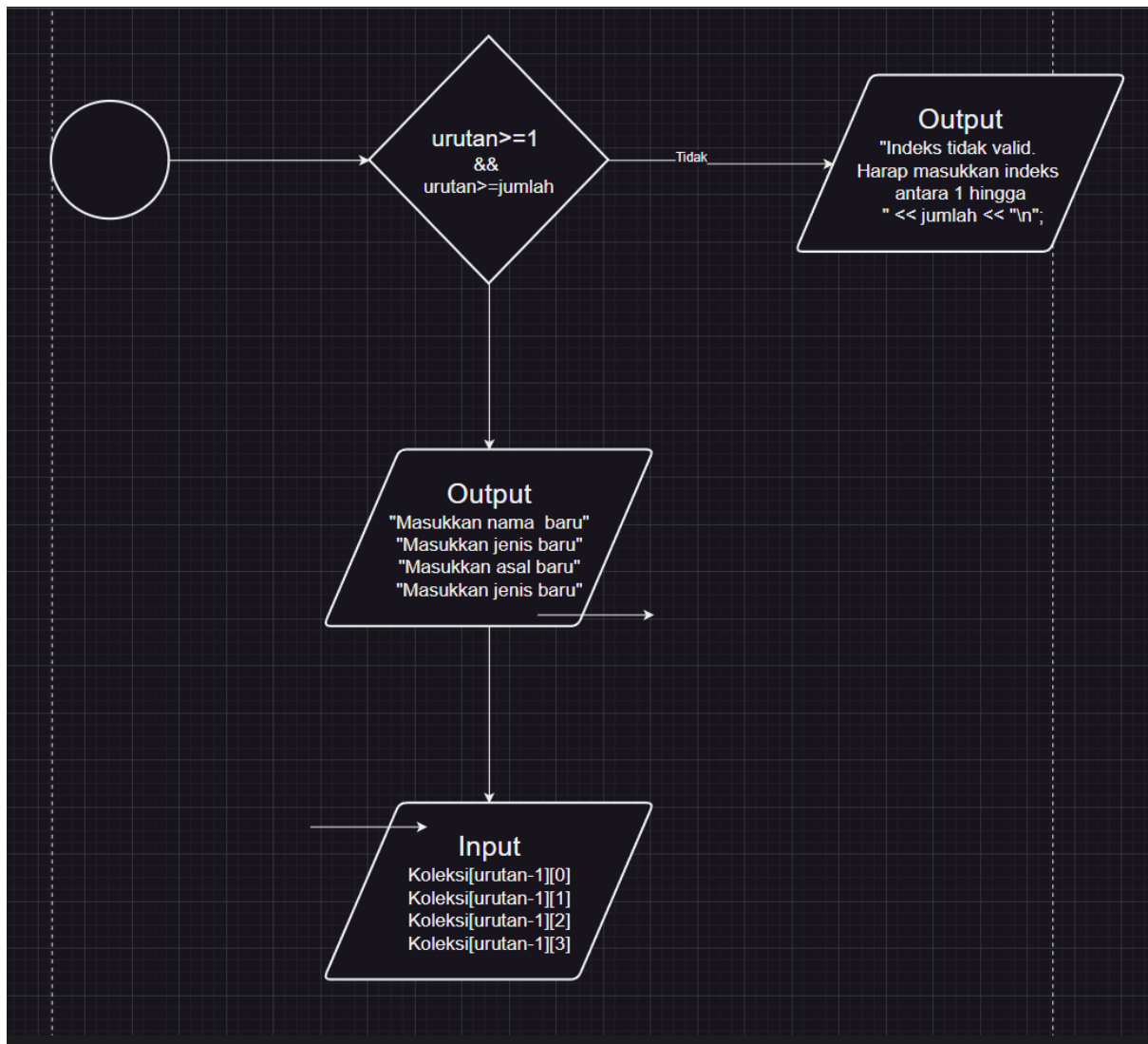
Gambar 1.4 case 1



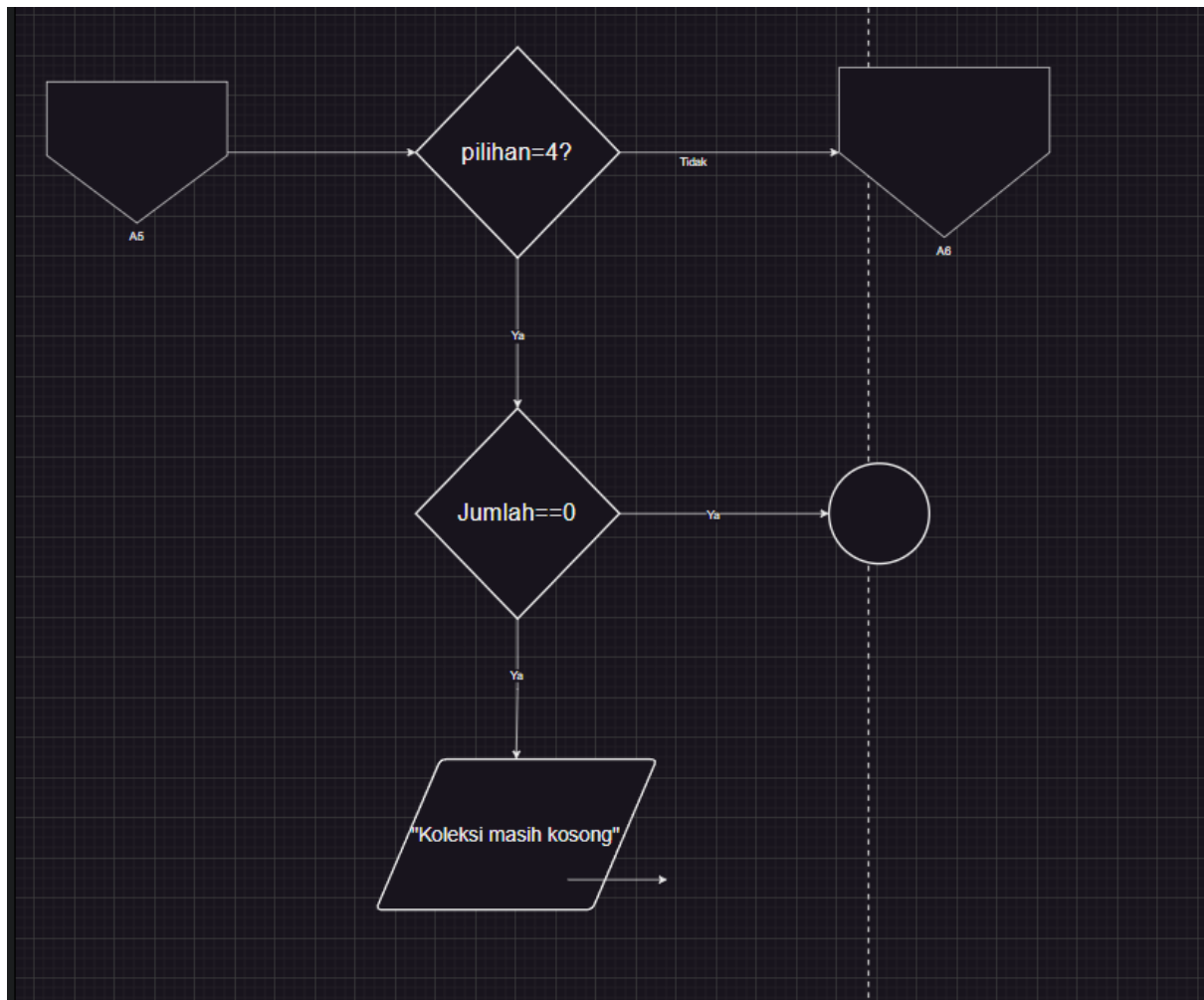
Gambar 1.5 case 2



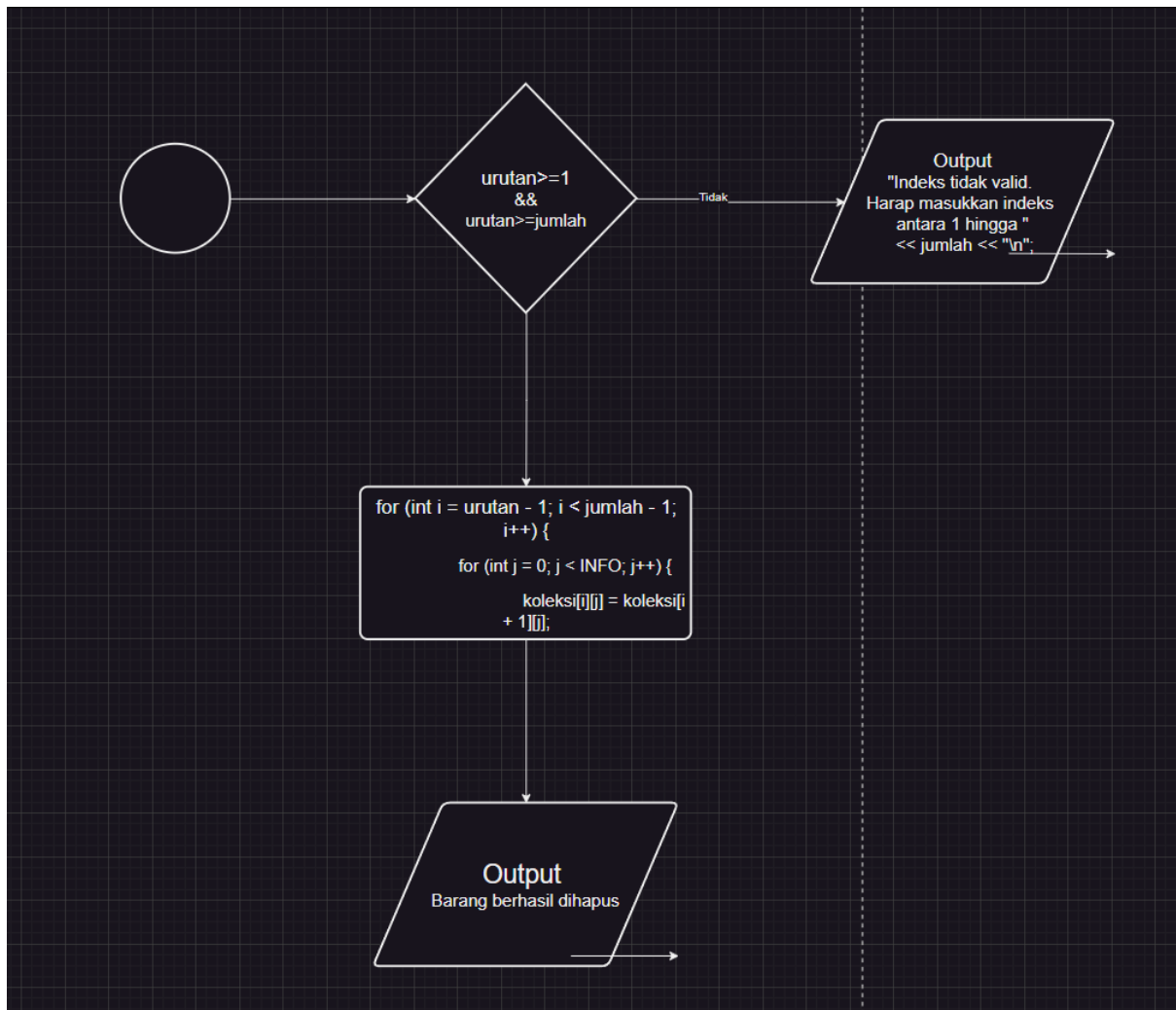
Gambar 1.6 case 3



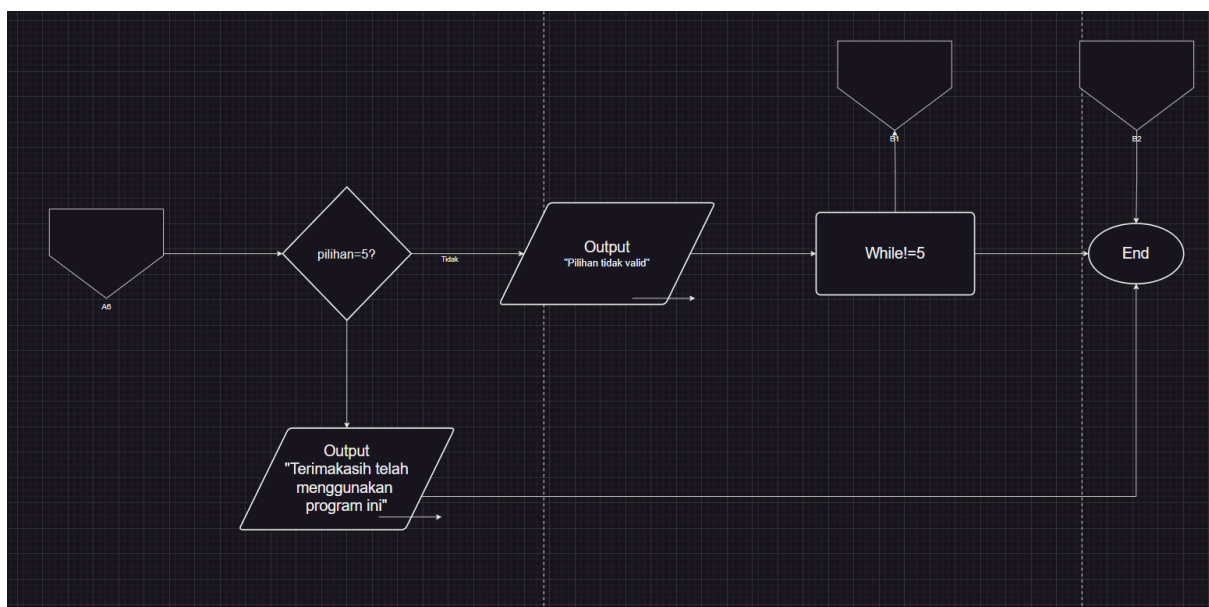
Gambar 1.7 on page case 3



Gambar 1.8 case 4



Gambar 1.9 on page case 4



Gambar 1.10 default to end



## **2. Analisis Program**

### **2.1 Deskripsi Singkat Program**

Program ini adalah manajemen koleksi barang antik berbasis C++, yang memungkinkan pengguna untuk menambah, melihat, memperbarui, dan menghapus data koleksi. Data yang disimpan meliputi nama barang, jenis, asal, dan tahun koleksi. Program juga dilengkapi dengan fitur login untuk keamanan akses.

#### Tujuan Program

1. Mempermudah pencatatan barang antik agar lebih terorganisir.
2. Menyediakan sistem sederhana untuk mengelola koleksi barang antik tanpa perlu menggunakan aplikasi kompleks.
3. Membantu kolektor dalam memantau dan memperbarui koleksinya secara sistematis.

### **2.2 Penjelasan Alur & Algoritma**

#### Alur Kerja Program

1. Autentikasi Pengguna
  - Pengguna harus memasukkan username dan password untuk mengakses program.
  - Program memberikan maksimal tiga kali percobaan login. Jika pengguna gagal dalam tiga percobaan, program akan berhenti.
  - Jika berhasil login, program akan menampilkan menu utama.
2. Menu Utama

Setelah login berhasil, pengguna dapat memilih salah satu dari lima opsi:

- Tambah Koleksi
  - Lihat Koleksi
  - Perbarui Koleksi
  - Hapus Koleksi
  - Keluar
3. Menambahkan Koleksi
    - Jika jumlah koleksi masih di bawah kapasitas maksimum (25 data), pengguna dapat memasukkan nama barang, jenis, asal, dan tahun koleksi.
    - Data akan disimpan dalam array dua dimensi koleksi[MAX][INFO], di mana setiap baris merepresentasikan satu barang antik.
    - Jika jumlah koleksi sudah penuh, program akan memberikan notifikasi.
  4. Melihat Koleksi
    - Program akan menampilkan semua koleksi yang telah dimasukkan dalam format tabel.
    - Jika tidak ada koleksi yang tersimpan, program akan menampilkan pesan bahwa koleksi masih kosong.
  5. Memperbarui Koleksi

- Pengguna diminta untuk memasukkan nomor indeks koleksi yang ingin diperbarui.
  - Jika indeks valid, pengguna dapat memasukkan data baru untuk barang tersebut.
  - Jika indeks tidak valid, program akan memberikan pesan kesalahan.
6. Menghapus Koleksi
- Pengguna diminta untuk memasukkan indeks koleksi yang ingin dihapus.
  - Jika indeks valid, data koleksi akan dihapus dan daftar koleksi akan diperbarui.
  - Jika indeks tidak valid, program akan memberikan pesan kesalahan.
7. Keluar dari Program
- Jika pengguna memilih opsi keluar, program akan menampilkan pesan terima kasih dan berhenti berjalan.

### 3. Source Code

#### A. Fitur Login

```
percobaan = 0;

while (percobaan < 3) {

    cout << "Masukkan username: ";

    cin >> username;

    cout << "Masukkan password: ";

    cin >> password;

    if (username == "Ahnaf" && password == "035") {

        cout << "Berhasil login!\n";

        break;

    } else {

        percobaan++;

        cout << "Gagal login. Username atau password salah. Percobaan ke-" << percobaan << " dari 3.\n";

    }

    if (percobaan == 3) {

        cout << "Anda telah mencoba login 3 kali. Program akan keluar.\n";

        return 0;

    }

}
```

#### B. Fitur Tambah Koleksi

```
if (jumlah < MAX) {

    cout << "Masukkan nama koleksi: ";
```

```

getline(cin, koleksi[jumlah][0]);
cout << "Masukkan jenis koleksi: ";
getline(cin, koleksi[jumlah][1]);
cout << "Masukkan asal koleksi: ";
getline(cin, koleksi[jumlah][2]);
cout << "Masukkan tahun koleksi: ";
getline(cin, koleksi[jumlah][3]);
jumlah++;
} else {
    cout << "Koleksi sudah penuh\n";
}

```

### C. Fitur Melihat Koleksi

```

if (jumlah == 0) {
    cout << "Koleksi masih kosong\n";
} else {
    for (int i = 0; i < jumlah; i++) {
        cout << "| " << setw(2) << i + 1 << " | "
        << setw(25) << left << koleksi[i][0] << " | "
        << setw(20) << left << koleksi[i][1] << " | "
        << setw(20) << left << koleksi[i][2] << " | "
        << setw(20) << left << koleksi[i][3] << " \n";
    }
}

```

### D. Fitur Mengupdate Koleksi

```

if (jumlah == 0) {
    cout << "Koleksi masih kosong\n";
} else {
    cout << "Masukkan indeks koleksi yang ingin diperbarui (1 - " << jumlah << "): ";
    cin >> urutan;
    cin.ignore();
    if (urutan >= 1 && urutan <= jumlah) {
        cout << "Masukkan nama baru: ";
        getline(cin, koleksi[urutan - 1][0]);
        cout << "Masukkan jenis baru: ";
        getline(cin, koleksi[urutan - 1][1]);
        cout << "Masukkan asal baru: ";
        getline(cin, koleksi[urutan - 1][2]);
        cout << "Masukkan tahun baru: ";
        getline(cin, koleksi[urutan - 1][3]);
    } else {
        cout << "Indeks tidak valid. Harap masukkan indeks antara 1 hingga " << jumlah << "\n";
    }
}

```

## E. Fitur Menghapus Koleksi

```
if (jumlah == 0) {
    cout << "Koleksi masih kosong\n";
} else {
    cout << "Masukkan indeks koleksi yang ingin dihapus (1 - " << jumlah << "): ";
    cin >> urutan;
    if (urutan >= 1 && urutan <= jumlah) {
        for (int i = urutan - 1; i < jumlah - 1; i++) {
            for (int j = 0; j < INFO; j++) {
                koleksi[i][j] = koleksi[i + 1][j];
            }
        }
        jumlah--;
        cout << "Barang berhasil dihapus\n";
    } else {
        cout << "Indeks tidak valid. Harap masukkan indeks antara 1 hingga " << jumlah << "\n";
    }
}
```

## 4. Uji Coba dan Hasil Output

### 4.1 Uji Coba

1. Skenario 1 : Salah menginput login sebanyak batas yang di tentukan
2. Skenario 2: Berhasil login tetapi salah menginput pilihan menu
3. Skenario 3: Berhasil login dan melakukan perulangan program
4. Skenario 4: Berhasil login dan tidak melakukan perulangan program

## 4.2 Hasil Output

```
PS C:\Users\Ahnaf GK\OneDrive\ドキュメント\Praktikum-apl\post-test\post-test-apl-2> g++ -o 2409106035-AhnaAliyyu-PT-2.exe 2409106035-AhnaAliyyu-PT-2.cpp
PS C:\Users\Ahnaf GK\OneDrive\ドキュメント\Praktikum-apl\post-test\post-test-apl-2> ./2409106035-AhnaAliyyu-PT-2.exe
Masukkan username: ahnaf
Masukkan password: ganteng
Gagal login. Username atau password salah. Percobaan ke-1 dari 3.
Masukkan username: gagal
Masukkan password: ganteng
Gagal login. Username atau password salah. Percobaan ke-2 dari 3.
Masukkan username: ganteng
Masukkan password: 123
Gagal login. Username atau password salah. Percobaan ke-3 dari 3.
Anda telah mencoba login 3 kali. Program akan keluar.
PS C:\Users\Ahnaf GK\OneDrive\ドキュメント\Praktikum-apl\post-test\post-test-apl-2> |
```

Gambar 4. 1 output skenario 1

Pada Skenario 1 jika kita gagal melakukan input login dengan benar sebanyak 3 kali maka akan menghasilkan output sebagai di atas.

```
PS C:\Users\Ahnaf GK\OneDrive\ドキュメント\Praktikum-apl\post-test\post-test-apl-2> ./2409106035-AhnaAliyyu-PT-2.exe
Masukkan username: Ahnaf
Masukkan password: 035
Berhasil login!

=== Manajemen Koleksi Barang Antik ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
7
Pilihan tidak valid

=== Manajemen Koleksi Barang Antik ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
9
Pilihan tidak valid

=== Manajemen Koleksi Barang Antik ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
|
```

Gambar 4. 2 output skenario 2

Pada scenario 2 kita dihadapkan pada permasalahan salah menginput program menu, dan program akan otoatis mengulang hingga inputnya benar.

```

PS C:\Users\Ahnaf GK\OneDrive\ドキュメント\Praktikum-apl\post-test\post-test-apl-2> ./2409106035-AhnafAliyyu-PT-2.exe
Masukkan username: Ahnaf
Masukkan password: 035
Berhasil login!

=== Manajemen Koleksi Barang Antik ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
7
Pilihan tidak valid

=== Manajemen Koleksi Barang Antik ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
9
Pilihan tidak valid

=== Manajemen Koleksi Barang Antik ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar

```

Gambar 4. 3 output skenario 3

```

Masukkan username: Ahnaf
Masukkan password: 035
Berhasil login!

=== Manajemen Koleksi Barang Antik ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
1
Masukkan nama koleksi: Keris segoro pitu
Masukkan jenis koleksi: Keris
Masukkan asal koleksi: Jawa Timur
Masukkan tahun koleksi: 1987

=== Manajemen Koleksi Barang Antik ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
2
| 1 | Keris segoro pitu      | Keris      | Jawa Timur  | 1987      |
| 1 | Keris segoro pitu      | Keris      | Jawa Timur  | 1987      |

=== Manajemen Koleksi Barang Antik ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
1
Masukkan nama koleksi: Badik
Masukkan jenis koleksi: Badik/senjata
Masukkan asal koleksi: Sulawesi
Masukkan tahun koleksi: 1789

```

Gambar 4.4 output skenario 3(2))

```

=== Manajemen Koleksi Barang Antik ===

=== Manajemen Koleksi Barang Antik ===
=== Manajemen Koleksi Barang Antik ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
2
| 1 | Keris segoro pitu      | Keris      | Jawa Timur      | 1987      |
| 2 | Badik                   | Badik/senjata | Sulawesi        | 1789      |

=== Manajemen Koleksi Barang Antik ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
3
Masukkan indeks koleksi yang ingin diperbarui (1 - 2): 1
Masukkan nama baru: Keris Adiluhung
Masukkan jenis baru: Keris/Senjata
Masukkan asal baru: Jawa Tengah
Masukkan tahun baru: 1765

```

Gambar 4.5 outputs kenario 3(3)

```

=== Manajemen Koleksi Barang Antik ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
2
| 1 | Keris Adiluhung      | Keris/Senjata | Jawa Tengah      | 1765      |
| 2 | Badik                 | Badik/senjata | Sulawesi          | 1789      |

=== Manajemen Koleksi Barang Antik ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
4
Masukkan indeks koleksi yang ingin dihapus (1 - 2): 1
Barang berhasil dihapus

=== Manajemen Koleksi Barang Antik ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
2
| 1 | Badik                 | Badik/senjata | Sulawesi          | 1789      |

=== Manajemen Koleksi Barang Antik ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
5
Terima kasih telah menggunakan program ini
PS C:\Users\Ahnaf GK\OneDrive\ドキュメント\Praktikum-apl\post-test\post-test-apl-2>

```

Gambar 4.6 output kenario 3(4)

Pada skenario 3 Kita akan melakukan input secara benar terus-menerus dan melakukan perulangan sehingga menghasilkan beragam output.



```
Terima kasih telah menggunakan program ini
PS C:\Users\Ahnaf_GK\OneDrive\ドキュメント\Praktikum-apl\post-test\post-test-apl-2> ./2409106035-AhnafAliyyu-PT-2.exe
Masukkan username: Ahnaf
Masukkan password: 035
Berhasil login!

=== Manajemen Koleksi Barang Antik ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
1
Masukkan nama koleksi: adiluhur
Masukkan jenis koleksi: keris
Masukkan asal koleksi: jawa
Masukkan tahun koleksi: 1563

=== Manajemen Koleksi Barang Antik ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
2
| 1 | adiluhur          | keris          | jawa          | 1563          |
=== Manajemen Koleksi Barang Antik ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
5
Terima kasih telah menggunakan program ini
PS C:\Users\Ahnaf_GK\OneDrive\ドキュメント\Praktikum-apl\post-test\post-test-apl-2>
```

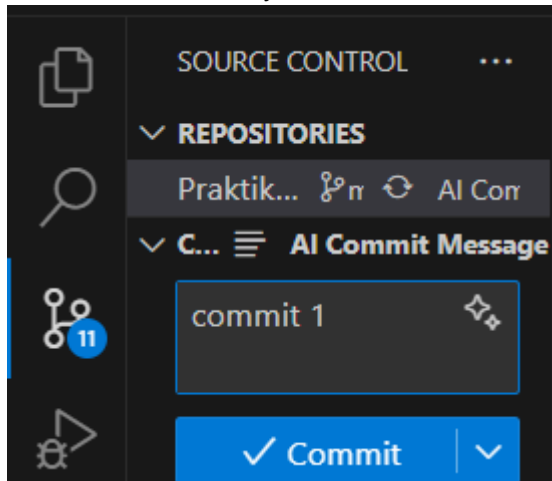
Gambar 4.7 output kenario 4

Dan yang terakhir adalah menjalankan program tanpa melakukan perulangan pada setiap menu.

## 5. Git

Pada bagian penjelasan git kali ini kita tidak akan terlalu berpanjang lebar karena pada posttest sebelumnya kita telah mensinkronisasikan akun github dan folder atau repository kita melalui git yang ada pada vs code

Untuk melakukan commit dan push pada kali ini cukup dengan mencari menu “source control” pada visual studio code kita dan melakukan commit beserta pesan commitnya, setelah melakukan commit maka lakukanlah sync atau sinkronisasi terhadap perubahan folder dan akun github. Jadi kita melakukan urutan dengan menambah file—source control—commit—sync/.



Gambar 5.1 GIT