

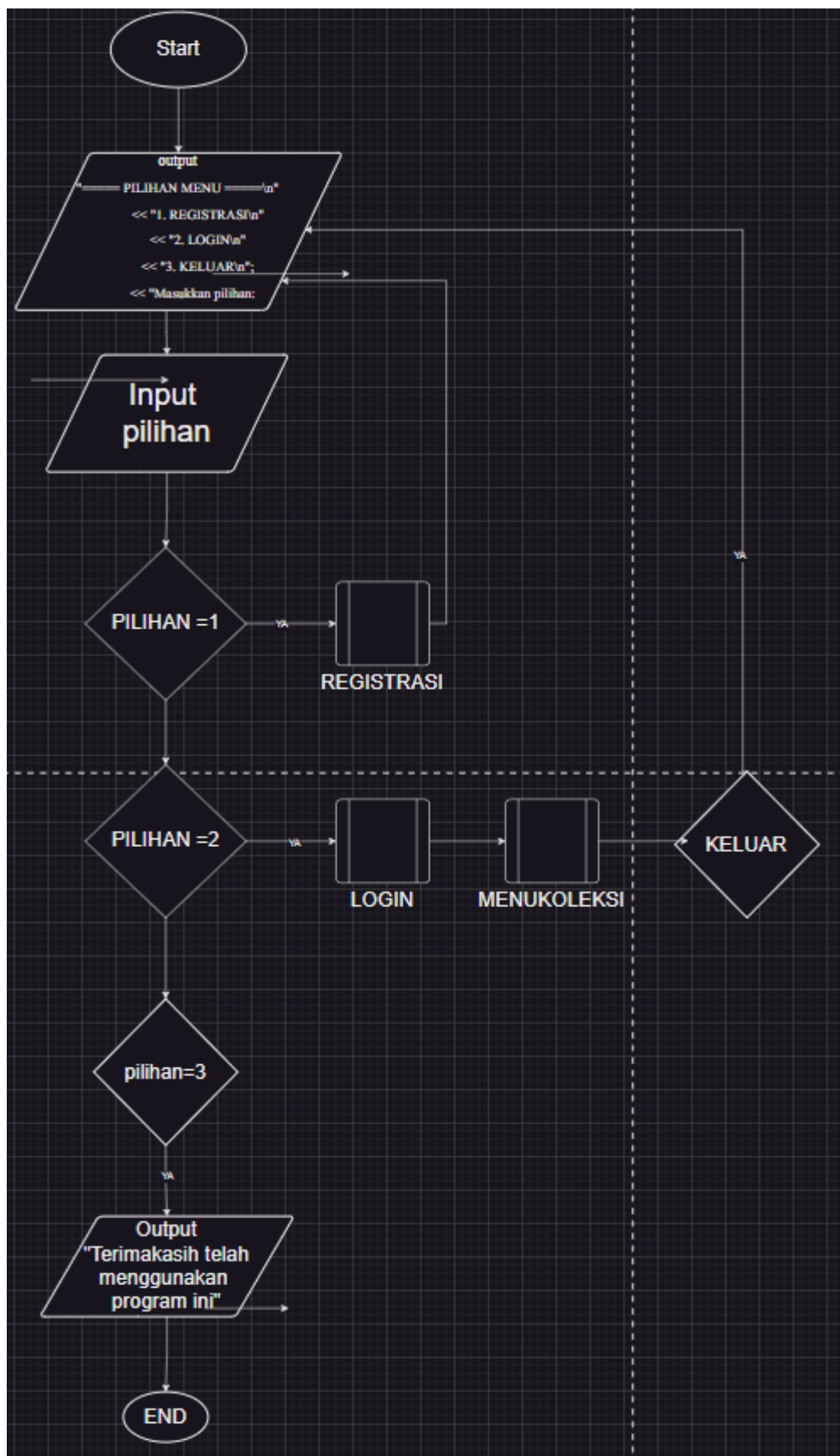
LAPORAN PRAKTIKUM
POSTTEST (VI)
ALGORITMA PEMROGRAMAN LANJUT



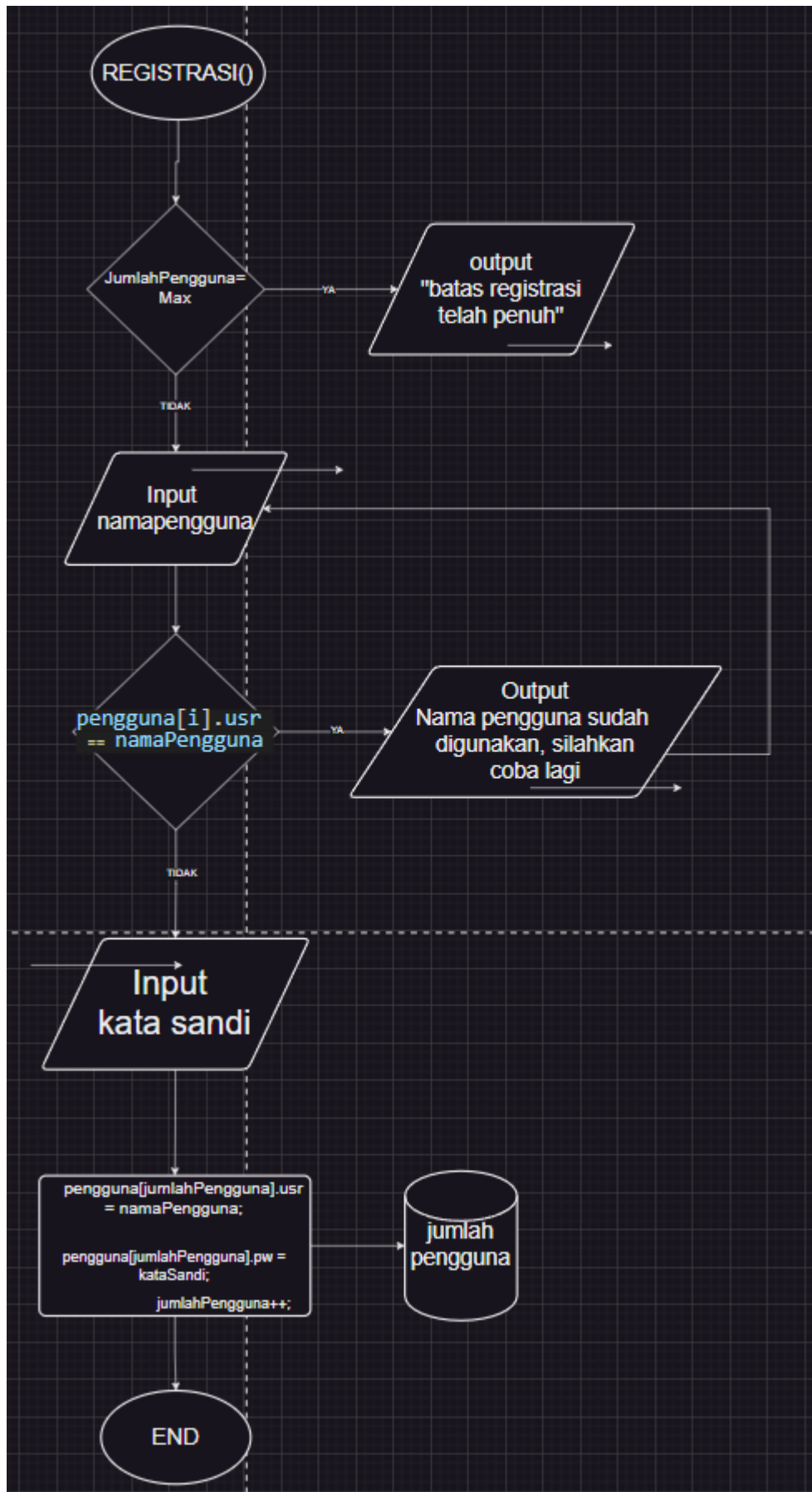
Disusun oleh:
Ahnaf Aliyyu (2409106035)
Kelas (A2 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

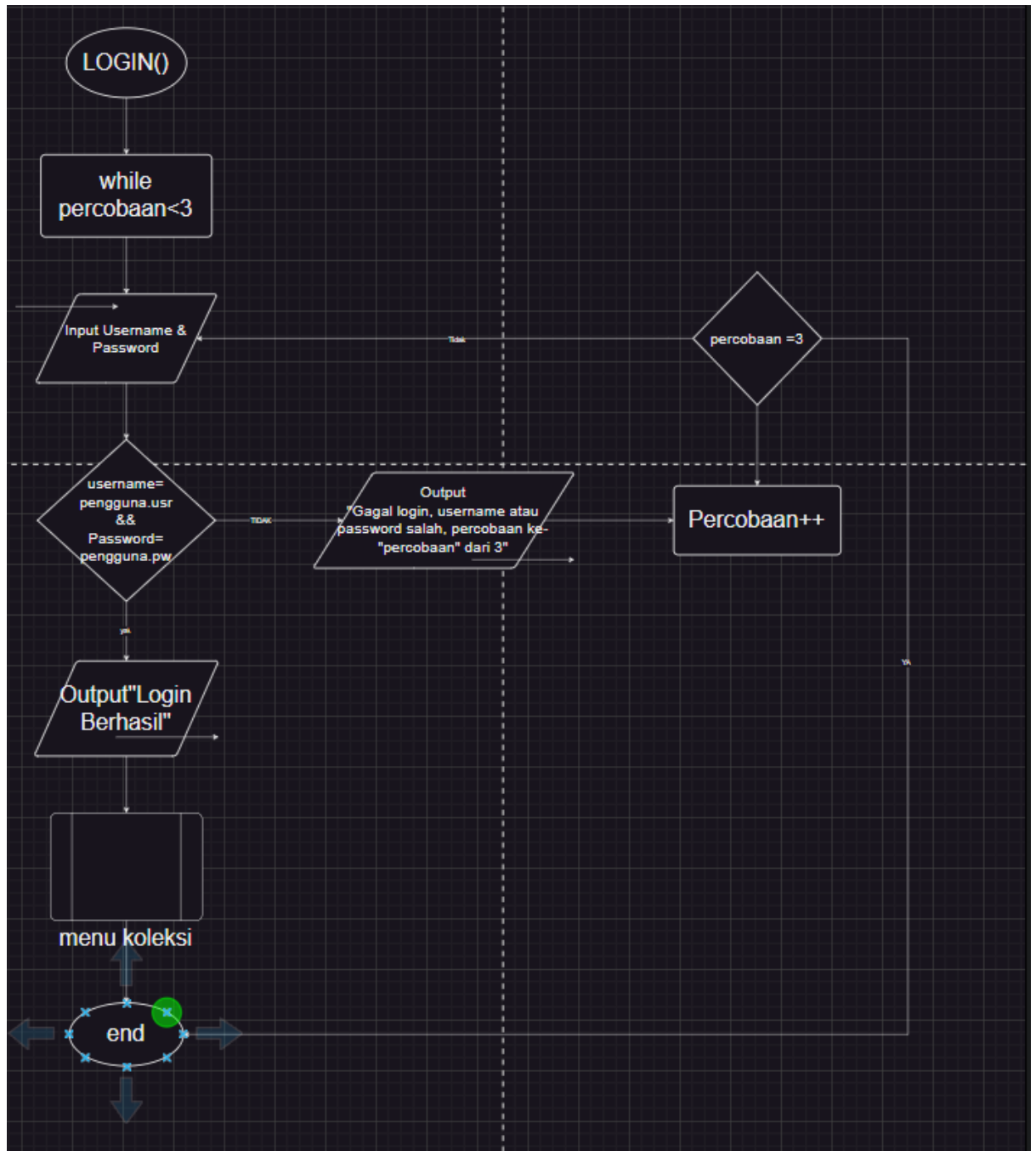
1. Flowchart



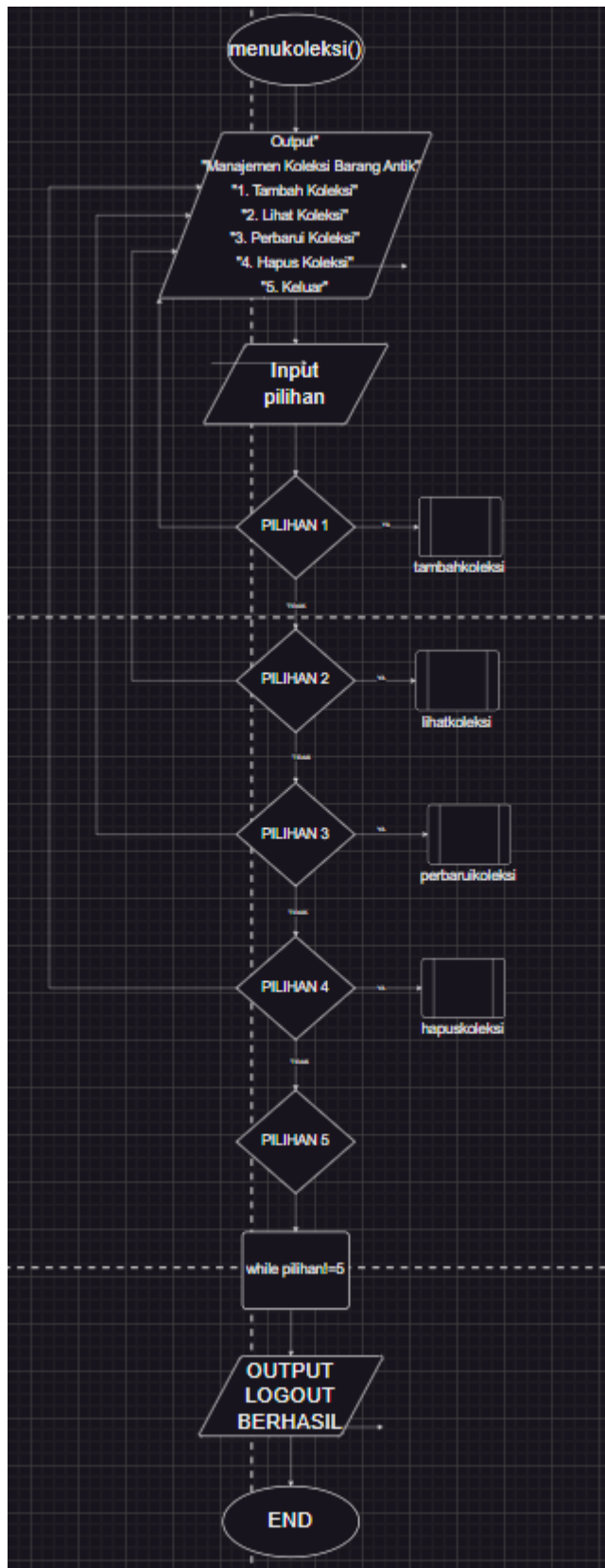
Gambar 1.1 MAIN FLOWCHART



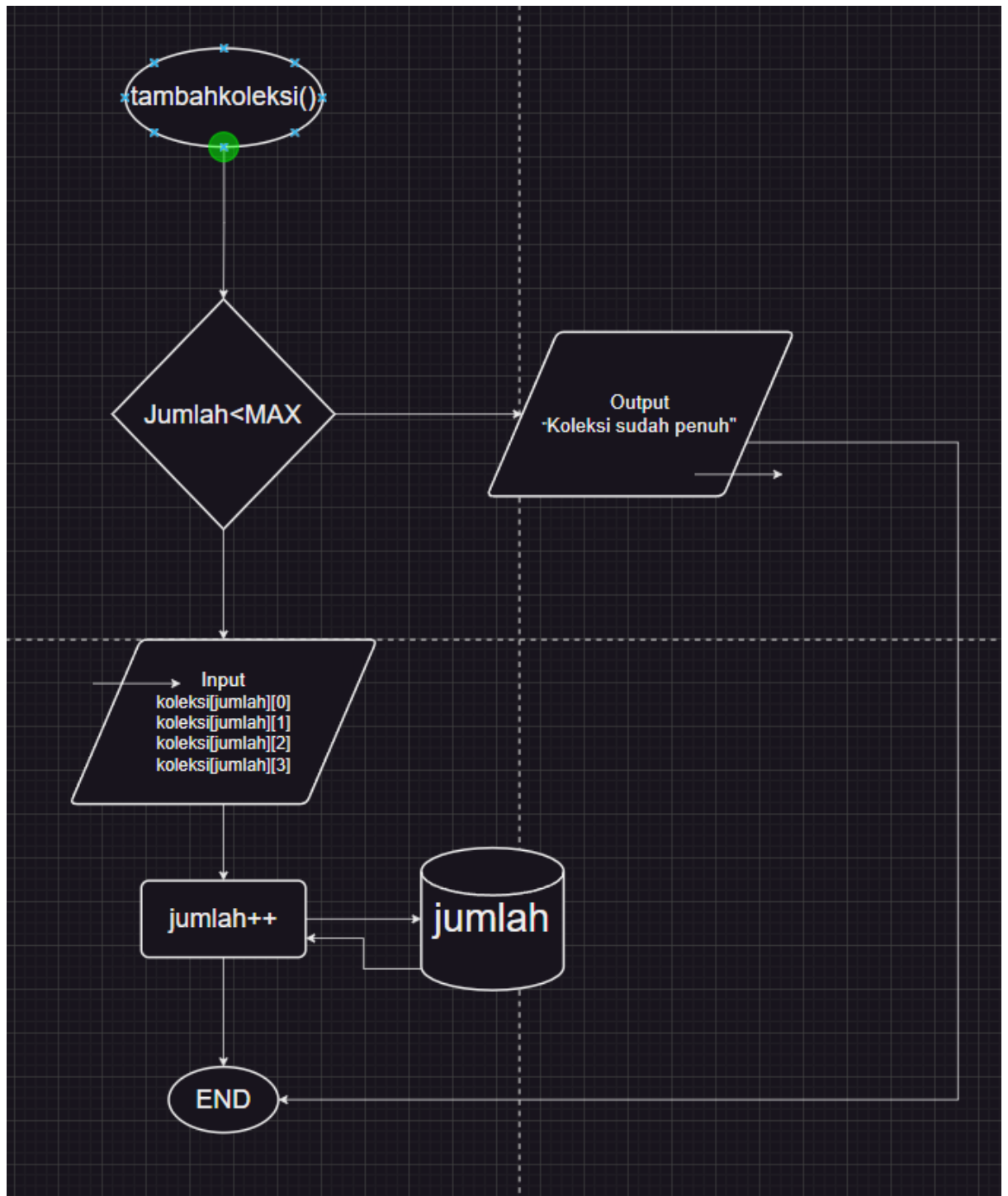
Gambar 1.2 registrasi



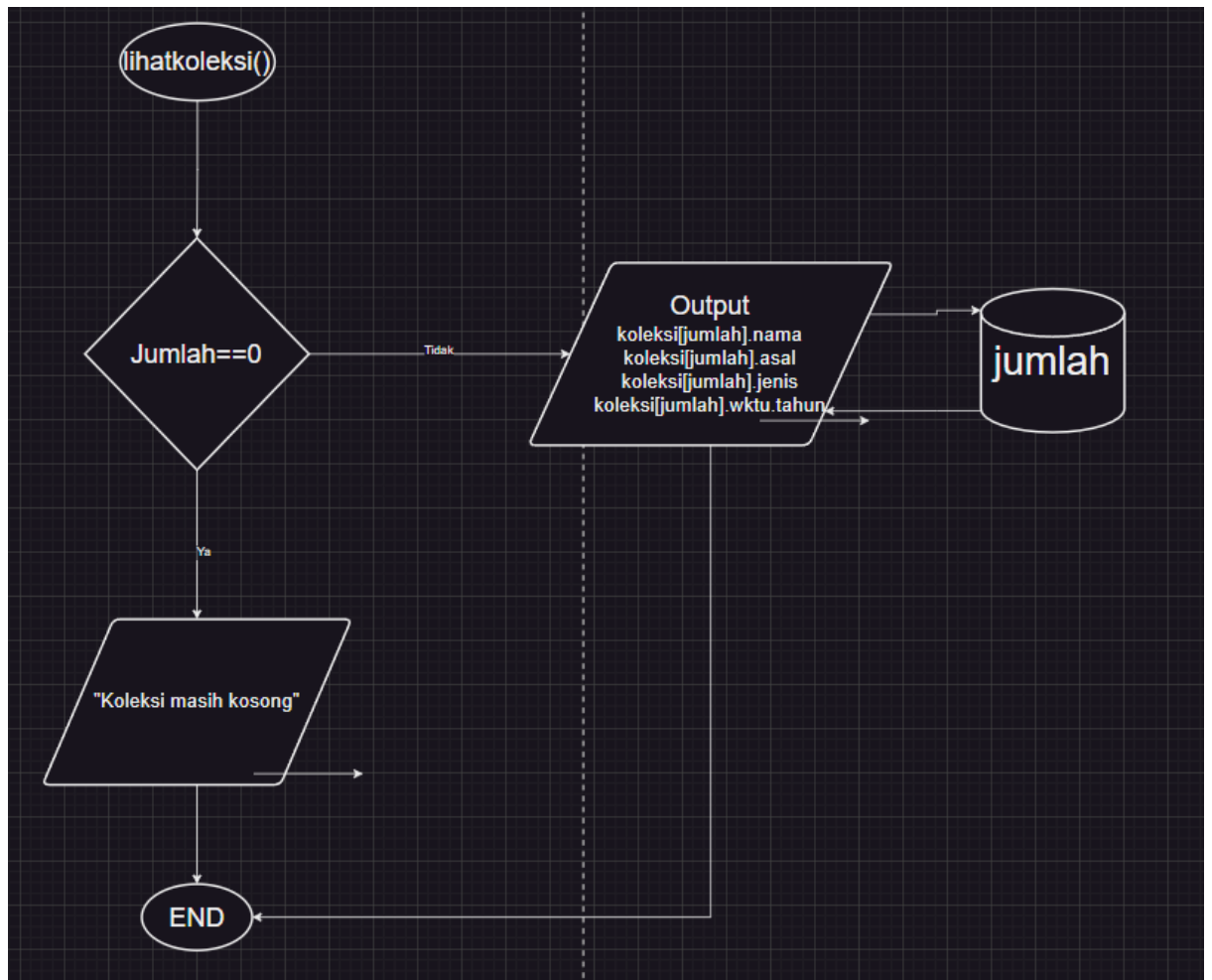
Gambar 1.3 Login



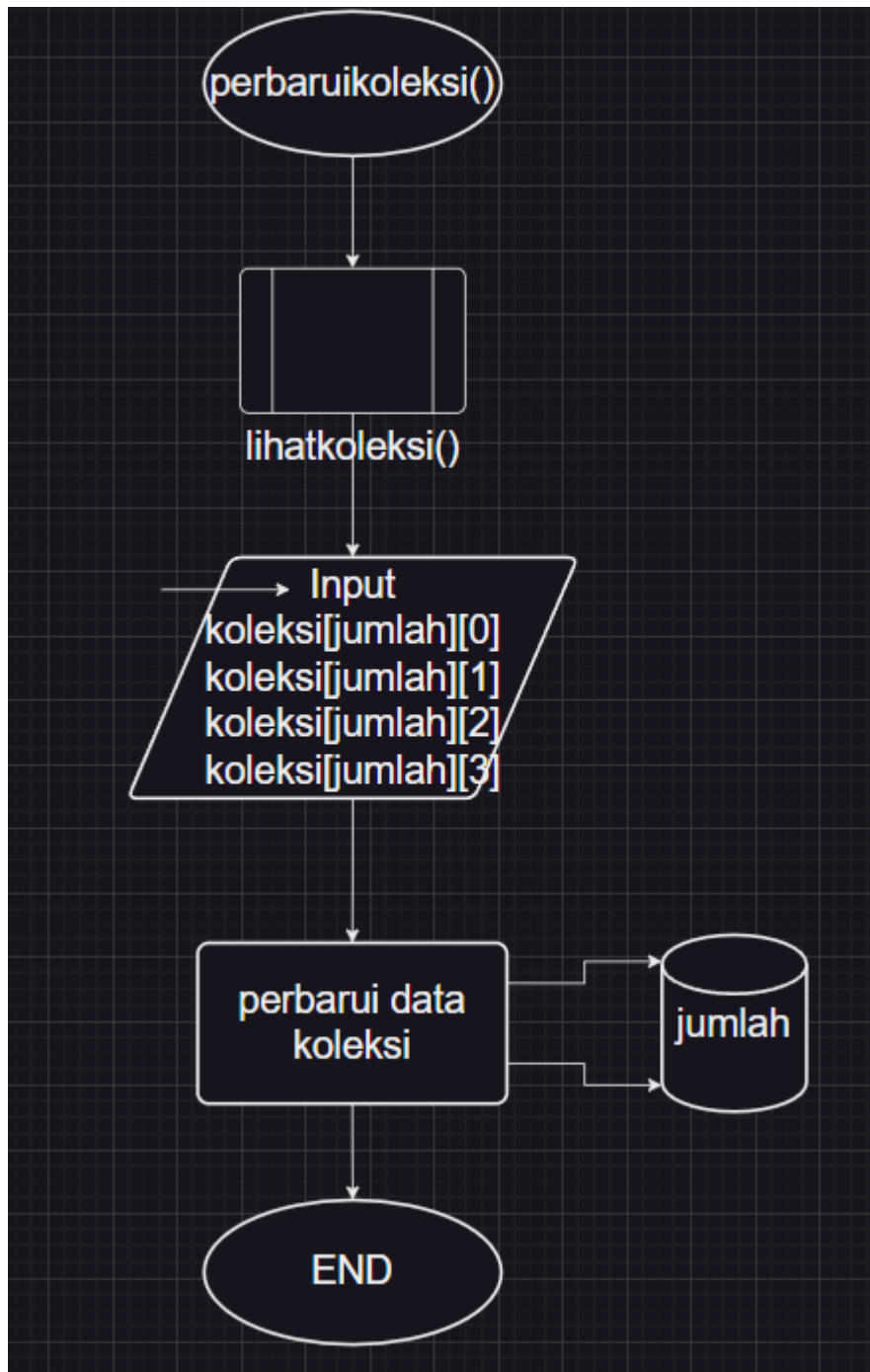
Gambar 1.4 menukoleksi



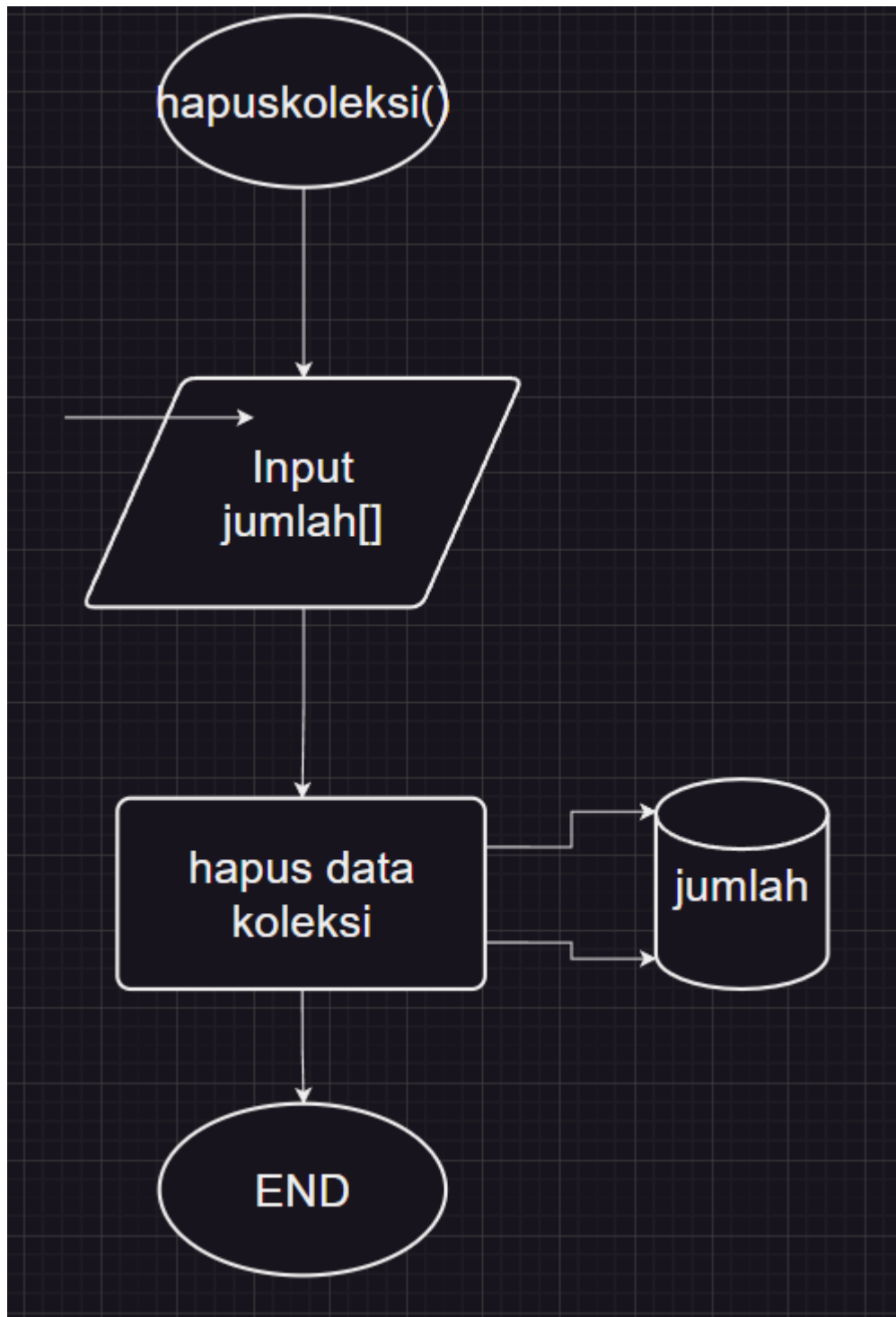
Gambar 1.5 tambahkoleksi



Gambar 1.6 lihatkoleksi



Gambar 1.7 perbaruikoleksi



Gambar 1.8 hapuskoleksi

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.

3. Source Code

A. Fitur Regis & Login

```
void registrasi(regis* pengguna, int* jumlahPengguna) {
    string namaPengguna, kataSandi;
    bool penggunaAda = false;

    cout << "==== REGISTRASI ==== \n";
    cout << "Masukkan nama pengguna: ";
    cin >> namaPengguna;

    for (int i = 0; i < *jumlahPengguna; i++) {
        if ((pengguna + i)->usr == namaPengguna) {
            penggunaAda = true;
            break;
        }
    }

    if (penggunaAda) {
        cout << "Nama pengguna sudah digunakan.\n";
    } else {
        cout << "Masukkan kata sandi: ";
        cin >> kataSandi;

        (pengguna + *jumlahPengguna)->usr = namaPengguna;
        (pengguna + *jumlahPengguna)->pw = kataSandi;
        (*jumlahPengguna)++;
    }
}
```

```

        cout << "Registrasi berhasil!\n";
    }
}

bool login(regis* pengguna, int jumlahPengguna, int* indeksPenggunaLogin) {
    string namaPengguna, kataSandi;
    bool berhasil = false;
    percobaan = 0;

    while (percobaan < 3 && !berhasil) {
        cout << "Masukkan nama pengguna: ";
        cin >> namaPengguna;
        cout << "Masukkan kata sandi: ";
        cin >> kataSandi;

        for (int i = 0; i < jumlahPengguna; i++) {
            if ((pengguna + i)->usr == namaPengguna && (pengguna + i)->pw ==
kataSandi) {
                berhasil = true;
                *indeksPenggunaLogin = i;
                cout << "Login berhasil!\n";
                break;
            }
        }

        if (!berhasil) {
            percobaan++;
            cout << "Gagal login. Percobaan ke-" << percobaan << " dari 3.\n";
        }

        if (percobaan == 3 && !berhasil) {
            cout << "Anda telah mencoba login 3 kali.\n";
        }
    }

    return berhasil;
}
}

```

B. Fitur Tambah Koleksi

```

void tambahKoleksi(rincian* koleksi, int* jumlahKoleksi) {
    if (*jumlahKoleksi < MAX) {
        cout << "Masukkan nama koleksi: ";
        cin.ignore();
        getline(cin, (koleksi + *jumlahKoleksi)->nama);
        cout << "Masukkan jenis koleksi: ";
        getline(cin, (koleksi + *jumlahKoleksi)->jenis);
        cout << "Masukkan asal koleksi: ";
        getline(cin, (koleksi + *jumlahKoleksi)->asal);
    }
}

```

```

        cout << "Masukkan tahun koleksi: ";
        getline(cin, (koleksi + *jumlahKoleksi)->wktu.tahun);
        (*jumlahKoleksi)++;
    } else {
        cout << "Koleksi penuh.\n";
    }
}
}

```

C. Fitur Melihat Koleksi

```

void tampilkanKoleksiRekursif(rincian* koleksi, int index, int jumlah) {
    if (index >= jumlah) {
        cout << setfill('=') << setw(83) << "=" << endl;
        return;
    }

    cout << "| " << setw(2) << right << index + 1 << " | "
        << setw(25) << left << (koleksi + index)->nama << "| "
        << setw(20) << left << (koleksi + index)->asal << "| "
        << setw(20) << left << (koleksi + index)->jenis << "| "
        << setw(6) << (koleksi + index)->wktu.tahun << " |\n";

    tampilkanKoleksiRekursif(koleksi, index + 1, jumlah);
}

void lihatKoleksi(rincian* koleksi, int jumlah) {
    if (jumlah == 0) {
        cout << "Koleksi kosong\n";
    } else {
        cout << setw(70) << "===== Koleksi Barang Antik =====\n";
        cout << setfill('=') << setw(83) << "=" << endl;
        cout << "| No | " << setw(25) << left << "Nama"
            << "| " << setw(20) << left << "Asal"
            << "| " << setw(20) << left << "Jenis"
            << "| Tahun |\n";
        cout << setfill('=') << setw(83) << "=" << endl;
        cout << setfill(' ');
        tampilkanKoleksiRekursif(koleksi, 0, jumlah);
    }
}

```

D. Fitur Mengupdate Koleksi

```

void perbaruiKoleksi(rincian* koleksi, int jumlah) {
    if (jumlah == 0) {
        cout << "Koleksi kosong\n";
        return;
    }

    lihatKoleksi(koleksi, jumlah);
}

```

```

int indeks;
cout << "Masukkan indeks koleksi yang ingin diperbarui: ";
cin >> indeks;
cin.ignore();

if (indeks >= 1 && indeks <= jumlah) {
    cout << "Masukkan nama baru: ";
    getline(cin, (koleksi + indeks - 1)->nama);
    cout << "Masukkan jenis baru: ";
    getline(cin, (koleksi + indeks - 1)->jenis);
    cout << "Masukkan asal baru: ";
    getline(cin, (koleksi + indeks - 1)->asal);
    cout << "Masukkan tahun baru: ";
    getline(cin, (koleksi + indeks - 1)->wktu.tahun);
    cout << "Data berhasil diperbarui.\n";
} else {
    cout << "Indeks tidak valid.\n";
}
}

```

E. Fitur Menghapus Koleksi

```

void hapusKoleksi(rincian* koleksi, int* jumlah) {
    if (*jumlah == 0) {
        cout << "Koleksi kosong\n";
        return;
    }

    int indeks;
    cout << "Masukkan indeks koleksi yang ingin dihapus: ";
    cin >> indeks;

    if (indeks >= 1 && indeks <= *jumlah) {
        for (int i = indeks - 1; i < *jumlah - 1; i++) {
            *(koleksi + i) = *(koleksi + i + 1);
        }
        (*jumlah)--;
        cout << "Data berhasil dihapus.\n";
    } else {
        cout << "Indeks tidak valid.\n";
    }
}

```

F. Program sorting

```
// Sorting berdasarkan Nama (Descending) menggunakan Bubble Sort
void sortNamaDescending(rincian* koleksi, int jumlah) {
    for (int i = 0; i < jumlah - 1; i++) {
        for (int j = i + 1; j < jumlah; j++) {
            if ((koleksi + i)->nama < (koleksi + j)->nama) {
                swap(*(koleksi + i), *(koleksi + j));
            }
        }
    }
    cout << "Sorting berdasarkan Nama (Descending) menggunakan Bubble Sort\n";
}

// Sorting berdasarkan Jenis (Descending) menggunakan Selection Sort
void sortJenisDescending(rincian* koleksi, int jumlah) {
    for (int i = 0; i < jumlah - 1; i++) {
        int maxx = i;
        for (int j = i + 1; j < jumlah; j++) {
            if ((koleksi + j)->jenis > (koleksi + maxx)->jenis) {
                maxx = j;
            }
        }
        if (maxx != i) {
            swap(*(koleksi + i), *(koleksi + maxx));
        }
    }
    cout << "Sorting berdasarkan Jenis (Descending) menggunakan Selection Sort\n";
}

// QuickSort berdasarkan Tahun (Ascending)
void quickSortTahunAscending(rincian* koleksi, int low, int high) {
    if (low < high) {
        int pivot = koleksi[high].wktu.tahun;
        int i = (low - 1);

        for (int j = low; j < high; j++) {
            if (koleksi[j].wktu.tahun <= pivot) {
                i++;
                swap(koleksi[i], koleksi[j]);
            }
        }
        swap(koleksi[i + 1], koleksi[high]);

        int pi = i + 1;
        quickSortTahunAscending(koleksi, low, pi - 1);
        quickSortTahunAscending(koleksi, pi + 1, high);
    }
}
```

```
    }  
}  
  
// Fungsi untuk melihat koleksi berdasarkan Tahun (Ascending)  
void lihatKoleksiBerdasarkanTahun(rincian* koleksi, int jumlah) {  
    quickSortTahunAscending(koleksi, 0, jumlah - 1);  
    lihatKoleksi(koleksi, jumlah);  
}
```

4. Uji Coba dan Hasil Output

4.2 Hasil Output

```

Login berhasil!

=== Manajemen Koleksi ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
Masukkan pilihan: 1
Masukkan nama koleksi: melki
Masukkan jenis koleksi: batak
Masukkan asal koleksi: sumtera
Masukkan tahun koleksi: 2003

=== Manajemen Koleksi ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
Masukkan pilihan: 1
Masukkan nama koleksi: rusdi
Masukkan jenis koleksi: dayak
Masukkan asal koleksi: kaltim
Masukkan tahun koleksi: 2006

=== Manajemen Koleksi ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
Masukkan pilihan: 2

=== Lihat Koleksi ===
1. Tampilkan Koleksi Asli
2. Tampilkan Urut Nama (Descending)
3. Tampilkan Urut Jenis (Descending)
4. Tampilkan Urut Tahun (Ascending)
5. Kembali
Masukkan pilihan: 1

===== Koleksi Barang Antik =====
=====
| No | Nama=====| Asal=====| Jenis=====| Tahun |
=====
| 1 | melki          | sumtera    | batak      | 2003 |
| 2 | rusdi          | kaltim     | dayak      | 2006 |
=====

=== Manajemen Koleksi ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
Masukkan pilihan: 2

=== Lihat Koleksi ===
1. Tampilkan Koleksi Asli
2. Tampilkan Urut Nama (Descending)
3. Tampilkan Urut Jenis (Descending)
4. Tampilkan Urut Tahun (Ascending)
5. Kembali

```

gambar 4.1 Hasil program 1


```

Masukkan pilihan: 3
Sorting berdasarkan Jenis (Descending) menggunakan Selection Sort
===== Koleksi Barang Antik =====
=====
| No | Nama=====| Asal=====| Jenis=====| Tahun |
=====
| 1 | rusdi          | kaltim      | dayak      | 2006 |
| 2 | melki          | sumtera     | batak      | 2003 |
=====

=== Manajemen Koleksi ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
Masukkan pilihan: 2

=== Lihat Koleksi ===
1. Tampilkan Koleksi Asli
2. Tampilkan Urut Nama (Descending)
3. Tampilkan Urut Jenis (Descending)
4. Tampilkan Urut Tahun (Ascending)
5. Kembali
Masukkan pilihan: 2
Sorting berdasarkan Nama (Descending) menggunakan Bubble Sort
===== Koleksi Barang Antik =====
=====
| No | Nama=====| Asal=====| Jenis=====| Tahun |
=====
| 1 | rusdi          | kaltim      | dayak      | 2006 |
| 2 | melki          | sumtera     | batak      | 2003 |
=====

=== Manajemen Koleksi ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
Masukkan pilihan: 2

=== Lihat Koleksi ===
1. Tampilkan Koleksi Asli
2. Tampilkan Urut Nama (Descending)
3. Tampilkan Urut Jenis (Descending)
4. Tampilkan Urut Tahun (Ascending)
5. Kembali
Masukkan pilihan: 4
===== Koleksi Barang Antik =====
=====
| No | Nama=====| Asal=====| Jenis=====| Tahun |
=====
| 1 | melki          | sumtera     | batak      | 2003 |
| 2 | rusdi          | kaltim      | dayak      | 2006 |
=====

=== Manajemen Koleksi ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi

```

gambar 4.2 Hasil program 2

```

=== Manajemen Koleksi ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
Masukkan pilihan: 3
===== Koleksi Barang Antik =====
=====
| No | Nama=====| Asal=====| Jenis=====| Tahun |
=====
| 1 | melki          | sumtera      | batak        | 2003 |
| 2 | rusdi          | kaltim       | dayak        | 2006 |
=====

Masukkan indeks koleksi yang ingin diperbarui: 1
Masukkan nama baru: rasyid
Masukkan jenis baru: banjar
Masukkan asal baru: kalbar
Masukkan tahun baru: 2005
Data berhasil diperbarui.

=== Manajemen Koleksi ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
Masukkan pilihan: 2

=== Lihat Koleksi ===
1. Tampilkan Koleksi Asli
2. Tampilkan Urut Nama (Descending)
3. Tampilkan Urut Jenis (Descending)
4. Tampilkan Urut Tahun (Ascending)
5. Kembali
Masukkan pilihan: 1
===== Koleksi Barang Antik =====
=====
| No | Nama=====| Asal=====| Jenis=====| Tahun |
=====
| 1 | rasyid        | kalbar       | banjar       | 2005 |
| 2 | rusdi         | kaltim       | dayak        | 2006 |
=====

=== Manajemen Koleksi ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
Masukkan pilihan: 2

=== Lihat Koleksi ===
1. Tampilkan Koleksi Asli
2. Tampilkan Urut Nama (Descending)
3. Tampilkan Urut Jenis (Descending)
4. Tampilkan Urut Tahun (Ascending)
5. Kembali
Masukkan pilihan: 2

```

gambar 4.3 Hasil program

```

Masukkan pilihan: 2
Sorting berdasarkan Nama (Descending) menggunakan Bubble Sort
===== Koleksi Barang Antik =====
=====
| No | Nama=====| Asal=====| Jenis=====| Tahun |
=====
| 1 | rusdi          | kaltim      | dayak      | 2006 |
| 2 | rasyid         | kalbar      | banjar     | 2005 |
=====

=== Manajemen Koleksi ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
Masukkan pilihan: 2

=== Lihat Koleksi ===
1. Tampilkan Koleksi Asli
2. Tampilkan Urut Nama (Descending)
3. Tampilkan Urut Jenis (Descending)
4. Tampilkan Urut Tahun (Ascending)
5. Kembali
Masukkan pilihan: 3
Sorting berdasarkan Jenis (Descending) menggunakan Selection Sort
===== Koleksi Barang Antik =====
=====
| No | Nama=====| Asal=====| Jenis=====| Tahun |
=====
| 1 | rusdi          | kaltim      | dayak      | 2006 |
| 2 | rasyid         | kalbar      | banjar     | 2005 |
=====

=== Manajemen Koleksi ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
Masukkan pilihan: 2

=== Lihat Koleksi ===
1. Tampilkan Koleksi Asli
2. Tampilkan Urut Nama (Descending)
3. Tampilkan Urut Jenis (Descending)
4. Tampilkan Urut Tahun (Ascending)
5. Kembali
Masukkan pilihan: 4
===== Koleksi Barang Antik =====
=====
| No | Nama=====| Asal=====| Jenis=====| Tahun |
=====
| 1 | rasyid         | kalbar      | banjar     | 2005 |
| 2 | rusdi          | kaltim      | dayak      | 2006 |
=====

=== Manajemen Koleksi ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
Masukkan pilihan:

```

gambar 4.4 Hasil program

```

=== Lihat Koleksi ===
1. Tampilkan Koleksi Asli
2. Tampilkan Urut Nama (Descending)
3. Tampilkan Urut Jenis (Descending)
4. Tampilkan Urut Tahun (Ascending)
5. Kembali
Masukkan pilihan: 4
===== Koleksi Barang Antik =====
=====
| No | Nama=====| Asal=====| Jenis=====| Tahun |
=====
| 1 | rasyid          | kalbar          | banjar          | 2005 |
| 2 | rusdi           | kaltim          | dayak           | 2006 |
=====

=== Manajemen Koleksi ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
Masukkan pilihan: 4
Masukkan indeks koleksi yang ingin dihapus: 1
Data berhasil dihapus.

=== Manajemen Koleksi ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
Masukkan pilihan: 2

=== Lihat Koleksi ===
1. Tampilkan Koleksi Asli
2. Tampilkan Urut Nama (Descending)
3. Tampilkan Urut Jenis (Descending)
4. Tampilkan Urut Tahun (Ascending)
5. Kembali
Masukkan pilihan: 2
Sorting berdasarkan Nama (Descending) menggunakan Bubble Sort
===== Koleksi Barang Antik =====
=====
| No | Nama=====| Asal=====| Jenis=====| Tahun |
=====
| 1 | rusdi           | kaltim          | dayak           | 2006 |
=====

=== Manajemen Koleksi ===
1. Tambah Koleksi
2. Lihat Koleksi
3. Perbarui Koleksi
4. Hapus Koleksi
5. Keluar
Masukkan pilihan: 

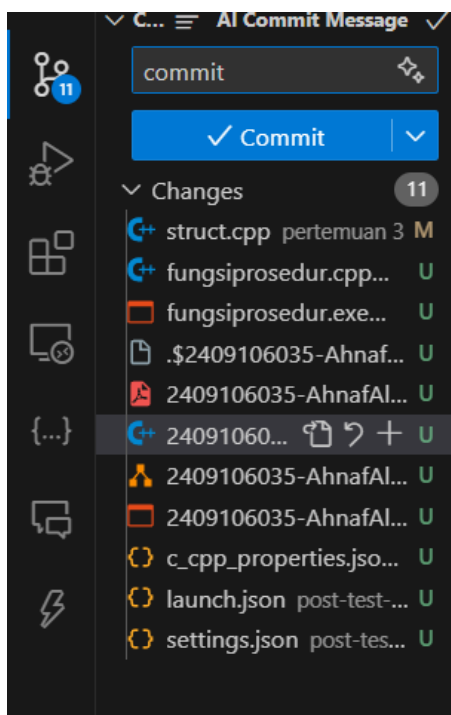
```

gambar 4.5 Hasil program

5. Git

Pada bagian penjelasan git kali ini kita tidak akan terlalu berpanjang lebar karena pada posttest sebelumnya kita telah mensinkronisaasikan 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

Selain melalui source control juga dapat menggunakan command git yang ada di vs code

- Git add
Menggunakan git add . untuk menginisiasi semua file

- `Git commit`
Menyimpan perubahan dengan pesan commit.
- `git push`
mengunggah (push) kode dari repository lokal ke repository remote (GitHub) dan menetapkan branch default.