LAPORAN PRAKTIKUM

POSTTEST 3

ALGORITMA PEMROGRAMAN LANJUT

Disusun oleh:

Elfin Sinaga (2409106024)

Kelas (A2 ‘24)

PROGRAM STUDI INFORMATIKA

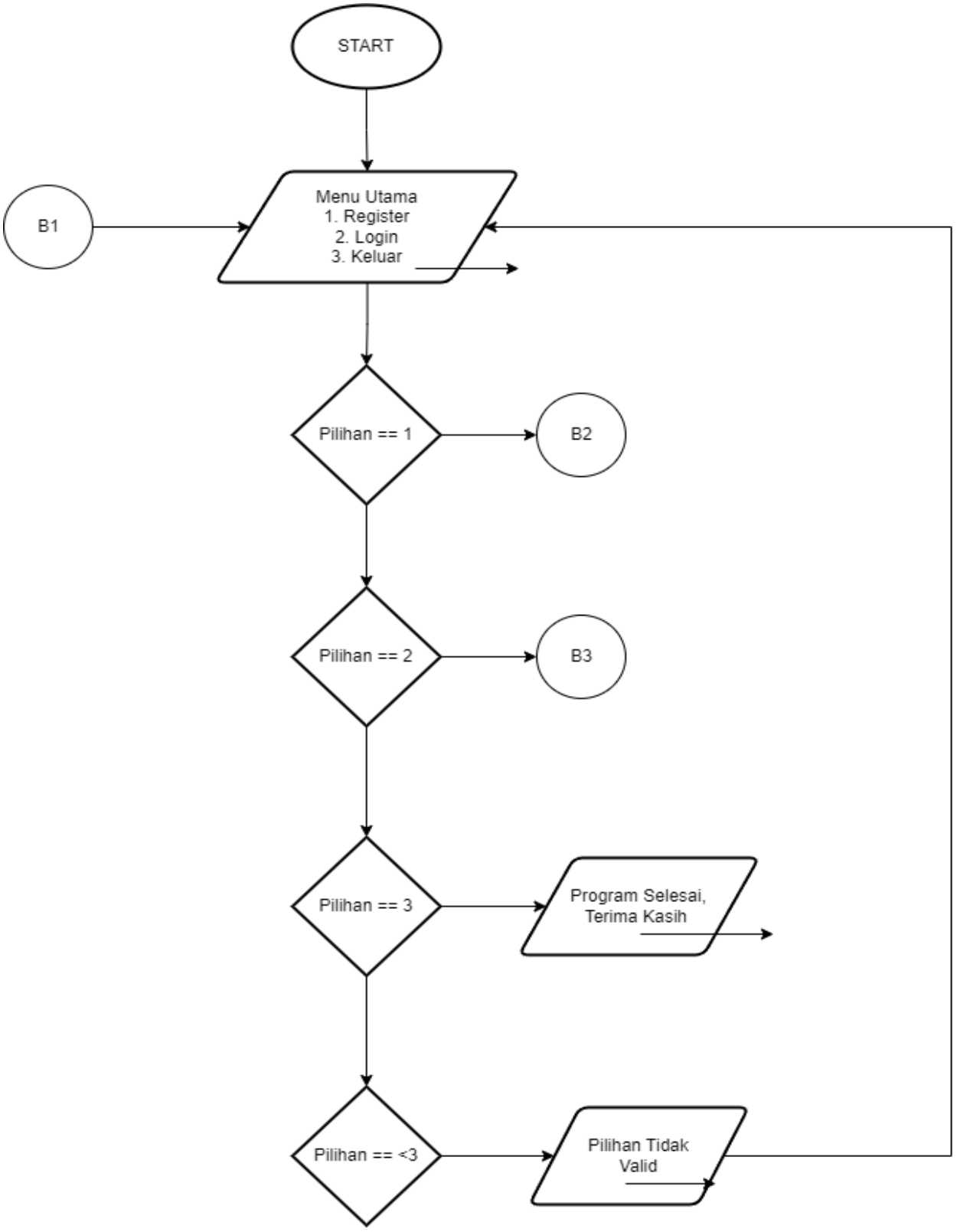
UNIVERSITAS MULAWARMAN

SAMARINDA

2025

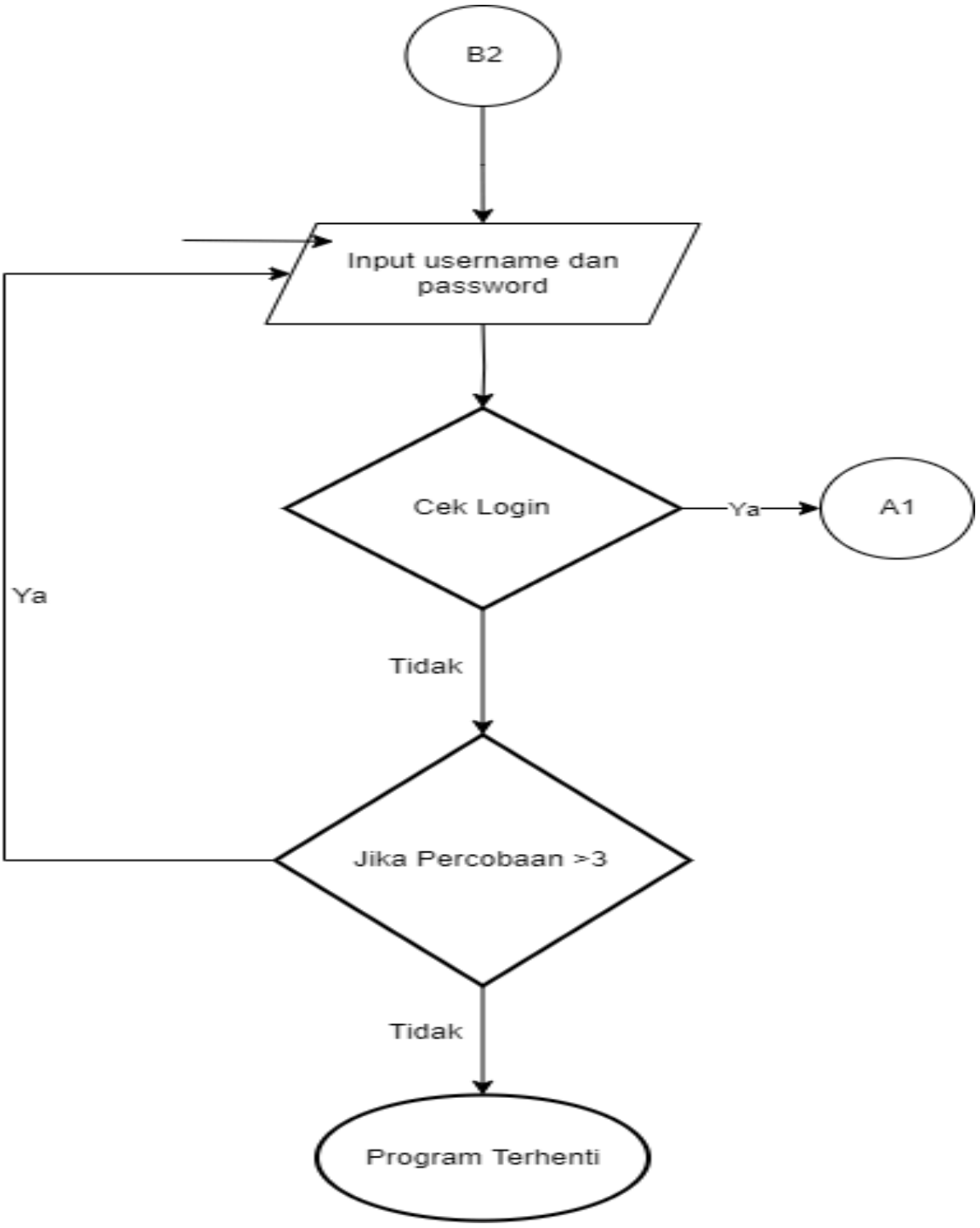
1. Flowchart

1.1 Menu Utama



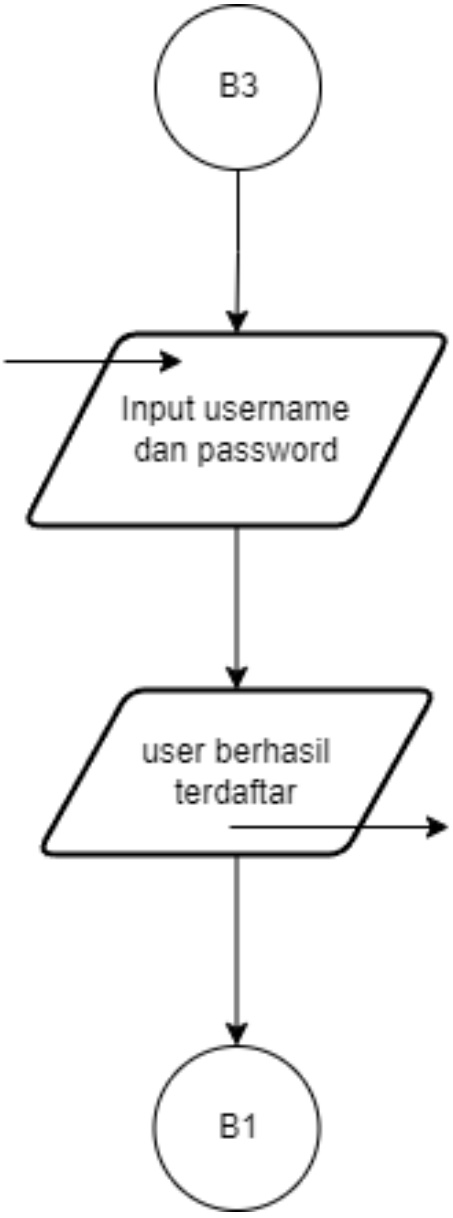
Gambar 1.1

1.2 Menu Login



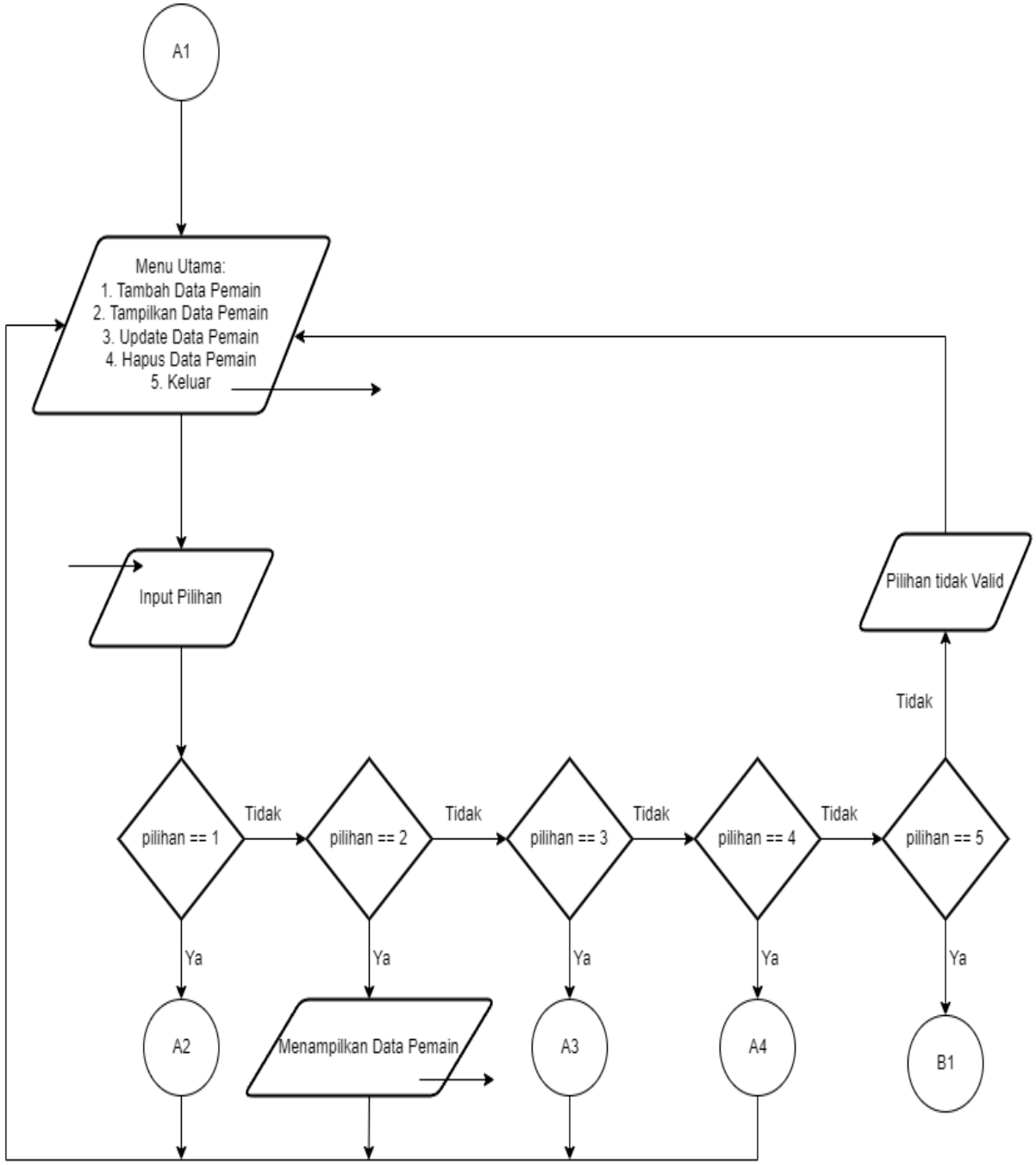
Gambar 1.2

1.3 Menu Register



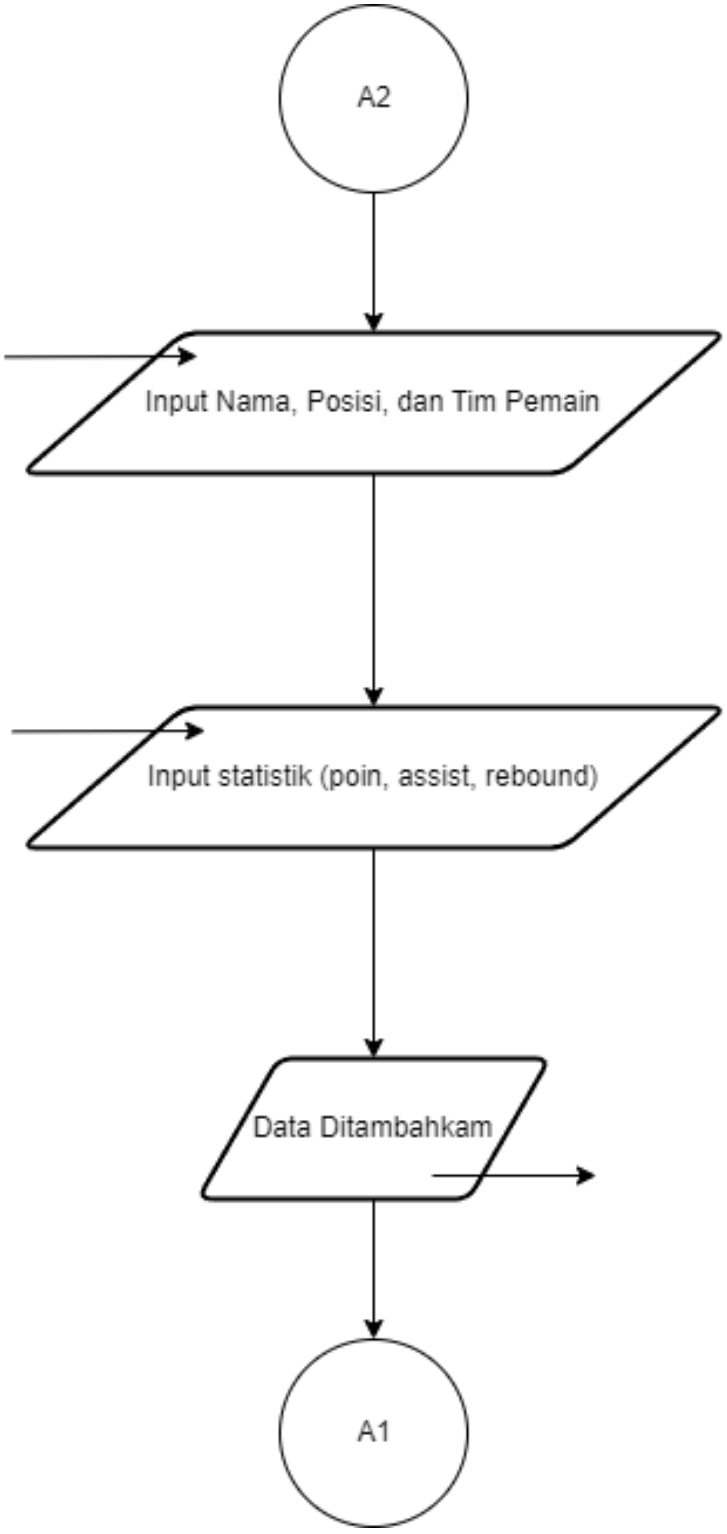
Gambar 1.3

1.4 Menu Tampilan



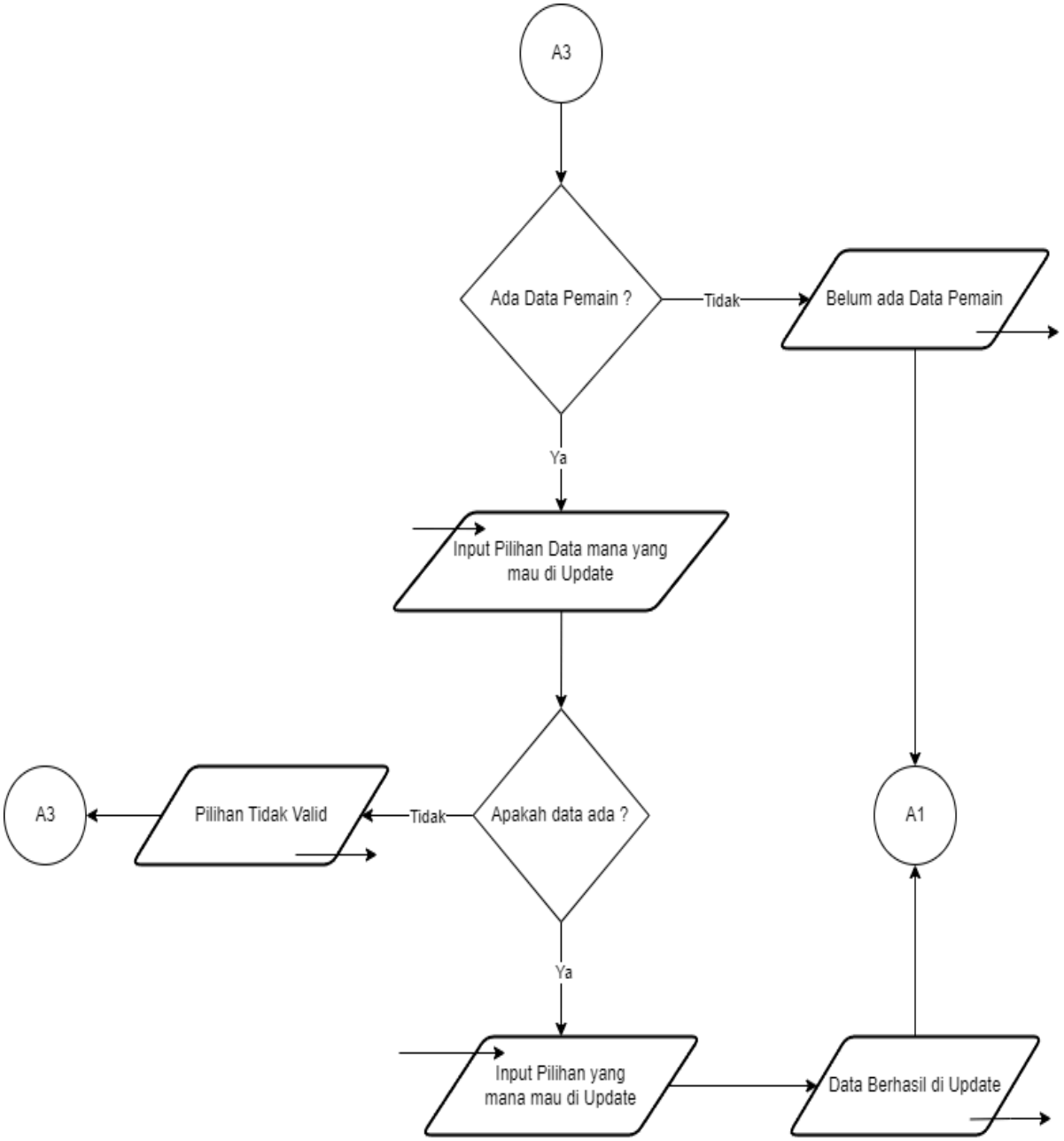
Gambar 1.4

1.5 Tambah Data



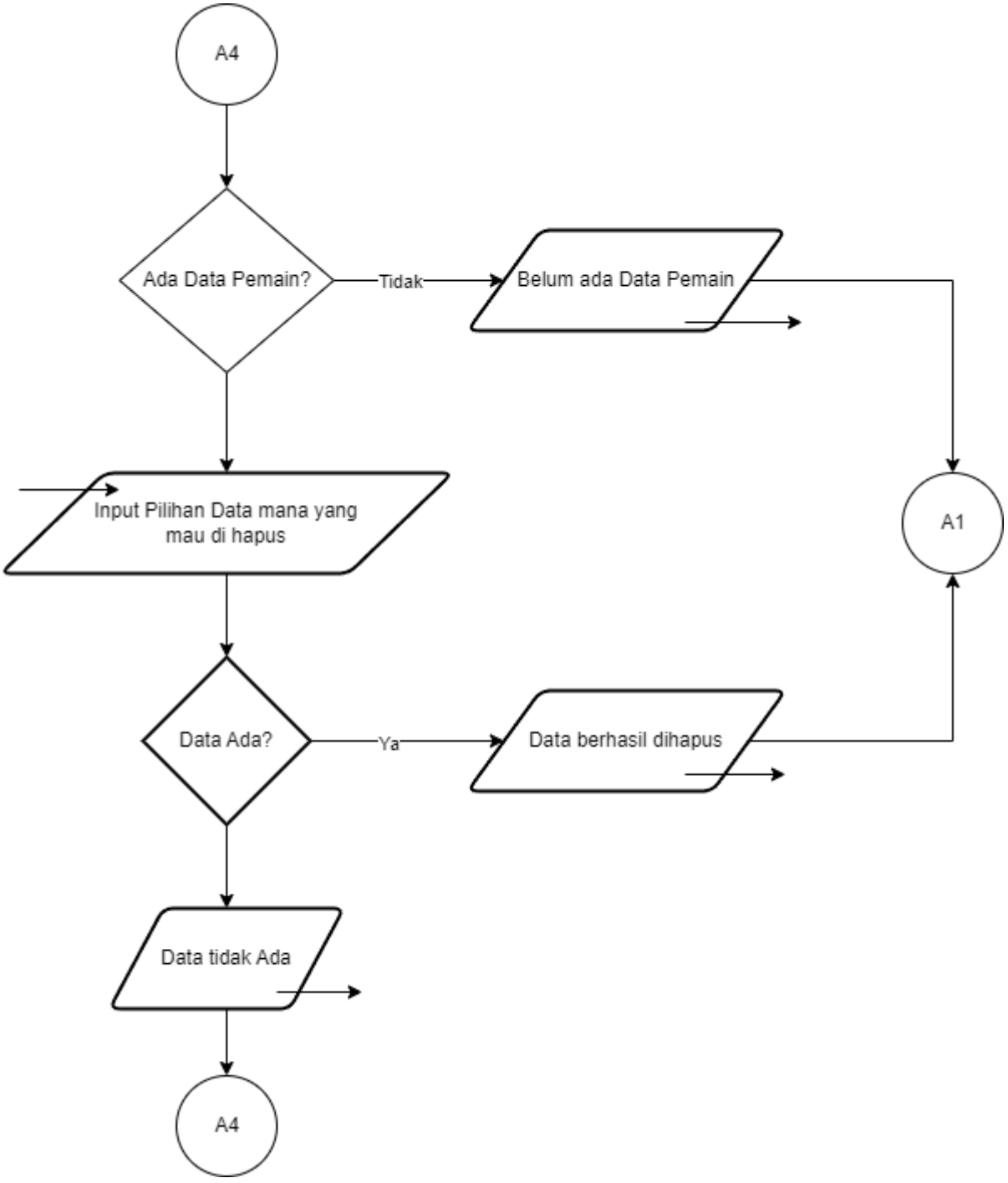
Gambar 1.5

1.6 Update Data



Gambar 1.6

1.7 Hapus Data



Gambar 1.7

2. Deskripsi Singkat Program

Program ini adalah aplikasi berbasis C++ untuk mengelola data pemain NBA. Pengguna dapat mendaftar dan login untuk mengakses fitur-fitur seperti:
1. Menambah data pemain (nama, posisi, tim, dan statistik poin, assist, rebound).
2. Menampilkan data pemain yang sudah tersimpan.
3. Mengupdate data pemain (nama, posisi, tim, dan statistik).
4. Menghapus data pemain dari data.

Program menggunakan array untuk menyimpan data pengguna dan pemain dengan kapasitas terbatas. Fitur login memiliki batasan 3 percobaan untuk keamanan.

3. Source Code

A. Menu Utama

```
    do {
        cout << "\n=== Menu Utama ===\n";
        cout << "1. Register\n";
        cout << "2. Login\n";
        cout << "3. Keluar\n";
        cout << "Pilih menu: ";
        cin >> pilihan;
```

B. Register

```
bool registerUser() {
    if (jumlahUser < MAX_USER) {
        cout << "\n=== Register User Baru ===\n";
        cout << "Masukkan Username: ";
        cin >> users[jumlahUser].username;
        cout << "Masukkan Password: ";
        cin >> users[jumlahUser].password;
        jumlahUser++;
        cout << "User berhasil terdaftar!\n";
        return true;
    } else {
        cout << "Kapasitas user penuh. Tidak bisa mendaftarkan user baru.\n";
        return false;
    }
```

C. Login

```
int loginUser() {
    string username, password;
    int percobaan = 0;

    while (percobaan < MAX_LOGIN_ATTEMPTS) {
        cout << "\n=== Login ===\n";
        cout << "Masukkan Username: ";
        cin >> username;
        cout << "Masukkan Password: ";
        cin >> password;

        for (int i = 0; i < jumlahUser; i++) {
            if (users[i].username == username && users[i].password == password) {
                cout << "Login berhasil! Selamat datang, " << username << "!\n";
                return i;
            }
        }

        percobaan++;
        cout << "Login gagal. Percobaan " << percobaan << " dari " <<
MAX_LOGIN_ATTEMPTS << ".\n";
    }

    cout << "Anda telah melebihi batas percobaan login. Program berhenti.\n";
    return -1;
}
```

D. Menu Manajemen

```
        switch (pilihan) {
            case 1:
                registerUser();
                break;
            case 2:
                loggedInUserIndex = loginUser();
                if (loggedInUserIndex != -1) {
                    int pilihanSetelahLogin;
                    do {
                        cout << "\n=== Manajemen Pemain NBA ===\n";
                        cout << "1. Tambah Data Pemain\n";
                        cout << "2. Tampilkan Data Pemain\n";
                        cout << "3. Update Data Pemain\n";
                        cout << "4. Hapus Data Pemain\n";
                        cout << "5. Logout\n";
```

```
                        cout << "Pilih menu: ";
                        cin >> pilihanSetelahLogin;

                        switch (pilihanSetelahLogin) {
                            case 1:
                                tambahDataPemain();
                                break;
                            case 2:
                                tampilkanDataPemain();
                                break;
                            case 3:
                                updateDataPemain();
                                break;
                            case 4:
                                hapusDataPemain();
                                break;
                            case 5:
                                cout << "Anda telah logout.\n";
                                loggedInUserIndex = -1;
                                break;
                            default:
                                cout << "Pilihan tidak valid.\n";
                                break;
                        }
                    } while (loggedInUserIndex != -1);
                }
                break;
            case 3:
                cout << "Program selesai. Terima kasih!\n";
                break;
            default:
                cout << "Pilihan tidak valid. Silakan coba lagi.\n";
                break;
        }
    } while (pilihan != 3);

    return 0;
}
```

E. Tambah Data

```
bool tambahDataPemain() {
    if (jumlahPemain < MAX_PEMAIN) {
        cout << "\n=== Tambah Data Pemain ===\n";
        cin.ignore();
        cout << "Masukkan Nama Pemain: ";
        getline(cin, dataPemain[jumlahPemain].nama);
        cout << "Masukkan Posisi Pemain: ";
        getline(cin, dataPemain[jumlahPemain].posisi);
        cout << "Masukkan Tim Pemain: ";
        getline(cin, dataPemain[jumlahPemain].tim);
        cout << "Masukkan Statistik Pemain (poin assist rebound)" << "\n";
        cout << "Poin : ";
        cin >> dataPemain[jumlahPemain].statistik.poin;
        cout << "Assist : ";
        cin >> dataPemain[jumlahPemain].statistik.assist;
        cout << "Rebound : ";
        cin >> dataPemain[jumlahPemain].statistik.rebound;
        jumlahPemain++;
        cout << "Data pemain berhasil ditambahkan.\n";
        return true;
    } else {
        cout << "Kapasitas pemain penuh. Tidak bisa menambah data lagi.\n";
        return false;
    }
}
```

F. Tampilkan Data

```
int tampilkanDataPemain() {
    if (jumlahPemain == 0) {
        cout << "\nBelum ada data pemain.\n";
        return 0;
    } else {
        cout << "\n=== Data Pemain NBA ===\n";
        cout << "-------------------------------------\n";
        for (int i = 0; i < jumlahPemain; i++) {
            cout << "Pemain ke-" << i + 1 << ":\n";
            cout << "Nama: " << dataPemain[i].nama << "\n";
            cout << "Posisi: " << dataPemain[i].posisi << "\n";
            cout << "Tim: " << dataPemain[i].tim << "\n";
            cout << "Statistik:\n";
            cout << "  Poin: " << dataPemain[i].statistik.poin << "\n";
            cout << "  Assist: " << dataPemain[i].statistik.assist << "\n";
```

```
            cout << "  Rebound: " << dataPemain[i].statistik.rebound << "\n";
            cout << "-------------------------------------\n";
        }
        return jumlahPemain;
    }
}
```

G. Update Data

```
bool updateDataPemain() {
    if (jumlahPemain == 0) {
        cout << "\nBelum ada data pemain untuk diupdate.\n";
        return false;
    } else {
        cout << "\n=== Update Data Pemain ===\n";
        int index;
        cout << "Masukkan nomor pemain yang ingin diupdate (1-" << jumlahPemain <<
"): ";
        cin >> index;

        if (index > 0 && index <= jumlahPemain) {
            int pilihanUpdate;
            cout << "\nPilih data yang ingin diupdate:\n";
            cout << "1. Nama\n";
            cout << "2. Posisi\n";
            cout << "3. Tim\n";
            cout << "4. Statistik\n";
            cout << "Pilihan: ";
            cin >> pilihanUpdate;

            cin.ignore();
            switch (pilihanUpdate) {
                case 1:
                    cout << "Masukkan Nama Pemain Baru: ";
                    getline(cin, dataPemain[index - 1].nama);
                    cout << "Nama pemain berhasil diupdate.\n";
                    break;
                case 2:
                    cout << "Masukkan Posisi Pemain Baru: ";
                    getline(cin, dataPemain[index - 1].posisi);
                    cout << "Posisi pemain berhasil diupdate.\n";
                    break;
                case 3:
                    cout << "Masukkan Tim Pemain Baru: ";
```

```
                    getline(cin, dataPemain[index - 1].tim);
                    cout << "Tim pemain berhasil diupdate.\n";
                    break;
                case 4:
                    cout << "Masukkan Statistik Pemain Baru (poin assist rebound)"
<< "\n";
                    cout << "Poin : ";
                    cin >> dataPemain[index - 1].statistik.poin;
                    cout << "Assist : ";
                    cin >> dataPemain[index - 1].statistik.assist;
                    cout << "Rebound : ";
                    cin >> dataPemain[index - 1].statistik.rebound;
                    cout << "Statistik pemain berhasil diupdate.\n";
                    break;
                default:
                    cout << "Pilihan tidak valid.\n";
                    return false;
            }
            return true;
        } else {
            cout << "Nomor pemain tidak valid.\n";
            return false;
        }
    }
}
```

H. Hapus Data

```
bool hapusDataPemain() {
    if (jumlahPemain == 0) {
        cout << "\nBelum ada data pemain untuk dihapus.\n";
        return false;
    } else {
        cout << "\n=== Hapus Data Pemain ===\n";
        int index;
        cout << "Masukkan nomor pemain yang ingin dihapus (1-" << jumlahPemain
<< "): ";
        cin >> index;

        if (index > 0 && index <= jumlahPemain) {
            for (int i = index - 1; i < jumlahPemain - 1; i++) {
                dataPemain[i] = dataPemain[i + 1];
            }
            jumlahPemain--;
```

```
            cout << "Data pemain berhasil dihapus.\n";
            return true;
        } else {
            cout << "Nomor pemain tidak valid.\n";
            return false;
        }
    }
}
```

4. Hasil Output

A. Menu Utama, Register, dan Login

```
=== Menu Utama ===
1. Register
2. Login
3. Keluar
Pilih menu: 1

=== Register User Baru ===
Masukkan Username: Elfin
Masukkan Password: 024
User berhasil terdaftar!

=== Menu Utama ===
1. Register
2. Login
3. Keluar
Pilih menu: 2

=== Login ===
Masukkan Username: Elfin
Masukkan Password: 024
Login berhasil! Selamat datang, Elfin!
```

Gambar 2.1

B. Menu Manajemen dan Tambah Data

```
=== Manajemen Pemain NBA ===
1. Tambah Data Pemain
2. Tampilkan Data Pemain
3. Update Data Pemain
4. Hapus Data Pemain
5. Logout
Pilih menu: 1

=== Tambah Data Pemain ===
Masukkan Nama Pemain: Lebron James
Masukkan Posisi Pemain: Forward
Masukkan Tim Pemain: Los Angles Lakers
Masukkan Statistik Pemain (poin assist rebound)
Poin : 234
Assist : 134
Rebound : 89
Data pemain berhasil ditambahkan.
```

Gambar 2.2

C. Tampilkan Data

```
=== Manajemen Pemain NBA ===
1. Tambah Data Pemain
2. Tampilkan Data Pemain
3. Update Data Pemain
4. Hapus Data Pemain
5. Logout
Pilih menu: 2

=== Data Pemain NBA ===
---------------------------------------
Pemain ke-1:
Nama: Lebron James
Posisi: Forward
Tim: Los Angles Lakers
Statistik:
  Poin: 234
  Assist: 134
  Rebound: 89
---------------------------------------
```

Gambar 2.3

D. Update Data

```
=== Manajemen Pemain NBA ===
1. Tambah Data Pemain
2. Tampilkan Data Pemain
3. Update Data Pemain
4. Hapus Data Pemain
5. Logout
Pilih menu: 3

=== Update Data Pemain ===
Masukkan nomor pemain yang ingin diupdate (1-1): 1

Pilih data yang ingin diupdate:
1. Nama
2. Posisi
3. Tim
4. Statistik
Pilihan: 3
Masukkan Tim Pemain Baru: Bucks
Tim pemain berhasil diupdate.

=== Manajemen Pemain NBA ===
1. Tambah Data Pemain
2. Tampilkan Data Pemain
3. Update Data Pemain
4. Hapus Data Pemain
5. Logout
Pilih menu: 2

=== Data Pemain NBA ===
---------------------------------------
Pemain ke-1:
Nama: Lebron James
Posisi: Forward
Tim: Bucks
Statistik:
  Poin: 234
  Assist: 134
  Rebound: 89
---------------------------------------
```

Gambar 2.4

E. Hapus Data

```
=== Manajemen Pemain NBA ===
1. Tambah Data Pemain
2. Tampilkan Data Pemain
3. Update Data Pemain
4. Hapus Data Pemain
5. Logout
Pilih menu: 4

=== Hapus Data Pemain ===
Masukkan nomor pemain yang ingin dihapus (1-1): 1
Data pemain berhasil dihapus.

=== Manajemen Pemain NBA ===
1. Tambah Data Pemain
2. Tampilkan Data Pemain
3. Update Data Pemain
4. Hapus Data Pemain
5. Logout
Pilih menu: 2

Belum ada data pemain.
```

Gambar 2.4

5. Langkah Langkah Git

```
PS C:\Kuliah\praktikum-apl> git add .
PS C:\Kuliah\praktikum-apl> git commit -m "test"
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
PS C:\Kuliah\praktikum-apl> git push origin main
Enumerating objects: 20, done.
Counting objects: 100% (20/20), done.
Delta compression using up to 16 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (14/14), 3.17 KiB | 1.58 MiB/s, done.
Total 14 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Symthi/praktikum-apl.git
   8b1abdb..5f1d327  main -> main
PS C:\Kuliah\praktikum-apl>
```

1. Menginput (git add .)
   Perintah ini digunakan untuk menambahkan semua perubahan yang ada di direktori kerja ke staging area. Ini berarti semua file yang telah dimodifikasi, dihapus, atau ditambahkan akan siap untuk di-commit.
2. Menginput (git commit -m “test”)
   Perintah ini membuat commit baru dengan pesan "test". Commit adalah cara untuk menyimpan perubahan yang telah dilakukan ke dalam repositori lokal.
3. Menginput (git push origin main)
   Perintah ini digunakan untuk mengirim commit yang ada di branch lokal main ke repositori remote (origin).