

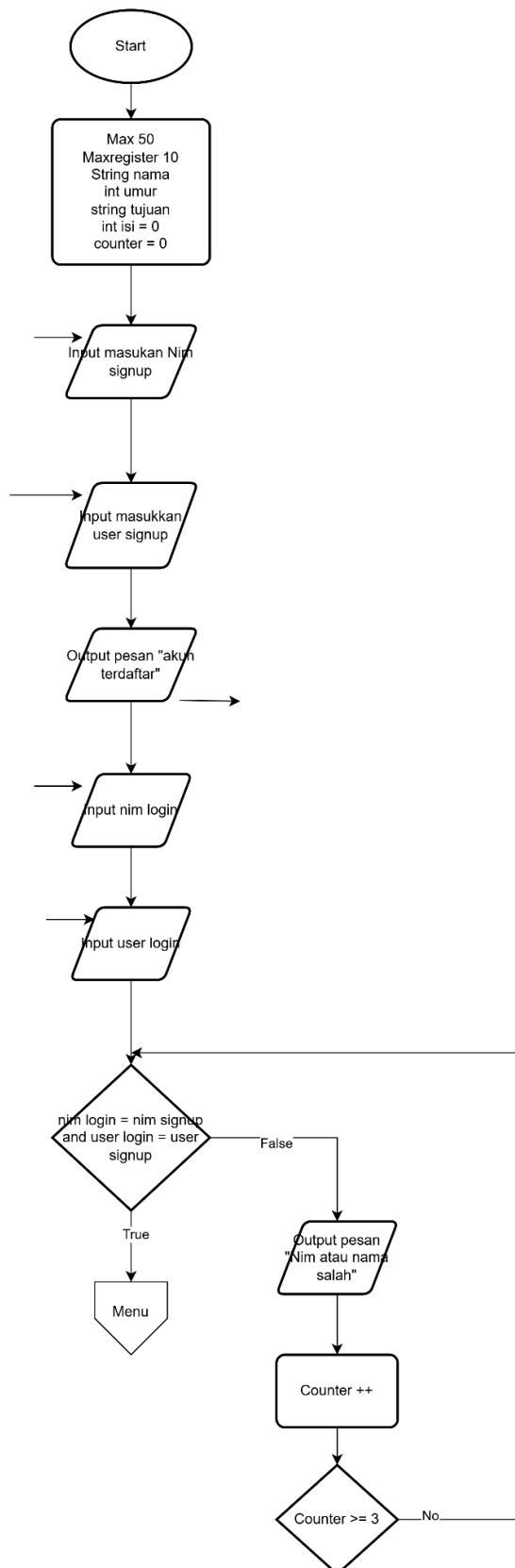
LAPORAN PRAKTIKUM
POSTTEST (6)
ALGORITMA PEMROGRAMAN LANJUT



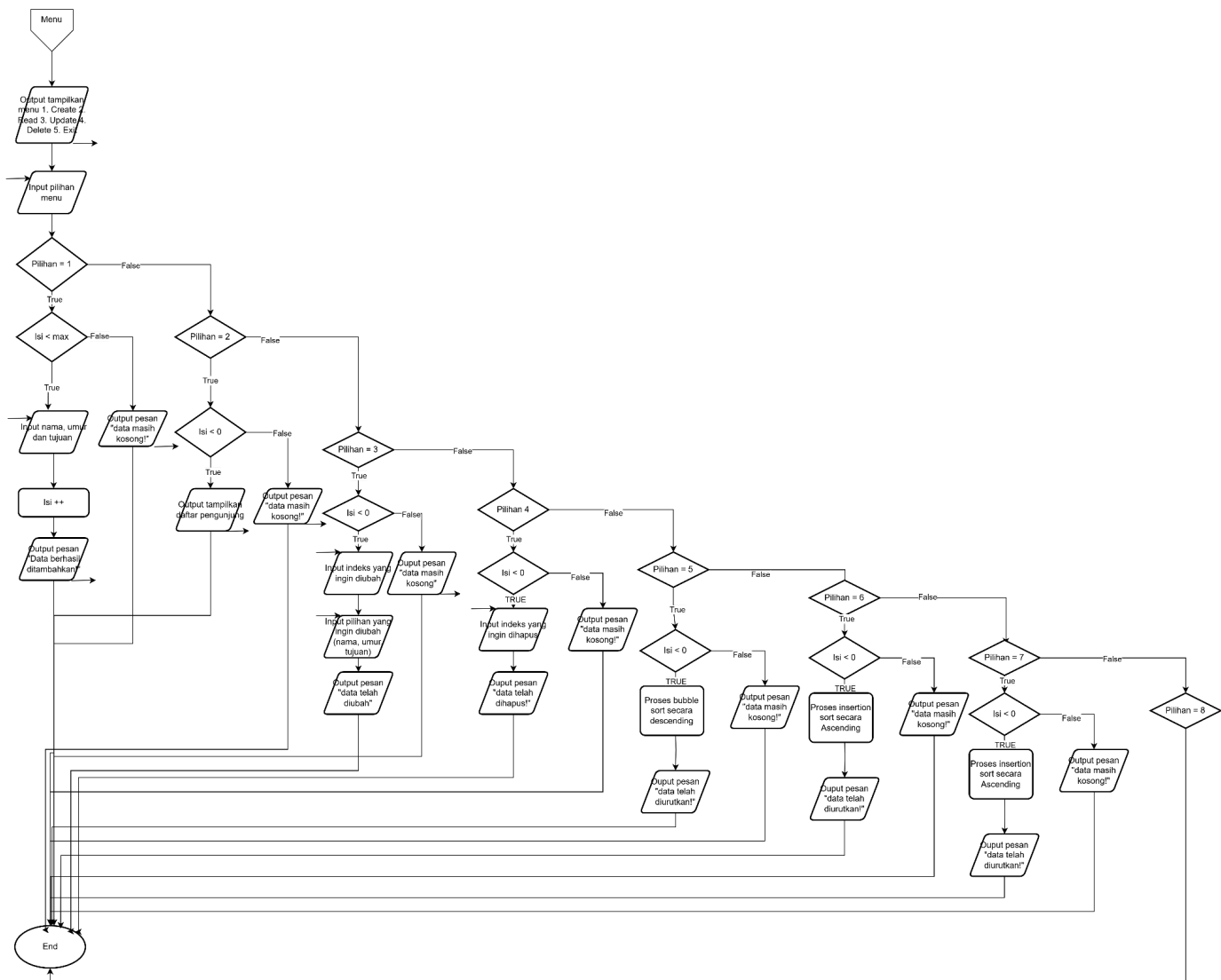
Disusun oleh:
Richard Dante Gunawan (2409106061)
Kelas (B1 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



Gambar 1.1 Flowchart login dan signup



Gambar 1.2 Flowchart menu utama

2. Analisis Program

Program manajemen data pengunjung mall dibuat dengan tujuan mensimulasikan konsep CRUD seperti menambah data pengunjung, membaca data pengunjung, mengubah data pengunjung serta menghapus data pengunjung.

3. Source Code

A. Signup

Fitur ini digunakan untuk mendaftarkan akun pengguna menggunakan nim dan nama.

Source Code:

```
void signup(regis* rgs, int awal) {
    cout << "=== Signup Akun ===" << endl;
    cout << "Silahkan masukkan NIM anda: ";
    cin >> rgs[awal].nimsignup;
    cin.ignore();

    cout << "Silahkan masukkan nama anda: ";
    getline(cin, rgs[awal].usersignup);

    cout << "\nAkun berhasil terdaftar!\n" << endl;
}
```

B. Login

Fitur ini digunakan untuk memvalidasi akun pengguna. Pengguna akan diberi kesempatan sebanyak 3 kali untuk mencoba.

Source Code:

```
bool login(regis* rgs, int awal) {
    cout << "=== Login Akun ===" << endl;
    cout << "Silahkan masukkan NIM anda: ";
    cin >> rgs[awal].nimlogin;
    cin.ignore();

    cout << "Silahkan masukkan nama Anda: ";
    getline(cin, rgs[awal].userlogin);

    if (rgs[awal].nimsignup == rgs[awal].nimlogin && rgs[awal].usersignup ==
    rgs[awal].userlogin) {
        cout << "\nLogin berhasil! Selamat datang, " << rgs[awal].usersignup <<
        "\n" << endl;
        return true;
    } else {
        cout << "\nLogin gagal! NIM atau Nama tidak sesuai.\n" << endl;
        return false;
    }
}
```

C. Menu

Fitur ini digunakan untuk menampilkan menu utama yang terdiri dari tambah data, baca data, update data, hapus data dan exit.

Source Code:

```
cout << "\n=== Silahkan pilih aksi yang ingin dilakukan! === " << endl;
cout << "1. Tambah Data Pengunjung" << endl;
cout << "2. Lihat Data Pengunjung" << endl;
cout << "3. Perbarui Data Pengunjung" << endl;
cout << "4. Hapus Data Pengunjung" << endl;
cout << "5. Exit" << endl;
cout << "Pilih menu: ";
cin >> pilihan;
cin.ignore();
```

D. Tambah data

Fitur ini digunakan untuk menambahkan data baru ke dalam array.

Source Code:

```
void Tambah(pengunjung *pgj[], int &isi) {
    cout << "Silahkan masukkan nama pengunjung: ";
    getline(cin, pgj[isi]->nama);

    cout << "Masukkan umur pengunjung: ";
    cin >> pgj[isi]->umur;
    cin.ignore();

    cout << "Masukkan tujuan kunjungan: ";
    getline(cin, pgj[isi]->tujuan);

    isi++;
    cout << "Data pengunjung berhasil ditambahkan!\n" << endl;
}
```

E. Baca data

Fitur ini digunakan untuk membaca data yang telah tersimpan di dalam array.

Source Code:

```

void Tampil(pengunjung *pgj[], int &isi) {
    cout << "Daftar Pengunjung" << endl;
    for (int i = 0; i < isi; i++) {
        cout << "Pengunjung ke-" << i + 1 << ": " << pgj[i]->nama
            << " | Umur: " << pgj[i]->umur
            << " | Tujuan: " << pgj[i]->tujuan << endl;
    }
}

```

F. Update data

Fitur ini digunakan untuk memperbarui data di dalam array. Pengguna diberi 3 pilihan untuk mengubah data yaitu nama, umur atau tujuan kunjungan.

Source Code:

```

void Ubah(pengunjung *pgj[], int &isi) {
    int index;
    cout << "Masukkan index pengunjung yang ingin diperbarui: ";
    cin >> index;
    cin.ignore();
    if (index > 0 && index <= isi) {
        int pilihanEdit;
        cout << "Pilih data yang ingin diperbarui:" << endl;
        cout << "1. Nama" << endl;
        cout << "2. Umur" << endl;
        cout << "3. Tujuan Kunjungan" << endl;
        cout << "Pilih: ";
        cin >> pilihanEdit;
        cin.ignore();

        switch (pilihanEdit) {
            case 1:
                cout << "Masukkan nama baru: ";
                getline(cin, pgj[index - 1]->nama);
                cout << "Nama berhasil diperbarui!\n";
                break;
            case 2:
                cout << "Masukkan umur baru: ";
                cin >> pgj[index - 1]->umur;
                cin.ignore();
                cout << "Umur berhasil diperbarui!\n";
                break;
            case 3:
                cout << "Masukkan tujuan kunjungan baru: ";
                getline(cin, pgj[index - 1]->tujuan);
                cout << "Tujuan kunjungan berhasil diperbarui!\n";
                break;
        }
    }
}

```

```

        break;
    default:
        cout << "Pilihan tidak valid!\n";
        break;
    }
} else {
    cout << "Index tidak valid!\n" << endl;
}
}

bool Hapus(pengunjung data[], int &jumlah, int idx) {
    if (idx > 0 && idx <= jumlah) {
        for (int i = idx - 1; i < jumlah - 1; i++) {
            data[i] = data[i + 1];
        }
        jumlah--;
        return true;
    } else {
        return false;
    }
}

```

G. Hapus data

Fitur ini digunakan untuk menghapus data dari indeks yang dipilih.

Source Code:

```

bool Hapus (pengunjung data [], int &jumlah, int idx){
    if (idx > 0 && idx <= jumlah) {
        for (int i = idx - 1; i < jumlah - 1; i++) {
            data[i] = data[i + 1];
        }
        jumlah--;
        return true;
    } else {
        return false;
    }
}

```

H. Bubble sort

```

void bubbleSort(pengunjung data[], int n) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n - i - 1; j++) {

```

```

        if (data[j].umur < data[j + 1].umur) {
            pengunjung temp = data[j];
            data[j] = data[j + 1];
            data[j + 1] = temp;
        }
    }
    cout << "\nProses [" << i + 1 << "] : ";
    for (int k = 0; k < n; k++) {
        cout << data[k].umur << " ";
    }
}
cout << endl;
}

```

I. Insertion sort

```

void insertionSort(pengunjung* data, int panjang)
{
    for (int i = 1; i < panjang; i++)
    {
        pengunjung key = *(data + i);
        int j = i - 1;

        while (j >= 0 && (data + j)->nama > key.nama)
        {
            *(data + j + 1) = *(data + j);
            j = j - 1;
        }
        *(data + j + 1) = key;

        cout << "Tahap ke-" << i << ": ";
        for (int k = 0; k < panjang; k++)
        {
            cout << (data + k)->nama << " ";
        }
        cout << endl;
    }
}

```

J. Selection sort

```

void selectionSort(pengunjung* data, int panjang)
{
    for (int i = 0; i < panjang - 1; i++)
    {
        int min = i;
        for (int j = i + 1; j < panjang; j++)
        {

```



```

        if ((data + j)->umur < (data + min)->umur)
        {
            min = j;
        }
    }

    pengunjung temp = *(data + i);
    *(data + i) = *(data + min);
    *(data + min) = temp;

    cout << "Tahap ke-" << i + 1 << ": ";
    for (int k = 0; k < panjang; k++)
    {
        cout << (data + k)->umur << " ";
    }
    cout << endl;
}
}

```

4. Uji Coba dan Hasil Output

Hasil Output

```

=== Signup Akun ===
Silahkan masukkan NIM anda: 61
Silahkan masukkan nama anda: dnt

Akun berhasil terdaftar!

=== Login Akun ===
Silahkan masukkan NIM anda: 61
Silahkan masukkan nama Anda: dnt

Login berhasil! Selamat datang, dnt!

```

Gambar 4.1 Login dan Signup

```

==== Silahkan pilih aksi yang ingin dilakukan! ====
1. Tambah Data Pengunjung
2. Lihat Data Pengunjung
3. Perbarui Data Pengunjung
4. Hapus Data Pengunjung
5. Exit
Pilih menu: 1
Silahkan masukkan nama pengunjung: dante
Masukkan umur pengunjung: 14
Masukkan tujuan kunjungan: ramayana
Data pengunjung berhasil ditambahkan!

```

Gambar 4.2 Fitur tambah data

```

==== Silahkan pilih aksi yang ingin dilakukan! ====
1. Tambah Data Pengunjung
2. Lihat Data Pengunjung
3. Perbarui Data Pengunjung
4. Hapus Data Pengunjung
5. Exit
Pilih menu: 2
Daftar Pengunjung
Pengunjung ke-1: dnt | Umur: 89 | Tujuan: uniqlo

```

Gambar 4.3 Fitur lihat data

```

==== Silahkan pilih aksi yang ingin dilakukan! ====
1. Tambah Data Pengunjung
2. Lihat Data Pengunjung
3. Perbarui Data Pengunjung
4. Hapus Data Pengunjung
5. Exit
Pilih menu: 3
Daftar Pengunjung
Pengunjung ke-1: dnte | Umur: 88 | Tujuan: uniqlo
Masukkan index pengunjung yang ingin diperbarui: 1
Pilih data yang ingin diperbarui:
1. Nama
2. Umur
3. Tujuan Kunjungan
Pilih: 3
Masukkan tujuan kunjungan baru: funcity
Tujuan kunjungan berhasil diperbarui!

```

Gambar 4.4 Fitur perbarui data

```

==== Silahkan pilih aksi yang ingin dilakukan! ====
1. Tambah Data Pengunjung
2. Lihat Data Pengunjung
3. Perbarui Data Pengunjung
4. Hapus Data Pengunjung
5. Exit
Pilih menu: 4
Daftar Pengunjung
Pengunjung ke-1: dnte | Umur: 88 | Tujuan: matahari
Masukkan index pengunjung yang ingin dihapus:
1
Data pengunjung berhasil dihapus!

```

Gambar 4.5 Fitur hapus data

```

Pilih menu: 5

Proses [1] : 72 106 65
Proses [2] : 106 72 65
Proses [3] : 106 72 65

Data pengunjung setelah diurutkan (Descending):
Daftar Pengunjung:
Pengunjung ke-1: gugun | Umur: 106 | Tujuan: celcius
Pengunjung ke-2: dante | Umur: 72 | Tujuan: uniqlo
Pengunjung ke-3: loki | Umur: 65 | Tujuan: pasar

```

Gambar 4.6 Fitur Bubble sort

```
Pilih menu: 6
Tahap ke-1: dante gugun loki
Tahap ke-2: dante gugun loki

Data pengunjung setelah diurutkan (Ascending):
Daftar Pengunjung:
Pengunjung ke-1: dante | Umur: 72 | Tujuan: uniqlo
Pengunjung ke-2: gugun | Umur: 106 | Tujuan: celcius
Pengunjung ke-3: loki | Umur: 65 | Tujuan: pasar
```

Gambar 4.7 Fitur Insertion sort

```
Pilih menu: 7
Tahap ke-1: 65 106 72
Tahap ke-2: 65 72 106

Data pengunjung setelah diurutkan (Ascending):
Daftar Pengunjung:
Pengunjung ke-1: loki | Umur: 65 | Tujuan: pasar
Pengunjung ke-2: dante | Umur: 72 | Tujuan: uniqlo
Pengunjung ke-3: gugun | Umur: 106 | Tujuan: celcius
```

Gambar 4.8 Fitur Selection sort

5.GIT

```
PS C:\Users\PC\Desktop\praktikum-apl\B1'24\post-test\post-test-apl-6> git add .
PS C:\Users\PC\Desktop\praktikum-apl\B1'24\post-test\post-test-apl-6> git commit -m "Finish Posttest 5"
[main 8838697] Finish Posttest 5
2 files changed, 299 insertions(+)
create mode 100644 B1'24/post-test/post-test-apl-6/2409106061-RichardDanteGunawan-PT-6.cpp
create mode 100644 B1'24/post-test/post-test-apl-6/2409106061-RichardDanteGunawan-PT-6.exe
PS C:\Users\PC\Desktop\praktikum-apl\B1'24\post-test\post-test-apl-6> git push -u origin main
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 678.64 KiB | 2.49 MiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/404-DevNotFound/praktikum-apl.git
aa59604..8838697 main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\PC\Desktop\praktikum-apl\B1'24\post-test\post-test-apl-6>
```

Gambar 5.1 langkah-langkah git

Git add: perintah untuk menambahkan file ke staging area

Git commit: perintah untuk menyimpan perubahan yang telah ditambahkan ke staging area ke dalam repository lokal.

Git push: perintah untuk mengirim commit dari repository lokal ke repository remote.