

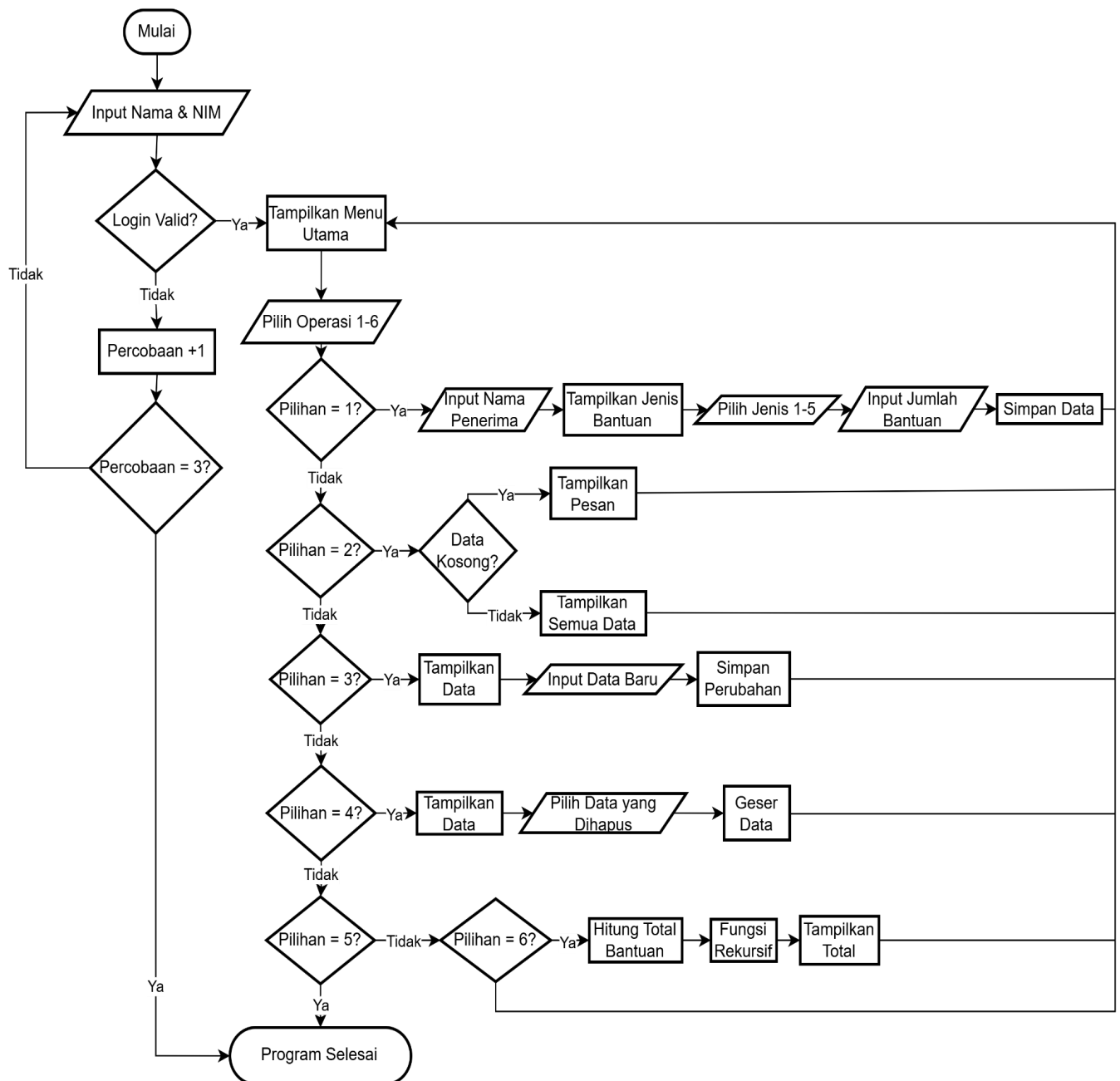
LAPORAN PRAKTIKUM
POSTTEST (5)
ALGORITMA PEMROGRAMAN LANJUT



Disusun oleh:
Fachlevi Muhammad (2409106059)
Kelas (B1 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



2. Analisis Program

Tujuan dan Manfaat Program

Tujuan dari pembuatan program ini adalah untuk membantu dalam pendataan dan pengelolaan informasi donasi secara lebih tertata dan efisien. Melalui sistem ini, pengguna dapat melakukan berbagai operasi seperti menambahkan data bantuan, melihat daftar penerima, mengubah informasi, hingga menghapus data jika diperlukan.

Program ini juga dilengkapi dengan fitur login pengguna untuk keamanan, serta memiliki kemampuan menghitung total donasi secara otomatis menggunakan fungsi rekursif. Selain itu, penerapan fungsi overloading juga digunakan untuk menampilkan informasi dengan fleksibel.

Adapun manfaat utama dari program ini yaitu:

1. Memudahkan pencatatan data donasi tanpa harus dilakukan secara manual.
2. Mencegah duplikasi dan kesalahan input.
3. Menyederhanakan proses pengelolaan bantuan.
4. Melatih pemahaman dalam membuat program modular dengan fungsi dan prosedur di C++.
5. Menjadi contoh nyata penerapan rekursi dan overloading dalam kasus sederhana.

3. Source Code

A. Login (Fitur Keamanan Awal)

```
bool login() {
    string nama, nim;
    int percobaan = 0;

    while (percobaan < 3) {
        cout << "Masukkan Nama: ";
        getline(cin, nama);
        cout << "Masukkan NIM: ";
        cin >> nim;
        cin.ignore();

        if (nama == "Fachlevi Muhammad" && nim == "059") {
            cout << "Login berhasil\n";
            return true;
        } else {
            percobaan++;
            cout << "Login gagal percobaan ke-" << percobaan << endl;
        }
    }
    cout << "Anda telah gagal login 3 kali. Program berhenti.\n";
    return false;
}
```

B. Menambah Data Donasi

```
void tambahData(Donasi dataDonasi[], int &jumlahData) {
    cin.ignore();
    cout << "Masukkan Nama Penerima: ";
    getline(cin, dataDonasi[jumlahData].namaPenerima);

    int pilihan;
    cout << "Pilih jenis bantuan (1-5): ";
    cin >> pilihan;

    switch (pilihan) {
        case 1: dataDonasi[jumlahData].jenisBantuan = "Makanan"; break;
        case 2: dataDonasi[jumlahData].jenisBantuan = "Pakaian"; break;
        case 3: dataDonasi[jumlahData].jenisBantuan = "Uang Tunai";
        break;
        case 4: dataDonasi[jumlahData].jenisBantuan = "Obat-obatan";
        break;
    }
```

```

        case 5: dataDonasi[jumlahData].jenisBantuan = "Buku"; break;
    }

    cout << "Masukkan Jumlah Bantuan: ";
    cin >> dataDonasi[jumlahData].jumlahBantuan;

    jumlahData++;
    cout << "Data berhasil ditambahkan\n";
}

```

C. Penerapan Pointer

```

void tambahData(Donasi dataDonasi[], int *jumlahData) {
    if (*jumlahData < 100) {
        cin.ignore();
        cout << "Masukkan Nama Penerima: ";
        getline(cin, dataDonasi[*jumlahData].namaPenerima);

        int pilihanBantuan;
        tampilkanJenisBantuan();
        cout << "Pilih jenis bantuan (1-5): ";
        while (!(cin >> pilihanBantuan) || pilihanBantuan < 1 ||
pilihanBantuan > 5) {
            cout << "Input tidak valid, Masukkan angka antara 1 sampai
5: ";

            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
        }

        switch (pilihanBantuan) {
            case 1: dataDonasi[*jumlahData].jenisBantuan = "Makanan";
break;
            case 2: dataDonasi[*jumlahData].jenisBantuan = "Pakaian";
break;
            case 3: dataDonasi[*jumlahData].jenisBantuan = "Uang Tunai";
break;
            case 4: dataDonasi[*jumlahData].jenisBantuan = "Obat
obatan"; break;
            case 5: dataDonasi[*jumlahData].jenisBantuan = "Buku";
break;
        }

        cout << "Masukkan Jumlah Bantuan: ";
        while (!(cin >> dataDonasi[*jumlahData].jumlahBantuan)) {
            cout << "Input tidak valid, Masukkan angka: ";

```

```

        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
    }

    (*jumlahData)++;
    cout << "Data berhasil ditambahkan\n";
} else {
    cout << "Kapasitas data penuh\n";
}
}

void ubahData(Donasi dataDonasi[], int *jumlahData) {
    int index;
    tampilkanData(dataDonasi, *jumlahData);
    cout << "Masukkan nomor data yang ingin diubah: ";
    while (!(cin >> index) || index < 1 || index > *jumlahData) {
        cout << "Nomor data tidak valid, Masukkan angka antara 1 sampai "
" << *jumlahData << ": ";
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
    }

    cin.ignore();
    cout << "Masukkan Nama Penerima Baru: ";
    getline(cin, dataDonasi[index - 1].namaPenerima);

    int pilihanBantuan;
    tampilkanJenisBantuan();
    cout << "Pilih jenis bantuan (1-5): ";
    while (!(cin >> pilihanBantuan) || pilihanBantuan < 1 ||
pilihanBantuan > 5) {
        cout << "Input tidak valid, Masukkan angka antara 1 sampai 5: ";
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
    }

    switch (pilihanBantuan) {
        case 1: dataDonasi[index - 1].jenisBantuan = "Makanan"; break;
        case 2: dataDonasi[index - 1].jenisBantuan = "Pakaian"; break;
        case 3: dataDonasi[index - 1].jenisBantuan = "Uang Tunai";
break;
        case 4: dataDonasi[index - 1].jenisBantuan = "Obat obatan";
break;
        case 5: dataDonasi[index - 1].jenisBantuan = "Buku"; break;
    }
}

```

```

        cout << "Masukkan Jumlah Bantuan Baru: ";
        while (!(cin >> dataDonasi[index - 1].jumlahBantuan)) {
            cout << "Input tidak valid, Masukkan angka: ";
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
        }

        cout << "Data berhasil diubah\n";
    }

void hapusData(Donasi dataDonasi[], int *jumlahData) {
    int index;
    tampilkanData(dataDonasi, *jumlahData);
    cout << "Masukkan nomor data yang ingin dihapus: ";
    while (!(cin >> index) || index < 1 || index > *jumlahData) {
        cout << "Nomor data tidak valid, Masukkan angka antara 1 sampai "
        << *jumlahData << ": ";
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
    }

    for (int i = index - 1; i < *jumlahData - 1; i++) {
        dataDonasi[i] = dataDonasi[i + 1];
    }
    (*jumlahData)--;
    cout << "Data berhasil dihapus\n";
}

```

D. Parameter Dengan Address-of

```

int main() {
    Donasi dataDonasi[100];
    int jumlahData = 0;

    if (!login()) {
        return 0;
    }

    int pilihan;
    do {
        tampilkanMenu();
        while (!(cin >> pilihan)) {
            cout << "Input tidak valid, Masukkan angka: ";
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
        }
    }
}

```

```

        switch (pilihan) {
            case 1:
                tambahData(dataDonasi, &jumlahData);
                break;
            case 2:
                tampilkanData(dataDonasi, jumlahData);
                break;
            case 3:
                ubahData(dataDonasi, &jumlahData);
                break;
            case 4:
                hapusData(dataDonasi, &jumlahData);
                break;
            case 5:
                cout << "Terima kasih, Program selesai\n";
                break;
            case 6:
                cout << "Total semua bantuan: " <<
hitungTotalBantuan(dataDonasi, jumlahData) << endl;
                break;
            default:
                cout << "Pilihan tidak valid\n";
        }
    } while (pilihan != 5);

    return 0;
}

```

E. Parameter Dengan Deference

```

void tambahData(Donasi dataDonasi[], int *jumlahData) {
    if (*jumlahData < 100) {
        cin.ignore();
        cout << "Masukkan Nama Penerima: ";
        getline(cin, dataDonasi[*jumlahData].namaPenerima);

        int pilihanBantuan;
        tampilkanJenisBantuan();
        cout << "Pilih jenis bantuan (1-5): ";
        while (!(cin >> pilihanBantuan) || pilihanBantuan < 1 ||
pilihanBantuan > 5) {
            cout << "Input tidak valid, Masukkan angka antara 1 sampai
5: ";

            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');

```



```

    }

    switch (pilihanBantuan) {
        case 1: dataDonasi[*jumlahData].jenisBantuan = "Makanan";
break;
        case 2: dataDonasi[*jumlahData].jenisBantuan = "Pakaian";
break;
        case 3: dataDonasi[*jumlahData].jenisBantuan = "Uang Tunai";
break;
        case 4: dataDonasi[*jumlahData].jenisBantuan = "Obat
obatan"; break;
        case 5: dataDonasi[*jumlahData].jenisBantuan = "Buku";
break;
    }

    cout << "Masukkan Jumlah Bantuan: ";
    while (!(cin >> dataDonasi[*jumlahData].jumlahBantuan)) {
        cout << "Input tidak valid, Masukkan angka: ";
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
    }

    (*jumlahData)++;
    cout << "Data berhasil ditambahkan\n";
} else {
    cout << "Kapasitas data penuh\n";
}
}
}

```

4. Uji Coba dan Hasil Output

Login

```
Masukkan Nama: Fachlevi Muhammad
Masukkan NIM: 059
Login berhasil
```

Tambah Data Donasi

```
=== MENU UTAMA ===
1. Tambah Data Donasi
2. Tampilkan Data Donasi
3. Ubah Data Donasi
4. Hapus Data Donasi
5. Keluar
6. Hitung Total Bantuan
Pilih menu: 1
Masukkan Nama Penerima: Loki

=== JENIS BANTUAN ===
1. Makanan
2. Pakaian
3. Uang Tunai
4. Obat-obatan
5. Buku
Pilih jenis bantuan (1-5): 4
Masukkan Jumlah Bantuan: 1000000
Data berhasil ditambahkan
```

Tampilkan Data Donasi

```
=== MENU UTAMA ===
1. Tambah Data Donasi
2. Tampilkan Data Donasi
3. Ubah Data Donasi
4. Hapus Data Donasi
5. Keluar
6. Hitung Total Bantuan
Pilih menu: 2

=== DATA DONASI ===
Data ke-1:
Nama Penerima : Loki
Jenis Bantuan : Obat obatan
Jumlah Bantuan: 1000000
-----
```

Ubah Data Donasi

```
=== MENU UTAMA ===
1. Tambah Data Donasi
2. Tampilkan Data Donasi
3. Ubah Data Donasi
4. Hapus Data Donasi
5. Keluar
6. Hitung Total Bantuan
Pilih menu: 3

=== DATA DONASI ===
Data ke-1:
Nama Penerima : Loki
Jenis Bantuan : Obat obatan
Jumlah Bantuan: 1000000
-----
Masukkan nomor data yang ingin diubah: 1
Masukkan Nama Penerima Baru: Hela

=== JENIS BANTUAN ===
1. Makanan
2. Pakaian
3. Uang Tunai
4. Obat-obatan
5. Buku
Pilih jenis bantuan (1-5): 2
Masukkan Jumlah Bantuan Baru: 1000000
Data berhasil diubah
```

Hapus Data Donasi

```
=== MENU UTAMA ===
1. Tambah Data Donasi
2. Tampilkan Data Donasi
3. Ubah Data Donasi
4. Hapus Data Donasi
5. Keluar
6. Hitung Total Bantuan
Pilih menu: 4

=== DATA DONASI ===
Data ke-1:
Nama Penerima : Hela
Jenis Bantuan : Pakaian
Jumlah Bantuan: 1000000
-----
Masukkan nomor data yang ingin dihapus: 1
Data berhasil dihapus
```

Hitung Total Bantuan

```
=== MENU UTAMA ===  
1. Tambah Data Donasi  
2. Tampilkan Data Donasi  
3. Ubah Data Donasi  
4. Hapus Data Donasi  
5. Keluar  
6. Hitung Total Bantuan  
Pilih menu: 6  
Total semua bantuan: 0
```

Keluar

```
=== MENU UTAMA ===  
1. Tambah Data Donasi  
2. Tampilkan Data Donasi  
3. Ubah Data Donasi  
4. Hapus Data Donasi  
5. Keluar  
6. Hitung Total Bantuan  
Pilih menu: 5  
Terima kasih, Program selesai
```

5. Langkah-Langkah Git pada VSCode

```
Microsoft Windows [Version 10.0.26100.3323]
C:\Users\ACER\Downloads\Praktikum-APL>git init
Initialized empty Git repository in C:/Users/ACER/Downloads/Praktikum-APL/.git/

C:\Users\ACER\Downloads\Praktikum-APL>git add .

C:\Users\ACER\Downloads\Praktikum-APL>git commit -m "Finish Post Test 4"
[main (root-commit) 4c95f06] Finish Post Test 4
 2 files changed, 73 insertions(+)
 create mode 100644 Post-Test/main.cpp
 create mode 100644 Post-Test/main.exe

C:\Users\ACER\Downloads\Praktikum-APL>git remote add origin https://github.com/Diego444666/Praktikum-APL.git

C:\Users\ACER\Downloads\Praktikum-APL>git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 671.23 KiB | 8.09 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Diego444666/Praktikum-APL.git
 * [new branch]      main -> main
 branch 'main' set up to track 'origin/main'.

C:\Users\ACER\Downloads\Praktikum-APL>
```

git init : Menginisialisasi repository Git di dalam folder proyek.

Perintah ini membuat folder git add : Menambahkan semua file dalam proyek ke staging area, yang berarti file sudah siap untuk dikomit

git commit -m "Finish Post Test 4" : Menyimpan perubahan ke dalam repository Git dengan sebuah pesan deskriptif.

git remote add origin https://github.com/Diego444666/Praktikum-APL.git : Menghubungkan repository lokal dengan repository yang ada di GitHub. URL repository harus sesuai dengan yang dibuat di GitHub

. git push -u origin main : Mengunggah semua file yang telah dikomit ke repository GitHub pada branch