

LAPORAN PRAKTIKUM
POSTTEST (3)
ALGORITMA PEMROGRAMAN LANJUT



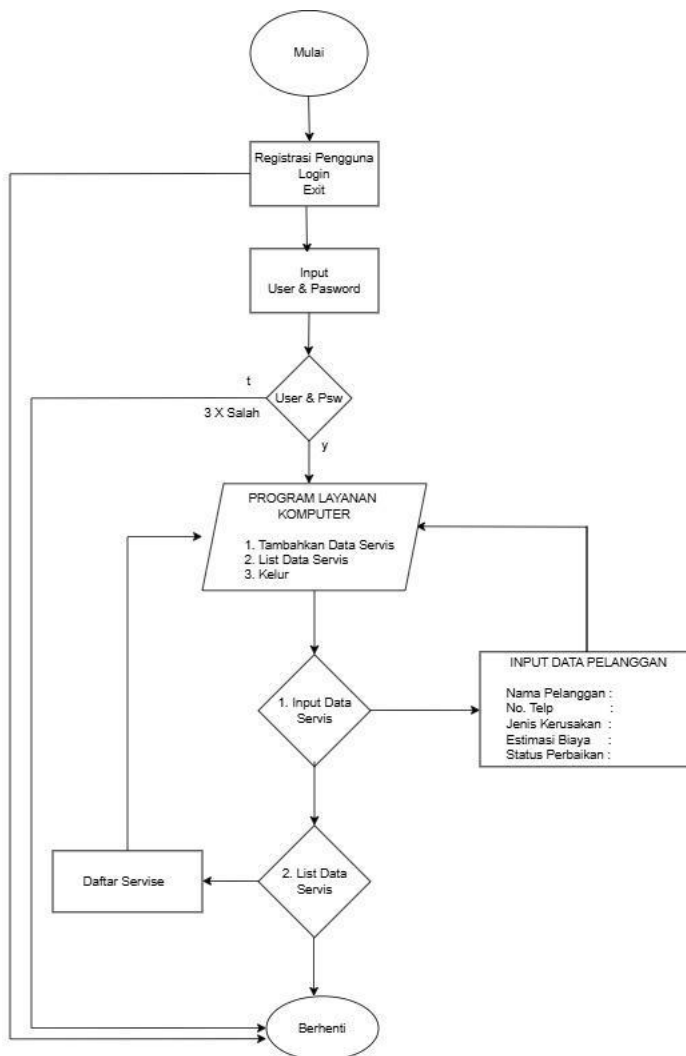
Disusun oleh:

Nama (2409106101)

Kelas (C1 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



2. Analisis Program

- Multiuser: Program saat dijalankan akan tampil registrasi pengguna terlebih dahulu sebelum login program. pada menu utama terdapat 3 submenu yaitu registrasi pengguna, login, dan keluar
- Create:
 1. Membuat User baru dengan memilih registrasi pengguna dimana akan tampil menu registrasi pengguna yang akan menambahkan user baru dan password pengguna.
 2. Menambahkan data servis dimana akan tampil nama pelanggan, nomor telepon, jenis kerusakan, dan estimasi biaya.
- Read: Membaca data hasil penambahan data servis, dimana akan menampilkan daftar servis.
- Update: Update Status Servis, dimana akan tampil nomor id servis, setelah memasukkan id servis akan tampil status saat ini (Diterima, Perbaikan, Selesai). Silahkan update status terbaru servis.
- Delete: Menu Delete akan menampilkan hapus data servis yang sudah selesai diperbaiki. dimana akan tampil nomor id servis yang akan di hapus.

3. Source Code

Source Code:

```
#include <iostream>
#include <string>
#include <stdlib.h>
using namespace std;

const int MAX_ACCOUNT = 10;
const int MAX_SERVIS = 100;

struct Account {
    string username;
    string password;
};

struct Servis {
    int idServis;
    string namaPelanggan;
    string jenisKerusakan;
    string nomorKontak;
    string status; // "Diterima", "Perbaikan", "Selesai"
    string biayaEstimasi;
};

// Data admin
const string ADMIN_USERNAME = "Aditya Mahyudi Ramadhan";
const string ADMIN_PASSWORD = "2409106101";

Account accounts[MAX_ACCOUNT];
int totalAccounts = 0; // jumlah akun yang sudah terdaftar
Servis dataServis[MAX_SERVIS];
int jumlahServis = 0;

int main() {
    int pilihanAwal;
    do {
        system("cls");
        cout << "=====\n";
        cout << "          SELAMAT DATANG DI SISTEM          \n";
        cout << "=====\n";
        cout << "1. Registrasi Account\n";
        cout << "2. Login\n";
        cout << "3. Keluar\n";
        cout << "=====\n";
        cout << "Pilih menu: ";
        cin >> pilihanAwal;
        cin.ignore();
    } while (pilihanAwal < 4);
}
```

```

    if (pilihanAwal == 1) {
        if (totalAccounts < MAX_ACCOUNT) {
            system("cls");
            cout << "=====\n";
            cout << "          REGISTRASI ACCOUNT BARU          \n";
            cout << "=====\n";
            cout << "Masukkan Username: ";
            getline(cin, accounts[totalAccounts].username);

            // Cek jika username adalah admin
            if (accounts[totalAccounts].username == ADMIN_USERNAME) {
                cout << "\nUsername 'admin' tidak dapat digunakan untuk
registrasi.\n";
                system("pause");
                continue;
            }

            cout << "Masukkan Password: ";
            getline(cin, accounts[totalAccounts].password);
            totalAccounts++;
            cout << "\nRegistrasi berhasil!\n";
        } else {
            cout << "\nKapasitas registrasi sudah penuh.\n";
        }
        system("pause");
    } else if (pilihanAwal == 2) {
        // Proses Login
        int attempts = 0;
        bool loginBerhasil = false;
        bool isAdmin = false;

        while (attempts < 3 && !loginBerhasil) {
            system("cls");
            cout << "=====\n";
            cout << "          LOGIN SISTEM          \n";
            cout << "=====\n";
            string inputUsername, inputPassword;
            cout << "Username: "; getline(cin, inputUsername);
            cout << "Password: "; getline(cin, inputPassword);

            // Cek Login admin terlebih dahulu
            if (inputUsername == ADMIN_USERNAME && inputPassword ==
ADMIN_PASSWORD) {
                loginBerhasil = true;
                isAdmin = true;
                break;
            }

            // Cek Login user biasa
            for (int i = 0; i < totalAccounts; i++) {
                if (accounts[i].username == inputUsername &&

```

```

accounts[i].password == inputPassword) {
    loginBerhasil = true;
    break;
}
}

if (loginBerhasil) {
    cout << "\nLogin berhasil! Selamat datang, " <<
inputUsername << "!\n";
    system("pause");
    break;
} else {
    attempts++;
    cout << "\nUsername atau password salah! Percobaan tersisa:
" << (3 - attempts) << "\n";
    system("pause");
}
}

if (!loginBerhasil) {
    cout << "\nTerlalu banyak percobaan gagal. Program keluar.\n";
    return 0;
}

// Menu utama sistem servis
int pilih;
do {
    system("cls");
    cout << "\n=====\\n";
    cout << "|          PROGRAM SERVIS KOMPUTER          |\\n";
    cout << "=====\\n";
    if (isAdmin) {
        cout << "|          ANDA LOGIN SEBAGAI ADMIN          |\\n";
        cout << "=====\\n";
    }
    cout << "| 1. Tambah Data Servis                      |\\n";
    cout << "| 2. Tampilkan Data Servis                   |\\n";
    cout << "| 3. Update Data Servis                     |\\n";
    cout << "| 4. Delete Data Servis                    |\\n";
    cout << "| 5. Keluar                                |\\n";
    cout << "=====\\n";
    cout << "  Pilih menu : ";
    cin >> pilih;
    cin.ignore();

    if (pilih == 1) {
        if (jumlahServis < MAX_SERVIS) {
            dataServis[jumlahServis].idServis = jumlahServis + 1;
            system("cls");
            cout <<
"\n=====\\n";

```

```

        cout << "                Input Data Pelanggan \n";
        cout << "=====\\n";
        cout << "    Masukkan Nama Pelanggan    : ";
        getline(cin, dataServis[jumlahServis].namaPelanggan);
        cout << "    Masukkan No. Telepon        : ";
        getline(cin, dataServis[jumlahServis].nomorKontak);
        cout << "    Masukkan Jenis Kerusakan    : ";
        getline(cin, dataServis[jumlahServis].jenisKerusakan);
        cout << "    Estimasi Biaya                Rp. ";
        getline(cin, dataServis[jumlahServis].biayaEstimasi);
        cout << "    Status Perbaikan              : ";
        getline(cin, dataServis[jumlahServis].status);
        jumlahServis++;
        cout << "=====\\n";
        cout << "    Data berhasil ditambahkan!\\n";
        cout << "=====\\n";
    } else {
        cout << "Data servis penuh, tidak dapat menambahkan
lagi.\\n";
    }
    system("pause");
} else if (pilih == 2) {
    if (jumlahServis == 0) {
        cout << "Tidak ada data servis.\\n";
    } else {
        cout << "\\nDaftar Servis Komputer:\\n";
        cout <<
"=====\\n";
        for (int i = 0; i < jumlahServis; i++) {
            cout << "| No. ID : " << dataServis[i].idServis
                << " | Pelanggan : " <<
dataServis[i].namaPelanggan
                << " | No. Tlp. : " <<
dataServis[i].nomorKontak
                << " | Kerusakan : " <<
dataServis[i].jenisKerusakan
                << " | Estimasi Biaya Rp. " <<
dataServis[i].biayaEstimasi
                << " | Status : " << dataServis[i].status <<
endl;
        }
        cout <<
"=====\\n";
    }
    system("pause");
} else if (pilih == 3) { // Update Data Servis
    if (jumlahServis == 0) {
        cout << "Tidak ada data servis yang dapat diupdate.\\n";
    } else {
        int idUpdate;
        cout << "Masukkan No. ID Servis yang akan diupdate: ";

```

```

        cin >> idUpdate;
        cin.ignore();
        bool found = false;
        for (int i = 0; i < jumlahServis; i++) {
            if (dataServis[i].idServis == idUpdate) {
                found = true;
                system("cls");
                cout <<
"\n=====\\n";
                cout << "                Update Data Servis
\\n";
                cout <<
"=====\\n";
                cout << "Data lama:\\n";
                cout << "Nama Pelanggan : " <<
dataServis[i].namaPelanggan << endl;
                cout << "No. Telepon : " <<
dataServis[i].nomorKontak << endl;
                cout << "Jenis Kerusakan : " <<
dataServis[i].jenisKerusakan << endl;
                cout << "Estimasi Biaya : Rp. " <<
dataServis[i].biayaEstimasi << endl;
                cout << "Status : " <<
dataServis[i].status << endl;

                cout << "\\nMasukkan data baru:\\n";
                cout << "Nama Pelanggan : "; getline(cin,
dataServis[i].namaPelanggan);
                cout << "No. Telepon : "; getline(cin,
dataServis[i].nomorKontak);
                cout << "Jenis Kerusakan : "; getline(cin,
dataServis[i].jenisKerusakan);
                cout << "Estimasi Biaya : Rp. "; getline(cin,
dataServis[i].biayaEstimasi);
                cout << "Status : "; getline(cin,
dataServis[i].status);

                cout << "\\nData berhasil diupdate!\\n";
                break;
            }
        }
        if (!found) {
            cout << "Data dengan ID tersebut tidak
ditemukan.\\n";
        }
    }
    system("pause");
} else if (pilih == 4) { // Delete Data Servis
    if (jumlahServis == 0) {
        cout << "Tidak ada data servis yang dapat dihapus.\\n";
    } else {
        int idDelete;

```



```

        cout << "Masukkan No. ID Servis yang akan dihapus: ";
        cin >> idDelete;
        cin.ignore();
        bool found = false;
        for (int i = 0; i < jumlahServis; i++) {
            if (dataServis[i].idServis == idDelete) {
                found = true;
                // Geser data untuk menimpa data yang dihapus
                for (int j = i; j < jumlahServis - 1; j++) {
                    dataServis[j] = dataServis[j + 1];
                }
                jumlahServis--;
                cout << "Data berhasil dihapus!\n";
                break;
            }
        }
        if (!found) {
            cout << "Data dengan ID tersebut tidak
ditemukan.\n";
        }
    }
    system("pause");
} else if (pilih == 5) {
    cout << "Terima kasih telah menggunakan sistem ini!\n";
    system("pause");
    break;
} else {
    cout << "Pilihan tidak valid! Silakan coba lagi.\n";
    system("pause");
}
} while (true);
break; // keluar dari menu registrasi/login setelah login dan
penggunaan sistem servis
} else if (pilihanAwal == 3) {
    cout << "\nTerima kasih. Program selesai.\n";
    system("pause");
    return 0;
} else {
    cout << "\nPilihan tidak valid! Silakan coba lagi.\n";
    system("pause");
}
} while (true);

return 0;
}

```

4. Uji Coba dan Hasil Output

```
=====
                        SELAMAT DATANG DI SISTEM
=====
1. Registrasi Account
2. Login
3. Keluar
=====
Pilih menu: █
```

Gambar 4.1 Tampilan Menu Registrasi

```
=====
                        REGISTRASI ACCOUNT BARU
=====
Masukkan Username: Adit
Masukkan Password: Adit101

Registrasi berhasil!
Press any key to continue . . . █
```

Gambar 4.2 Registrasi Akun User

```
=====
                        LOGIN SISTEM
=====
Username: Adit
Password: Adit101

Login berhasil! Selamat datang, Adit!
Press any key to continue . . . █
```

Gambar 4.3 Login User

```
=====
|          PROGRAM SERVIS KOMPUTER          |
=====
| 1. Tambah Data Servis                    |
| 2. Tampilkan Data Servis                 |
| 3. Update Data Servis                   |
| 4. Delete Data Servis                   |
| 5. Keluar                               |
=====
Pilih menu : █
```

Gambar 4.4 Tampilan Menu Utama

```
=====
                        Input Data Pelanggan
=====
Masukkan Nama Pelanggan : Adit
Masukkan No. Telepon    : 0812359817
Masukkan Jenis Kerusakan : Mati
Estimasi Biaya          Rp. 105000
Status Perbaikan        : Diterima
=====
                        Data berhasil ditambahkan!
=====
Press any key to continue . . .
```

Gambar 4.5 Menambahkan Data Pelanggan

```
Daftar Servis Komputer:
=====
| No. ID : 1 | Pelanggan : Adit | No. Tlp. : 0812359817 | Kerusakan : Mati | Estimasi Biaya Rp. 105000 | Status : Diterima
=====
Press any key to continue . . .
```

Gambar 4.6 Menampilkan Daftar Servis

```
=====
                        Update Data Servis
=====
Data lama:
Nama Pelanggan : Adit
No. Telepon    : 0812359817
Jenis Kerusakan : Mati
Estimasi Biaya : Rp. 105000
Status         : Diterima

Masukkan data baru:
Nama Pelanggan : Adit
No. Telepon    : 085715749760
Jenis Kerusakan : Laptop Mati
Estimasi Biaya : Rp. 125000
Status         : Perbaikan

Data berhasil diupdate!
Press any key to continue . . .
```

Gambar 4.7 Mengubah Data Servis

```
=====
|          PROGRAM SERVIS KOMPUTER          |
=====
| 1. Tambah Data Servis                    |
| 2. Tampilkan Data Servis                 |
| 3. Update Data Servis                   |
| 4. Delete Data Servis                   |
| 5. Keluar                               |
=====
Pilih menu : 4
Masukkan No. ID Servis yang akan dihapus: 1
Data berhasil dihapus!
Press any key to continue . . .
```

Gambar 4.8 Menghapus Data Servis

5. Langkah-Langkah Git pada VSCode

```
USER@DESKTOP-NDADPFO MINGW64 /d/Tugas KK Adit/pratikum-apl/post-test/post-test-3 (master)
$ git add .

USER@DESKTOP-NDADPFO MINGW64 /d/Tugas KK Adit/pratikum-apl/post-test/post-test-3
(master)
$ git commit -m "finish posttest 3"
[master b60c20c] finish posttest 3
2 files changed, 265 insertions(+)
create mode 100644 post-test/post-test-3/2409106101-AdityaMahyudiRamadhan-PT-3.
cpp
create mode 100644 post-test/post-test-3/2409106101-AdityaMahyudiRamadhan-PT-3.
exe

USER@DESKTOP-NDADPFO MINGW64 /d/Tugas KK Adit/pratikum-apl/post-test/post-test-3
(master)
$ git push
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 680.12 KiB | 2.42 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Adityamr2005/pratikum-apl.git
d6b6b38..b60c20c master -> master

USER@DESKTOP-NDADPFO MINGW64 /d/Tugas KK Adit/pratikum-apl/post-test/post-test-3
(master)
$ git push -u origin master
branch 'master' set up to track 'origin/master'.
Everything up-to-date

USER@DESKTOP-NDADPFO MINGW64 /d/Tugas KK Adit/pratikum-apl/post-test/post-test-3
(master)
$ |
```

Langkah-Langkah:

1. Masuk ke Folder Repository
2. Menambahkan Perubahan ke Staging Area
3. Melakukan Commit
4. Mengunggah Perubahan ke Repository Remote
5. Menetapkan Tracking Branch ke Remote Repository

Penjelasan:

- Menambahkan perubahan ke Git (`git add .`).
- Membuat commit (`git commit -m "finish posttest 3"`).
- Mengunggah commit ke GitHub (`git push`).
- Menghubungkan branch lokal dengan remote (`git push -u origin master`).