

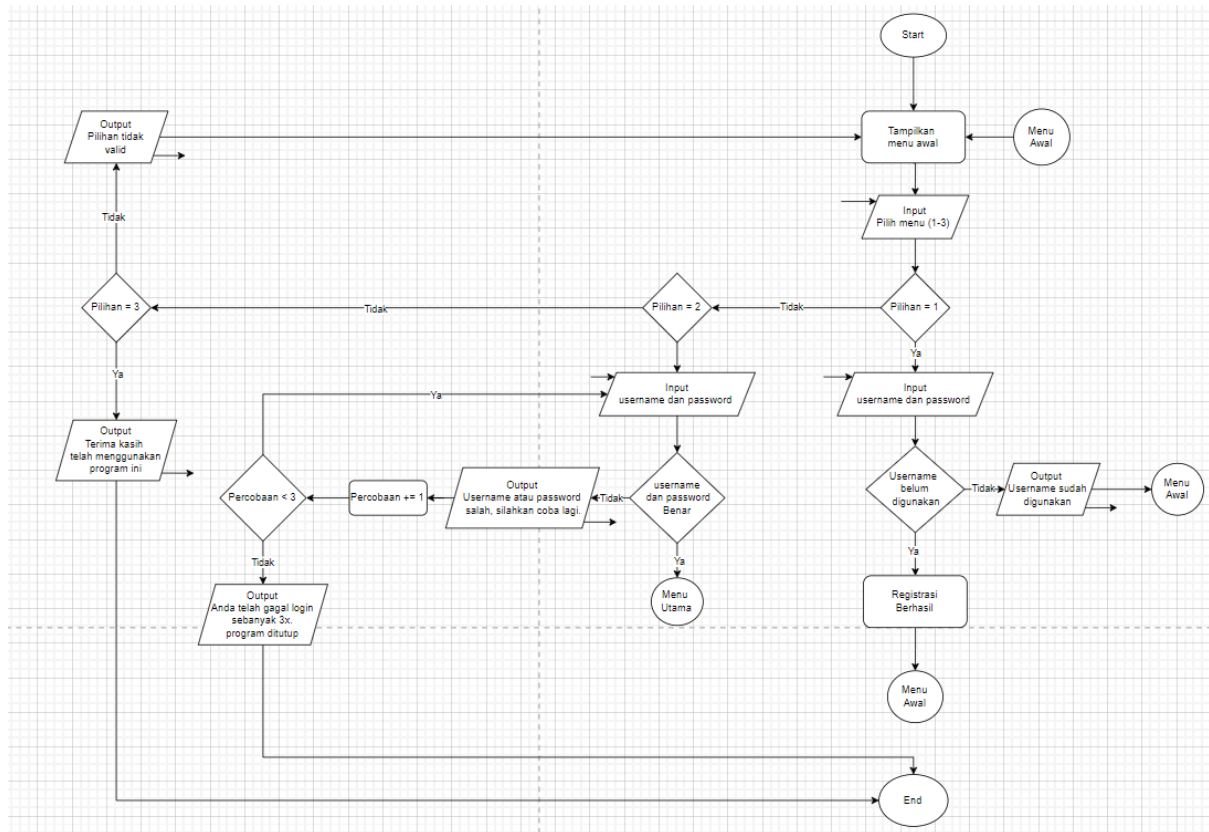
LAPORAN PRAKTIKUM
POSTTEST 4
ALGORITMA PEMROGRAMAN LANJUT



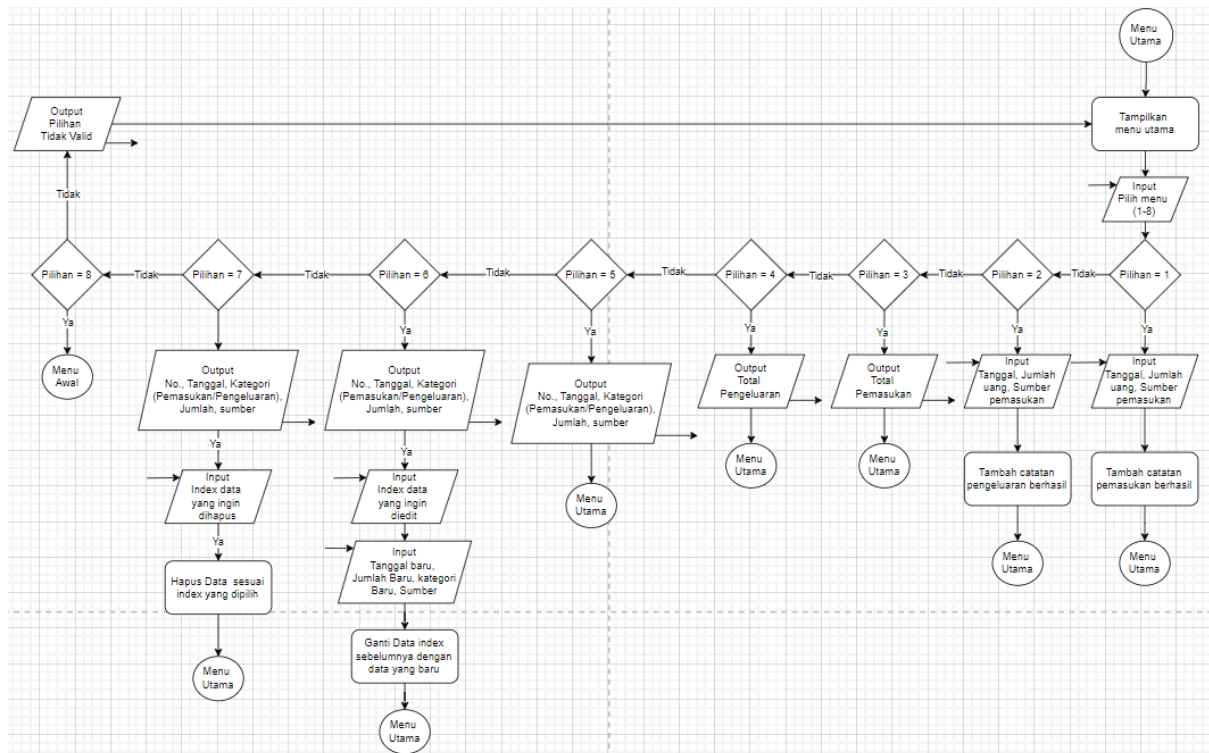
Disusun oleh:
Muhammad Rafli Pernanda 2409106040
Kelas A2 '24

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



Gambar 1.1 Flowchart Menu Awal



Gambar 1.2 Flowchart Menu Utama

2. Analisis Program

Program ini dirancang untuk membantu pengguna dalam mencatat dan mengelola keuangan secara lebih terstruktur dan efisien. Setiap pengguna dapat melakukan registrasi akun terlebih dahulu, kemudian login untuk mengakses menu utama program. Terdapat pembatasan jumlah percobaan login maksimal tiga kali sebagai upaya pengamanan agar akun tidak disalahgunakan.

Setelah berhasil login, pengguna dapat mengakses berbagai fitur yang tersedia seperti menambahkan catatan pemasukan atau pengeluaran, melihat total dari masing-masing jenis transaksi, serta melihat seluruh catatan yang telah dibuat. Setiap catatan terdiri dari informasi seperti tanggal transaksi, jumlah uang, kategori (pemasukan/pengeluaran), dan sumber dana.

Program ini juga dilengkapi dengan fitur untuk mengelola data yang telah dicatat. Pengguna dapat memperbarui catatan yang telah dimasukkan jika terdapat kesalahan, maupun menghapus catatan yang sudah tidak diperlukan. Sebelum melakukan proses edit atau hapus, program akan menampilkan seluruh catatan terlebih dahulu agar mempermudah pengguna dalam memilih indeks catatan yang ingin dikelola.

Secara teknis, program dibuat lebih modular dengan memanfaatkan fungsi-fungsi terpisah seperti untuk menampilkan menu awal, login, registrasi, hingga manajemen catatan keuangan. Hal ini bertujuan agar struktur kode menjadi lebih rapi dan mudah dibaca.

Dengan berbagai fitur ini, diharapkan pengguna dapat lebih mudah dalam mencatat transaksi keuangan secara berkala, menjaga kerapian data, dan memahami kondisi finansial mereka dengan lebih baik.

3. Source Code

A. Deklarasi Variable

Bagian ini berisi deklarasi struct, array global, dan konstanta yang digunakan untuk menyimpan data pengguna serta catatan keuangan.

Source Code:

```
#include <iostream>
#include <iomanip>
#include <string>

using namespace std;

const int MAX_USERS = 100;
const int MAX_CATATAN = 100;

struct User {
    string username;
    string password;
};

struct Catatan {
    string tanggal;
    string kategori;
    double jumlah;
    string sumber;
};

User users[MAX_USERS] = {"pernanda", "040"};
Catatan catatan[MAX_CATATAN];

int totalUsers = 1;
int totalCatatan = 0;
```

B. Menu Awal

Menampilkan tampilan awal program dan meminta input dari user untuk memilih registrasi, login, atau keluar dari program.

Source Code:

```
void MenuAwal() {
    cout << "=====\n";
    cout << "|          MENU AWAL          |\n";
    cout << "=====\n";
}
```

```

    cout << "| 1. Registrasi           |\n";
    cout << "| 2. Login             |\n";
    cout << "| 3. Keluar            |\n";
    cout << "===== \n";
}

```

C. Registrasi User

Digunakan untuk menambahkan akun user baru ke dalam array *users* jika username belum pernah digunakan

Source Code:

```

void registrasi() {
    if (totalUsers >= MAX_USERS) {
        cout << "User sudah penuh!\n";
        return;
    }

    string uname, pass;
    cout << "Masukkan username baru: ";
    cin >> uname;

    if (usernameSudahAda(uname)) {
        cout << "Username sudah digunakan!\n";
        return;
    }

    cout << "Masukkan password: ";
    cin >> pass;

    users[totalUsers++] = {uname, pass};
    cout << "Registrasi berhasil!\n\n";
}

```

D. Login User

Memverifikasi username dan password, jika username dan password benar, maka akan masuk ke dalam menu utama, serta membatasi percobaan login maksimal tiga kali.

Source Code:

```

bool login(string& username) {
    string inputUser, inputPass;
    int attempts = 0;
}

```

```

while (attempts < 3) {
    cout << "Masukkan Username: ";
    cin >> inputUser;
    cout << "Masukkan Password: ";
    cin >> inputPass;

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

    attempts++;
    cout << "Login gagal! Sisa percobaan: " << (3 - attempts) << "\\n\\n";
}

return false;
}

```

E. Menambah Catatan

Digunakan untuk mencatat transaksi baru, baik pemasukan maupun pengeluaran .

Source Code:

```

void tambahCatatan(bool pemasukan) {
    if (totalCatatan >= MAX_CATATAN) {
        cout << "Penyimpanan penuh!\\n";
        return;
    }

    cin.ignore();
    cout << "Masukkan tanggal (YYYY-MM-DD): ";
    getline(cin, catatan[totalCatatan].tanggal);
    cout << "Masukkan jumlah uang: ";
    cin >> catatan[totalCatatan].jumlah;
    cin.ignore();
    cout << "Masukkan sumber: ";
    getline(cin, catatan[totalCatatan].sumber);
    catatan[totalCatatan].kategori = pemasukan ? "Pemasukan" : "Pengeluaran";

    totalCatatan++;
    cout << "Catatan berhasil ditambahkan!\\n\\n";
}

```

F. Melihat Total Pemasukan / Pengeluaran

Menjumlahkan semua nilai jumlah dari catatan berdasarkan kategori tertentu.

Source Code:

```
void lihatTotal(bool pemasukan) {
    double total = 0;
    string tipe = pemasukan ? "Pemasukan" : "Pengeluaran";

    for (int i = 0; i < totalCatatan; i++) {
        if (catatan[i].kategori == tipe) {
            total += catatan[i].jumlah;
        }
    }

    cout << "Total " << tipe << ": Rp" << total << "\n\n";
}
```

G. Menampilkan Semua Catatan

Menampilkan seluruh catatan yang telah ditambahkan dalam bentuk tabel.

Source Code:

```
void tampilkanSemuaCatatan() {
    if (totalCatatan == 0) {
        cout << "Belum ada catatan.\n\n";
        return;
    }

    cout << "\n===== Daftar Catatan =====\n";
    cout <<
    "+-----+-----+-----+-----+-----+\n";
    cout << "| No | Tanggal | Kategori | Jumlah | Sumber\n";
    cout <<
    "+-----+-----+-----+-----+-----+\n";

    for (int i = 0; i < totalCatatan; i++) {
        cout << "| " << setw(2) << i + 1
        << " | " << setw(10) << catatan[i].tanggal
        << " | " << setw(10) << catatan[i].kategori
        << " | " << setw(10) << catatan[i].jumlah
        << " | " << setw(16) << catatan[i].sumber
        << " |\n";
    }
}
```

```

        cout <<
        "+-----+-----+-----+-----+-----+\n\n";
    }

```

H. Memperbarui Catatan

Menampilkan semua catatan terlebih dahulu, lalu memperbarui data sesuai indeks yang dipilih user.

Source Code:

```

void editCatatan() {
    tampilkanSemuaCatatan();

    if (totalCatatan == 0) return;

    int index;
    cout << "Masukkan nomor catatan yang ingin diedit: ";
    cin >> index;

    if (index < 1 || index > totalCatatan) {
        cout << "Nomor tidak valid!\n\n";
        return;
    }

    index--;
    cin.ignore();
    cout << "Masukkan tanggal baru: ";
    getline(cin, catatan[index].tanggal);
    cout << "Masukkan jumlah baru: ";
    cin >> catatan[index].jumlah;
    cin.ignore();
    cout << "Masukkan sumber baru: ";
    getline(cin, catatan[index].sumber);

    cout << "Catatan berhasil diperbarui!\n\n";
}

```

I. Menghapus Catatan

Menampilkan semua catatan terlebih dahulu, lalu menghapus catatan sesuai indeks yang dimasukkan user.

Source Code:

```

void hapusCatatan() {

```



```

    tampilkanSemuaCatatan();

    if (totalCatatan == 0) return;

    int index;
    cout << "Masukkan nomor catatan yang ingin dihapus: ";
    cin >> index;

    if (index < 1 || index > totalCatatan) {
        cout << "Nomor tidak valid!\n\n";
        return;
    }

    for (int i = index - 1; i < totalCatatan - 1; i++) {
        catatan[i] = catatan[i + 1];
    }
    totalCatatan--;

    cout << "Catatan berhasil dihapus!\n\n";
}

```

J. Tampilan Menu Utama

Menampilkan menu utama setelah user berhasil login. Berisi berbagai pilihan fitur pengelolaan catatan keuangan.

Source Code:

```

void menuUtama() {
    int pilihan;
    while (true) {
        cout << "=====\n";
        cout << "|    Aplikasi Catatan Keuangan    |\n";
        cout << "=====\n";
        cout << "| 1. Tambah Catatan Pemasukan      |\n";
        cout << "| 2. Tambah Catatan Pengeluaran    |\n";
        cout << "| 3. Lihat Total Pemasukan          |\n";
        cout << "| 4. Lihat Total Pengeluaran        |\n";
        cout << "| 5. Lihat Semua Catatan            |\n";
        cout << "| 6. Perbarui/Edit Catatan          |\n";
        cout << "| 7. Hapus Catatan                  |\n";
        cout << "| 8. Logout                         |\n";
        cout << "=====\n";
        cout << "Pilih menu (1-8): ";
        cin >> pilihan;

        switch (pilihan) {
            case 1: tambahCatatan(true); break;

```

```

        case 2: tambahCatatan(false); break;
        case 3: lihatTotal(true); break;
        case 4: lihatTotal(false); break;
        case 5: tampilkanSemuaCatatan(); break;
        case 6: editCatatan(); break;
        case 7: hapusCatatan(); break;
        case 8: cout << "\nLogout berhasil!\n\n"; return;
        default: cout << "Pilihan tidak valid!\n\n";
    }
}
}

```

K. Fungsi main()

Fungsi main() merupakan titik awal program berjalan. DI dalamnya terdapat loop utama yang akan terus menampilkan menu awal hingga pengguna memilih untuk keluar. Jika pengguna berhasil login, maka akan diarahkan ke menuUtama() untuk mengakses fitur-fitur utama.

Source Code:

```

int main() {
    int pilihan;
    string usernameLogin;

    while (true) {
        MenuAwal();
        cout << "Pilih menu (1-3): ";
        cin >> pilihan;

        switch (pilihan) {
            case 1: registrasi(); break;
            case 2:
                if (login(usernameLogin)) {
                    menuUtama();
                } else {
                    cout << "Anda gagal login 3 kali. Program dihentikan.\n";
                    return 0;
                }
                break;
            case 3:
                cout << "Terima kasih karena telah menggunakan program ini\n";
                cout << "-----Muhammad Rafli Pernanda-----\n";
                cout << "-----2409106040-----\n";
                return 0;
            default: cout << "Pilihan tidak valid!\n\n";
        }
    }
}

```

}

4. Uji Coba dan Hasil Output

```
=====
|          MENU AWAL          |
=====
| 1. Registrasi               |
| 2. Login                   |
| 3. Keluar                   |
=====
Pilih menu (1-3): 1
Masukkan username baru: Rafli
Masukkan password: 123
Registrasi berhasil!

=====
|          MENU AWAL          |
=====
| 1. Registrasi               |
| 2. Login                   |
| 3. Keluar                   |
=====
Pilih menu (1-3): 2
Masukkan Username: Rafli
Masukkan Password: 123
Login berhasil! Selamat datang, Rafli!

=====
| Aplikasi Catatan Keuangan   |
=====
| 1. Tambah Catatan Pemasukan |
| 2. Tambah Catatan Pengeluaran |
| 3. Lihat Total Pemasukan    |
| 4. Lihat Total Pengeluaran  |
| 5. Lihat Semua Catatan     |
| 6. Perbarui/Edit Catatan   |
| 7. Hapus Catatan           |
| 8. Logout                   |
=====
Pilih menu (1-8): █
```

Gambar 4.1 Menu awal, registrasi, dan login

```
=====
|      Aplikasi Catatan Keuangan      |
=====
| 1. Tambah Catatan Pemasukan          |
| 2. Tambah Catatan Pengeluaran        |
| 3. Lihat Total Pemasukan              |
| 4. Lihat Total Pengeluaran            |
| 5. Lihat Semua Catatan                |
| 6. Perbarui/Edit Catatan              |
| 7. Hapus Catatan                     |
| 8. Logout                            |
=====
Pilih menu (1-8): 1
Masukkan tanggal (YYYY-MM-DD): 2025-04-10
Masukkan jumlah uang: 150000
Masukkan sumber: THR
Catatan berhasil ditambahkan!

=====
|      Aplikasi Catatan Keuangan      |
=====
| 1. Tambah Catatan Pemasukan          |
| 2. Tambah Catatan Pengeluaran        |
| 3. Lihat Total Pemasukan              |
| 4. Lihat Total Pengeluaran            |
| 5. Lihat Semua Catatan                |
| 6. Perbarui/Edit Catatan              |
| 7. Hapus Catatan                     |
| 8. Logout                            |
=====
Pilih menu (1-8): 2
Masukkan tanggal (YYYY-MM-DD): 2025-04-10
Masukkan jumlah uang: 20000
Masukkan sumber: makan
Catatan berhasil ditambahkan!
```

Gambar 4.2 Menambah Catatan Pemasukan & Pengeluaran

```
=====
|      Aplikasi Catatan Keuangan      |
=====
| 1. Tambah Catatan Pemasukan          |
| 2. Tambah Catatan Pengeluaran        |
| 3. Lihat Total Pemasukan              |
| 4. Lihat Total Pengeluaran            |
| 5. Lihat Semua Catatan                |
| 6. Perbarui/Edit Catatan              |
| 7. Hapus Catatan                     |
| 8. Logout                            |
=====
Pilih menu (1-8): 3
Total Pemasukan: Rp150000

=====
|      Aplikasi Catatan Keuangan      |
=====
| 1. Tambah Catatan Pemasukan          |
| 2. Tambah Catatan Pengeluaran        |
| 3. Lihat Total Pemasukan              |
| 4. Lihat Total Pengeluaran            |
| 5. Lihat Semua Catatan                |
| 6. Perbarui/Edit Catatan              |
| 7. Hapus Catatan                     |
| 8. Logout                            |
=====
Pilih menu (1-8): 4
Total Pengeluaran: Rp20000
```

Gambar 4.3 Melihat total pemasukan & pengeluaran

```
=====
|      Aplikasi Catatan Keuangan      |
=====
| 1. Tambah Catatan Pemasukan          |
| 2. Tambah Catatan Pengeluaran        |
| 3. Lihat Total Pemasukan              |
| 4. Lihat Total Pengeluaran            |
| 5. Lihat Semua Catatan                |
| 6. Perbarui/Edit Catatan              |
| 7. Hapus Catatan                      |
| 8. Logout                            |
=====
Pilih menu (1-8): 5

===== Daftar Catatan =====
+---+-----+-----+-----+-----+
| No | Tanggal | Kategori | Jumlah | Sumber |
+---+-----+-----+-----+-----+
| 1 | 2025-04-10 | Pemasukan | 150000 | THR |
| 2 | 2025-04-10 | Pengeluaran | 20000 | makan |
+---+-----+-----+-----+-----+
```

Gambar 4.4 Menampilkan semua catatan

```
=====
|      Aplikasi Catatan Keuangan      |
=====
| 1. Tambah Catatan Pemasukan          |
| 2. Tambah Catatan Pengeluaran        |
| 3. Lihat Total Pemasukan             |
| 4. Lihat Total Pengeluaran           |
| 5. Lihat Semua Catatan              |
| 6. Perbarui/Edit Catatan            |
| 7. Hapus Catatan                   |
| 8. Logout                          |
=====
Pilih menu (1-8): 6

===== Daftar Catatan =====
+---+---+---+---+---+
| No | Tanggal | Kategori | Jumlah | Sumber |
+---+---+---+---+---+
| 1 | 2025-04-10 | Pemasukan | 150000 | THR |
| 2 | 2025-04-10 | Pengeluaran | 20000 | makan |
+---+---+---+---+---+

Masukkan nomor catatan yang ingin diedit: 1
Masukkan tanggal baru: 2025-04-09
Masukkan jumlah baru: 50000
Masukkan sumber baru: THR
Catatan berhasil diperbarui!
```

Gambar 4.5 Mengedit catatan


```
=====
|      Aplikasi Catatan Keuangan      |
=====
| 1. Tambah Catatan Pemasukan         |
| 2. Tambah Catatan Pengeluaran       |
| 3. Lihat Total Pemasukan            |
| 4. Lihat Total Pengeluaran          |
| 5. Lihat Semua Catatan              |
| 6. Perbarui/Edit Catatan            |
| 7. Hapus Catatan                   |
| 8. Logout                          |
=====
Pilih menu (1-8): 7

===== Daftar Catatan =====
+---+-----+-----+-----+-----+
| No | Tanggal | Kategori | Jumlah | Sumber |
+---+-----+-----+-----+-----+
| 1 | 2025-04-09 | Pemasukan | 50000 | THR |
| 2 | 2025-04-10 | Pengeluaran | 20000 | makan |
+---+-----+-----+-----+-----+

Masukkan nomor catatan yang ingin dihapus: 1
Catatan berhasil dihapus!
```

Gambar 4.6 Menghapus catatan

```

=====
|      Aplikasi Catatan Keuangan      |
=====
|  1. Tambah Catatan Pemasukan        |
|  2. Tambah Catatan Pengeluaran      |
|  3. Lihat Total Pemasukan           |
|  4. Lihat Total Pengeluaran         |
|  5. Lihat Semua Catatan             |
|  6. Perbarui/Edit Catatan           |
|  7. Hapus Catatan                  |
|  8. Logout                          |
=====
Pilih menu (1-8): 8

Logout berhasil!

=====
|              MENU AWAL              |
=====
|  1. Registrasi                      |
|  2. Login                          |
|  3. Keluar                         |
=====
Pilih menu (1-3): 3
Terima kasih karena telah menggunakan program ini
-----Muhammad Rafli Pernanda-----
-----2409106040-----
PS D:\praktikum-apl>

```

Gambar 4.7 Logout dan keluar dari program

5. Langkah-Langkah Git pada VSCode

1. Git add .

Digunakan untuk menambahkan semua perubahan (termasuk file baru, modifikasi, dan penghapusan) dalam direktori kerja ke staging area, yang berarti file tersebut siap untuk di commit

```
● PS D:\praktikum-apl> git add .
```

Gambar 5.1 git add .

2. Git commit -m "Selesai source code dan 90% Laporan"

Digunakan untuk menyimpan perubahan yang ada di staging area ke repositori lokal, dengan menyertakan pesan commit yang menjelaskan perubahan tersebut.

```
● PS D:\praktikum-apl> git commit -m "Selesai source code dan 90% Laporan"
[master f2dd33d] Selesai source code dan 90% Laporan
3 files changed, 257 insertions(+)
create mode 100644 post-test/post-test-4/2409106040-MuhammadRaflipernanda-PT-4.cpp
create mode 100644 post-test/post-test-4/2409106040-MuhammadRaflipernanda-PT-4.exe
create mode 100644 post-test/post-test-4/2409106040-MuhammadRaflipernanda-PT-4.pdf
```

Gambar 5.2 git commit -m

3. Git push

Digunakan untuk mengirimkan perubahan file setelah di commit ke remote repository.

```
● PS D:\praktikum-apl> git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 12 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 1.28 MiB | 403.00 KiB/s, done.
Total 7 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/Raflipernanda/praktikum-apl.git
32112b6..f2dd33d master -> master
```

Gambar 5.3 Git push