

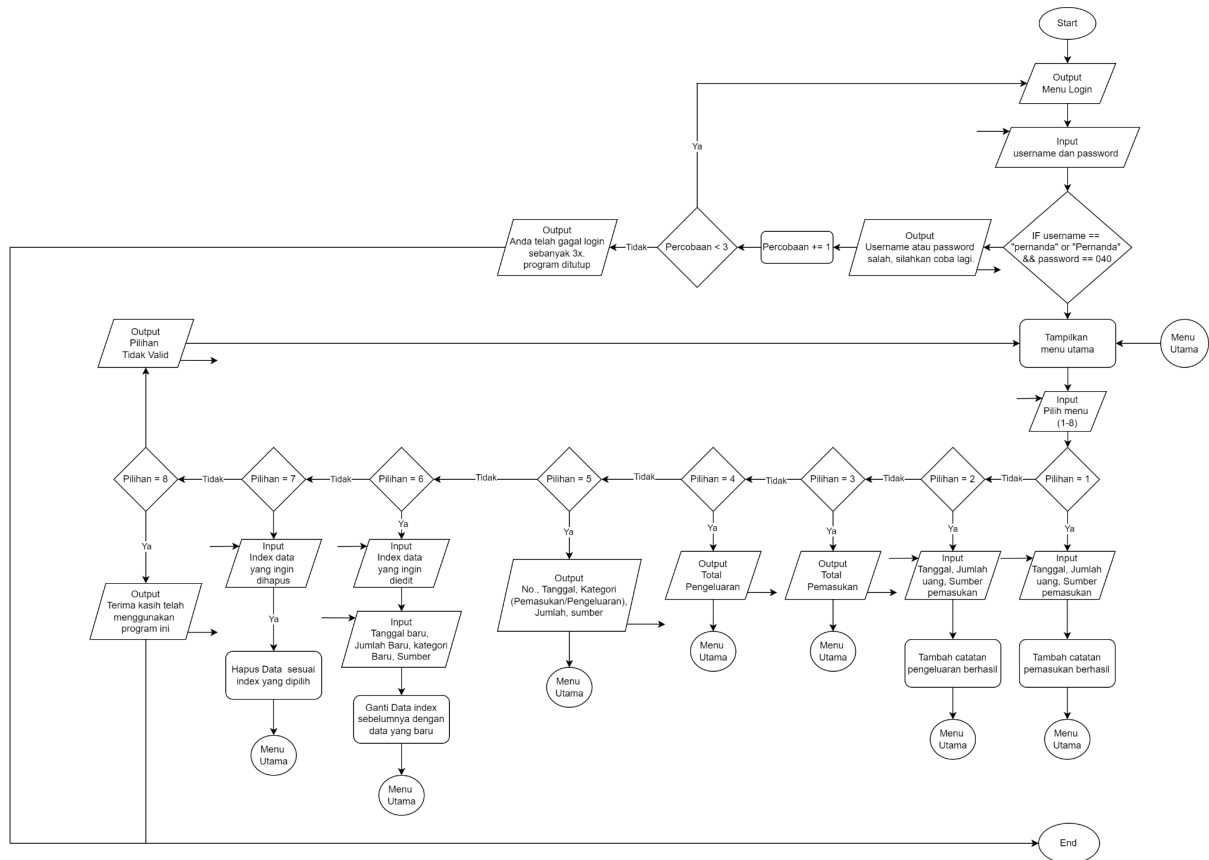
LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN LANJUT



Disusun oleh:
Muhammad Rafli Pernanda 2409106040
Kelas A2'24

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



Gambar 1.1 Flowchart

2. Analisis Program

2.1 Deskripsi Singkat Program

Program ini bertujuan untuk membantu pengguna dalam mencatat dan mengelola keuangan pribadi dengan lebih terstruktur. Dengan fitur seperti pencatatan pemasukan dan pengeluaran, perhitungan total secara otomatis, serta kemampuan untuk menambahkan, mengedit, dan menghapus catatan, program ini memudahkan pengguna dalam memantau kondisi keuangan mereka. Selain itu, adanya sistem login meningkatkan keamanan data, sementara tampilan tabel memungkinkan informasi lebih mudah dibaca dan dianalisis. Dengan demikian, pengguna dapat mengelola keuangan secara lebih efektif dan terorganisir.

2.2 Penjelasan Alur & Algoritma

Program ini adalah aplikasi pencatatan keuangan berbasis konsol yang memungkinkan pengguna mencatat pemasukan dan pengeluaran serta melihat rekap keuangan mereka. Berikut adalah tahapan alur kerja program secara berurutan:

1. Memulai Program dan Login

Saat program dijalankan, fungsi utama `main()` akan dipanggil. Program pertama-tama menampilkan menu login, yang diproses oleh fungsi `login()`.

- Pengguna diminta memasukkan username dan password.
- Program memeriksa apakah username dan password yang dimasukkan sesuai dengan data yang telah ditentukan ("pernanda" atau "Pernanda" sebagai username, serta "040" sebagai password).
- Jika benar, pengguna diberikan akses ke menu utama.
- Jika salah, pengguna diberikan kesempatan login hingga tiga kali. Jika gagal tiga kali, program menampilkan pesan error dan keluar.

2. Menampilkan Menu Utama

Setelah login berhasil, program menjalankan fungsi `menu()`, yang menampilkan daftar pilihan sebagai berikut:

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. Keluar dari Program

Pengguna diminta untuk memasukkan angka yang sesuai dengan pilihan yang diinginkan.

3. Memasukkan Data Keuangan

Jika pengguna memilih opsi 1 (pemasukan) atau 2 (pengeluaran), maka program akan menjalankan fungsi `tambahCatatan(string tipe)`, yang bekerja sebagai berikut:

- Memeriksa apakah jumlah catatan belum mencapai batas maksimum (`MAX_CATATAN = 100`).
- Meminta pengguna memasukkan tanggal transaksi (YYYY-MM-DD).
- Meminta pengguna memasukkan jumlah uang.
- Meminta pengguna memasukkan sumber pemasukan atau pengeluaran (contoh: "Gaji" atau "Belanja bulanan").
- Data yang dimasukkan akan disimpan dalam array `tanggal[]`, `jumlah[]`, `kategori[]`, dan `sumber[]`.
- Setelah semua data dimasukkan, program menampilkan pesan sukses bahwa catatan telah ditambahkan.

4. Menghitung Total Pemasukan dan Pengeluaran

Jika pengguna memilih opsi 3 (melihat total pemasukan) atau 4 (melihat total pengeluaran), maka program akan memanggil fungsi `hitungTotalPemasukan()` atau `hitungTotalPengeluaran()`.

- Fungsi ini melakukan iterasi pada array `kategori[]` untuk menjumlahkan semua nilai `jumlah[]` yang memiliki kategori "Pemasukan" atau "Pengeluaran".
- Hasil perhitungan ditampilkan di layar dalam bentuk total rupiah.

5. Menampilkan Semua Catatan Keuangan

Jika pengguna memilih opsi 5, program akan menjalankan fungsi `lihatSemuaCatatan()`.

- Program akan memeriksa apakah ada catatan yang tersimpan. Jika tidak ada, akan muncul pesan "Belum ada catatan keuangan."

- Jika ada data, program akan menampilkan semua catatan dalam bentuk tabel dengan kolom nomor, tanggal, kategori, jumlah uang, dan sumber transaksi.
- Data ditampilkan menggunakan fungsi `setw()` dari pustaka `<iomanip>` agar tampilan tabel lebih rapi.

6. Mengedit Catatan Keuangan

Jika pengguna memilih opsi 6, program akan menjalankan fungsi `editCatatan(int index)`.

- Pengguna diminta untuk memasukkan nomor catatan yang ingin diedit.
- Program memeriksa apakah nomor catatan valid. Jika tidak valid, program menampilkan pesan error.
- Jika valid, program meminta pengguna memasukkan ulang tanggal, jumlah uang, kategori, dan sumber transaksi yang baru.
- Setelah data diperbarui, program menampilkan pesan sukses bahwa catatan telah berhasil diedit.

7. Menghapus Catatan Keuangan

Jika pengguna memilih opsi 7, program akan menjalankan fungsi `hapusCatatan(int index)`.

- Pengguna diminta untuk memasukkan nomor catatan yang ingin dihapus.
- Program memeriksa apakah nomor yang dimasukkan valid. Jika tidak valid, program menampilkan pesan error.
- Jika valid, program akan menghapus data pada indeks tersebut dengan cara menggeser seluruh data di array agar tetap berurutan.
- `totalCatatan` akan dikurangi satu untuk memastikan catatan terakhir tidak diakses secara salah.
- Setelah penghapusan berhasil, program menampilkan pesan sukses.

8. Keluar dari Program

Jika pengguna memilih opsi 8, program menampilkan pesan terima kasih dan keluar.

3. Source Code

A. Fitur Menu Login

Fitur ini digunakan sebelum user masuk ke menu utama, user akan diminta untuk menginput username dan password, jika user salah menginput username atau password sebanyak 3 kali, maka program akan dihentikan.

Source Code:

```
bool login() {
    string username, password;
    int attempts = 0;

    while (attempts < 3) {
        cout << "=====\n";
        cout << "|          MENU LOGIN          |\n";
        cout << "=====\n";
        cout << "Masukkan Username: ";
        cin >> username;
        cout << "Masukkan Password: ";
        cin >> password;
        cout << endl;

        if ((username == "pernanda" || username == "Pernanda") && password == "040") {
            cout << "=====" << endl;
            cout << "|      Login Berhasil!      |" << endl;
            cout << "| Selamat datang, Pernanda! |" << endl;
            cout << "=====" << endl << endl << endl << endl;
            return true;
        } else {
            cout << "Login gagal! Username atau password salah.\n";
            attempts++;
            cout << "Percobaan tersisa: " << (3 - attempts) << "\n";
        }
    }

    cout << "Anda telah gagal login sebanyak 3 kali. Program ditutup.\n";
    return false;
}
```

B. Fitur Menu Utama

Fitur ini digunakan untuk menampilkan menu utama

Source Code:

```
int menu() {
    int pilihan;
    do {
        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. Keluar dari Program                |\n";
        cout << "=====\n";
        cout << "Pilih menu (1-8): ";
        cin >> pilihan;

        switch (pilihan) {
            case 1: tambahCatatan("Pemasukan"); break;
            case 2: tambahCatatan("Pengeluaran"); break;
            case 3: cout << "Total Pemasukan: Rp" << hitungTotalPemasukan() <<
endl << endl; break;
            case 4: cout << "Total Pengeluaran: Rp" << hitungTotalPengeluaran()
<< endl << endl; break;
            case 5:
                if (!lihatSemuaCatatan()) {
                    cout << "Tidak ada catatan untuk ditampilkan.\n" << endl;
                }
                break;
            case 6:
                int editIndex;
                cout << "Masukkan nomor catatan yang ingin diedit: ";
                cin >> editIndex;
                if (editCatatan(editIndex)) {
                    cout << "Catatan berhasil diperbarui!\n" << endl;
                } else {
                    cout << "Nomor catatan tidak valid!\n" << endl;
                }
                break;
            case 7:
                int hapusIndex;
                cout << "Masukkan nomor catatan yang ingin dihapus: ";
                cin >> hapusIndex;
                if (hapusCatatan(hapusIndex)) {
                    cout << "Catatan berhasil dihapus!\n" << endl;
                } else {
                    cout << "Nomor catatan tidak valid!\n" << endl;
                }
            }
        }
    } while (pilihan != 8);
}
```

```

        }
        break;
    case 8: cout << "Terima kasih telah menggunakan program ini!\n" <<
endl; break;
    default: cout << "Pilihan tidak valid!\n" << endl; break;
    }
} while (pilihan != 8);

return 0;
}

```

C. Fitur CRUD

Fitur ini digunakan untuk Create, Read, Update, dan Delete data yang ada pada program

Source Code:

```

int tambahCatatan(string tipe) {
    if (totalCatatan >= MAX_CATATAN) {
        cout << "Penyimpanan penuh!\n";
        return 0;
    }

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

    totalCatatan++;
    cout << "Catatan berhasil ditambahkan!\n" << endl;
    return 1;
}

// Fungsi menghitung total pemasukan
double hitungTotalPemasukan() {
    double total = 0;
    for (int i = 0; i < totalCatatan; i++) {
        if (kategori[i] == "Pemasukan") {
            total += jumlah[i];
        }
    }
    return total;
}

// Fungsi menghitung total pengeluaran

```



```

double hitungTotalPengeluaran() {
    double total = 0;
    for (int i = 0; i < totalCatatan; i++) {
        if (kategori[i] == "Pengeluaran") {
            total += jumlah[i];
        }
    }
    return total;
}

// Fungsi untuk menampilkan semua catatan dalam bentuk tabel
bool lihatSemuaCatatan() {
    if (totalCatatan == 0) {
        cout << "Belum ada catatan keuangan.\n";
        return false;
    }

    cout << "\n===== Daftar Catatan Keuangan\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) << tanggal[i]
        << " | " << setw(10) << kategori[i]
        << " | " << setw(10) << jumlah[i]
        << " | " << setw(16) << sumber[i]
        << " |\n";
    }

    cout << "+-----+-----+-----+-----+-----+\n"
    << endl;
    return true;
}

// Fungsi untuk mengedit catatan
bool editCatatan(int index) {
    if (index < 1 || index > totalCatatan) return false;

    index--;
    cout << "Masukkan tanggal baru: ";
    cin >> tanggal[index];
    cout << "Masukkan jumlah uang baru: ";
    cin >> jumlah[index];
    cout << "Masukkan kategori baru (Pemasukan/Pengeluaran): ";
    cin >> kategori[index];
}

```

```

    cout << "Masukkan sumber baru: ";
    cin.ignore();
    getline(cin, sumber[index]);

    return true;
}

// Fungsi untuk menghapus catatan
bool hapusCatatan(int index) {
    if (index < 1 || index > totalCatatan) return false;

    index--;
    for (int i = index; i < totalCatatan - 1; i++) {
        tanggal[i] = tanggal[i + 1];
        jumlah[i] = jumlah[i + 1];
        kategori[i] = kategori[i + 1];
        sumber[i] = sumber[i + 1];
    }
    totalCatatan--;

    return true;
}

```

4. Uji Coba dan Hasil Output

4.1 Uji Coba

1. User memasukkan username dan password yang benar.
2. User memasukkan username dan password yang salah sebanyak 3x.
3. User memilih pilihan 1 lalu 3 di menu utama.
4. User memilih pilihan 2 lalu 4 di menu utama.
5. User memilih pilihan 5 di menu utama setelah menambahkan pemasukan dan pengeluaran.
6. User mengedit catatan yang telah ditambahkan sebelumnya.
7. User menghapus catatan yang telah di edit sebelumnya.
8. User keluar dari program.

4.2 Hasil Output

```
=====
|           MENU LOGIN           |
=====
Masukkan Username: pernanda
Masukkan Password: 040

=====
|      Login Berhasil!          |
| Selamat datang, Pernanda!    |
=====

=====
|   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. Keluar dari Program       |
=====
Pilih menu (1-8): █
```

Gambar 4.1 Login Berhasil

```

=====
|           MENU LOGIN           |
=====
Masukkan Username: prenanda
Masukkan Password: 004

Login gagal! Username atau password salah.
Percobaan tersisa: 2
=====
|           MENU LOGIN           |
=====
Masukkan Username: pernanad
Masukkan Password: 040

Login gagal! Username atau password salah.
Percobaan tersisa: 1
=====
|           MENU LOGIN           |
=====
Masukkan Username: pernanda
Masukkan Password: 400

Login gagal! Username atau password salah.
Percobaan tersisa: 0
Anda telah gagal login sebanyak 3 kali. Program ditutup.
PS D:\praktikum-apl>

```

Gambar 4.2 Login gagal

```

=====
|      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. Keluar dari Program              |
=====
Pilih menu (1-8): 1
Masukkan tanggal (YYYY-MM-DD): 2025-03-17
Masukkan jumlah uang: 15000
Masukkan sumber Pemasukan: Ngojek
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. Keluar dari Program              |
=====
Pilih menu (1-8): 3
Total Pemasukan: Rp15000

```

Gambar 4.3 Menambahkan catatan pemasukan

```
=====
|      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. Keluar dari Program              |
=====
Pilih menu (1-8): 2
Masukkan tanggal (YYYY-MM-DD): 2025-03-17
Masukkan jumlah uang: 5000
Masukkan sumber Pengeluaran: Beli es teh
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. Keluar dari Program              |
=====
Pilih menu (1-8): 4
Total Pengeluaran: Rp5000
```

Gambar 4.4 Menambahkan catatan 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. Keluar dari Program                |
=====
Pilih menu (1-8): 5

===== Daftar Catatan Keuangan =====
+---+-----+-----+-----+-----+
| No | Tanggal | Kategori | Jumlah | Sumber |
+---+-----+-----+-----+-----+
| 1  | 2025-03-17 | Pemasukan | 15000 | Ngojek |
| 2  | 2025-03-17 | Pengeluaran | 5000 | Beli es teh |
+---+-----+-----+-----+-----+
```

Gambar 4.5 Menampilkan semua catatan


```

===== Daftar Catatan Keuangan =====
+---+-----+-----+-----+-----+
| No | Tanggal | Kategori | Jumlah | Sumber |
+---+-----+-----+-----+-----+
| 1 | 2025-03-17 | Pemasukan | 15000 | Ngojek |
| 2 | 2025-03-17 | Pengeluaran | 5000 | Beli es teh |
+---+-----+-----+-----+-----+

=====
| 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. Keluar dari Program |
=====
Pilih menu (1-8): 6
Masukkan nomor catatan yang ingin diedit: 2
Masukkan tanggal baru: 2025-03-17
Masukkan jumlah uang baru: 10000
Masukkan kategori baru (Pemasukan/Pengeluaran): Pengeluaran
Masukkan sumber baru: Beli Bakso
Catatan berhasil diperbarui!

```

Gambar 4.6 Mengedit catatan

```

===== Daftar Catatan Keuangan =====
+---+-----+-----+-----+-----+
| No | Tanggal | Kategori | Jumlah | Sumber |
+---+-----+-----+-----+
| 1 | 2025-03-17 | Pemasukan | 15000 | Ngojek |
| 2 | 2025-03-17 | Pengeluaran | 10000 | Beli Bakso |
+---+-----+-----+-----+

=====
| 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. Keluar dari Program |
=====
Pilih menu (1-8): 7
Masukkan nomor catatan yang ingin dihapus: 2
Catatan berhasil dihapus!

```

Gambar 4.7 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. Keluar dari Program |
=====
Pilih menu (1-8): 8
Terima kasih telah menggunakan program ini!

PS D:\praktikum-apl> █

```

Gambar 4.8 User keluar dari program

5. Git

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 .  
warning: in the working copy of '.vscode/launch.json', LF will be replaced by CRLF the next time Git touches it  
warning: in the working copy of '.vscode/settings.json', LF will be replaced by CRLF the next time Git touches it  
PS D:\praktikum-apl>
```

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 b16f0df] Selesai Source code dan 90% laporan  
10 files changed, 379 insertions(+), 3 deletions(-)  
create mode 100644 kelas/Pertemuan-1/2409106040-MuhammadRaflipernanda-Pertemuan-1.cpp  
create mode 100644 kelas/Pertemuan-2/2409106040-MuhammadRaflipernanda-Pertemuan-2.cpp  
create mode 100644 kelas/Pertemuan-2/2409106040-MuhammadRaflipernanda-Pertemuan-2.exe  
create mode 100644 post-test/post-test-2/2409106040-MuhammadRaflipernanda-PT-2.cpp  
create mode 100644 post-test/post-test-2/2409106040-MuhammadRaflipernanda-PT-2.exe  
create mode 100644 post-test/post-test-2/2409106040-MuhammadRaflipernanda-PT-2.pdf  
create mode 100644 post-test/post-test-2/build/Debug/2409106040-MuhammadRaflipernanda-PT-2.o  
create mode 100644 post-test/post-test-2/build/Debug/outDebug.exe  
PS D:\praktikum-apl>
```

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: 24, done.  
Counting objects: 100% (24/24), done.  
Delta compression using up to 12 threads  
Compressing objects: 100% (17/17), done.  
Writing objects: 100% (18/18), 2.00 MiB | 909.00 KiB/s, done.  
Total 18 (delta 5), reused 0 (delta 0), pack-reused 0 (from 0)  
remote: Resolving deltas: 100% (5/5), completed with 2 local objects.  
To https://github.com/Raflipernanda/praktikum-apl.git  
94937a0..b16f0df master -> master  
PS D:\praktikum-apl>
```

Gambar 5.3 Git push