

LAPORAN PRAKTIKUM
POSTTEST (1)
ALGORITMA PEMROGRAMAN LANJUT



Disusun oleh:
Dinda Shasha Amaranggana (2509106063)
Kelas B '25

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. FLOWCHART

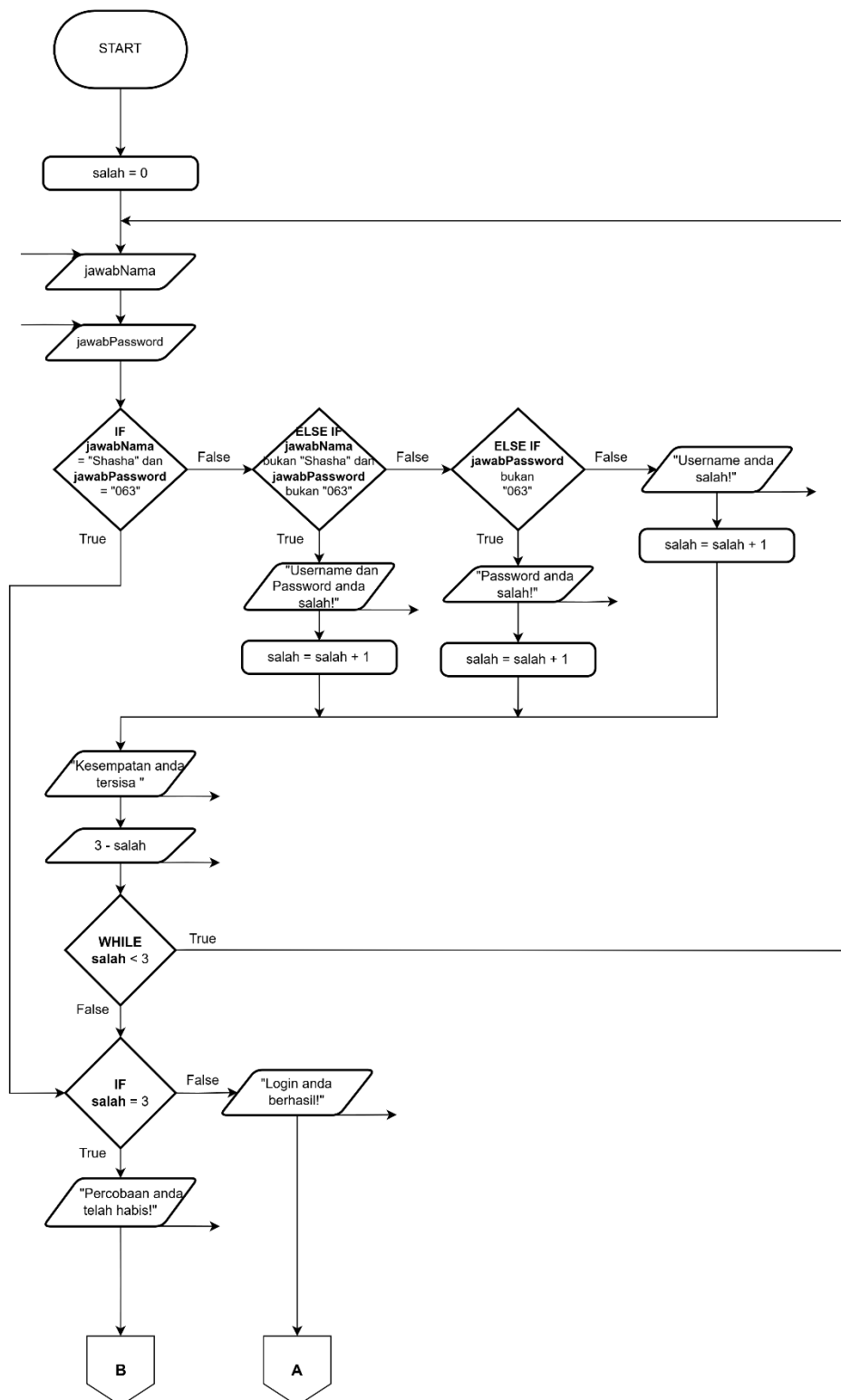


Diagram 1.1 *Flowchart Menu Login*

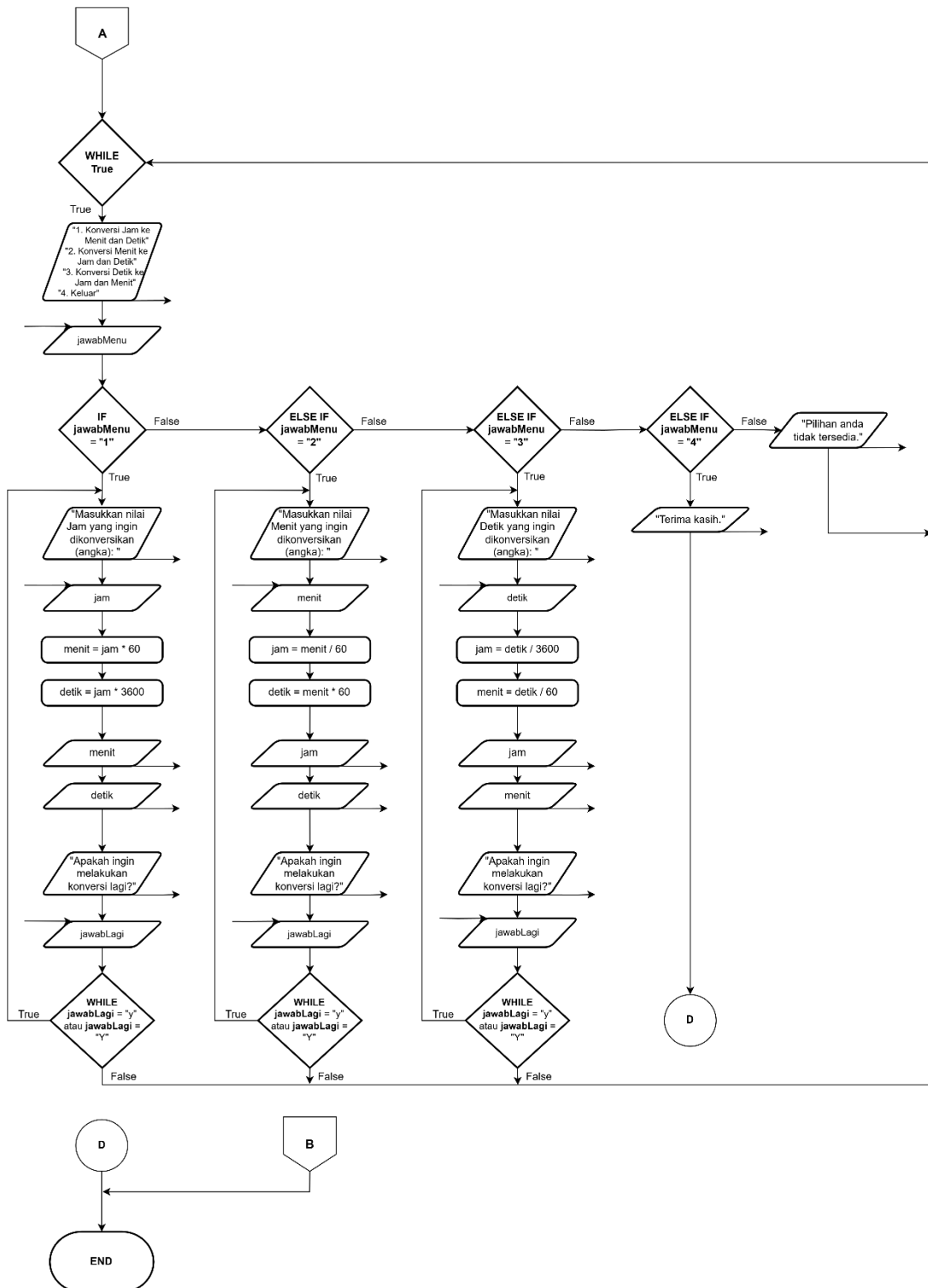


Diagram 1.2 *Flowchart* Menu Konversi

- 1) Program dimulai (**Start**);
- 2) inisialisasi variabel **salah** = 0 (**Process**);
- 3) Proses *login* dilakukan (**Do-While**):
 - a) Pengguna meng-*input* **username** dan **password** (**Input**);

- b) pengecekan terhadap *input* (**Decision**):
 - i) Jika **username** adalah “Shasha” dan **password** adalah “063”, maka *login* berhasil dan keluar dari perulangan proses *login* menuju menu utama konversi;
 - ii) jika **username** dan/atau **password** salah, maka *login* tidak berhasil, pesan peringatan sisa kesempatan dikeluarkan (**Output**), dan **salah** bertambah 1 (**Process**).
 - c) proses *login* akan diulang terus selama **salah** < 3 (**Do-While**);
 - d) jika **salah** bernilai 3, maka program berhenti.
- 4) Menu utama konversi ditampilkan (**While**):
- a) Pengguna diberikan pilihan untuk melakukan jenis konversi apa atau keluar dari program (**Output**);
 - b) Pengguna meng-*input* **jawabMenu** (**Input**);
 - c) Pengecekan terhadap *input* (**Decision**);
 - i) Jika **jawabMenu** adalah “1”, maka pengguna diarahkan menuju konversi Jam ke Menit dan Detik;
 - ii) Jika **jawabMenu** adalah “2”, maka pengguna diarahkan menuju konversi Menit ke Jam dan Detik;
 - iii) Jika **jawabMenu** adalah “3”, maka pengguna diarahkan menuju konversi Detik ke Jam dan Menit;
 - iv) Jika **jawabMenu** adalah “4”, maka pengguna keluar dari program dan program berhenti;
 - v) Jika **jawabMenu** adalah lainnya, maka pesan peringatan dikeluarkan dan pengguna diarahkan untuk meng-*input* **jawabMenu** ulang;
 - d) Menu utama konversi akan ditampilkan berulang terus selama pengguna tidak memilih keluar dari program;
- 5) Proses konversi ditampilkan jika pengguna meng-*input* **jawabMenu** dengan “1”, “2”, atau “3” (**Do-While**):
- a) Pengguna meng-*input* **jam** (**jawabMenu** adalah “1”) atau **menit** (**jawabMenu** adalah “2”) atau **detik** (**jawabMenu** adalah “3”) untuk dikonversikan (**Input**);
 - b) Dihitung **menit** = **jam** * 60 dan **detik** = **jam** * 3600 (**jawabMenu** adalah “1”) atau **jam** = **menit** / 60 dan **detik** = **menit** * 60 (**jawabMenu** adalah “2”)

- atau **jam** = **detik** / 3600 dan **menit** = **detik** / 60 (**jawabMenu** adalah “3”)
(*Process*);
- c) Hasil konversi ditampilkan (*Output*);
 - d) Pengguna diberikan pilihan untuk melanjutkan konversi jenis tersebut atau tidak (*Output*);
 - e) Pengguna meng-input **jawabLagi** (*Input*);
 - f) Proses konversi akan diulang terus selama **jawabLagi** bernilai “Y” atau “y”;
- 6) Program selesai (*End*).

2. DESKRIPSI SINGKAT PROGRAM

Program ini berfungsi untuk mengonversi nilai waktu; dari Jam ke Menit dan Detik, dari Menit ke Jam dan Detik, serta dari Detik ke Jam dan Menit. Fokus dalam program ini adalah setelah pengguna *login* ke dalam program, program akan terus mengulang selama kondisi perulangan terpenuhi.

Untuk masuk ke dalam program, pengguna harus *login* terlebih dahulu dengan meng-input *username* dan *password* yang telah diprogramkan. Pengguna dibatasi kesempatan untuk mencoba *login* sebanyak tiga kali sebelum berhasil masuk. Jika di percobaan ketiga dan pengguna masih melakukan kesalahan, maka pengguna tidak bisa melakukan *login* kembali.

Setelah berhasil masuk, menu utama konversi akan ditampilkan dan pengguna dapat memilih menu konversi nilai waktu yang meliputi konversi Jam ke Menit dan Detik, Menit ke Jam dan Detik, serta Detik ke Jam dan Menit, ataupun menu keluar yang membuat pengguna dapat keluar dari program. Program juga memungkinkan pengguna untuk mengulang di menu utama konversi maupun menu proses konversi ketiganya dan dapat keluar sesuai kemauan pengguna.

Pada dasarnya program ini bermanfaat untuk mengonversikan nilai satuan waktu dari ketiga satuan waktu ke bentuk satu sama lain.

3. SOURCE CODE

```
#include <iostream>
using namespace std;
int main() {
    int salah = 0;
    string jawabNama;
    string jawabPassword;
    string jawabMenu;
```

```

float menit;
float detik;
float jam;
string jawabLagi;
do {
    cout << "\nUsername: ";
    cin >> jawabNama;
    cout << "Password: ";
    cin >> jawabPassword;
    if (jawabNama == "Shasha" && jawabPassword == "063") {
        break;
    }
    else if (jawabNama != "Shasha" && jawabPassword != "063") {
        cout << "\nUsername dan Password anda salah!" << endl;
        salah++;
    }
    else if (jawabPassword != "063") {
        cout << "\nPassword anda salah!" << endl;
        salah++;
    }
    else {
        cout << "\nUsername anda salah!" << endl;
        salah++;
    }
    cout << "Kesempatan anda tersisa " << 3 - salah << " kali." <<
endl;
} while (salah < 3);
if (salah == 3) {
    cout << "\nPercobaan anda telah habis!";
    return 0;
}
cout << "\nLogin anda berhasil!" << endl;
while (true) {
    cout << "\n1. Konversi Jam ke Menit dan Detik.";
    cout << "\n2. Konversi Menit ke Jam dan Detik.";
    cout << "\n3. Konversi Detik ke Jam dan Menit.";
    cout << "\n4. Keluar.";
    cout << "\n\nSilakan anda ingin melakukan apa? (input angka): ";
    cin >> jawabMenu;
    if (jawabMenu == "1") {
        do {
            cout << "\nMasukkan nilai Jam yang ingin dikonversikan
(angka): ";
            cin >> jam;
            menit = jam * 60;
            detik = jam * 3600;
            cout << jam << " jam = " << menit << " menit atau " <<
detik << " detik." << endl;
            cout << "\nApakah ingin melakukan konversi Jam lagi?
(y/lainnya): ";
            cin >> jawabLagi;
        } while (jawabLagi == "y" || jawabLagi == "Y");
    }
    else if (jawabMenu == "2") {
        do {
            cout << "\nMasukkan nilai Menit yang ingin dikonversikan
(angka): ";
            cin >> menit;

```

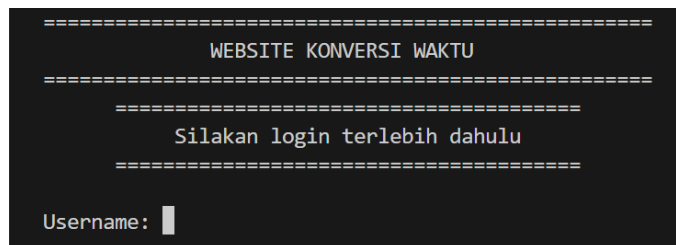
```

        jam = menit / 60;
        detik = menit * 60;
        cout << menit << " menit = " << jam << " jam atau " <<
detik << " detik." << endl;
        cout << "\nApakah ingin melakukan konversi Menit lagi?
(y/lainnya)";
        cin >> jawabLagi;
        } while (jawabLagi == "y" || jawabLagi == "Y");
    }
    else if (jawabMenu == "3") {
        do {
            cout << "\nMasukkan nilai Detik yang ingin dikonversikan
(angka): ";
            cin >> detik;
            jam = detik / 3600;
            menit = detik / 60;
            cout << detik << " detik = " << jam << " jam atau " <<
menit << " menit." << endl;
            cout << "\nApakah ingin melakukan konversi Detik lagi?
(y/lainnya)" << endl;
            cin >> jawabLagi;
        } while (jawabLagi == "y" || jawabLagi == "Y");
    }

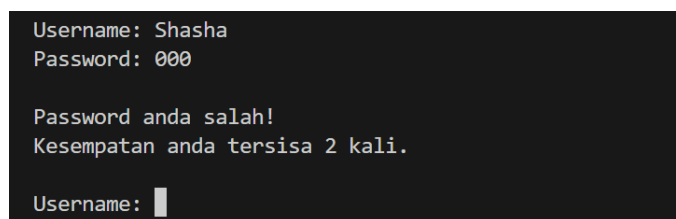
    else if (jawabMenu == "4") {
        cout << "                Terima kasih, " << jawabNama << endl;
        break;
    }
    else {
        cout << "\nPilihan anda tidak tersedia. Isi yang benar!" <<
endl;
    }
}
return 0;
}

```

4. HASIL OUTPUT



Gambar 4.1 *Screenshot Output Menu Awal*



Gambar 4.2 *Screenshot Output Password Salah*

```

Username: Sha
Password: 063

Username anda salah!
Kesempatan anda tersisa 1 kali.

Username: █

```

Gambar 4.3 *Screenshot Output Username Salah*

```

Username: Sha
Password: 000

Username dan Password anda salah!

```

Gambar 4.4 *Screenshot Output Username dan Password Salah*

```

Kesempatan anda tersisa 0 kali.

=====
Percobaan anda telah habis! YAHAAH KASIAN.
=====

```

Gambar 4.5 *Screenshot Output Kesempatan Habis dan Program Terhenti*

```

=====
WEBSITE KONVERSI WAKTU
=====

=====
Silakan login terlebih dahulu
=====

Username: Shasha
Password: 063

-----
Login anda berhasil!
-----

```

Gambar 4.6 *Screenshot Output Login Berhasil*

```

=====
WEBSITE KONVERSI WAKTU
=====

=====
Selamat menjelajah, Shasha
=====

1. Konversi Jam ke Menit dan Detik.
2. Konversi Menit ke Jam dan Detik.
3. Konversi Detik ke Jam dan Menit.
4. Keluar.

Silakan anda ingin melakukan apa? (input angka): █

```

Gambar 4.7 *Screenshot Output Menu Utama Konversi*


```

Silakan anda ingin melakukan apa? (input angka): 1

=====
Konversi Jam ke Menit dan Detik
=====

Masukkan nilai Jam yang ingin dikonversikan (angka): 1
1 jam = 60 menit atau 3600 detik.

Apakah ingin melakukan konversi Jam lagi? (y/lainnya): y

```

Gambar 4.8 *Screenshot Output* Konversi Jam

```

Apakah ingin melakukan konversi Jam lagi? (y/lainnya): y

=====
Konversi Jam ke Menit dan Detik
=====

Masukkan nilai Jam yang ingin dikonversikan (angka): 2
2 jam = 120 menit atau 7200 detik.

```

Gambar 4.9 *Screenshot Output* Lanjut Konversi di Jenis Konversi yang Sama

```

Apakah ingin melakukan konversi Jam lagi? (y/lainnya): t

=====
WEBSITE KONVERSI WAKTU
=====

Selamat menjelajah, Shasha

=====
1. Konversi Jam ke Menit dan Detik.
2. Konversi Menit ke Jam dan Detik.
3. Konversi Detik ke Jam dan Menit.
4. Keluar.

Silakan anda ingin melakukan apa? (input angka): █

```

Gambar 4.10 *Screenshot Output* Berhenti Konversi di Jenis Konversi yang Sama

```

Silakan anda ingin melakukan apa? (input angka): 2

=====
Konversi Menit ke Jam dan Detik
=====

Masukkan nilai Menit yang ingin dikonversikan (angka): 60
60 menit = 1 jam atau 3600 detik.

Apakah ingin melakukan konversi Menit lagi? (y/lainnya): █

```

Gambar 4.11 *Screenshot Output* Konversi Menit

```

Silakan anda ingin melakukan apa? (input angka): 3

=====
                Konversi Detik ke Jam dan Menit
=====

Masukkan nilai Detik yang ingin dikonversikan (angka): 3600
3600 detik = 1 jam atau 60 menit.

Apakah ingin melakukan konversi Detik lagi? (y/lainnya)

```

Gambar 4.12 *Screenshot Output* Konversi Detik

```

Silakan anda ingin melakukan apa? (input angka): 5

Pilihan anda tidak tersedia. Isi yang benar!

=====

                WEBSITE KONVERSI WAKTU
=====

                Selamat menjelajah, Shasha
=====

1. Konversi Jam ke Menit dan Detik.
2. Konversi Menit ke Jam dan Detik.
3. Konversi Detik ke Jam dan Menit.
4. Keluar.

Silakan anda ingin melakukan apa? (input angka): 

```

Gambar 4.13 *Screenshot Output Input* Menu Konversi di luar Pilihan

```

=====
                WEBSITE KONVERSI WAKTU
=====

                Selamat menjelajah, Shasha
=====

1. Konversi Jam ke Menit dan Detik.
2. Konversi Menit ke Jam dan Detik.
3. Konversi Detik ke Jam dan Menit.
4. Keluar.

Silakan anda ingin melakukan apa? (input angka): 4

=====
                Terima kasih, Shasha
=====

```

Gambar 4.14 *Screenshot Output Input* Keluar Program

5. LANGKAH-LANGKAH GIT

```
PS D:\COLLEGE\APL (S-2)\praktikum-apl> git init
Initialized empty Git repository in D:/COLLEGE/APL (S-2)/praktikum-apl/.git/
PS D:\COLLEGE\APL (S-2)\praktikum-apl> git add .
PS D:\COLLEGE\APL (S-2)\praktikum-apl> git commit -m "Up source code c++"
[main (root-commit) 04f367a] Up source code c++
 1 file changed, 133 insertions(+)
   create mode 100644 post-test/post-test-apl-1/2509106063-DindaShashaAmaranggana-PT-1.cpp
PS D:\COLLEGE\APL (S-2)\praktikum-apl> git remote add origin https://github.com/Shasha-Amaranggana/praktikum-apl.git
PS D:\COLLEGE\APL (S-2)\praktikum-apl> git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 1.28 KiB | 1.28 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Shasha-Amaranggana/praktikum-apl.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS D:\COLLEGE\APL (S-2)\praktikum-apl> █
```

Gambar 5.1 *Screenshot* Langkah-langkah Git

Untuk mengunggah *file* dari penyimpanan lokal komputer ke GitHub, langkah-langkah Git digunakan. Hal ini bertujuan agar semua perubahan yang kita lakukan pada *file* tersebut tercatat dan dapat kita lihat kembali bahkan kita bisa mengembalikannya ke versi perubahan sebelumnya.

5.1. GIT INIT

```
PS D:\COLLEGE\APL (S-2)\praktikum-apl> git init
Initialized empty Git repository in D:/COLLEGE/APL (S-2)/praktikum-apl/.git/
```

Gambar 5.2 *Screenshot* Git Init

Git Init berfungsi untuk mengubah *folder* yang berisi *file-file* yang ingin kita unggah ke GitHub, menjadi *Repository* lokal dalam komputer. Biasanya hanya dilakukan sekali saja, yaitu saat mengunggah *file* ke GitHub. Hal ini berbeda dengan *Repository* GitHub yang bersifat *online* dan bisa diakses oleh siapapun. Langkah ini juga yang bisa membuat kita dapat memantau perubahan-perubahan yang ada pada *folder/file* tersebut.

5.2. GIT ADD

```
PS D:\COLLEGE\APL (S-2)\praktikum-apl> git add .
```

Gambar 5.3 *Screenshot* Git Add

Git Add berfungsi untuk menandai *file* mana yang ingin kita unggah/ubah. Jika ingin mengunggah/mengubah seluruh *file* dalam folder maka gunakan *git add .*, dan jika hanya satu *file* maka gunakan *git add Nama_File*.

5.3. GIT COMMIT

```
PS D:\COLLEGE\APL (S-2)\praktikum-apl> git commit -m "Up source code c++"
[main (root-commit) 04f367a] Up source code c++
1 file changed, 133 insertions(+)
create mode 100644 post-test/post-test-apl-1/2509106063-DindaShashaAmaranggana-PT-1.cpp
```

Gambar 5.4 Screenshot Git Commit

Git Commit berfungsi untuk mengonfirmasi pengunggahan/perubahan file yang kita inginkan, dan juga pemberian keterangan pada penambahan tersebut sesuai yang kita mau.

5.4. GIT REMOTE

```
PS D:\COLLEGE\APL (S-2)\praktikum-apl> git remote add origin https://github.com/Shasha-Amaranggana/praktikum-apl.git
```

Gambar 5.5 Screenshot Git Remote

Git Remote berfungsi untuk menghubungkan *Repository* lokal kita dengan *Repository* di GitHub. Biasanya hanya dilakukan sekali saja, yaitu saat mengunggah *file* ke GitHub. Hal inilah yang dapat membuat kita melakukan penambahan apapun pada file lewat terminal lokal tanpa melakukannya secara manual di GitHub.

5.5. GIT PUSH

```
PS D:\COLLEGE\APL (S-2)\praktikum-apl> git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 1.28 KiB | 1.28 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Shasha-Amaranggana/praktikum-apl.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Gambar 5.6 Screenshot Git Push

Git Push berfungsi sebagai tahapan akhir untuk mengunggah/mengubah *file* kita. Dengan langkah ini, maka pengunggahan/perubahan yang kita inginkan akan masuk ke dalam GitHub.