

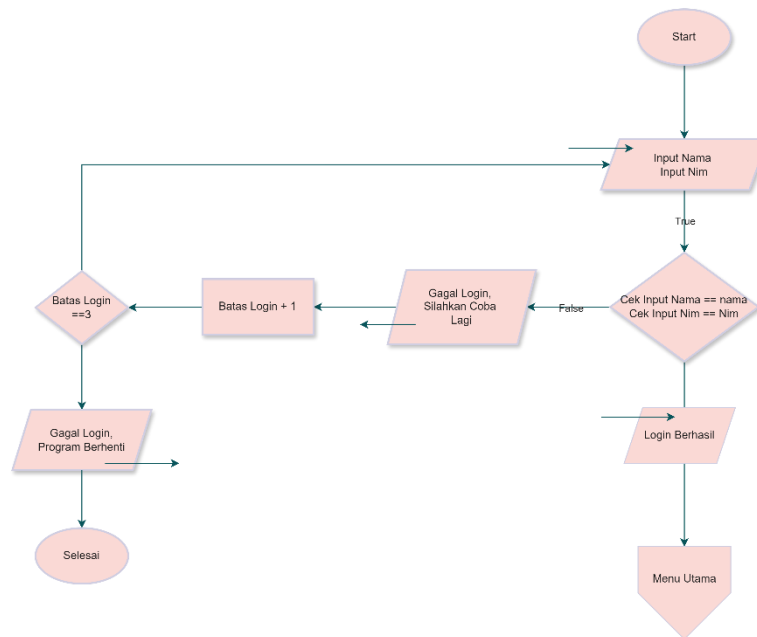
LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN LANJUT



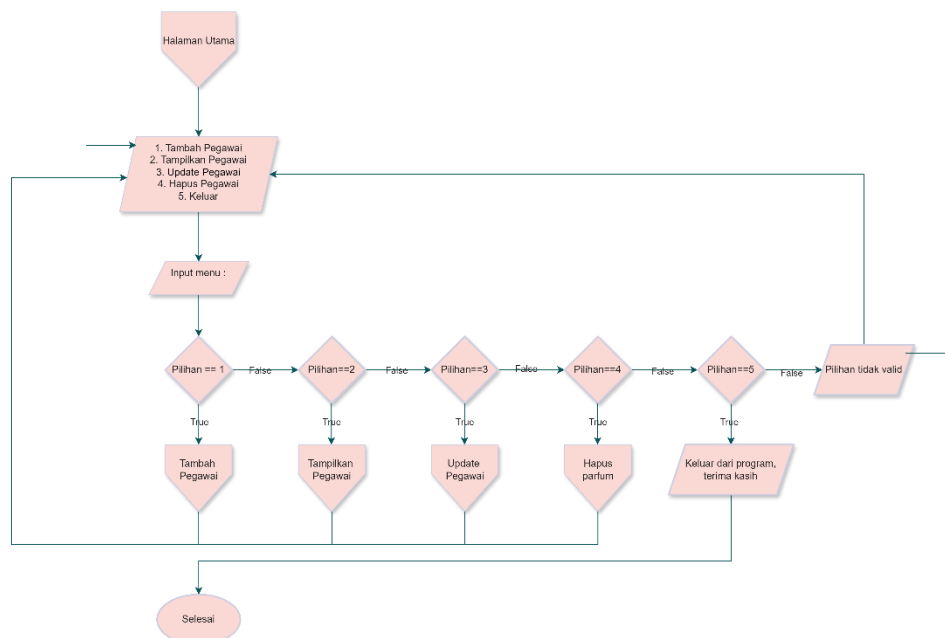
Disusun oleh:
Renaya Putri Alike (2409106002)
Kelas A1'24

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

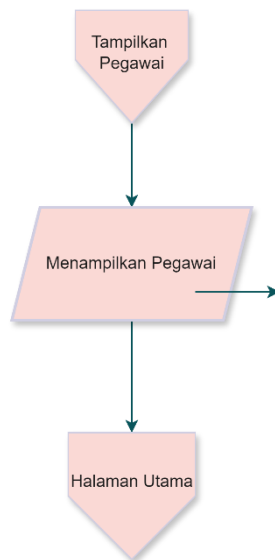
1. Flowchart



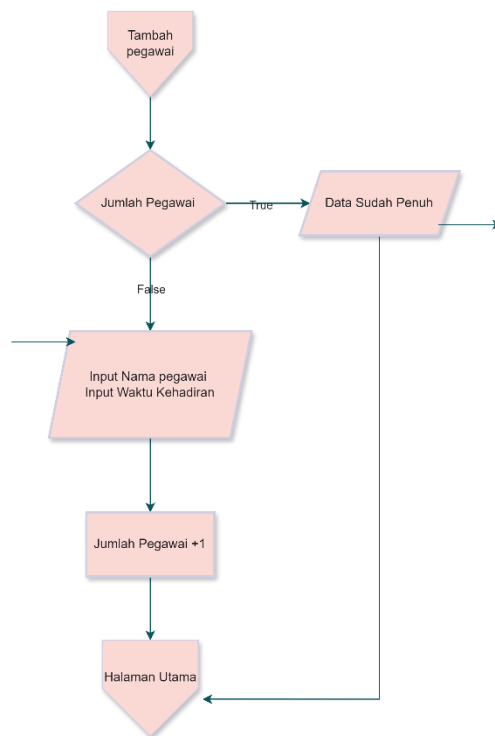
Flowchart 1.1 Menu Login



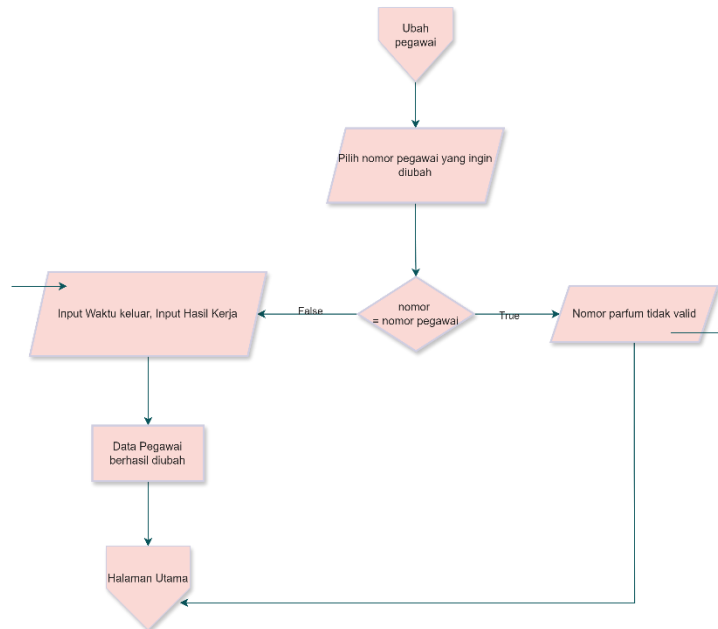
Flowchart 1.2 Menu Utama



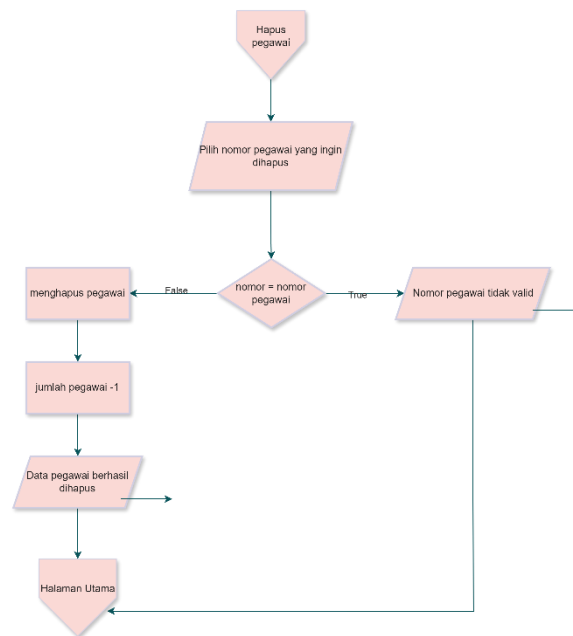
Flowchart 1.3 Tampilkan Pegawai



Flowchart 1.4 Tambah Pegawai



Flowchart 1.5 Ubah Pegawai



Flowchart 1.6 Hapus Pegawai

2. Analisis Program

2.1 Deskripsi Singkat Program

Program ini bertujuan untuk mengelola data pegawai secara efektif dengan menyediakan fitur CRUD (Create, Read, Update, Delete). Program ini memungkinkan pengguna untuk menambahkan pegawai baru beserta waktu kehadirannya, menampilkan daftar pegawai dalam

format tabel, memperbarui data pegawai dengan waktu keluar dan hasil kerja mereka, serta menghapus pegawai yang tidak mencapai target. Selain itu, program ini dilengkapi dengan sistem login yang hanya memberikan tiga kali kesempatan untuk masuk agar meningkatkan keamanan akses. Dengan tampilan yang rapi dan sistem yang sederhana, program ini membantu dalam pencatatan kehadiran, evaluasi kinerja, dan pengelolaan pegawai di lingkungan kerja yang membutuhkan sistem pencatatan dasar namun efektif.

2.2 Penjelasan Alur & Algoritma

2.2.1 Penjelasan Alur

1. Login Pengguna

- Program meminta **Nama** dan **NIM**.
- Jika input benar, program melanjutkan ke **menu utama**.
- Jika input salah 3 kali, program menampilkan pesan gagal dan berhenti.

2. Menampilkan Menu Utama

- Program menampilkan pilihan:
 1. **Tambah Pegawai**
 2. **Tampilkan Pegawai**
 3. **Update Pegawai**
 4. **Hapus Pegawai**
 5. **Keluar**
- Pengguna memilih opsi sesuai kebutuhan.

3. Tambah Pegawai (Create)

- Pengguna memasukkan **nama pegawai** dan **waktu masuk**.
- Waktu keluar dan hasil kerja awalnya diisi dengan tanda " - ".
- Data disimpan ke dalam **array**.

4. Tampilkan Pegawai (Read)

- Program menampilkan daftar pegawai dalam **bentuk tabel**.
- Setiap pegawai memiliki informasi: **Nama, Waktu Masuk, Waktu Keluar, dan Hasil Kerja**.

5. Update Pegawai (Update)

- Pengguna memilih nomor pegawai yang ingin diperbarui.
- Program meminta **waktu keluar** dan **hasil kerja** untuk pegawai tersebut.
- Data diperbarui dalam **array**.

6. Hapus Pegawai (Delete)

- Pengguna memilih nomor pegawai yang akan dihapus.
- Program menggeser data pegawai lain untuk menutupi data yang dihapus.
- Jumlah pegawai dikurangi.

7. Keluar dari Program

- Jika pengguna memilih **5. Keluar**, program menampilkan pesan **Keluar dari program**. Maka program berhenti berjalan.

2.2.2 Algoritma

1. Mulai
2. Login pengguna (maksimal 3 kali percobaan, jika gagal program berhenti)
3. Tampilkan menu utama, pengguna memilih:
 - Tambah Pegawai → Input nama dan waktu masuk, simpan ke array
 - Tampilkan Pegawai → Cetak daftar pegawai dalam tabel
 - Update Pegawai → Pilih pegawai, input waktu keluar & hasil kerja, perbarui data
 - Hapus Pegawai → Pilih pegawai, hapus data, geser array
 - Keluar → Program berhenti
4. Ulangi menu utama sampai pengguna memilih keluar
5. Selesai

2.3 Blok Kode

A. Import Library dan Deklarasi Konstanta

- `#include <iostream>` → Digunakan untuk input dan output.
- `#include <iomanip>` → Digunakan untuk format tampilan tabel agar rapi.
- `#include <string>` → Digunakan untuk manipulasi string.
- `const int MAX_PEGAWAI = 100;` → Konstanta yang menentukan jumlah maksimum pegawai

B. Login Pengguna

- Mendeklarasikan variabel login untuk menyimpan username dan NIM.
- Menentukan username dan NIM yang benar sebagai autentikasi pengguna.
- Loop **while** berjalan hingga login benar atau gagal 3 kali.
- Jika input benar, program lanjut ke menu utama.
- Jika salah, program memberi peringatan dan menambah jumlah percobaan
- Jika pengguna salah login 3 kali, program menampilkan pesan dan berhenti.

C. Deklarasi Variabel untuk Data Pegawai

- **nama[MAX_PEGAWAI]** → Menyimpan nama pegawai.
- **waktu_masuk[MAX_PEGAWAI]** → Menyimpan jam masuk pegawai.
- **waktu_keluar[MAX_PEGAWAI]** → Menyimpan jam keluar pegawai.
- **hasil_kerja[MAX_PEGAWAI]** → Menyimpan hasil kerja pegawai.
- **jumlah_pegawai** → Menyimpan jumlah pegawai yang ada.
- **pilihan** → Menyimpan pilihan menu pengguna.

D. Looping Menu Utama

- Loop **do-while** memastikan menu terus muncul sampai pengguna memilih keluar.
- Menampilkan menu utama yang memungkinkan pengguna memilih fitur CRUD atau keluar.

E. Menambah Pegawai (Create)

- Memasukkan nama dan waktu kehadiran pegawai.
- Waktu keluar dan hasil kerja diisi dengan "-" karena belum diperbarui.
- Jika kapasitas penuh, program menampilkan pesan "Data penuh!"

F. Menampilkan Data Pegawai (Read)

- Menampilkan daftar pegawai dalam tabel.
- Menggunakan **setw()** untuk merapikan tampilan tabel.

G. Memperbarui Data Pegawai (Update)

- Meminta nomor pegawai untuk diperbarui.
- Jika nomor valid, meminta input waktu keluar dan hasil kerja.

H. Menghapus Pegawai (Delete)

- Meminta nomor pegawai yang ingin dihapus.

- Jika valid, data pegawai setelahnya digeser ke atas untuk menutupi yang dihapus.

I. Keluar dari Program

- Menghentikan program jika pengguna memilih opsi keluar.

3. Source Code

3.1 Login Pengguna

Pada awal pengerjaan, program meminta pengguna untuk memasukkan **Nama** dan **NIM**. Jika input sesuai dengan yang telah ditentukan (**Renaya Putri Alike** dan **2409106002**), Maka program akan melanjutkan ke menu utama. Namun, Jika input salah sebanyak **3 kali**, Maka program akan berhenti secara otomatis.

```
string username, nim;
int login_attempts = 0;
const string correct_username = "Renaya Putri Alike";
const string correct_nim = "2409106002";

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

    if (username == correct_username && nim == correct_nim) {
        break;
    } else {
        cout << "Login gagal! Coba lagi.\n";
        login_attempts++;
    }
}

if (login_attempts == 3) {
    cout << "Anda telah gagal login sebanyak 3 kali. Program berhenti.\n";
    return 0;
}
```

3.2 Menu Utama

Setelah login berhasil, pengguna akan disajikan **menu pilihan** untuk mengelola data pegawai dan Menu akan terus berulang hingga pengguna memilih opsi **5. Keluar**.


```
do {  
    cout << "\n=== Menu Manajemen Pegawai ===" << endl;  
    cout << "1. Tambah Pegawai" << endl;  
    cout << "2. Tampilkan Pegawai" << endl;  
    cout << "3. Update Pegawai" << endl;  
    cout << "4. Hapus Pegawai" << endl;  
    cout << "5. Keluar" << endl;  
    cout << "Pilihan: ";  
    cin >> pilihan;  
    cin.ignore();  
}
```

3.3 CRUD Data Pegawai

Program menggunakan **array satu dimensi** untuk menyimpan data pegawai, yaitu:

a. CREATE (Menambahkan Pegawai)

Saat pengguna memilih **1. Tambah Pegawai**, program akan meminta nama pegawai dan waktu masuknya. Lalu, Data pegawai baru akan dimasukkan ke dalam array.

```
if (jumlah_pegawai < MAX_PEGAWAI) {
    cout << "Masukkan Nama Pegawai: ";
    getline(cin, nama[jumlah_pegawai]);
    cout << "Masukkan Waktu Kehadiran: ";
    getline(cin, waktu_masuk[jumlah_pegawai]);
    waktu_keluar[jumlah_pegawai] = "-";
    hasil_kerja[jumlah_pegawai] = "-";
    jumlah_pegawai++;
} else {
    cout << "Data penuh!" << endl;
}
```

b. READ (Menampilkan Daftar Pegawai)

Saat memilih **2. Tampilkan Pegawai**, program mencetak seluruh pegawai dalam bentuk tabel dan program menggunakan `setw()` untuk merapikan tampilan tabel.

```
cout << "\nDaftar Pegawai:\n";  
        cout << "+-----+-----+-----+-----+-----+-----+-----+-----+\n";  
    -----+-----+-----+" << endl;  
            cout << "| No | Nama                      | Waktu Masuk      | Waktu  
Keluar   | Hasil Kerja       |" << endl;
```



```

int index_hapus;

    cout << "Masukkan nomor pegawai yang ingin dihapus: ";
    cin >> index_hapus;
    cin.ignore();
    if (index_hapus > 0 && index_hapus <= jumlah_pegawai) {
        for (int i = index_hapus - 1; i < jumlah_pegawai - 1; i++) {
            nama[i] = nama[i + 1];
            waktu_masuk[i] = waktu_masuk[i + 1];
            waktu_keluar[i] = waktu_keluar[i + 1];
            hasil_kerja[i] = hasil_kerja[i + 1];
        }
        jumlah_pegawai--;
        cout << "Pegawai berhasil dihapus." << endl;
    } else {
        cout << "Nomor pegawai tidak valid!" << endl;
    }
}

```

e. Keluar dari Program

Jika pengguna memilih **5. Keluar**, program akan menampilkan pesan dan berhenti.

```

cout << "Keluar dari program." << endl;
    break;

```

4. Uji Coba dan Hasil Output

4.1 Uji Coba

Skenario 1 : Input Valid

➤ Login Berhasil

- Masukkan Nama: Renaya Putri Alika
- Masukkan NIM: 2409106002
- Program menampilkan menu utama.

➤ Tambah Pegawai

- Pilih menu 1. Tambah Pegawai.

- Masukkan Nama Pegawai: Thor Odinson
- Masukkan Waktu Kehadiran: 07.30
- Program menyimpan data pegawai.
- **Tampilkan Data Pegawai**
 - Pilih menu 2. Tampilkan Pegawai
- **Update Pegawai**
 - Pilih menu 3. Update Pegawai.
 - Masukkan nomor pegawai: 1
 - Masukkan Waktu Keluar: 17:20
 - Masukkan Hasil Kerja: Menyelesaikan laporan bulanan.
 - Program memperbarui data pegawai.
- **Hapus Pegawai**
 - Pilih menu 4. Hapus Pegawai.
 - Masukkan nomor pegawai: 1.
 - Program menghapus data pegawai.
- **Keluar dari Program**
 - Pilih menu 5. Keluar.
 - Program menampilkan pesan **Keluar dari program.** dan berhenti.

Skenario 2 : Input Tidak Valid

- **Login Gagal 3 Kali**
 - Masukkan Nama: Putri Renaya
 - Masukkan NIM: 2409106003
- **Login gagal!**
 - Ulangi kesalahan hingga 3 kali.
 - Program menampilkan **Anda telah gagal login sebanyak 3 kali. Program berhenti.** dan keluar.
- **Menambah Pegawai dengan Data Kosong**
 - Login berhasil.
 - Pilih menu 1. Tambah Pegawai.
 - Masukkan Nama Pegawai: *(tekan Enter tanpa mengetik apa pun)*
 - Masukkan Waktu Kehadiran: *(tekan Enter tanpa mengetik apa pun)*

- Program tetap menerima input kosong karena tidak ada validasi (bug yang bisa diperbaiki).
- **Menampilkan Pegawai Saat Data Kosong**
- Pilih menu 2. Tampilkan Pegawai.
 - Bug: Data kosong tetap ditampilkan dalam tabel.
- **Memperbarui Data Pegawai yang Tidak Ada**
- Pilih menu 3. Update Pegawai.
 - Masukkan nomor pegawai: **10** (padahal hanya ada 1 pegawai).
 - Program menampilkan pesan **Nomor pegawai tidak valid!**
- **Menghapus Pegawai yang Tidak Ada**
- Pilih menu 4. Hapus Pegawai.
 - Masukkan nomor pegawai: **10**.
 - Program menampilkan pesan **Nomor pegawai tidak valid!**
- **Keluar dari Program**
- Pilih menu 5. Keluar.
 - Program berhenti.

4.2 Hasil Output

Skenario 1 :

```
Masukkan Nama: Renaya Putri Alika
Masukkan NIM: 2409106002

=== Menu Manajemen Pegawai ===
1. Tambah Pegawai
2. Tampilkan Pegawai
3. Update Pegawai
4. Hapus Pegawai
5. Keluar
Pilihan: 1
Masukkan Nama Pegawai: Thor Odinson
Masukkan Waktu Kehadiran: 07.30

=== Menu Manajemen Pegawai ===
1. Tambah Pegawai
2. Tampilkan Pegawai
3. Update Pegawai
4. Hapus Pegawai
5. Keluar
Pilihan: 2

Daftar Pegawai:
+-----+-----+-----+-----+
| No | Nama           | Waktu Masuk | Waktu Keluar | Hasil Kerja |
+-----+-----+-----+-----+
| 1  | Thor Odinson  | 07.30      | -            | -            |
+-----+-----+-----+-----+
```

Output 4.2.1

```

=== Menu Manajemen Pegawai ===
1. Tambah Pegawai
2. Tampilkan Pegawai
3. Update Pegawai
4. Hapus Pegawai
5. Keluar
Pilihan: 3
Masukkan nomor pegawai yang ingin diperbarui: 1
Masukkan Waktu Keluar: 17.20
Masukkan Hasil Kerja: 10.000

=== Menu Manajemen Pegawai ===
1. Tambah Pegawai
2. Tampilkan Pegawai
3. Update Pegawai
4. Hapus Pegawai
5. Keluar
Pilihan: 2

Daftar Pegawai:
+---+-----+-----+-----+
| No | Nama           | Waktu Masuk | Waktu Keluar | Hasil Kerja |
+---+-----+-----+-----+
| 1  | Thor Odinson  | 07.30      | 17.20       | 10.000      |
+---+-----+-----+-----+

```

Output 4.2.2

```

=== Menu Manajemen Pegawai ===
1. Tambah Pegawai
2. Tampilkan Pegawai
3. Update Pegawai
4. Hapus Pegawai
5. Keluar
Pilihan: 4
Masukkan nomor pegawai yang ingin dihapus: 1
Pegawai berhasil dihapus.

=== Menu Manajemen Pegawai ===
1. Tambah Pegawai
2. Tampilkan Pegawai
3. Update Pegawai
4. Hapus Pegawai
5. Keluar
Pilihan: 2

Daftar Pegawai:
+---+-----+-----+-----+
| No | Nama           | Waktu Masuk | Waktu Keluar | Hasil Kerja |
+---+-----+-----+-----+
+---+-----+-----+-----+

=== Menu Manajemen Pegawai ===
1. Tambah Pegawai
2. Tampilkan Pegawai
3. Update Pegawai
4. Hapus Pegawai
5. Keluar
Pilihan: 5
Keluar dari program.
PS C:\Users\LENOVO\OneDrive\Documents\praktikum-apl\post-test\post-test-2>

```

Output 4.2.3

Skenario 2 :

```
Masukkan Nama: rena
Masukkan NIM: 02
Login gagal! Coba lagi.
Masukkan Nama: naya
Masukkan NIM: 07
Login gagal! Coba lagi.
Masukkan Nama: nayul
Masukkan NIM: 06
Login gagal! Coba lagi.
Anda telah gagal login sebanyak 3 kali. Program berhenti.
PS C:\Users\LENOVO\OneDrive\Documents\praktikum-apl\post-test\post-test-2>
```

Output 4.2.4

```
Masukkan Nama: Renaya Putri Alika
Masukkan NIM: 2409106002

=== Menu Manajemen Pegawai ===
1. Tambah Pegawai
2. Tampilkan Pegawai
3. Update Pegawai
4. Hapus Pegawai
5. Keluar
Pilihan: 1
Masukkan Nama Pegawai:
Masukkan Waktu Kehadiran:

=== Menu Manajemen Pegawai ===
1. Tambah Pegawai
2. Tampilkan Pegawai
3. Update Pegawai
4. Hapus Pegawai
5. Keluar
Pilihan: 2

Daftar Pegawai:
+-----+-----+-----+-----+-----+
| No | Nama                | Waktu Masuk | Waktu Keluar | Hasil Kerja |
+-----+-----+-----+-----+-----+
| 1  |                      |              |              | -           |
+-----+-----+-----+-----+-----+
```

Output 4.2.5

```

+-----+-----+-----+-----+-----+
| 1 |           |           |           |           |
+-----+-----+-----+-----+-----+

=== Menu Manajemen Pegawai ===
1. Tambah Pegawai
2. Tampilkan Pegawai
3. Update Pegawai
4. Hapus Pegawai
5. Keluar
Pilihan: 3
Masukkan nomor pegawai yang ingin diperbarui: 10
Nomor pegawai tidak valid!

=== Menu Manajemen Pegawai ===
1. Tambah Pegawai
2. Tampilkan Pegawai
3. Update Pegawai
4. Hapus Pegawai
5. Keluar
Pilihan: 4
Masukkan nomor pegawai yang ingin dihapus: 10
Nomor pegawai tidak valid!

=== Menu Manajemen Pegawai ===
1. Tambah Pegawai
2. Tampilkan Pegawai
3. Update Pegawai
4. Hapus Pegawai
5. Keluar
Pilihan: 5
Keluar dari program.
PS C:\Users\LENOVO\OneDrive\Documents\praktikum-apl\post-test\post-test-2>

```

Output 4.2.6

5. GIT

1. Git Init

Git Init adalah perintah Git yang digunakan untuk menginisialisasi repository Git baru dalam sebuah folder. Cukup ketik “git init” pada terminal VSCode

```

PS D:\praktikum-apl> git init
Reinitialized existing Git repository in D:/praktikum-apl/.git/

```

- ### 2. Git commit
- Git commit adalah perintah Git yang digunakan untuk menyimpan perubahan yang telah ditambahkan ke staging area ke dalam repository. Setiap commit akan memiliki hash unik, pesan commit, dan menyimpan snapshot dari perubahan yang dilakukan.

```

PS C:\Users\LENOVO\Downloads\praktikum apl> git commit -m "finish Post Test 2"
[main (root-commit) e61d745] finish Post Test 2
2 files changed, 137 insertions(+)
create mode 100644 kelas/pertemuan-2/pertemuan2.cpp
create mode 160000 post-test

```

- ### 3. Git remote
- Git remote adalah perintah Git yang digunakan untuk menghubungkan repository lokal dengan repository remote (misalnya di GitHub, GitLab, atau Bitbucket). Dengan perintah ini, kita bisa mengelola koneksi ke repository jarak jauh, memungkinkan push, pull, dan fetch dari repository tersebut.

```

PS C:\Users\LENOVO\Downloads\praktikum apl> git remote set-url origin https://github.com/Renaya0207/praktikum-apl.git
>>

```


4. Git Push Git push adalah perintah Git yang digunakan untuk mengirim (mengunggah) perubahan dari repository lokal ke repository remote (seperti GitHub, GitLab, atau Bitbucket). Perintah ini memastikan bahwa perubahan yang sudah dikomit di lokal tersedia di repository jarak jauh sehingga bisa diakses oleh orang lain atau untuk cadangan.

```
PS C:\Users\LENOVO\Downloads\praktikum apl> git push origin main
>>
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 1.41 KiB | 481.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Renaya0207/praktikum-apl.git
14d9b60..268bafd main -> main
PS C:\Users\LENOVO\Downloads\praktikum apl>
```

5. Sdh berhasil

