**LAPORAN PRAKTIKUM**

**POSTTEST 5**

**ALGORITMA PEMROGRAMAN LANJUT**



**Disusun oleh:**

**Gadis Wulandari (2409106026)**

**Kelas (A2‘24)**

**PROGRAM STUDI INFORMATIKA**

**UNIVERSITAS MULAWARMAN**

**SAMARINDA**

**2025**

# 1. Flowchart

A diagram of a process

AI-generated content may be incorrect.

Gambar 1.1 Flowchart Menu Awal

A diagram of a diagram

AI-generated content may be incorrect.

Gambar 1.2 Flowchart Menu Registrasi

A diagram of a process

AI-generated content may be incorrect.

Gambar 1.3 Flowchart Menu Login

A diagram of a diagram

AI-generated content may be incorrect.

Gambar 1.4 Flowchart Halaman Utama

A diagram of a perfume

AI-generated content may be incorrect.

Gambar 1.4 Flowchart Menampilkan Data Parfum

A diagram of a diagram

AI-generated content may be incorrect.

Gambar 1.4 Flowchart Menambahkan Parfum

A diagram of a diagram

AI-generated content may be incorrect.

Gambar 1.5 Flowchart Mengubah Data Parfum

A diagram of a perfume

AI-generated content may be incorrect.

Gambar 1.5.1 Flowchart Mengubah Aroma Parfum

A diagram of a diagram

AI-generated content may be incorrect.

Gambar 1.5.2 Flowchart Mengubah Semua Data Parfum

A diagram of a diagram

AI-generated content may be incorrect.

Gambar 1.6 Flowchart Pencarian Parfum Termahal

A diagram of a diagram

AI-generated content may be incorrect.

Gambar 1.7 Flowchart Hapus Parfum

# 2. Deskripsi Singkat Program

## **Tujuan:**

DissScent adalah program manajemen toko parfum berbasis C++ yang dirancang untuk membantu pengelolaan data parfum secara efisien dan terstruktur. Tujuan dari program ini adalah untuk mempermudah pengelolaan stok parfum, mulai dari menambahkan, mengubah, menampilkan, hingga menghapus data parfum yang tersedia di toko. Program ini juga bertujuan untuk mengoptimalkan pencatatan data parfum berdasarkan merk, aroma, harga, dan ukuran, serta meminimalisir kesalahan dalam pendataan produk.

## **Fungsi/Manfaat:**

1. **Registrasi User –** User dapat mendaftar dengan memasukkan nama dan NIM. Program membatasi jumlah user yang dapat terdaftar hingga 50 orang.
2. **Keamanan Login –** Dengan membatasi jumlah percobaan login, program memberikan lapisan keamanan dasar untuk melindungi data user.
3. **Inisialisasi Data Parfum –** Menyediakan data awal parfum dari berbagai brand ternama seperti Alchemist, YSL, Gucci, Chanel, Dior dan lainnya.
4. **CRUD (Create, Read, Update, Delete)**

* Menampilkan Data Parfum : Menyajikan daftar parfum lengkap beserta merk, aroma, harga, dan ukuran.
* Menambah Parfum Baru : Menambahkan produk parfum yang belum tersedia di sistem.
* Mengubah Data Parfum : Mengedit informasi parfum seperti nama, merk, aroma, harga, dan ukuran.
* Menghapus Data Parfum : Menghapus parfum yang sudah tidak tersedia di toko

1. **Penggunaan Array Of Struct –** Data parfum dan user disimpan dalam array of struct, yang memudahkan pengelolaan dan akses data.
2. **Pengulangan Opsi –** Program berjalan secara terus-menerus hingga user memilih untuk keluar, sehingga lebih fleksibel dalam penggunaan.
3. **Fungsi Modular dan Parameter –** Setiap fitur utama dipecah menjadi fungsi/prosedur terpisah dan semuanya menggunakan parameter sesuai instruksi tugas.

8. **Fungsi Address-Of –** Menerima alamat memori dari suatu variabel, bukan nilainya langsung

9. **Fungsi dereference –** fungsi menggunakan nilai dari alamat memori (pointer) yang diterima, lalu mengakses atau memodifikasinya lewat operator \*.

# 3. Source Code

## **A. Fungsi Registrasi**

Untuk mendaftarkan pengguna baru ke dalam sistem.

**Source Code:**

|  |
| --- |
| bool validasiNIM(const string& nim) {     return nim.length() == 3 && nim.find\_first\_not\_of("0123456789") == string::npos; }  void registrasiPengguna() {     if (jumlahUser >= MAX\_USER) {         cout << "Tidak bisa mendaftar lagi, kapasitas sudah penuh.\n";         return;     }     cout << "\nRegistrasi Pengguna Baru\n";     cout << "Nama: "; getline(cin, parfums[jumlahUser].pengguna.nama);      string nimInput;     do {         cout << "NIM (3 digit): "; getline(cin, nimInput);         if (!validasiNIM(nimInput)) {             cout << "NIM harus terdiri dari 3 digit angka!\n";         } else {             break;         }     } while (true);     parfums[jumlahUser].pengguna.nim = nimInput;      jumlahUser++;     cout << "Registrasi berhasil! Anda sekarang dapat login.\n"; } |

Gambar 3.A Fungsi Registrasi

## **B. Fungsi Login**

Memverifikasi nama dan nim (maks 3 percobaan). Jika gagal, program berhenti.

**Source Code:**

|  |
| --- |
| int cariUser(string nama, string nim) {     for (int i = 0; i < jumlahUser; i++) {         if (parfums[i].pengguna.nama == nama && parfums[i].pengguna.nim == nim) {             return i;         }     }     return -1; }              } else if (pilihan == 2) {                 int percobaan = 0;                 cout << "\n=================================================\n";                 cout << "||           MANAJEMEN DISSCENT STORE          ||\n";                 cout << "=================================================\n";                     while (percobaan < 3) {                     cout << "\nSelamat datang di Manajemen DisScent Store!\n";                     cout << "Masukkan Nama: "; getline(cin, nama);                     cout << "Masukkan NIM (3 digit): "; getline(cin, nim);                         if (cariUser(nama, nim) != -1) {                         cout << "Login berhasil! Selamat datang, " << nama << " NIM: " << nim << ".\n";                      } else {                         percobaan++;                         cout << "\nLogin gagal, silahkan coba lagi! Percobaan ke-" << percobaan << " dari 3.\n";                     }                 }                     if (percobaan == 3) {                     cout << "\nYahh login sudah gagal sebanyak 3 kali. Program berhenti.\n";                     exit(0);                 }             } else if (pilihan == 3) {                 cout << "Keluar dari program, Terima kasih!\n";             } else {                 cout << "Pilihan tidak valid!\n";             }         } while (pilihan != 3);      } while (true);           return 0; } |

Gambar 3.B Fungsi Login

## **C. Fungsi Menu Utama**

Memberikan struktur dan navigasi bagi pengguna. Ini memungkinkan pengguna untuk dengan mudah mengakses berbagai fitur yang tersedia dalam sistem manajemen parfum.

**Source Code:**

|  |
| --- |
| do {                         cout << "\n=================================================\n";                         cout << "||                MANAJEMEN PARFUM             ||\n";                         cout << "=================================================\n";                         cout << "1. Tampilkan Parfum\n";                         cout << "2. Tambah Parfum\n";                         cout << "3. Ubah Parfum\n";                         cout << "4. Hapus Parfum\n";                         cout << "5. Pencarian Parfum Termahal\n";                         cout << "6. Kembali\n";                         cout << "=================================================\n";                         cout << "Pilih menu: "; cin >> menuPilihan;                         cin.ignore();                          switch (menuPilihan) {                             case 1: tampilkanParfum(); break;                             case 2: tambahParfum(parfums, jumlahParfum); break;                             case 3: ubahParfum(parfums, jumlahParfum); break;                             case 4: hapusParfum(parfums, jumlahParfum); break;                             case 5: tampilkanParfumTermahal(parfums, jumlahParfum); break;                             case 6: cout << "Logout berhasil.\n"; break;                             default: cout << "Pilihan tidak valid!\n"; break;                         }                     } while (menuPilihan != 6);                     break;                 } else {                     percobaan++;                     cout << "\nLogin gagal, silahkan coba lagi! Percobaan ke-" << percobaan << " dari 3.\n";                 }             }              if (percobaan == 3) {                 cout << "\nYahh login sudah gagal sebanyak 3 kali. Program berhenti.\n";                 break; *// keluar dari menu utama*             }          } else if (pilihan == 3) {             cout << "Keluar dari program, Terima kasih!\n";             break;         } else {             cout << "Pilihan tidak valid!\n";         }      } while (true);      return 0; } |

Gambar 3.C Fungsi Menu Utama

## **D. Fungsi Tampilkan Data Parfum**

Menampilkan seluruh data parfum yang tersedia, seperti merk, nama parfum, aroma, harga, dan ukuran parfum dalam bentuk tabel.

**Source Code:**

|  |
| --- |
| void tampilkanParfum(int index = 0) {     if (index == 0) {         cout << "\n+----+---------------------------+---------------------------+--------------------+------------+------------+" << endl;         cout << "| No | Merk                     | Nama Parfum               | Aroma             | Harga      | Ukuran     |" << endl;         cout << "+----+---------------------------+---------------------------+--------------------+------------+------------+" << endl;     }     if (index >= jumlahParfum) {         cout << "+----+---------------------------+---------------------------+--------------------+------------+------------+\n";         return;     }     cout << "| " << (index + 1) << "  | " << parfums[index].merk;     cout << string(25 - parfums[index].merk.length(), ' ') << "| " << parfums[index].nama;     cout << string(25 - parfums[index].nama.length(), ' ') << "| " << parfums[index].aroma;     cout << string(18 - parfums[index].aroma.length(), ' ') << "| " << parfums[index].harga;     cout << " | " << parfums[index].ukuran << " |" << endl;     tampilkanParfum(index + 1); } |

Gambar 3.D Tampilkan Data

## **E. Fungsi Tambah Parfum**

Menambahkan data parfum baru ke dalam daftar parfum yang sudah ada, seperti merk, nama parfum, aroma, harga, dan ukuran parfum.

**Source Code:**

|  |
| --- |
| void tambahParfum(Parfum data[], int &jumlah) {     if (jumlah >= MAX\_PARFUM) {         cout << "Tidak bisa menambah parfum lagi, kapasitas sudah penuh.\n";         return;     }     cout << "\nTambah Parfum Baru\n";     inputDataParfum(data[jumlah], false);     jumlah++;     cout << "Parfum berhasil ditambahkan!\n";     tampilkanParfum(); } |

Gambar 3.E Tambah Data

## **F. Fungsi Ubah Data Parfum**

Mengedit atau memperbarui data parfum yang sudah ada sesuai dengan input user.

**Source Code:**

|  |
| --- |
| void ubahParfum(Parfum data[], int jumlah) {     tampilkanParfum();     int index;     cout << "Pilih nomor parfum yang ingin diubah: "; cin >> index; cin.ignore();     if (index > 0 && index <= jumlah) {         cout << "Ubah Data Parfum:\n";         int pilihan;         do {             cout << "\n=================================================\n";             cout << "||                  UBAH PARFUM                ||\n";             cout << "=================================================\n";             cout << "1. Ubah Aroma Parfum\n";             cout << "2. Ubah Semua Data Parfum\n";             cout << "3. Kembali ke Menu Utama\n";             cout << "=================================================\n";             cout << "Pilih menu: "; cin >> pilihan; cin.ignore();             switch (pilihan) {                 case 1:                     cout << "Aroma sebelumnya: " << data[index - 1].aroma << endl;                     cout << "Masukkan aroma baru: ";                     getline(cin, data[index - 1].aroma);                     cout << "Aroma parfum berhasil diubah!\n";                     tampilkanParfum();                     break;                 case 2:                     inputDataParfum(data[index - 1]);                     cout << "Semua data parfum berhasil diubah!\n";                     tampilkanParfum();                     break;                 case 3:                     cout << "Kembali ke menu utama.\n";                     break;                 default:                     cout << "Pilihan tidak valid.\n";             }         } while (pilihan != 3);     } else {         cout << "Nomor parfum tidak ada!\n";     } } |

Gambar 3.5.F Ubah Data

## **G. Fungsi Hapus Parfum**

Menghapus data parfum yang dipilih oleh pengguna dari daftar parfum yang tersedia.

**Source Code:**

|  |
| --- |
| void hapusParfum(Parfum data[], int &jumlah) {     tampilkanParfum();     int index;     cout << "Pilih nomor parfum yang ingin dihapus: ";     cin >> index;     cin.ignore();     if (index > 0 && index <= jumlah) {         for (int i = index - 1; i < jumlah - 1; i++) {             data[i] = data[i + 1];         }         jumlah--;         cout << "Parfum berhasil dihapus.\n";     } else {         cout << "Nomor parfum tidak valid!\n";     }     tampilkanParfum(); } |

Gambar 3.G Hapus Data

**H. Fungsi Struct**

• User dipakai untuk menyimpan data pengguna (nama dan NIM).

• Parfum menyimpan info parfum (merk, nama, aroma, harga, ukuran) plus info pengguna yang menyimpannya (User pengguna;) → ini contoh nested struct (struct di dalam struct).

**Source Code:**

struct User { string nama; string nim;

}; struct Parfum { string merk; string nama; string aroma; string harga; string ukuran; User pengguna;

};

Gambar 3.H Fungsi Struct

## **I. Fungsi Overloading**

Nama sama (pilihUkuran) tapi parameter beda, dan dipakai sesuai kebutuhan.

**Source Code:**

|  |
| --- |
| string pilihUkuran() {     int pilihanUkuran;     do {         cout << "Ukuran Parfum:\n1. 50 ml\n2. 100 ml\nPilih ukuran (1/2): ";         cin >> pilihanUkuran;         cin.ignore();         if (pilihanUkuran == 1) return "50 ml";         else if (pilihanUkuran == 2) return "100 ml";         else cout << "Pilihan tidak valid, silakan pilih 1 atau 2.\n";     } while (true); }  string pilihUkuran(string defaultUkuran) {     cout << "Ukuran sebelumnya: " << defaultUkuran << endl;     cout << "Ingin ubah ukuran? (y/n): ";     char ubah;     cin >> ubah;     cin.ignore();     if (ubah == 'y' || ubah == 'Y') return pilihUkuran();     return defaultUkuran; } |

Gambar 3.I Fungsi Overloading

## **J. Fungsi Pencarian Parfum Termahal**

Fungsi ini menampilkan parfum dengan harga tertinggi dari seluruh data yang tersedia.

**Source Code:**

|  |
| --- |
| void tampilkanParfumTermahal(Parfum \*data, int jumlah) {     if (jumlah == 0) {         cout << "Belum ada parfum yang tersedia.\n";         return;     }      int maxHarga = -1;     int indexTermahal = -1;      for (int i = 0; i < jumlah; i++) {         if (data[i].harga.substr(0, 3) == "Rp ") {             int harga = stoi(data[i].harga.substr(3));             if (harga > maxHarga) {                 maxHarga = harga;                 indexTermahal = i;             }         }     }      if (indexTermahal != -1) {         cout << "\n====== Parfum Termahal ======\n";         cout << "Merk  : " << data[indexTermahal].merk << endl;         cout << "Nama  : " << data[indexTermahal].nama << endl;         cout << "Aroma : " << data[indexTermahal].aroma << endl;         cout << "Harga : " << data[indexTermahal].harga << endl;         cout << "Ukuran: " << data[indexTermahal].ukuran << endl;         cout << "==============================\n";     } else {         cout << "Tidak ada parfum dengan format harga yang valid.\n";     } } |

Gambar 3.J Fungsi Overloading

## **K. Fungsi Parameter Address-of (&)**

Fungsi yang menerima referensi langsung ke variabel, sehingga perubahan yang dilakukan di dalam fungsi akan memengaruhi nilai asli variabel tersebut.

**Source Code:**

|  |
| --- |
| void tambahParfum(Parfum data[], int &jumlah) { |

Gambar 3.K Fungsi Parameter Address-of

## **L. Fungsi Parameter Dereference (\*)**

Fungsi yang menerima alamat memori sebagai parameter, lalu mengakses atau mengubah data yang ditunjuk oleh alamat tersebut.

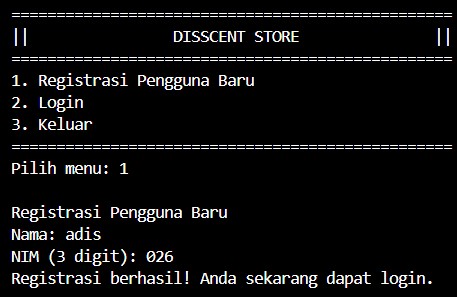
**Source Code:**

|  |
| --- |
| void tampilkanParfumTermahal(Parfum \*data, int jumlah) { |

Gambar 3.L Fungsi Parameter Dereference

# 4. Hasil Output

## A. Hasil Output Registrasi



Gambar 4.A Registrasi

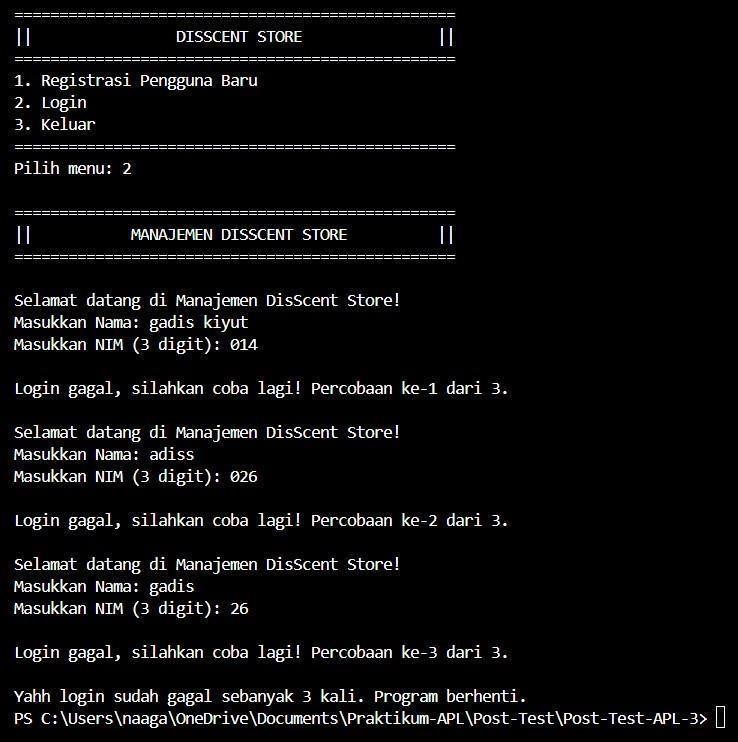
## B. Hasil Output Login Berhasil

A screenshot of a computer screen

AI-generated content may be incorrect.

Gambar 4.B Login Berhasil

## C. Hasil Output Login Gagal



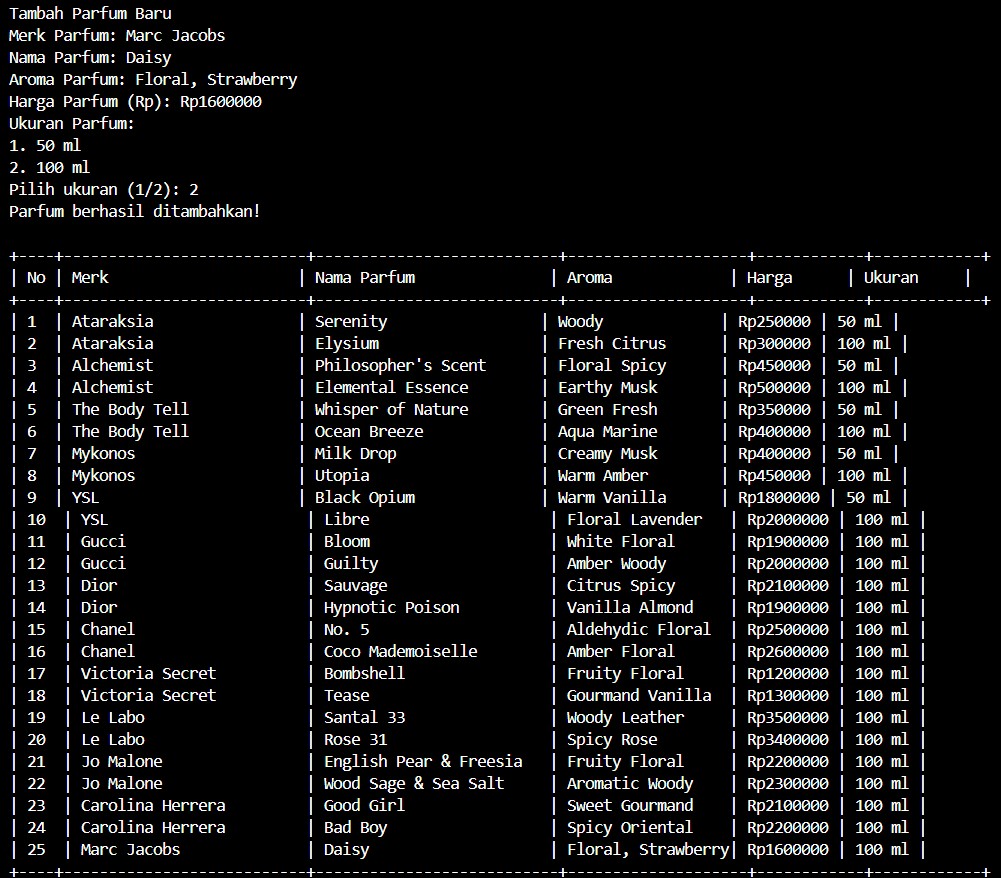
Gambar 4.C Login Gagal

## D. Hasil Output Menampilkan Data Parfum



Gambar 4.2.D Menampilkan Data parfum

## E. Hasil Output Menambah Parfum



Gambar 4.E Menambah Parfum

## F. Hasil Output Mengubah Data Parfum

A screenshot of a computer

AI-generated content may be incorrect.

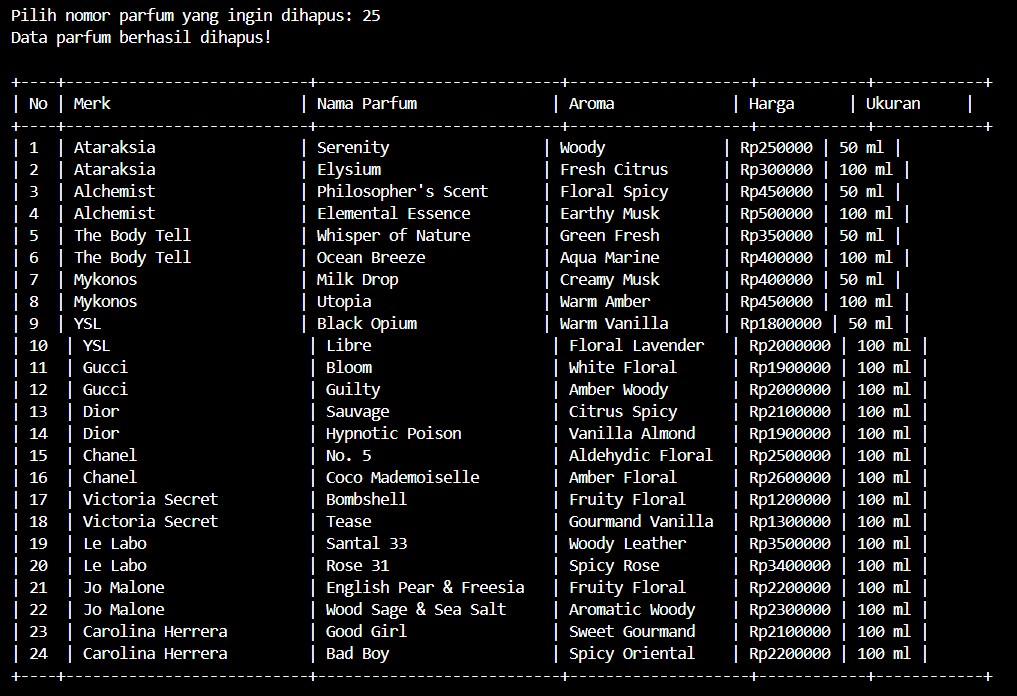
Gambar 4.F.1 Mengubah Semua Data Parfum

A screenshot of a computer

AI-generated content may be incorrect.

Gambar 4.F.2 Mengubah Aroma Parfum

## G. Hasil Output Menghapus Parfum



Gambar 4.G Menghapus Parfum

## H. Pencarian Parfum Termahal

A screenshot of a computer

AI-generated content may be incorrect.

Gambar 4.H Pencarian Parfum Termahal

## I. Hasil Output Keluar Dari Program

A screenshot of a computer

AI-generated content may be incorrect.

## Gambar 4.H Keluar dari Program

# 5. Penggunaan Git

## **1) Git Add .**

Menambahkan semua perubahan file (baru, diubah, atau dihapus) ke dalam staging area sebelum dikomit. Titik (.) berarti semua file dalam folder proyek akan ditambahkan ke staging area.



Gambar 5.1 Git Add .

## **2) Git Commit**

Menyimpan perubahan yang ada di staging area ke dalam repository dengan pesan deskriptif. Setiap commit mencatat perubahan sebagai versi yang bisa dikembalikan nanti.

A screen shot of a computer

AI-generated content may be incorrect.

Gambar 5.2 Git Commit

## **3) Git Push**

Mengirim (upload) commit dari branch lokal ke repository remote yang ada di server github.

A screen shot of a computer program

AI-generated content may be incorrect.

Gambar 5.3 Git Push