**LAPORAN PRAKTIKUM POSTTEST 3**

**ALGORITMA PEMROGRAMAN LANJUT**



**Disusun oleh:**

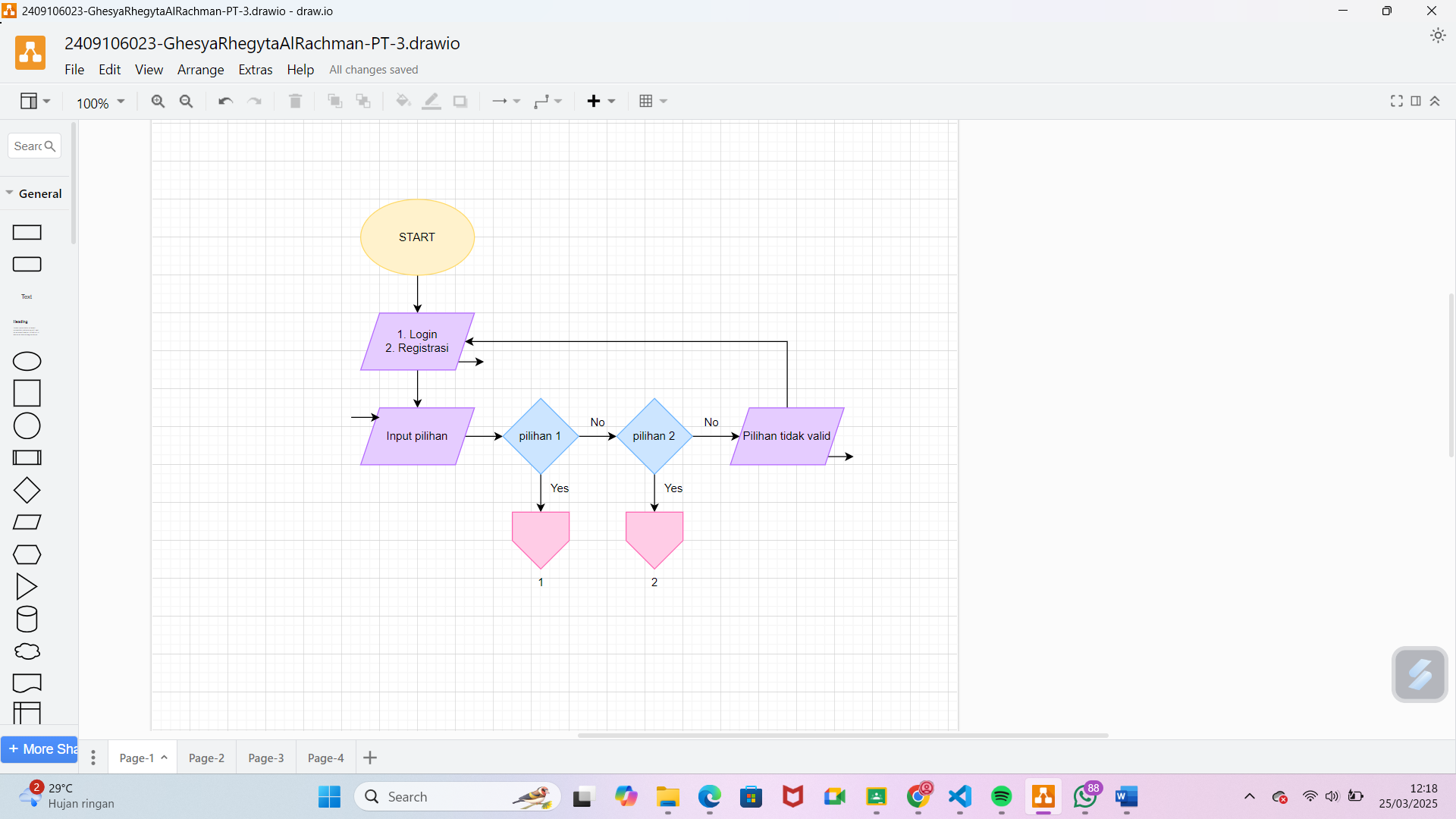
**Ghesya Rhegyta Al Rachman (2409106023)**

**Kelas (A2’24)**

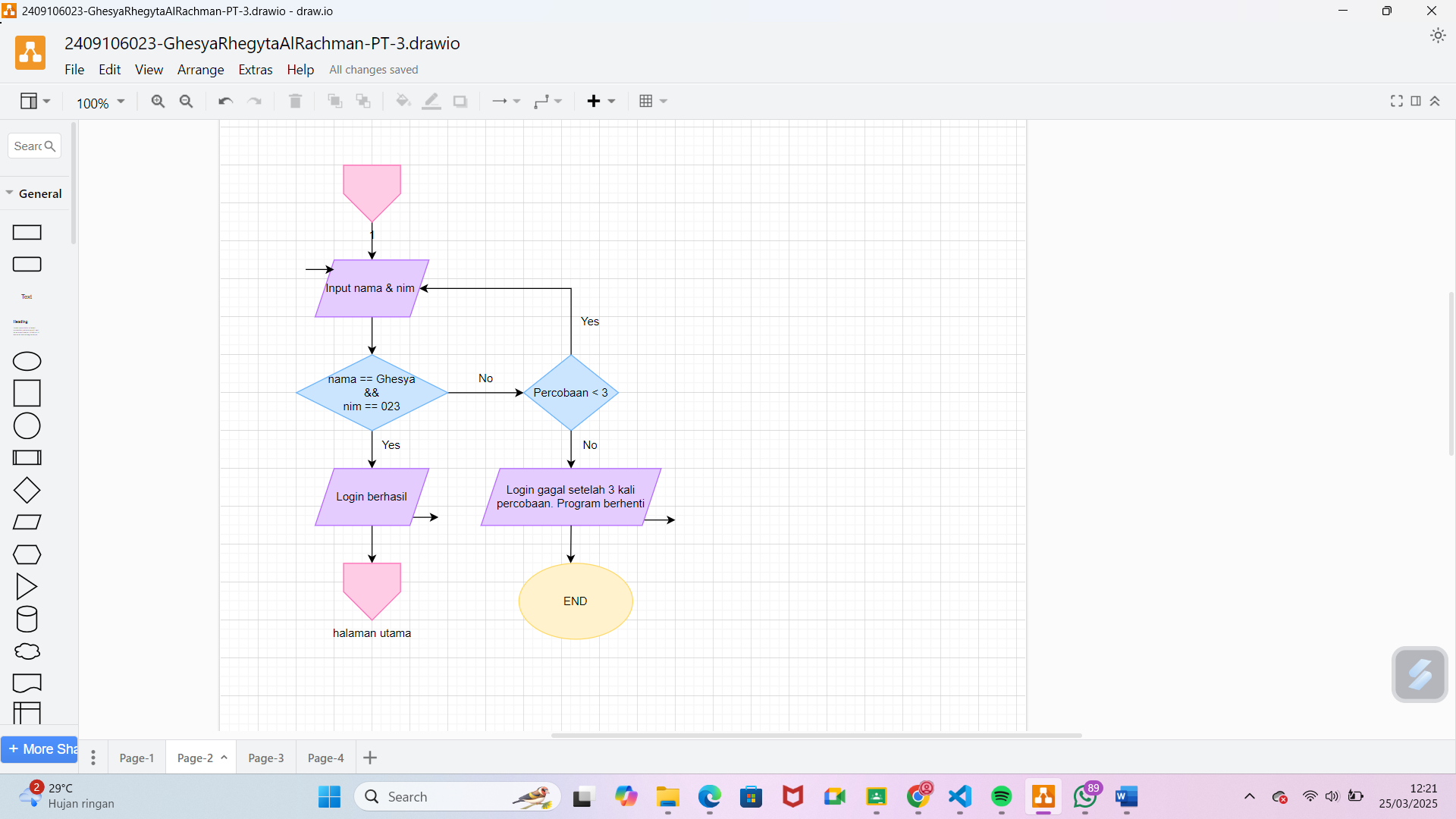
**PROGRAM STUDI INFORMATIKA UNIVERSITAS MULAWARMAN**

**SAMARINDA 2025**

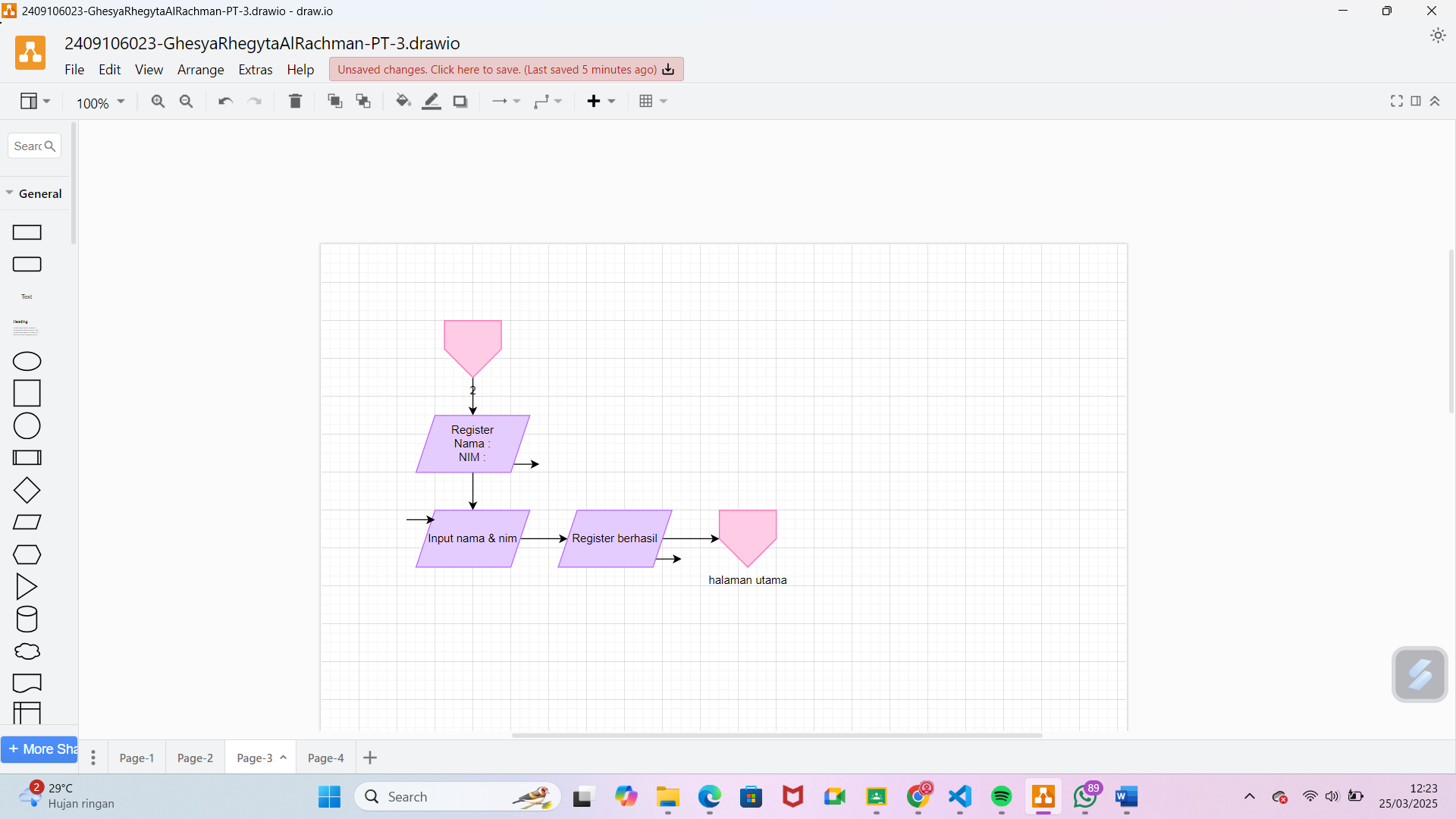
# Flowchart



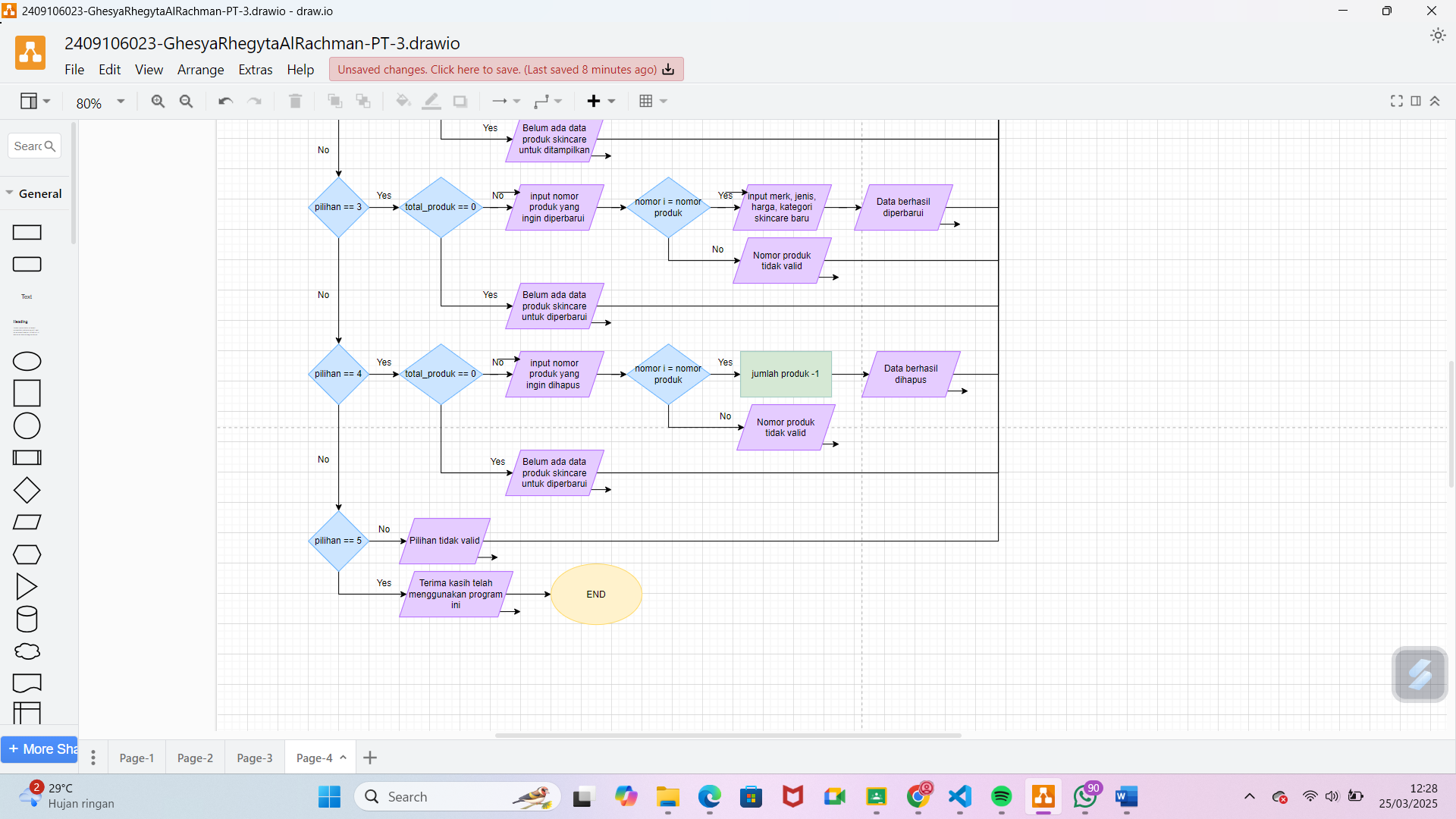
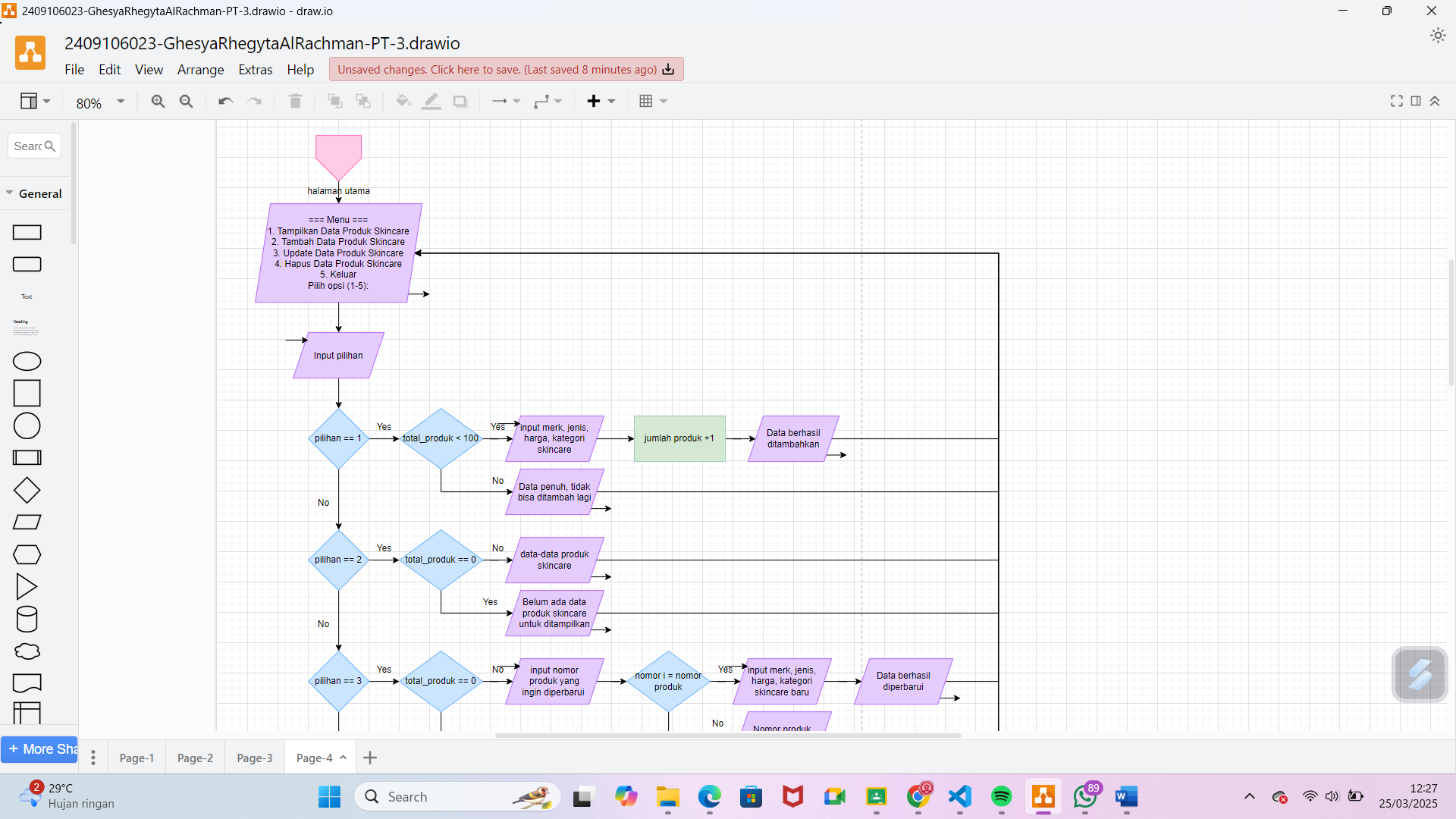
Gambar 1.1 Flowchart Pilihan Menu



Gambar 1.2 Flowchart Login



Gambar 1.3 Flowchart Register



Gambar 1.4 Flowchart Halaman Utama

# Analisis Program

**2.1 Deskripsi Singkat Program**

**Tujuan :**

Program ini bertujuan untuk membantu pengelolaan data produk skincare. Dengan adanya fitur multiuser, pengguna harus login atau registrasi sebelum mengakses program ini. Pengguna dapat menambah, melihat, memperbarui, serta menghapus data produk skincare. Program ini dapat membantu bisnis kecil untuk mencatat produk skincare.

**Manfaat :**

* Mudah digunakan – Dengan adanya menu yang jelas, pengguna dapat mengelola data skincare tanpa kesulitan
* Dapat dikembangkan lebih lanjut – Dapat ditambahkan fitur seperti pencarian produk
* Perulangan – Program terus berjalan hingga pengguna memilih untuk berhenti

# Source Code

* 1. **Fungsi Login**

Memverifikasi nama dan 3 digit terakhir NIM dengan batas maksimal 3 kali percobaan. Jika gagal, maka program berhenti.

|  |
| --- |
| int main() {  string nama, inputNama;  string nim, inputNIM;  int attempts = 0;   cout << "=== LOGIN ===" << endl;  cout << "Masukkan nama: ";  cin >> nama;  cout << "Masukkan 3 digit terakhir NIM: ";  cin >> nim;   if (nama == "Ghesya" && nim == "023") {  cout << "Login berhasil!" << endl;  } else {  attempts++;  while (attempts < 3) {  cout << "\nLogin Ulang" << endl;  cout << "Masukkan nama: ";  cin >> inputNama;  cout << "Masukkan 3 digit terakhir NIM: ";  cin >> inputNIM;   if (inputNama == "Ghesya" && inputNIM == "023") {  cout << "Login berhasil!" << endl;  break;  } else {  attempts++;  cout << "Login gagal! Percobaan ke-" << attempts << " dari 3" << endl;  }  }  }   if (attempts == 3) {  cout << "Login gagal setelah 3 kali percobaan. Program berhenti." << endl;  return 0;  } |

Gambar 3.1.1 Fungsi Login

* 1. **Menu Utama**

Menampilkan pilihan menu utama produk skincare

|  |
| --- |
| while (true) {  int pilihan;  cout << "\n-----------------------------------------" << endl;  cout << "|| === Menu === ||" << endl;  cout << "-----------------------------------------" << endl;  cout << "|| 1. Tambah Data Produk Skincare ||" << endl;  cout << "|| 2. Tampilkan Data Produk Skincare ||" << endl;  cout << "|| 3. Perbarui Data Produk Skincare ||" << endl;  cout << "|| 4. Hapus Data Produk Skincare ||" << endl;  cout << "|| 5. Keluar dari program ||" << endl;  cout << "-----------------------------------------" << endl;  cout << "Pilih opsi (1-5): ";  cin >> pilihan; |

Gambar 3.2.1 Menu Utama Produk Skincare

* 1. **CRUD Manajemen Produk Skincare**

Fitur CRUD dalam program ini yaitu Create (membuat atau menambahkan data), Read (menampilkan data yang tersimpan), Update (memperbarui data yang sudah ada), dan Delete (menghapus data yang sudah tidak diperlukan)

|  |
| --- |
| if (pilihan == 1) {  if (total\_produk < 100) {  cout << "Masukkan merk skincare: ";  cin >> merk[total\_produk];  cout << "Masukkan jenis skincare: ";  cin >> jenis[total\_produk];  cout << "Masukkan harga skincare: ";  cin >> harga[total\_produk];   total\_produk++;  cout << "Data berhasil ditambahkan!" << endl;  } else {  cout << "Jumlah data sudah penuh >\_< huhuhuu tidak bisa ditambah lagi" << endl;  }  }  else if (pilihan == 2) {  if (total\_produk == 0) {  cout << "Belum ada data produk skincare untuk ditampilkan" << endl;  } else {  cout << "\n-----------------------------------------------------" << endl;  cout << "=== Data Produk Skincare ===" << endl;  cout << "-----------------------------------------------------" << endl;  for (int i = 0; i < total\_produk; ++i) {  cout << i + 1 << " | Merk : " << merk[i] << ", Jenis : " << jenis[i] << ", Harga : " << harga[i] << endl;  }  cout << "-----------------------------------------------------" << endl;  }  }  else if (pilihan == 3) {  if (total\_produk == 0) {  cout << "Belum ada data produk skincare untuk diperbarui" << endl;  } else {  int no;  cout << "\n=== Data Produk Skincare ===" << endl;  cout << "-----------------------------------------------------" << endl;  for (int i = 0; i < total\_produk; ++i) {  cout << i + 1 << " | Merk : " << merk[i] << ", Jenis : " << jenis[i] << ", Harga : " << harga[i] << endl;  }  cout << "-----------------------------------------------------" << endl;  cout << "\nMasukkan nomor produk yang ingin diperbarui: ";  cin >> no;   if (no < 1 || no > total\_produk) {  cout << "Nomor produk tidak valid" << endl;  } else {  cout << "Masukkan merk baru: ";  cin >> merk[no - 1];  cout << "Masukkan jenis baru: ";  cin >> jenis[no - 1];  cout << "Masukkan harga baru: ";  cin >> harga[no - 1];  cout << "Data berhasil diperbarui!" << endl;  }  }  }  else if (pilihan == 4) {  if (total\_produk == 0) {  cout << "Belum ada data produk skincare untuk dihapus" << endl;  } else {  int no;  cout << "\n=== Data Produk Skincare ===" << endl;  cout << "-----------------------------------------------------" << endl;  for (int i = 0; i < total\_produk; ++i) {  cout << i + 1 << " | Merk : " << merk[i] << ", Jenis : " << jenis[i] << ", Harga : " << harga[i] << endl;  }  cout << "-----------------------------------------------------" << endl;  cout << "\nMasukkan nomor produk yang ingin dihapus: ";  cin >> no;   if (no < 1 || no > total\_produk) {  cout << "Nomor produk tidak valid" << endl;  } else {  for (int i = no - 1; i < total\_produk - 1; ++i) {  merk[i] = merk[i + 1];  jenis[i] = jenis[i + 1];  harga[i] = harga[i + 1];  }  total\_produk--;  cout << "Data berhasil dihapus!" << endl;  }  }  }  else if (pilihan == 5) {  cout << "Timaaci telah menggunakan program ini!!! papayyy" << endl;  break;  } else {  cout << "Pilihan anda tidak valid!" << endl;  }  }   return 0; } |

Gambar 3.3.1 CRUD Manajemen Produk Skincare

# Uji Coba dan Hasil Output

* 1. **Uji Coba**

1. Skenario 1 : Login dengan nama atau NIM yang salah

* Input nama atau NIM yang salah sebanyak 3 kali
* Jika pengguna menginputkan data yang benar sebelum 3 kali percobaan, maka pengguna dapat mengakses ke menu utama
* Setelah 3 kali percobaan gagal, maka program akan berhenti

1. Skenario 2 : Login berhasil

* Input nama, yaitu “Ghesya”
* Input 3 digit terakhir NIM, yaitu “023”
* Login berhasil dan program menampilkan menu utama

1. Skenario 3 : Menambahkan data produk skincare

* Memilih opsi 1 yaitu “Tambah Data Produk Skincare”
* Input data produk yaitu merk, jenis, dan harga skincare
* Program akan menampilkan “Data berhasil ditambahkan!”
* Jika data yang tersimpan sudah 100 (maksimal), maka program menampilkan pesan “Jumlah data sudah penuh >\_< huhuhuu tidak bisa ditambah lagi”

1. Skenario 4 : Menampilkan data produk skincare

* Memilih opsi 2 yaitu “Tampilkan Data Produk Skincare”
* Program akan menampilkan seluruh data produk skincare
* Jika tidak ada data yang tersimpan, maka program menampilkan pesan “Belum ada data produk skincare untuk ditampilkan”

1. Skenario 5 : Memperbarui data produk skincare

* Memilih opsi 3 yaitu “Perbarui Data Produk Skincare”
* Input nomor produk yang ingin diperbarui
* Input data baru untuk menggantikan data lama produk skincare
* Jika tidak ada data yang tersimpan, maka program menampilkan pesan “Belum ada data produk skincare untuk diperbarui”

1. Skenario 6 : Menghapus data produk skincare

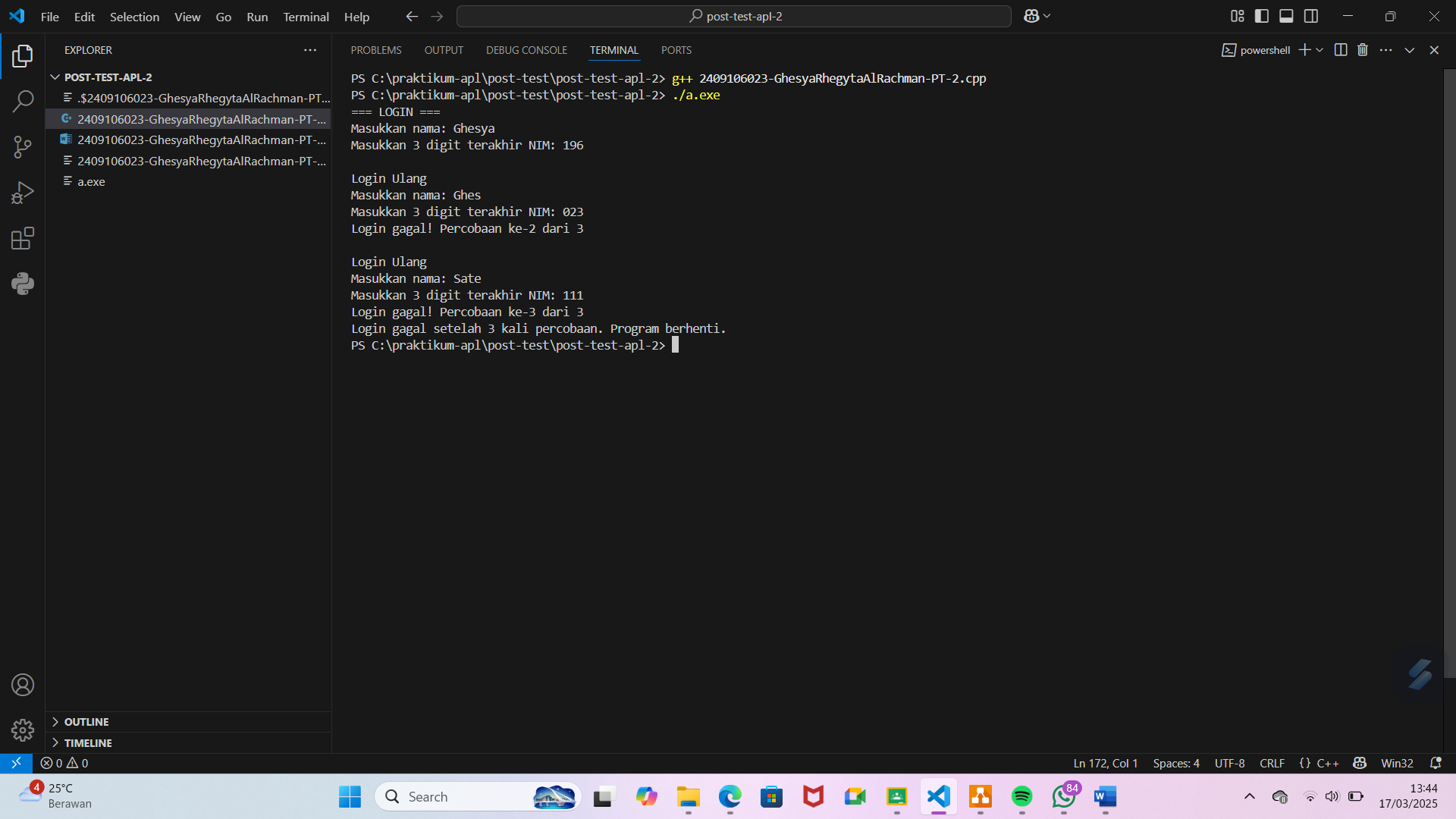
* Memilih opsi 4 yaitu “Hapus Data Produk Skincare”
* Input nomor produk yang ingin dihapus
* Program akan menampilkan “Data berhasil dihapus!”
* Jika tidak ada data yang tersimpan, maka program menampilkan pesan “Belum ada data produk skincare untuk dihapus”

1. Skenario 7 : Keluar dari program

* Memilih opsi 4 yaitu “Keluar dari program”
* Program akan menampilkan “Timaaci telah menggunakan program ini!!! papayyy”
* Program berhenti

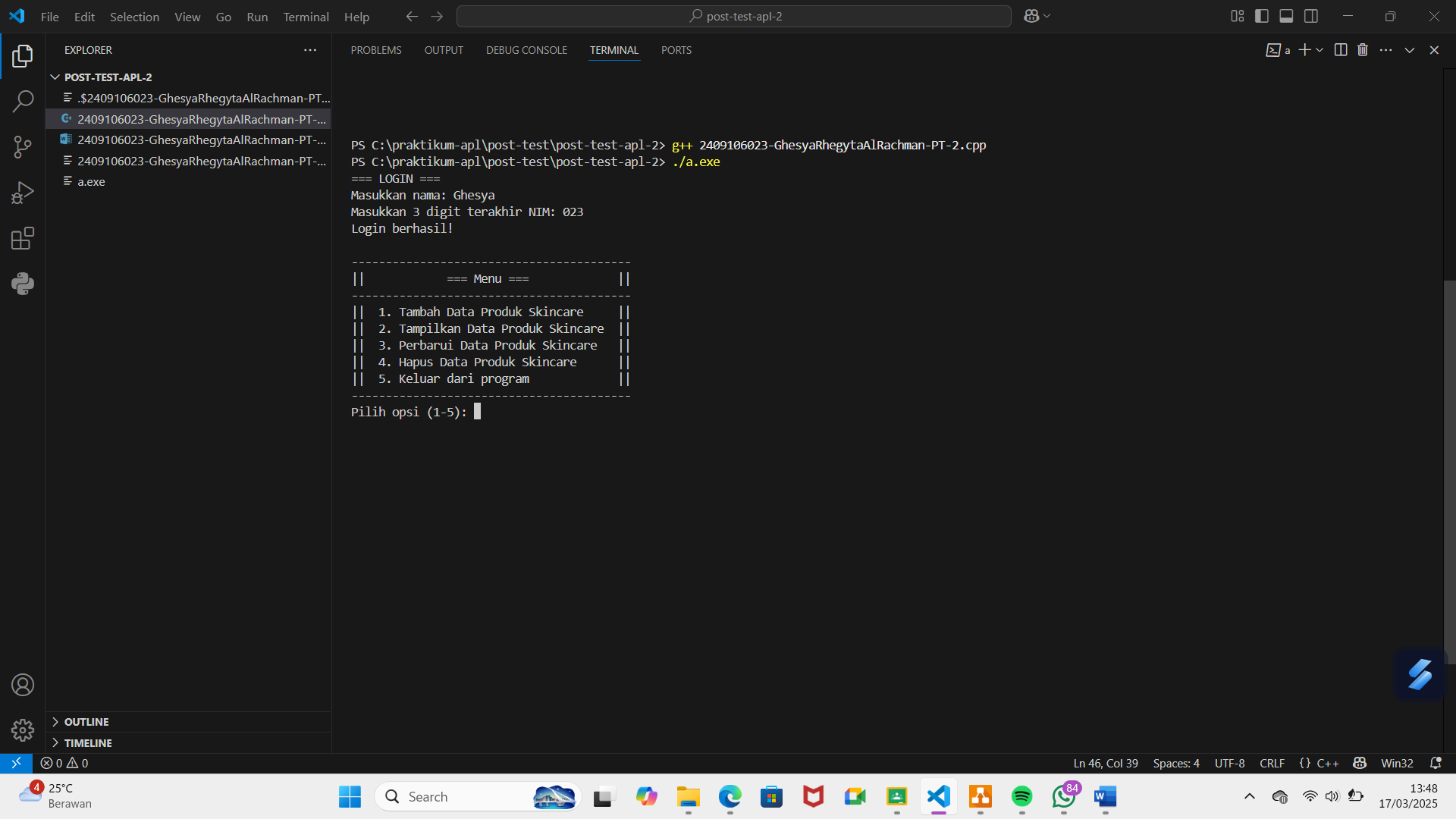
# Hasil Output

* Hasil Output Uji Coba Skenario 1



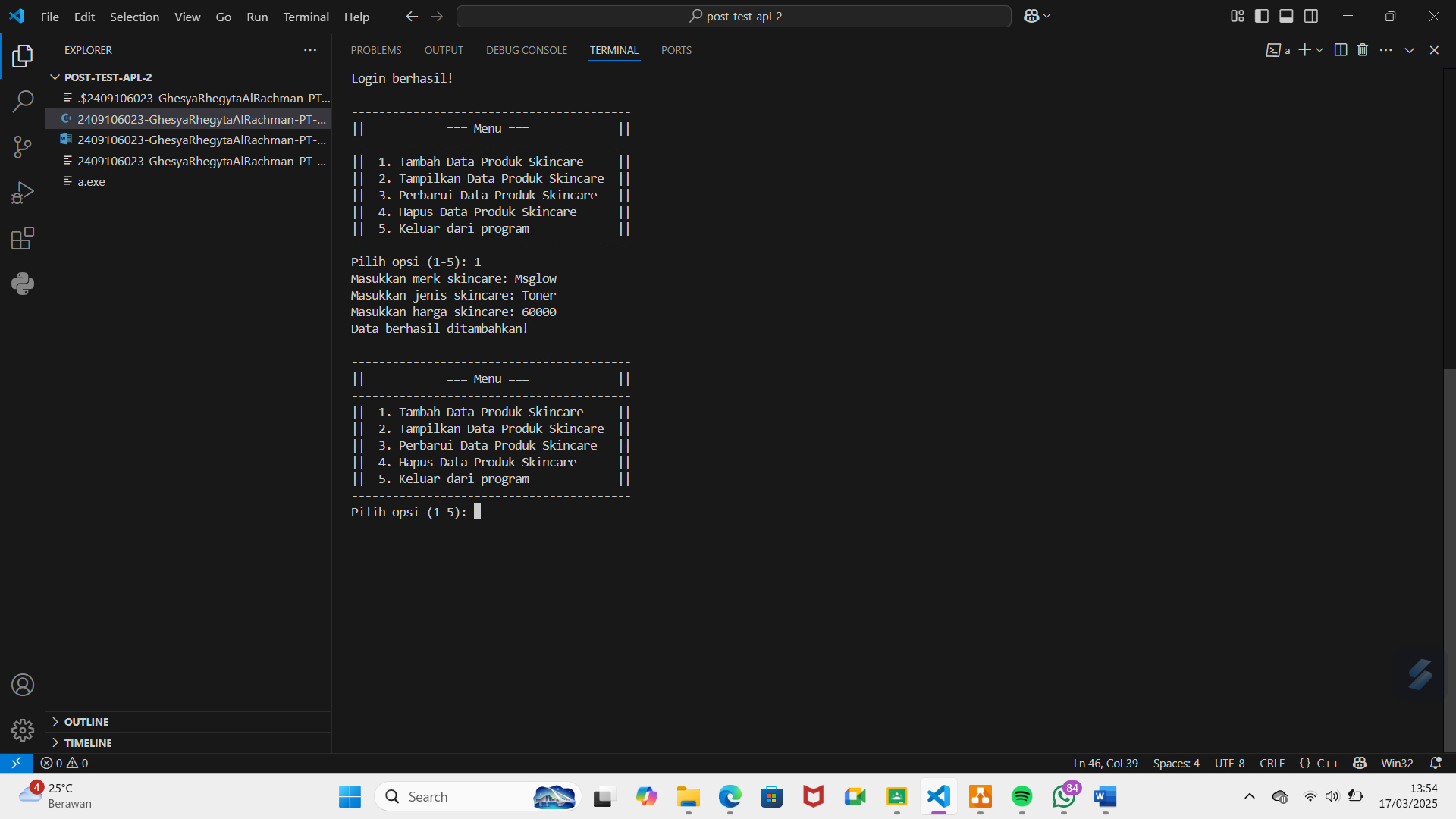
Gambar 4.2.1 Login gagal dan program berhenti

* Hasil Output Uji Coba Skenario 2



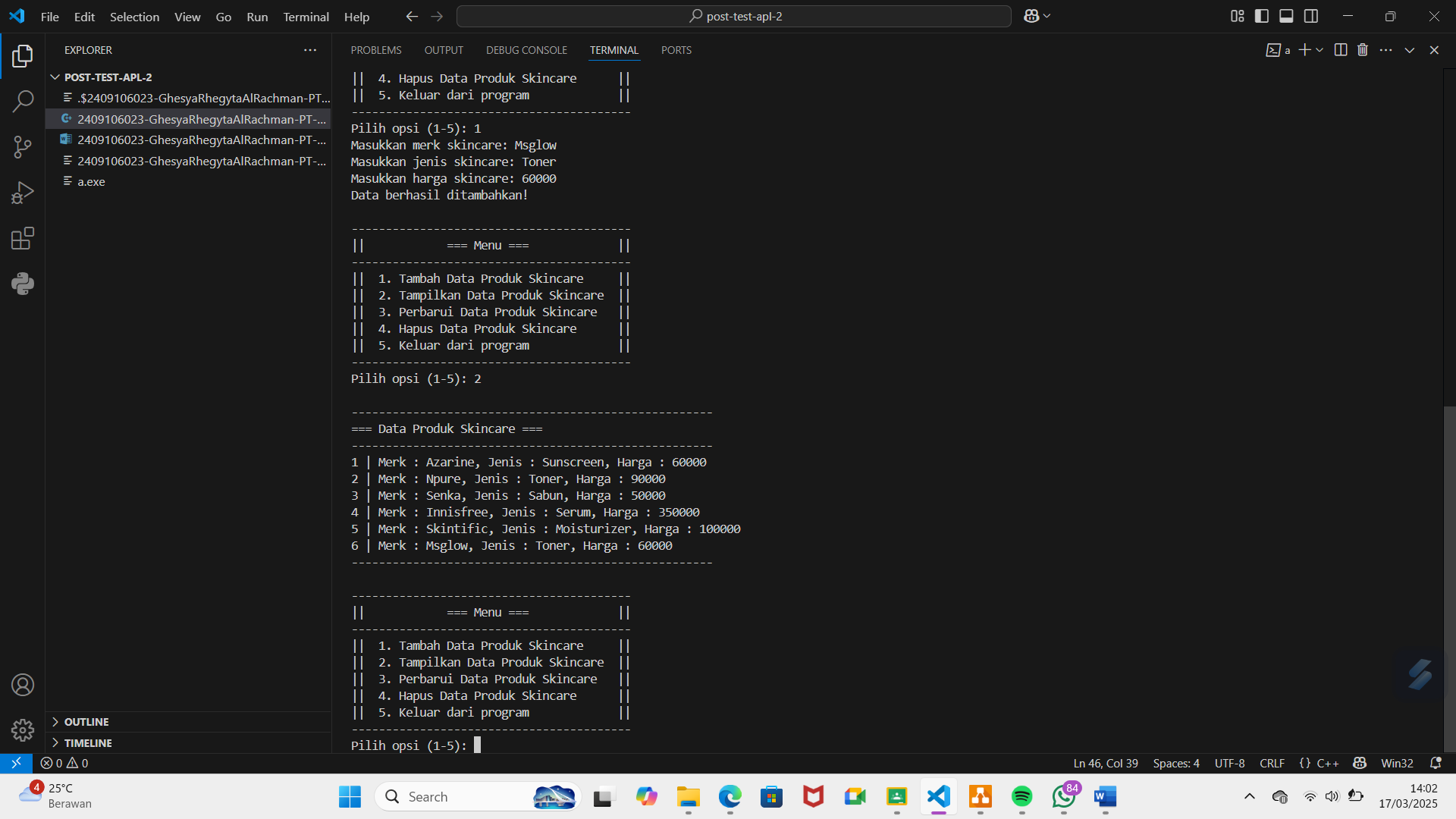
Gambar 4.2.2 Login berhasil dan program menampilkan menu utama

* Hasil Output Uji Coba Skenario 3



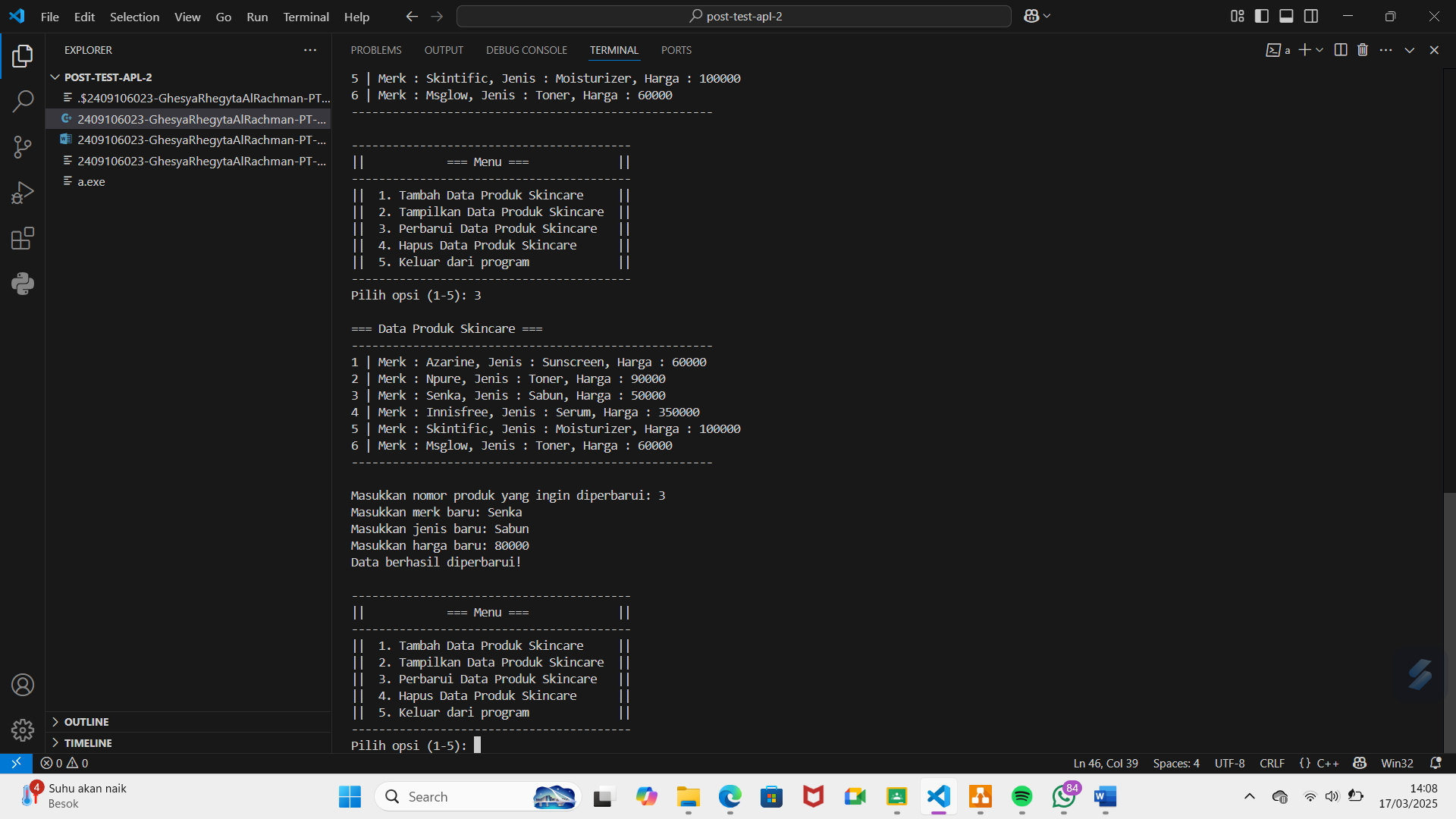
Gambar 4.2.3 Tambah data produk

* Hasil Output Uji Coba Skenario 4

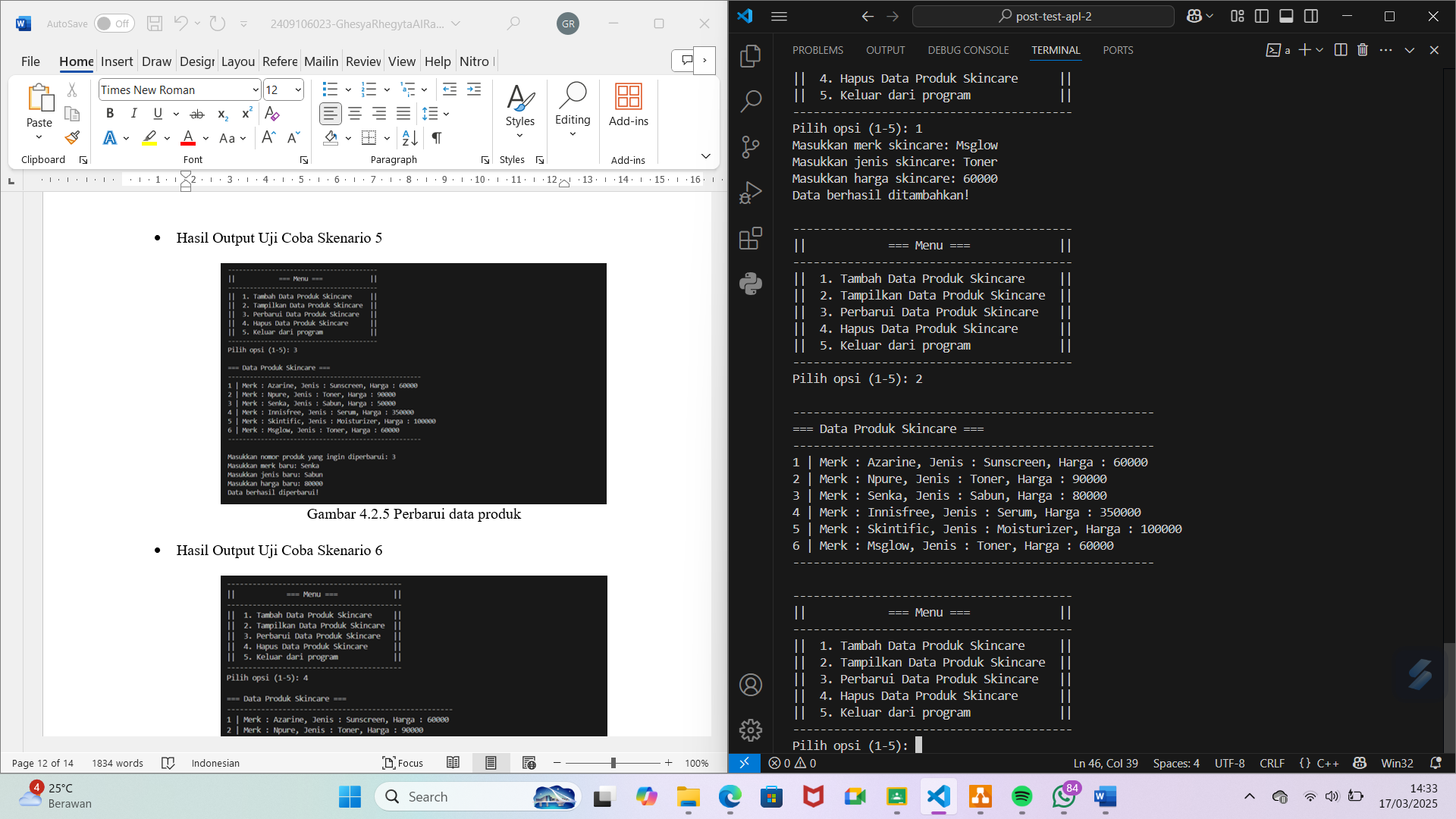


Gambar 4.2.4 Tampilkan data produk

* Hasil Output Uji Coba Skenario 5

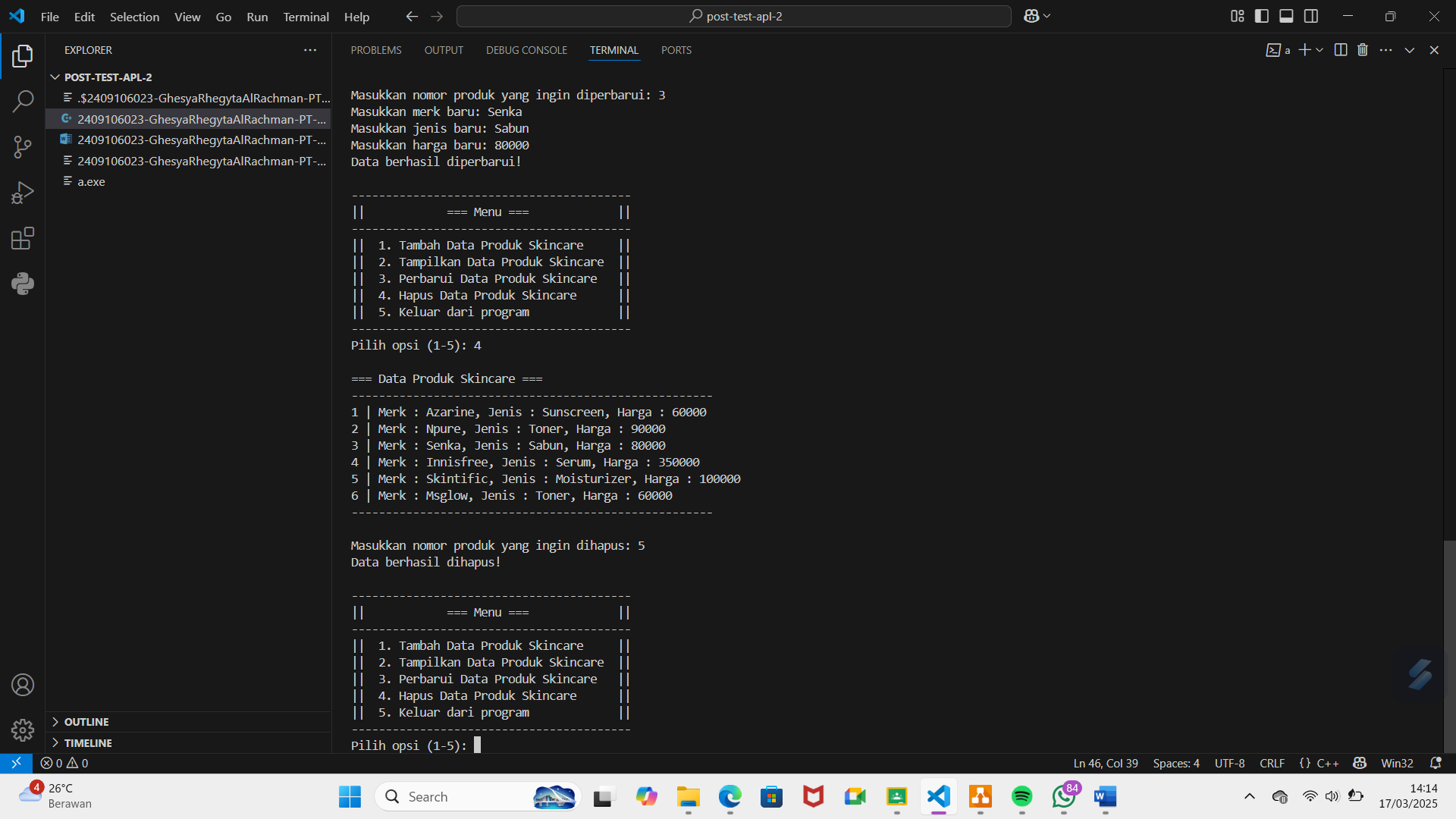


Gambar 4.2.5 Perbarui data produk



Gambar 4.2.6 Data setelah diperbarui

* Hasil Output Uji Coba Skenario 6

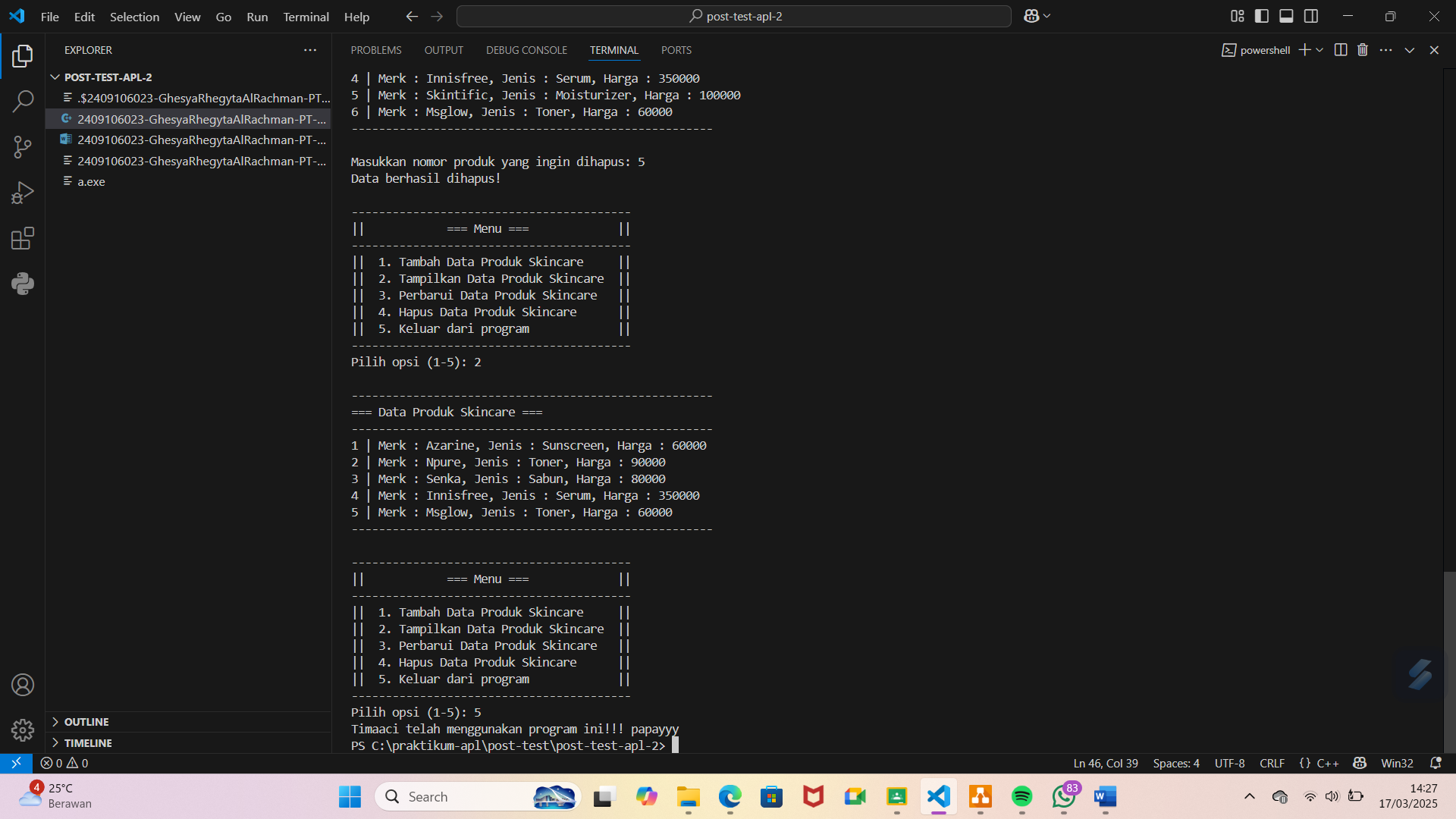


Gambar 4.2.7 Hapus data produk



Gambar 4.2.8 Data setelah dihapus

* Hasil Output Uji Coba Skenario 7

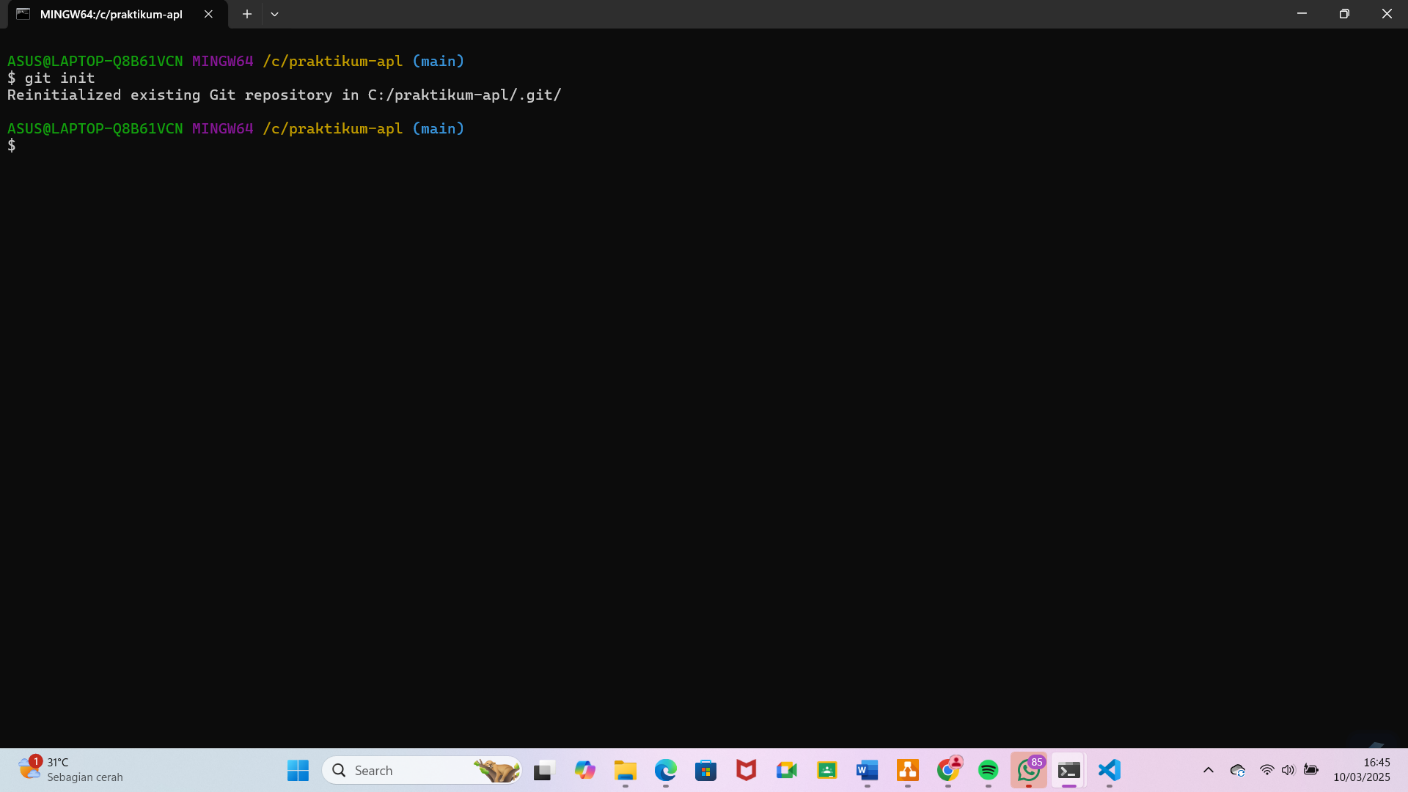


Gambar 4.2.9 Keluar dari program

# Langkah-Langkah GIT

* + - 1. Git Init

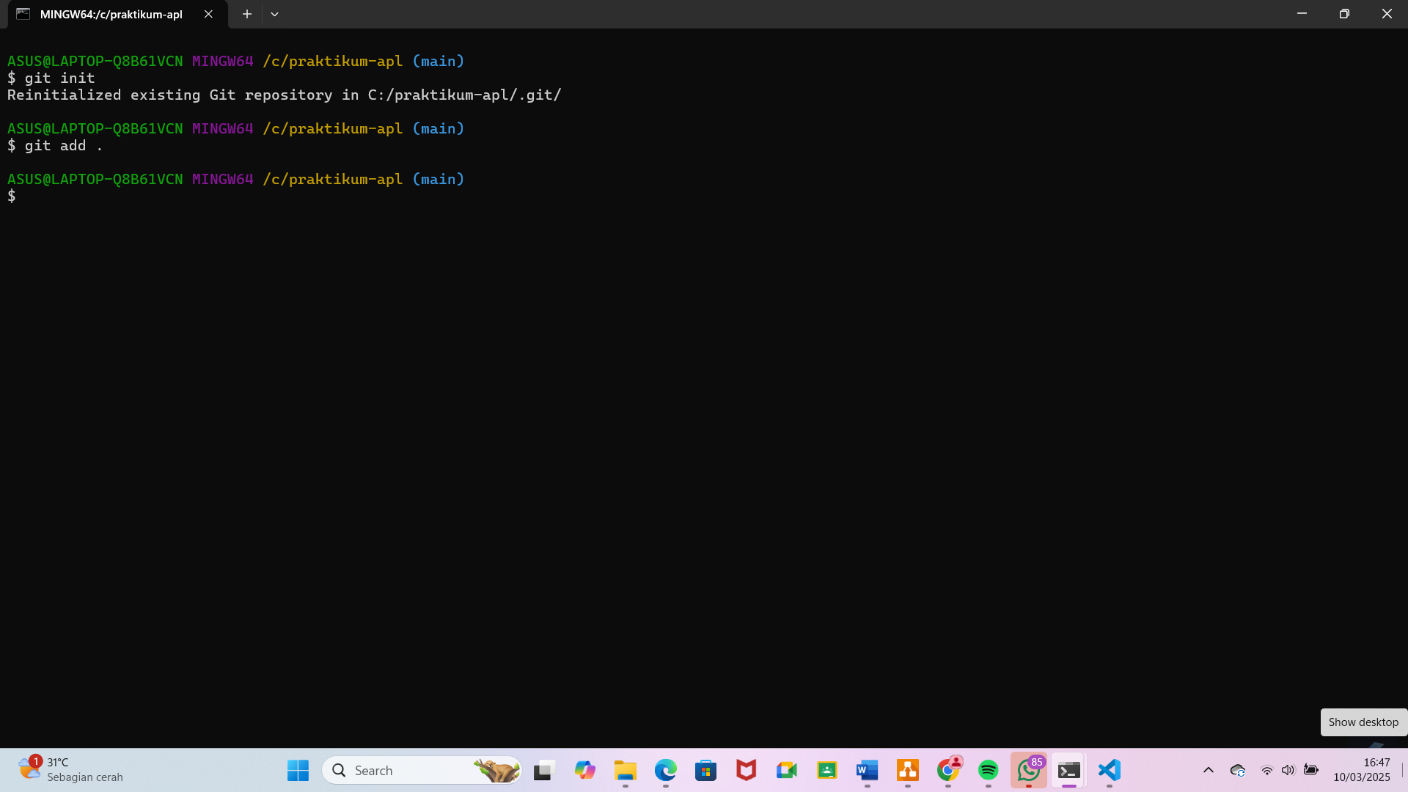
Menginisiasi repository Git di dalam folder proyek. Setelah perintah ini dijalankan, Git mulai melacak perubahan pada folder tersebut



Gambar 5.1 Git Init

* + - 1. Git Add .

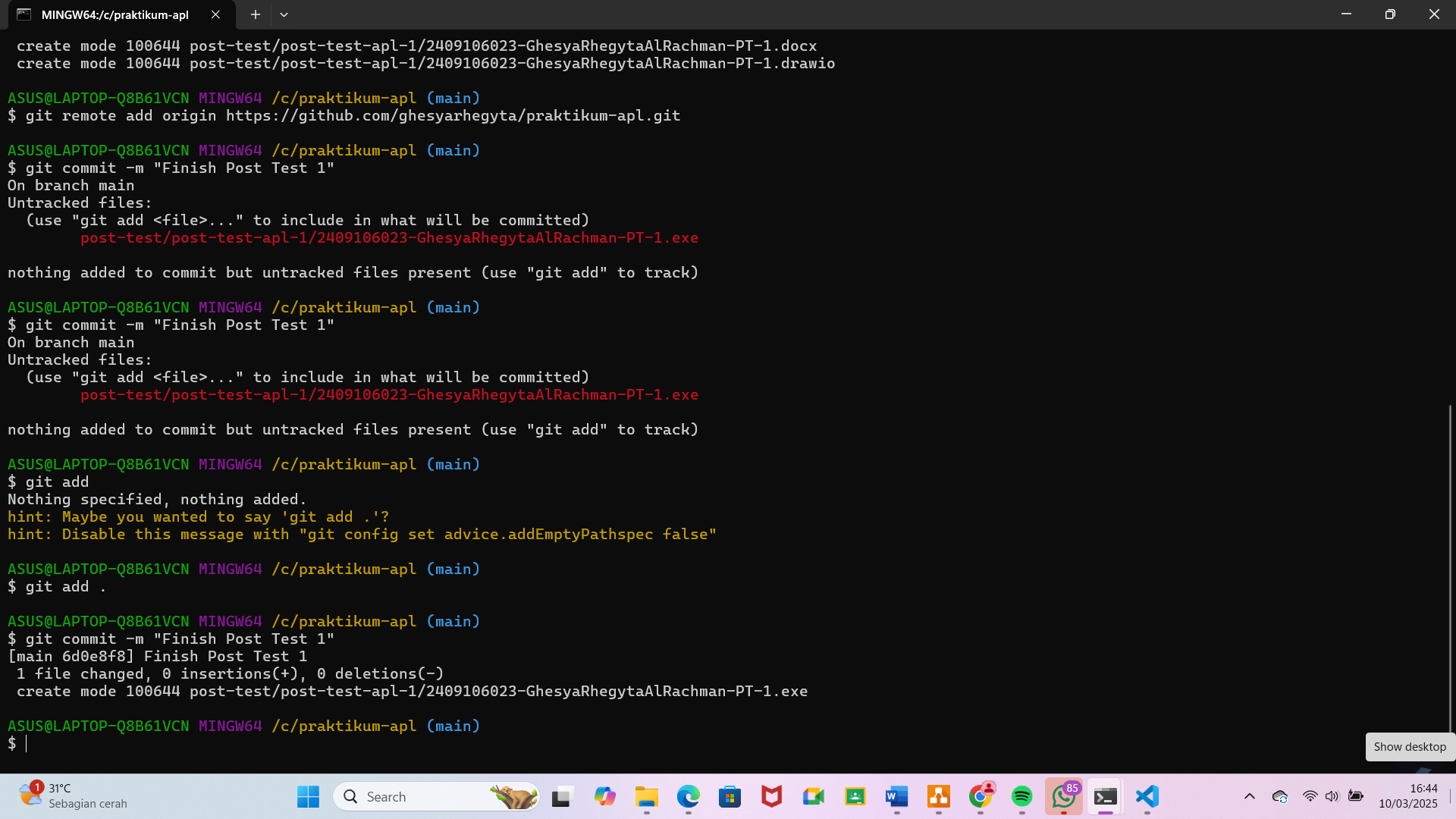
Menambahkan semua perubahan seperti file baru, file yang dimodifikasi, maupun file yang dihapus ke dalam staging area Git sebelum dikomit. Tanda titik (.) menunjukkan semua file dalam folder proyek akan ditambahkan ke staging area



Gambar 5.2 Git Add .

1. Git Commit

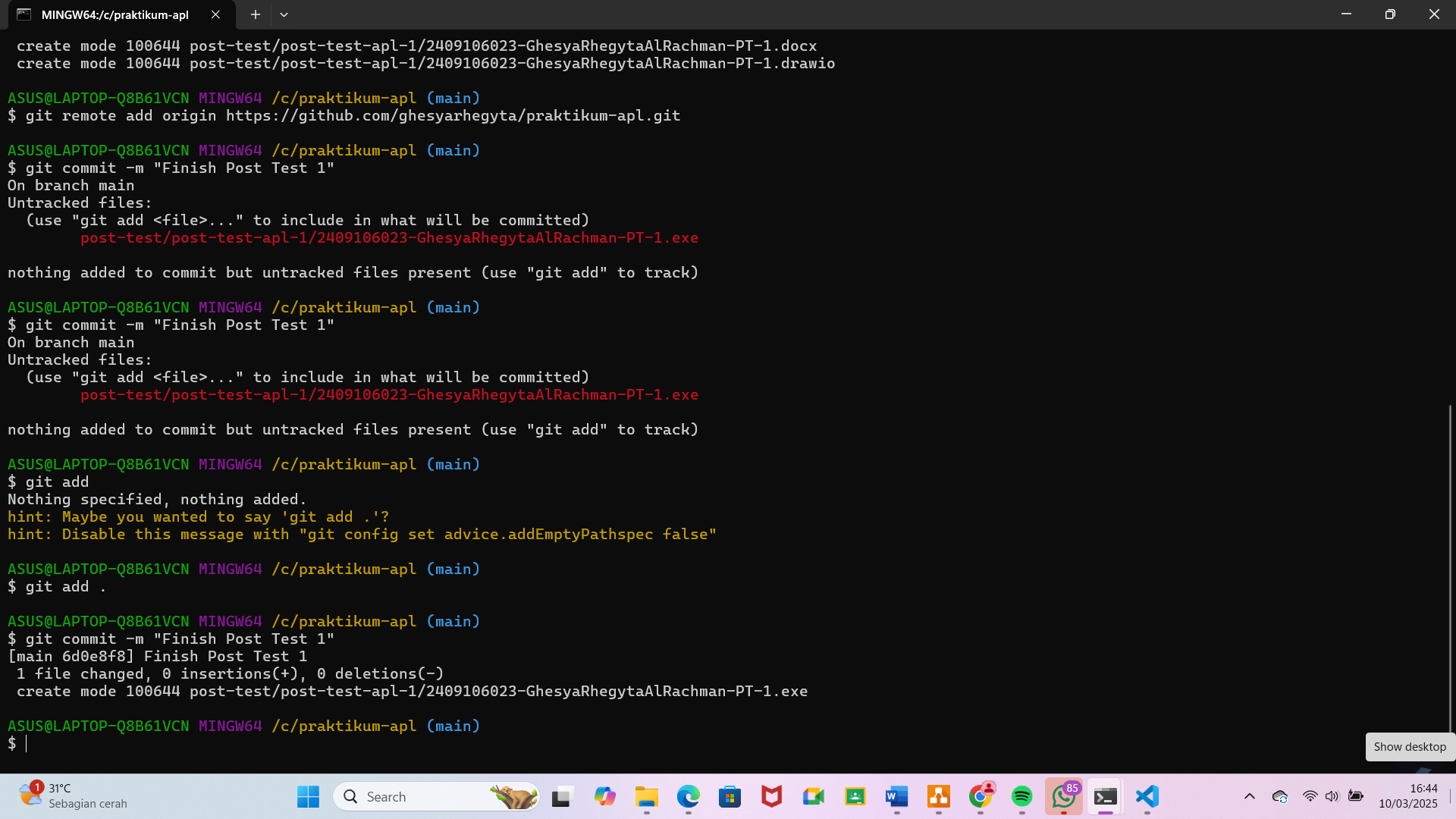
Menyimpan perubahan yang ada di staging area ke dalam commit history. Setiap commit mencatat perubahan apa yang dilakukan pada kode. (-m) digunakan untuk menyertakan pesan commit secara langsung



Gambar 5.3 Git Commit

1. Git Remote

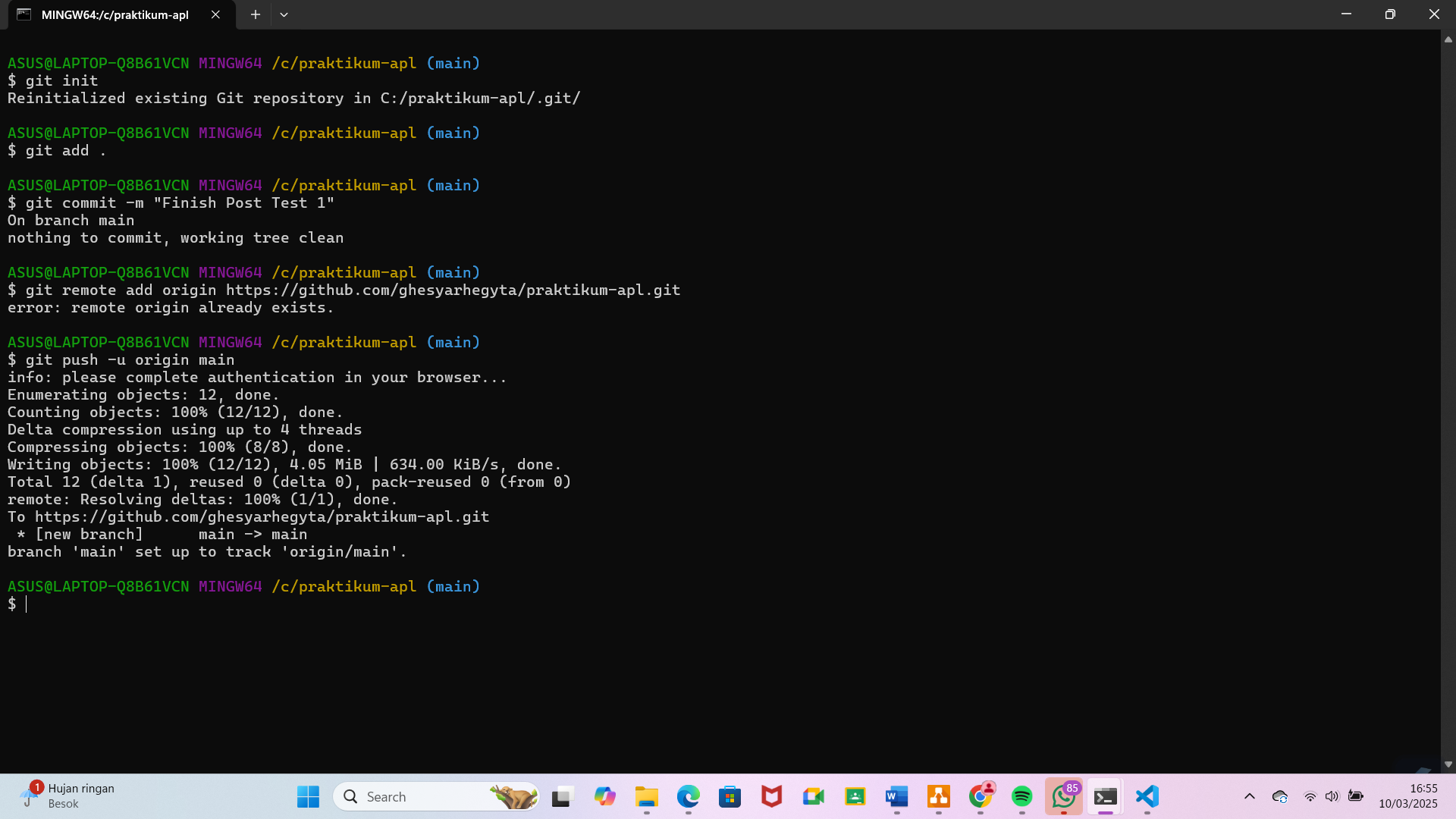
Digunakan untuk mengelola repository lokal dengan server github yang sudah ada. Kita dapat melihat daftar, menambahkan, menghapus, serta mengubah URL remote



Gambar 5.4 Git Remote

1. Git Push

Mengirim commit yang ada di lokal ke server seperti GitHub. Digunakan agar dapat diakses oleh orang lain atau untuk kolaborasi



Gambar 5.5 Git Push