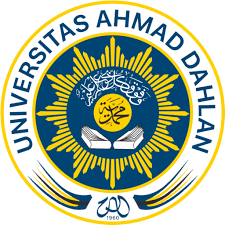
**Tugas Akhir**

**BMI (Body Mass Index)**

Disusun untuk memenuhi tugas akhir Dasar Sistem Komputer

Dosen Pengampu : Ali Tarmuji, S.T., M.Cs.

****

**Disusun oleh:**

**NAMA : Daffa Nur Hafidz**

**NIM : 2400018127**

**KELAS : C**

**Link Github**

<https://github.com/DaffaNurHafidz2400018127/Aplikasi-BMI>

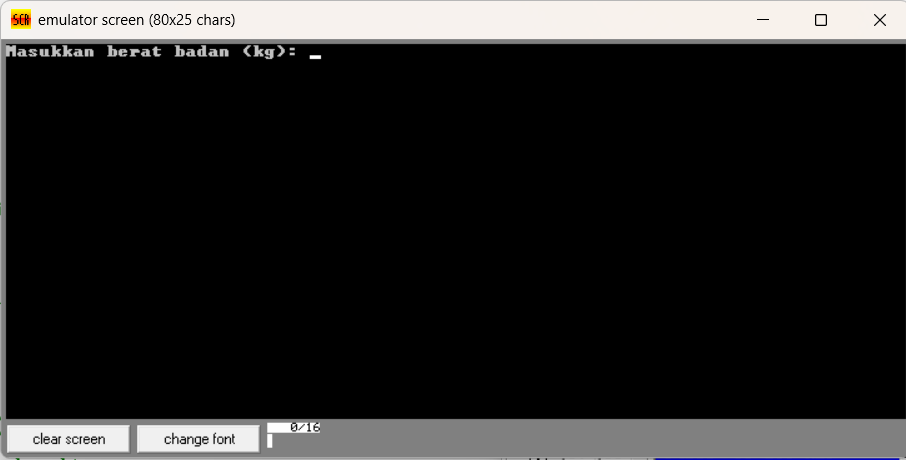
**PROGRAM STUDI S1 INFORMATIKA**

**FAKULTAS TEKNOLOGI INDUSTRI**

**UNIVERSITAS AHMAD DAHLAN**

**2024/2025**

1. **Tampilan Awal Aplikasi**

****

1. **RUANG LINGKUP**

* Masyarakat, Tenaga Kesehatan, Peneliti, Programmer dll

1. **DESKRIPSI DARI PROJEK**

Program ini adalah aplikasi untuk menghitung **Body Mass Index (BMI)** yang digunakan untuk menilai status berat badan seseorang. Program ini mempermudah pengguna untuk mengetahui kategori BMI mereka (Kurus, Normal, Gemuk) berdasarkan berat badan dan tinggi badan yang dimasukkan. Dengan menggunakan program ini, pengguna dapat lebih cepat dan mudah mengetahui status kesehatan tubuh mereka, serta mendapatkan informasi yang berguna untuk menjaga kesehatan. Program ini juga dirancang untuk memberikan hasil perhitungan secara langsung dan menampilkan kategori BMI secara jelas, sehingga memberikan pengalaman pengguna yang efisien dan informatif.

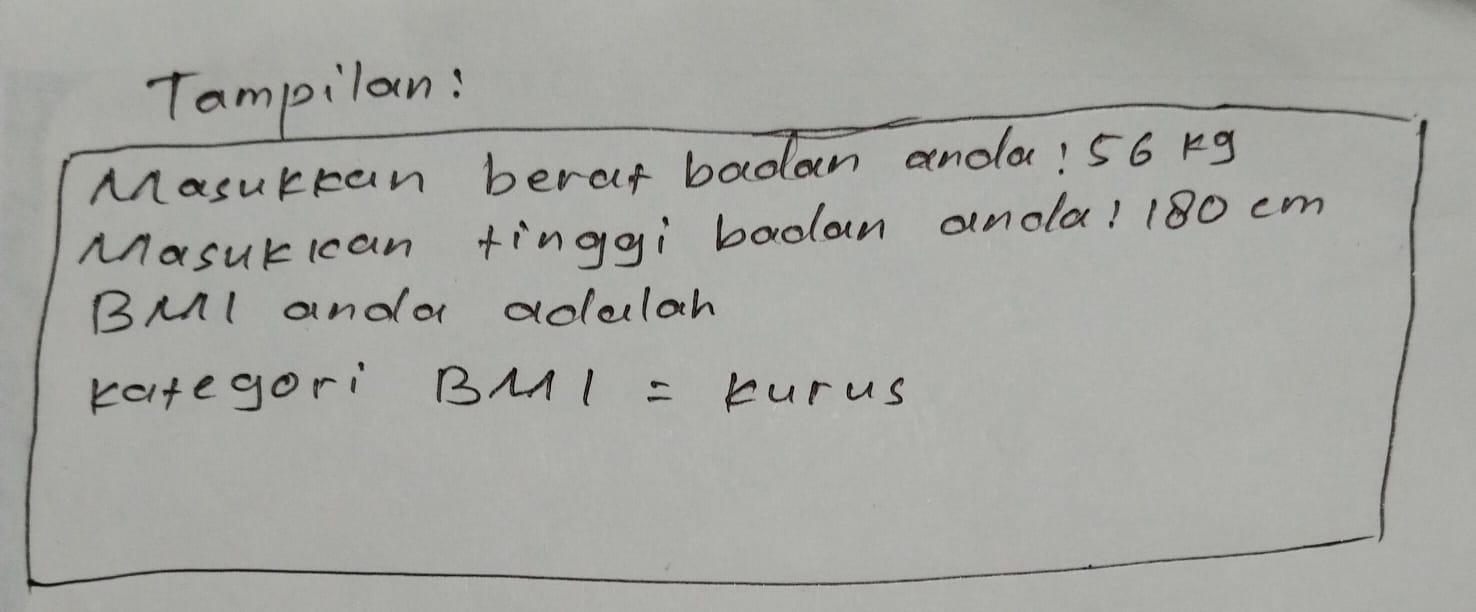
1. **DAFTAR SELURUH SPESIFIKASI APLIKASI SESUAI STUDI KASUS DAN SPESIFIKSI APLIKASI**

* **Penyimpanan Nama Pengguna  
  Program dapat menyimpan nama pengguna yang dimasukkan untuk digunakan pada tampilan hasil perhitungan BMI dan untuk memberi pengalaman personalisasi kepada pengguna.**
* **Pemasukan Berat Badan dan Tinggi Badan  
  Pengguna dapat memasukkan berat badan dalam kilogram dan tinggi badan dalam sentimeter. Program akan menggunakan data ini untuk menghitung BMI secara otomatis.**
* **Perhitungan BMI  
  Setelah data berat badan dan tinggi badan dimasukkan, program akan menghitung BMI menggunakan rumus standar dan menampilkan hasilnya kepada pengguna.**
* **Penampilan Kategori BMI  
  Program menampilkan kategori BMI berdasarkan hasil perhitungan, seperti Kurus, Normal, atau Gemuk, yang membantu pengguna memahami status berat badan mereka.**
* **Pesan Error  
  Program akan menampilkan pesan kesalahan jika ada input yang tidak valid, seperti nilai yang tidak sesuai format atau tidak dapat dihitung.**

**SPESIFIKASI APLIKASI**

* **Sistem Operasi: Windows**
* **Bahasa Pemrograman: Assembly (TASM)**

1. **RANCANGAN ANTAR MUKA**



1. **Penjelasan Code Program BMI**

**Contoh Kodingan :**

**.model small**

**.stack 100h**

**.data**

**prompt db 'Masukkan berat badan (kg): $'**

**prompt2 db 0Ah, 0Dh, 'Masukkan tinggi badan (cm): $'**

**hasil db 0Ah, 0Dh, 'BMI Anda adalah: $', 0Ah, 0Dh, '$'**

**kategori db 0Ah, 0Dh, 'Kategori BMI: $', 0Ah, 0Dh, '$'**

**newline db 0Ah, 0Dh, '$'**

**error\_msg db 0Ah, 0Dh, 'Input tidak valid!$'**

**; Variabel untuk menyimpan input dan hasil**

**berat dw 0 ; Menggunakan 2 byte untuk menyimpan nilai berat badan (integer)**

**tinggi dw 0 ; Menggunakan 2 byte untuk menyimpan tinggi badan (integer)**

**bmi dw 0 ; Variabel untuk menyimpan hasil BMI**

**.code**

**main:**

**; Inisialisasi segment data**

**mov ax, @data**

**mov ds, ax**

**; Menampilkan prompt untuk memasukkan berat badan**

**lea dx, prompt**

**mov ah, 09h**

**int 21h**

**; Input berat badan**

**call get\_input**

**mov [berat], ax ; Menyimpan berat badan ke dalam variabel berat**

**; Menampilkan prompt untuk memasukkan tinggi badan dalam cm**

**lea dx, prompt2**

**mov ah, 09h**

**int 21h**

**; Input tinggi badan**

**call get\_input**

**mov [tinggi], ax ; Menyimpan tinggi badan ke dalam variabel tinggi**

**; Menghitung tinggi badan dalam meter (tinggi / 100)**

**mov ax, [tinggi] ; Salin tinggi badan dalam cm ke register ax**

**mov bx, 100 ; Menyimpan angka 100 ke register bx**

**xor dx, dx ; Bersihkan dx sebelum pembagian**

**div bx ; Membagi ax dengan 100, hasilnya dalam ax (meter)**

**; Menghitung tinggi \* tinggi (dalam meter)**

**mov bx, ax ; Salin hasil tinggi dalam meter ke bx**

**mov ax, [berat] ; Salin berat badan ke dalam ax**

**mov cx, ax ; Simpan berat badan untuk pembagian nanti**

**mul bx ; ax = tinggi \* tinggi (meter^2)**

**; Pastikan hasil tinggi \* tinggi tidak 0**

**cmp ax, 0**

**je invalid\_input ; Jika ax = 0, tampilkan pesan error**

**; Sekarang ax berisi tinggi \* tinggi (m^2), bagi dengan berat badan**

**mov ax, cx ; Pindahkan berat badan ke ax**

**xor dx, dx ; Clear dx untuk hasil pembagian**

**div bx ; Membagi ax dengan hasil tinggi \* tinggi**

**; Simpan hasil BMI**

**mov [bmi], ax**

**; Menampilkan hasil BMI**

**lea dx, hasil**

**mov ah, 09h**

**int 21h**

**; Menampilkan kategori BMI**

**lea dx, kategori**

**mov ah, 09h**

**int 21h**

**; Tentukan kategori BMI berdasarkan hasil**

**mov ax, [bmi]**

**cmp ax, 1850 ; BMI < 18.5 -> Kurus**

**jb kurang**

**cmp ax, 2500 ; 18.5 <= BMI < 24.9 -> Normal**

**ja gemuk**

**jmp normal**

**kurang:**

**lea dx, kurang\_msg**

**mov ah, 09h**

**int 21h**

**jmp end\_program**

**normal:**

**lea dx, normal\_msg**

**mov ah, 09h**

**int 21h**

**jmp end\_program**

**gemuk:**

**lea dx, gemuk\_msg**

**mov ah, 09h**

**int 21h**

**jmp end\_program**

**invalid\_input:**

**; Menampilkan pesan error jika input tidak valid**

**lea dx, error\_msg**

**mov ah, 09h**

**int 21h**

**end\_program:**

**; Menampilkan newline setelah output**

**lea dx, newline**

**mov ah, 09h**

**int 21h**

**mov ah, 4Ch**

**int 21h**

**get\_input proc**

**xor ax, ax ; Clear ax**

**xor cx, cx ; Clear cx**

**input\_loop:**

**mov ah, 01h ; Input karakter**

**int 21h**

**cmp al, 0Dh ; Jika Enter ditekan**

**je done\_input**

**sub al, '0' ; Convert ASCII ke angka**

**mov dx, ax**

**shl ax, 1 ; ax = ax \* 2**

**shl ax, 1 ; ax = ax \* 4**

**add ax, dx ; ax = ax \* 10**

**add ax, dx ; ax = ax \* 10 + digit**

**jmp input\_loop**

**done\_input:**

**ret**

**get\_input endp**

**; Pesan kategori BMI**

**kurang\_msg db 'Kurus$', 0**

**normal\_msg db 'Normal$', 0**

**gemuk\_msg db 'Gemuk$', 0**

**end main**

Program di atas bertujuan untuk menghitung Body Mass Index (BMI) seseorang berdasarkan input berat badan (kg) dan tinggi badan (cm). Berikut adalah penjelasan rinci dari setiap bagian kode:

**1. Segmen Data (.data):**

* prompt, prompt2: Menyimpan teks yang digunakan untuk meminta input dari pengguna, yaitu berat badan dan tinggi badan.
* hasil, kategori: Menyimpan teks yang digunakan untuk menampilkan hasil perhitungan BMI dan kategori BMI (misalnya "Kurus", "Normal", atau "Gemuk").
* newline: Digunakan untuk menambahkan baris baru setelah output.
* error\_msg: Pesan yang ditampilkan jika input tidak valid.
* berat, tinggi, bmi: Variabel untuk menyimpan input berat badan, tinggi badan, dan hasil perhitungan BMI. Setiap variabel menggunakan tipe data dw yang memerlukan 2 byte untuk menyimpan nilai integer.

**2. Segmen Code (.code):**

* main: Label utama program yang mengendalikan alur eksekusi.
* mov ax, @data dan mov ds, ax: Menginisialisasi segmen data agar variabel dapat diakses.
* **Menampilkan Prompt**:
  + Program menampilkan pesan untuk meminta input berat badan menggunakan lea dx, prompt dan int 21h untuk menampilkan teks.
* **Input Berat Badan**:
  + Fungsi get\_input dipanggil untuk mengambil input angka dari pengguna (fungsi ini disediakan di bagian bawah program).
  + Hasil input disimpan dalam variabel berat dengan mov [berat], ax.
* **Menampilkan Prompt untuk Tinggi Badan**:
  + Setelah berat badan dimasukkan, program meminta pengguna untuk memasukkan tinggi badan dengan cara yang sama.
  + Hasilnya disimpan dalam variabel tinggi.
* **Menghitung BMI**:
  + BMI dihitung dengan rumus: BMI=Berat (kg)(Tinggi (m))2\text{BMI} = \frac{\text{Berat (kg)}}{(\text{Tinggi (m)})^2}BMI=(Tinggi (m))2Berat (kg)​
  + Tinggi badan yang diinputkan dalam cm dibagi dengan 100 untuk mengubahnya menjadi meter.
  + Program kemudian mengalikan tinggi badan dalam meter dengan dirinya sendiri untuk mendapatkan tinggi dalam meter kuadrat.
  + Hasil perhitungan tinggi² disimpan di bx, kemudian berat badan dibagi dengan hasil ini untuk menghitung BMI.
  + BMI disimpan dalam variabel bmi.

**3. Menentukan Kategori BMI:**

* Setelah BMI dihitung, program akan memeriksa kategori berdasarkan nilai BMI:
  + **BMI < 18.5**: Kategori "Kurus".
  + **18.5 <= BMI < 24.9**: Kategori "Normal".
  + **BMI >= 25**: Kategori "Gemuk".
* Hasil kategori BMI ditampilkan setelah perhitungan dengan menggunakan instruksi lea dx, kategori untuk menampilkan teks.

**4. Menangani Input yang Tidak Valid:**

* Jika hasil perhitungan tinggi² adalah 0, program akan menganggap input tidak valid dan menampilkan pesan error (invalid\_input).

**5. Subrutin get\_input:**

* Fungsi ini digunakan untuk membaca input angka yang dimasukkan pengguna.
* Setiap karakter yang dimasukkan akan diubah dari format ASCII menjadi angka melalui operasi aritmatika yang ada di dalam get\_input.
* Setelah input selesai, program melanjutkan ke langkah berikutnya.

**6. Penutupan Program:**

* Setelah BMI dihitung dan kategori ditentukan, program akan menampilkan hasilnya dan kemudian mengakhiri eksekusi dengan mov ah, 4Ch yang memanggil fungsi exit DOS.

**7. Pesan Kategori BMI:**

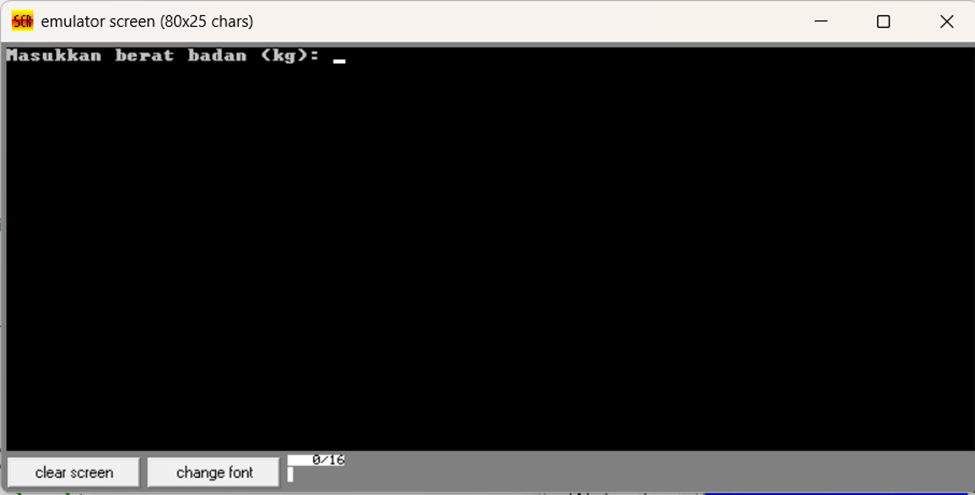
* Tiga kategori pesan untuk BMI:
  + kurang\_msg: Menampilkan "Kurus".
  + normal\_msg: Menampilkan "Normal".
  + gemuk\_msg: Menampilkan "Gemuk".

**Contoh Alur Program:**

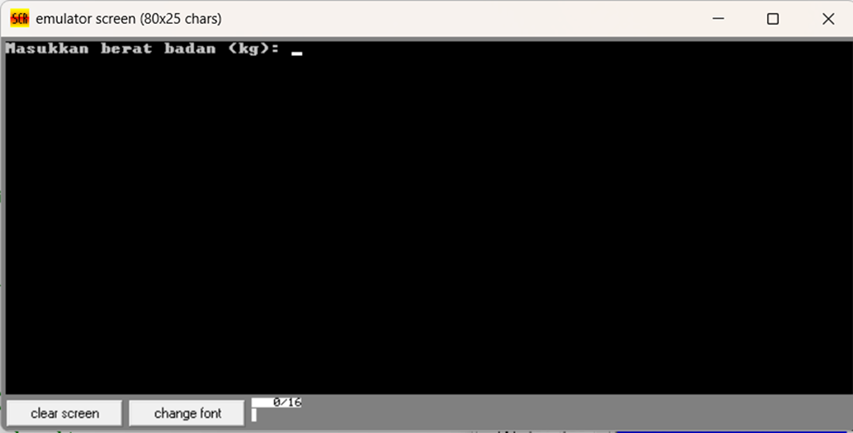
1. Program meminta input berat badan dan tinggi badan.
2. BMI dihitung berdasarkan rumus BMI.
3. Program menentukan kategori BMI berdasarkan hasil perhitungan.
4. Jika input tidak valid, pesan error akan ditampilkan.
5. Program menampilkan hasil BMI dan kategori yang sesuai (misalnya, "Kurus", "Normal", atau "Gemuk").

Dengan langkah-langkah ini, program dapat menghitung dan memberi tahu pengguna kategori BMI mereka berdasarkan data yang mereka input.

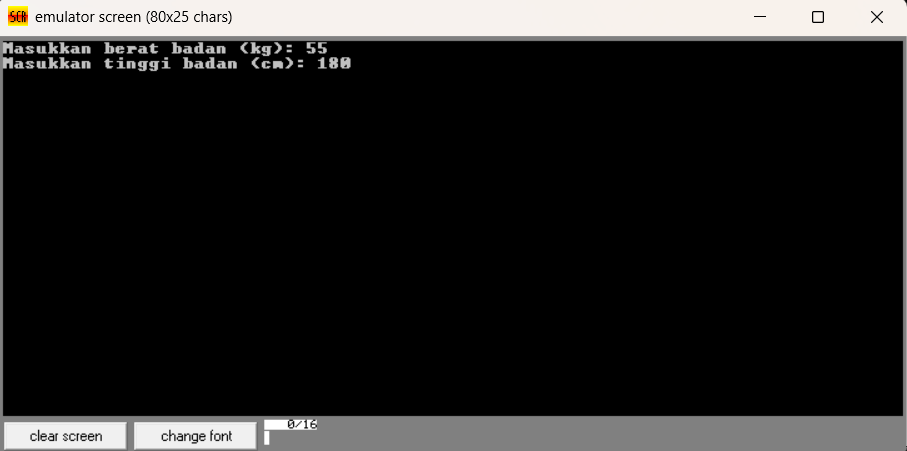
1. **DESKRIPSI TAMPILAN**
2. Program yang di jalankan.



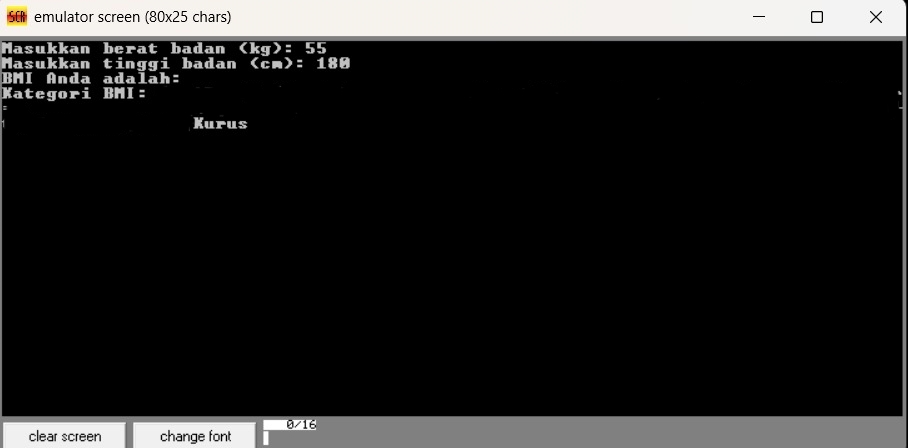
1. Pengguna diminta untuk memasukkan berat badan.



1. Pengguna diminta memasukkan tinggi badan.

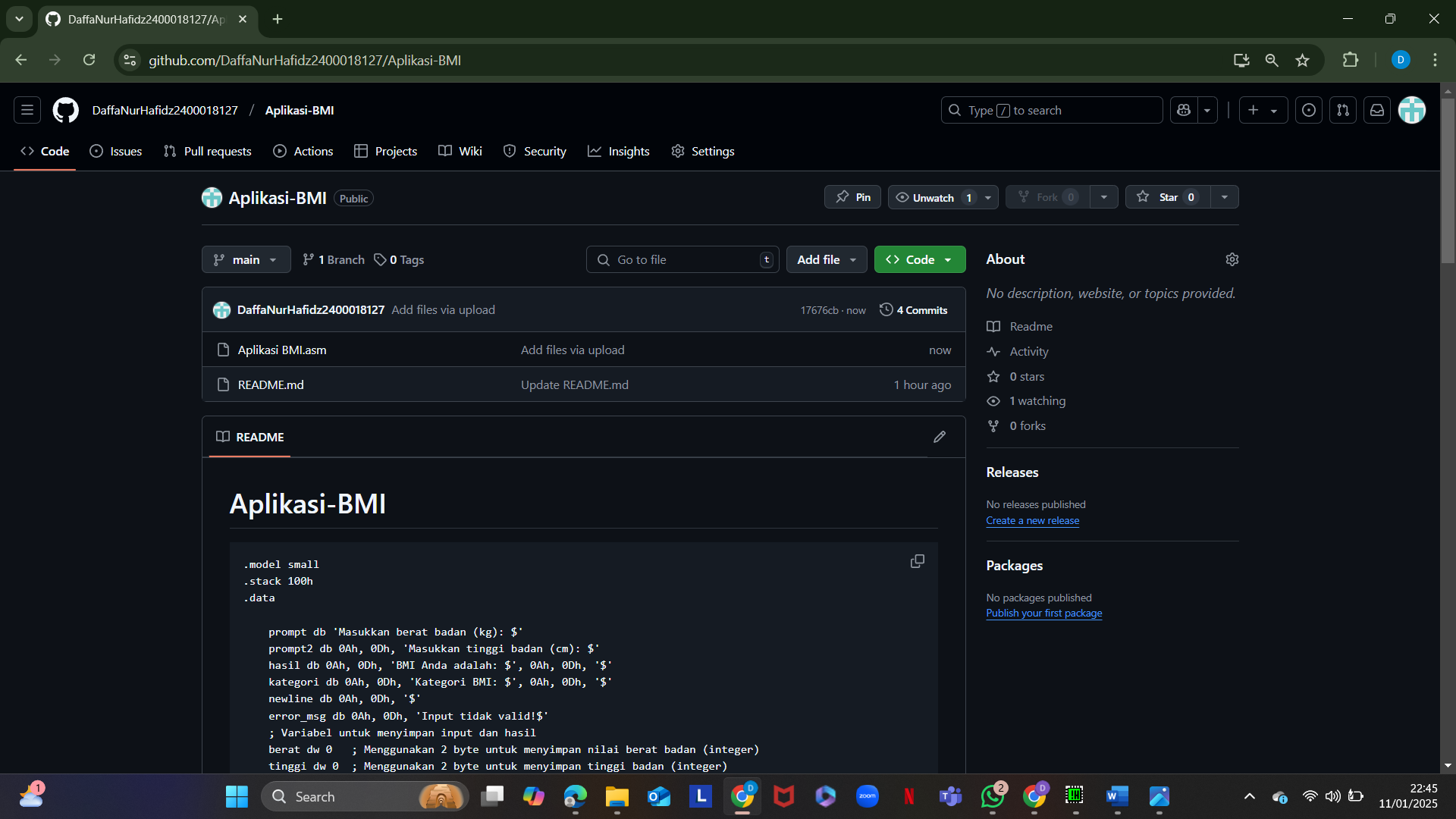


1. Pada tampilan tipe kamar, customer memasukkan kode nomor yang sudah ada di pilihan dengan memasukkan nomor tipe kamar yang diinginkan.

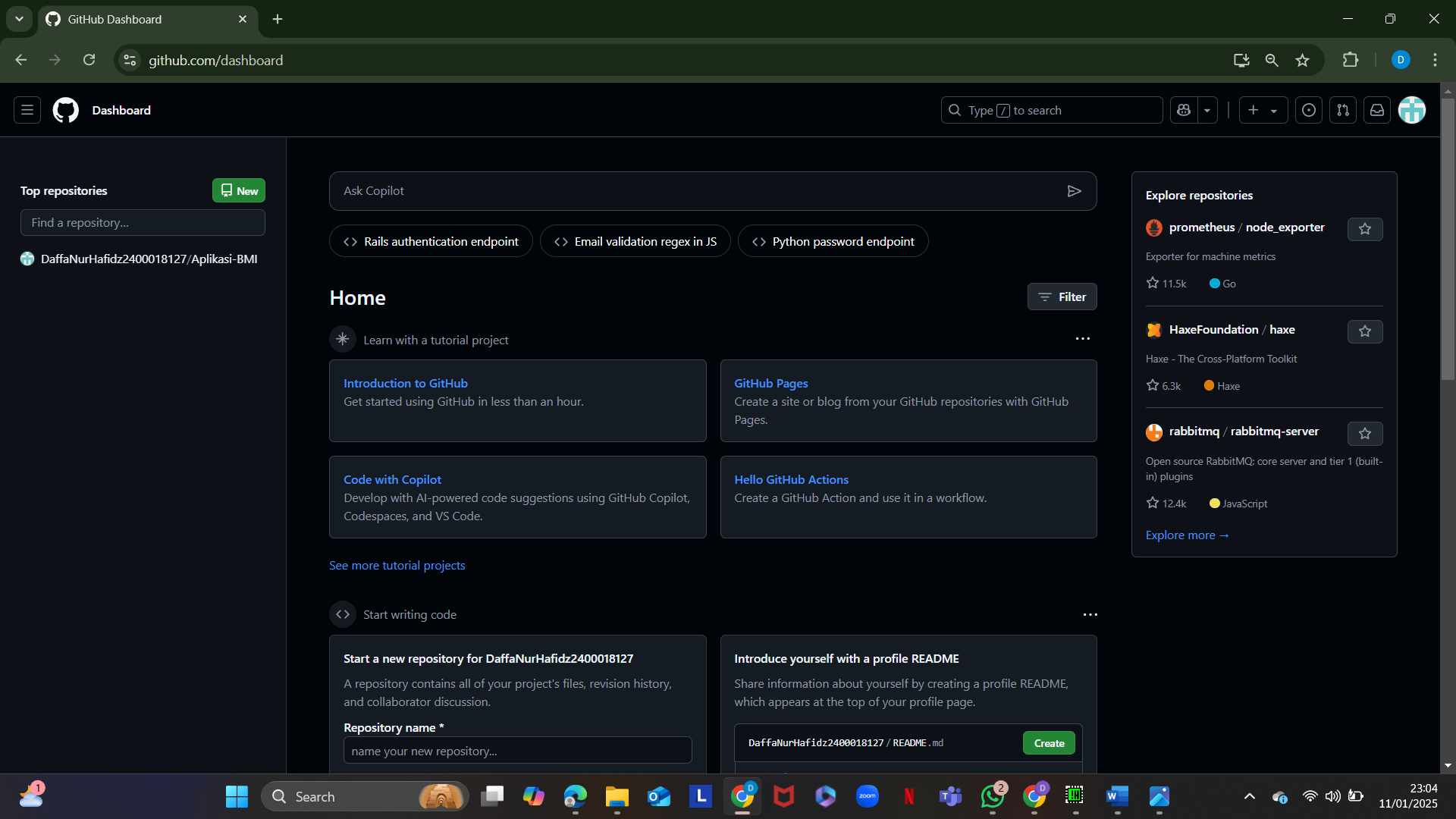


1. Program Selesai.
2. **DESKRIPSI SCREENSHOOT GITHUB**

* Tampilan Unggahan awal pada GitHub yang berisi file Aplikasi BMI dan file readme.



* Tampilan unggahan GitHub setelah diperbarui.



1. **ANALISIS PENGERJAAN TUGAS PROJEK**

Tugas Projek ini saya kerjakan dengan waktu sekitar 3 hari dalam pengerjaannya. Membuat Program ini juga melihat beberapa referensi dari internet, YouTube, dan juga bertanya kepada kakak tingkat yang sudah berpengalaman dalam membuat program sadar sistem computer ini. Untuk biaya yang di butuhkan untuk membuat program ini tidak ada. Ada beberapa hal yang menyulitkan dalam pengerjaan program ini yang dimana keterbatasan pengetahuan dalam mempelajari Bahasa assembly ini, kurangnya pengetahuan dalam pengerjaan ini membutuhkan waktu yang lumayan lama karena banyak nya juga tugas dan aktivitas kuliah, tetapi saya cukup puas dengan tugas projek ini yang dimana menambah pengetahuan lebih dalam lagi tentang Bahasa assembly.