

T3 ALJABAR LINEAR: *GUI*



Disusun Oleh :

Muhammad Arrafi Reva Razqana Arassy

NIM

241524017

**POLITEKNIK NEGERI BANDUNG
PROGRAM STUDI D4 TEKNIK INFORMATIKA
2025**

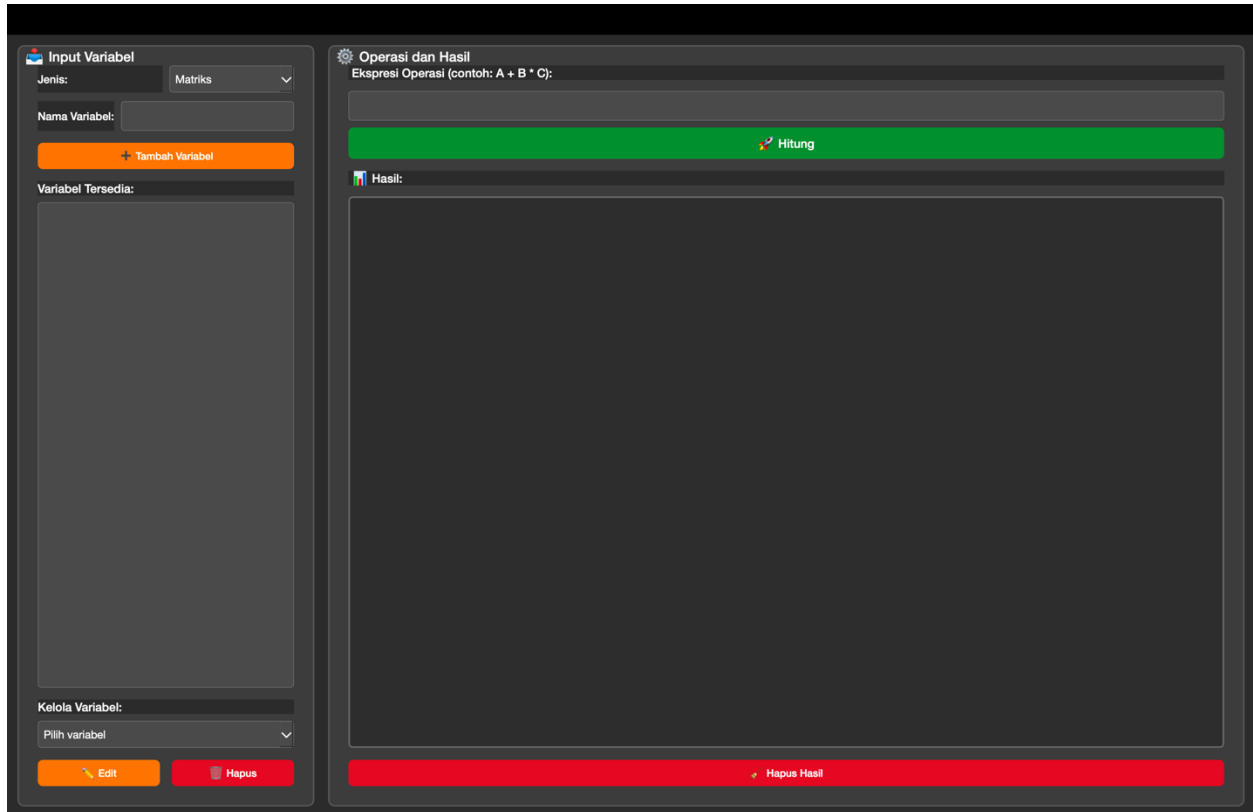
DAFTAR ISI

| | |
|-----------------------------------|---|
| DAFTAR ISI..... | 2 |
| I TAMPILAN..... | 3 |
| 1.1 Tampilan Awal..... | 3 |
| 1.2 Input Matriks..... | 4 |
| 1.3 Tampilan Variabel | 6 |
| 1.4 Output hasil Operasi | 7 |

I

TAMPILAN

1.1 Tampilan Awal



Deskripsi singkat:

Tampilan antarmuka utama aplikasi ini dibangun di dalam kelas MainWindow pada file UI.py. Struktur visualnya dibagi menjadi dua panel utama menggunakan QHBoxLayout untuk menampung dua buah QGroupBox. Tema gelap dan gaya visual yang konsisten pada setiap komponen, seperti tombol dan group box, diterapkan secara langsung di dalam fungsi init_ui menggunakan beberapa panggilan metode setStyleSheet dengan properti QSS (Qt Style Sheets) yang spesifik.

1.2 Input Matriks dan Skalar

Input Variabel

Jenis: Matriks

Nama Variabel:

+ Tambah Variabel

Variabel Tersedia:

Kelola Variabel:

Pilih variabel

Edit Hapus

Operasi dan Hasil

Eksprei Operasi (contoh: $A + B * C$):

Hitung

Hasil:

Input Matriks

Baris: Kolom: Buat Matriks

Tips: Gunakan Arrow Keys, Tab, atau Enter untuk navigasi antar sel

| | Kolom 1 | Kolom 2 | Kolom 3 |
|---------|---------|---------|---------|
| Baris 1 | 1 | 2 | 3 |
| Baris 2 | 4 | 5 | 6 |
| Baris 3 | 7 | 8 | 9 |

Konfirmasi Batal

Hapus Hasil

Input Matriks

Baris: Kolom: Buat Matriks

Tips: Gunakan Arrow Keys, Tab, atau Enter untuk navigasi antar sel

| | Kolom 1 | Kolom 2 | Kolom 3 |
|---------|---------|---------|---------|
| Baris 1 | 1 | 2 | 3 |
| Baris 2 | 4 | 5 | 6 |
| Baris 3 | 7 | 8 | 9 |

Konfirmasi Batal

 **Input Variabel**

Jenis:

Skalar

Nama Variabel:

i

Nilai Skalar:

10.5

+ Tambah Variabel

Variabel Tersedia:

C: Skalar (0.0)
Nilai: 5.0

D: Matriks (5, 5)
Elemen:
[1, 2, 3, 4, 5]
[2, 3, 2, 2, 2]
[3, 3, 3, 3, 3]
[4, 4, 4, 4, 4]
[5, 5, 5, 5, 5]

E: Matriks (2, 2)
Elemen:
[0, 0]
[0, 0]

F: Matriks (2, 2)
Elemen:
[1, 1]
[1, 1]

G: Matriks (3, 2)
Elemen:
[1, 2]
[3, 4]
[1, 1]

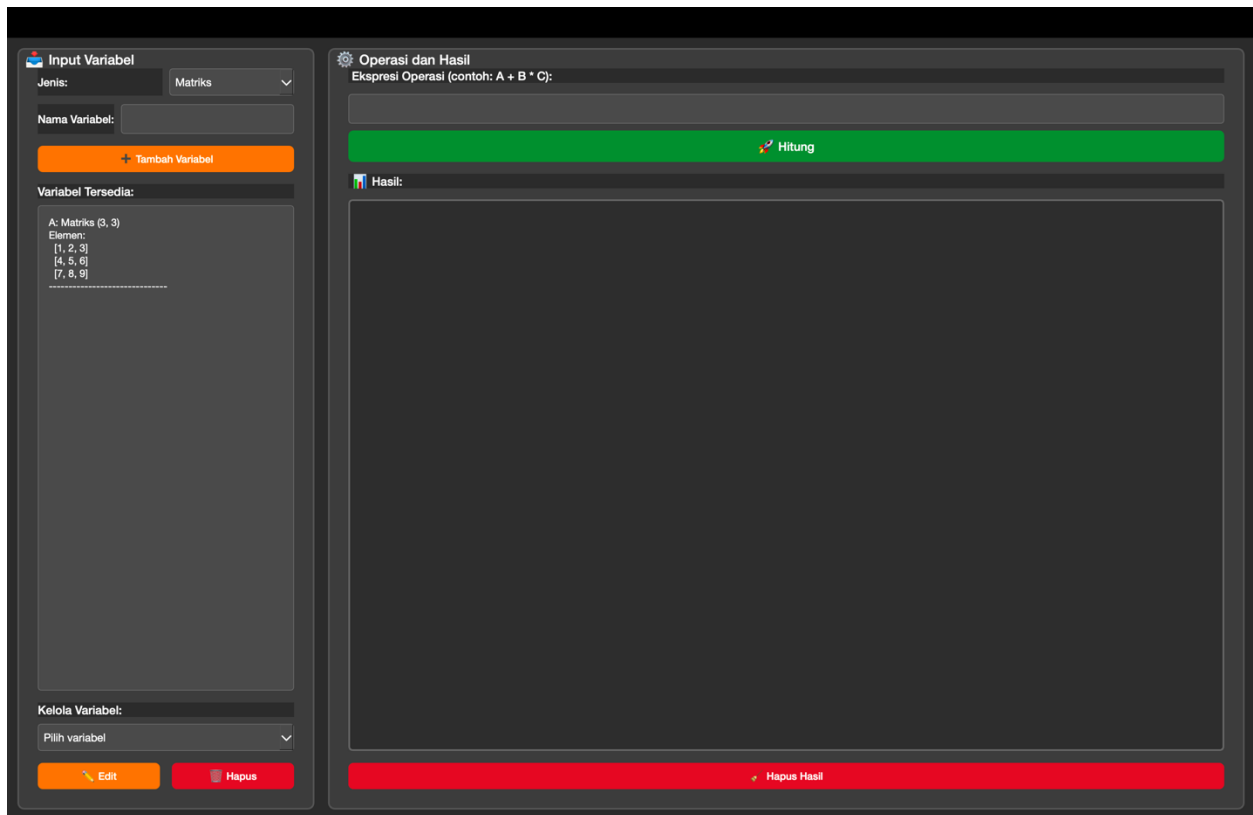
H: Skalar (0.5)
Nilai: 0.5

I: Skalar (10.5)
Nilai: 10.5

Deskripsi singkat:

Untuk memastikan pengalaman input data yang terfokus, aplikasi menggunakan jendela dialog modal yang dibuat dari kelas `MatrixInputDialog` di `UI.py`. Jendela ini, yang mewarisi `QDialog`, berisi widget `QSpinBox` untuk menentukan dimensi matriks dan sebuah `QTableWidget` yang dibuat secara dinamis oleh fungsi `create_matrix_table`. Saat tombol 'Konfirmasi' ditekan, metode `confirm_matrix` akan membaca setiap sel, memvalidasi inputnya, dan mengubahnya menjadi array NumPy sebelum mengirimkannya kembali ke jendela utama.

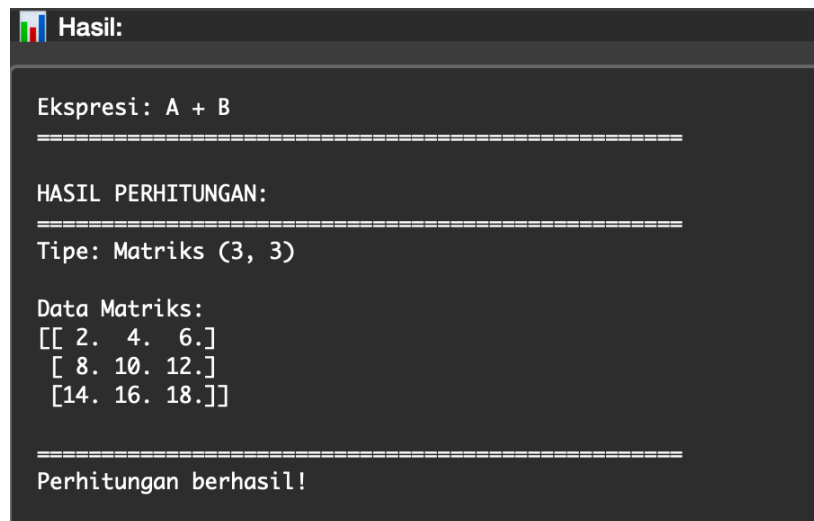
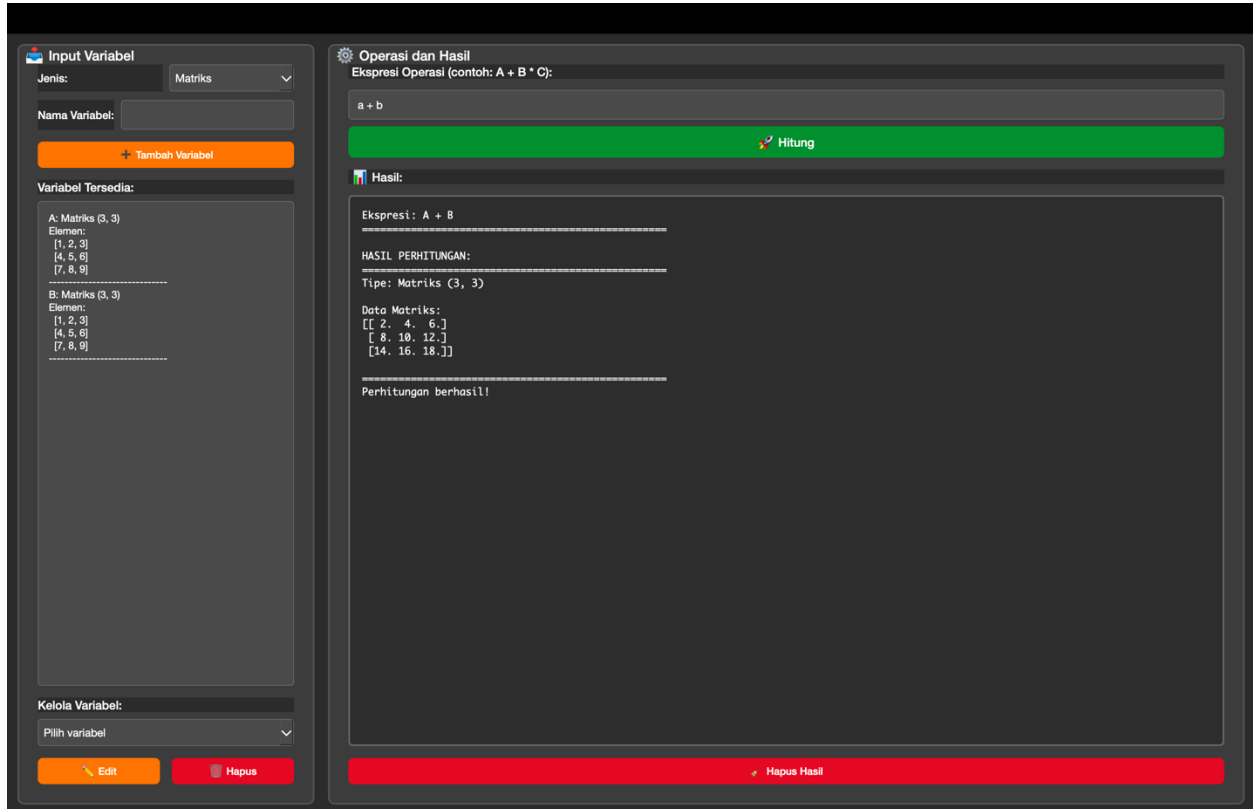
1.3 Tampilan Variabel





Deskripsi singkat:

Setelah pengguna menambahkan variabel, panel kiri akan diperbarui. Proses ini dikelola oleh fungsi `update_variables_list` di dalam kelas `MainWindow`. Fungsi ini mengambil data dari sebuah dictionary Python bernama `self.variables`, lalu mengiterasinya untuk membuat sebuah string yang diformat dengan baik. String ini kemudian ditampilkan dalam widget `QTextEdit` (`self.vars_list`).

1.4 Output hasil Operasi




 Input Variabel

Jenis: Skalar 

Nama Variabel:

Nilai Skalar:

 Tambah Variabel

Variabel Tersedia:


A: Matriks (3, 3)
Elemen:
[1, 2, 3]
[4, 5, 6]
[7, 8, 9]


B: Matriks (3, 3)
Elemen:
[1, 2, 3]
[4, 5, 6]
[7, 8, 9]


C: Skalar (5.0)
Nilai: 5.0


Kelola Variabel:

Pilih variabel




 Edit


 Hapus

 Operasi dan Hasil

Ekspresi Operasi (contoh: A + B * C):

(a + b) * c

 Hitung

 Hasil:


Ekspresi: (A + B) * C


HASIL PERHITUNGAN:

Tipe: Matriks (3, 3)

Data Matriks:
[[10. 20. 30.]
[40. 50. 60.]
[70. 80. 90.]]

Perhitungan berhasil!

 Hapus Hasil

 Hasil:

Ekspresi: (A + B) * C

=====

HASIL PERHITUNGAN:

=====

Tipe: Matriks (3, 3)

Data Matriks:
[[10. 20. 30.]
[40. 50. 60.]
[70. 80. 90.]]

=====

Perhitungan berhasil!

Input Variabel

Jenis:

Skalar

Nama Variabel:

1

Nilai Skalar:

0

+ Tambah Variabel

Variabel Tersedia:

A: Matriks (3, 3)
Elemen:
[1, 2, 3]
[4, 5, 6]
[7, 8, 9]

B: Matriks (3, 3)
Elemen:
[1, 2, 3]
[4, 5, 6]
[7, 8, 9]

C: Skalar (5,0)
Nilai: 5.0

D: Matriks (5, 5)
Elemen:
[1, 2, 3, 4, 5]
[2, 3, 2, 2, 2]
[3, 3, 3, 3, 3]
[4, 4, 4, 4, 4]
[5, 5, 5, 5, 5]

E: Matriks (2, 2)
Elemen:
[0, 0]
[0, 0]

F: Matriks (2, 2)
Elemen:
[1, 1]
[1, 1]

C: Matriks (5, 5)

Kelola Variabel:

H

Edit

Hapus

Operasi dan Hasil

Ekspresi Operasi (contoh: A + B * C):

c * d

Hitung

Hasil:

Ekspresi: C * D

HASIL PERHITUNGAN:

Tipe: Matriks (5, 5)

Data Matriks:
[[5. 10. 15. 20. 25.]
[10. 15. 10. 10. 10.]
[15. 15. 15. 15. 15.]
[20. 20. 20. 20. 20.]
[25. 25. 25. 25. 25.]]

Perhitungan berhasil!

Hapus Hasil

Hasil:

Ekspresi: C * D

HASIL PERHITUNGAN:

Tipe: Matriks (5, 5)

Data Matriks:
[[5. 10. 15. 20. 25.]
[10. 15. 10. 10. 10.]
[15. 15. 15. 15. 15.]
[20. 20. 20. 20. 20.]
[25. 25. 25. 25. 25.]]

Perhitungan berhasil!

Deskripsi singkat:

Proses perhitungan adalah contoh utama dari pemisahan tugas dalam aplikasi ini. Ketika tombol 🚀 'Hitung' (self.calc_btn) ditekan, ia memicu metode `calculate_expression` di `UI.py`. Namun, metode ini tidak melakukan perhitungan itu sendiri. Sebaliknya, ia mendelegasikan tugas tersebut dengan memanggil fungsi evaluasi yang diimpor dari file backend `T2_2A_017.py`. Setelah evaluasi mengembalikan hasilnya, `UI.py` Kembali bertugas untuk memformat dan menampilkannya di widget `QTextEdit` (self.result_display).