

## **LAPORAN PRAKTIKUM 2**

### **ALJABAR LINIER**



Nama : Alya Gustiani Nur 'Afifah

NIM : 231511035

Kelas/Prodi : 2B/D3 Teknik Informatika

**Politeknik Negeri Bandung**

**2024**

1. Link Repository GitHub :

[https://github.com/alyagustiani/AljabarLinear/tree/master/Praktikum5\\_ALIN\\_035\\_Alya%20Gustiani%20Nur%20Afifah](https://github.com/alyagustiani/AljabarLinear/tree/master/Praktikum5_ALIN_035_Alya%20Gustiani%20Nur%20Afifah)

2. Penjelasan tentang kalkulator ini

Kalkulator ini merupakan implementasi metode **eliminasi Gauss-Jordan**, yang digunakan untuk menyelesaikan sistem persamaan linear. Metode ini melibatkan manipulasi matriks augmented (gabungan matriks koefisien dan matriks hasil) melalui serangkaian operasi baris elementer sehingga matriks tersebut dikonversi menjadi bentuk **identitas**, dengan solusi yang dapat diperoleh langsung. Kalkulator ini memfasilitasi proses tersebut, mengambil input berupa matriks augmented, lalu melakukan eliminasi untuk menghasilkan solusi, baik berupa variabel terikat maupun variabel bebas (solusi parametrik).

3. Penjelasan masing-masing fungsi

- a. **fpb(int a, int b)**

- **Parameter:**

int a: Bilangan pertama.

int b: Bilangan kedua.

- **Proses:** Fungsi ini menggunakan metode **rekursif** untuk mencari **Faktor Persekutuan Terbesar (FPB)** dari dua bilangan dengan prinsip algoritma Euclid. Jika  $b == 0$ , maka a adalah FPB-nya. Jika tidak, fungsi akan dipanggil ulang dengan parameter b dan  $a \% b$ .

- **Nilai Kembalian:** Mengembalikan FPB dari dua bilangan (a dan b).

- b. **desimalKePecahan(double desimal, int &pembilang, int &penyebut)**

- **Parameter:**

double desimal: Angka desimal yang akan diubah menjadi pecahan.

int &pembilang: Referensi untuk menyimpan hasil pembilang dari pecahan.

int &penyebut: Referensi untuk menyimpan hasil penyebut dari pecahan.

- **Proses:** Fungsi ini menerima bilangan desimal dan mengalikan desimal tersebut dengan 10 hingga mencapai presisi yang diinginkan. Kemudian, bilangan tersebut diubah menjadi bentuk pecahan dengan mencari FPB antara pembilang dan penyebut, sehingga hasilnya disederhanakan.

- **Nilai Kembalian:** Fungsi ini tidak mengembalikan nilai secara langsung, tetapi mengubah nilai pembilang dan penyebut yang diberikan sebagai referensi.

**c. cetakPecahanSolusi(double desimal)**

- **Parameter:**  
double desimal: Nilai desimal yang akan ditampilkan dalam bentuk pecahan.
- **Proses:** Fungsi ini mengonversi nilai desimal menjadi pecahan dengan menggunakan fungsi desimalKePecahan. Jika hasilnya adalah bilangan bulat (penyebutnya 1), fungsi akan mencetak bilangan bulat. Jika tidak, fungsi mencetak pecahan dalam bentuk pembilang/penyebut.
- **Nilai Kembalian:** Fungsi ini tidak mengembalikan nilai. Outputnya berupa hasil yang ditampilkan pada layar.

**d. tampilkanSolusiParametrik(double matriks[MAKS][MAKS], int baris, int kolom)**

- **Parameter:**  
double matriks[MAKS][MAKS]: Matriks augmented yang berisi koefisien dan konstanta dari sistem persamaan linear.  
int baris: Jumlah baris dalam matriks.  
int kolom: Jumlah kolom dalam matriks.
- **Proses:** Fungsi ini mendeteksi variabel bebas (yang tidak memiliki solusi unik) dengan memeriksa baris yang seluruh elemennya nol. Jika variabel bebas ditemukan, hasilnya ditampilkan sebagai "free". Untuk variabel terikat, fungsi ini mencetak solusi parametrik, di mana variabel terikat ditampilkan sebagai fungsi dari variabel bebas. Solusi ditampilkan dalam dua format: desimal dan pecahan.
- **Nilai Kembalian:** Fungsi ini tidak mengembalikan nilai. Output ditampilkan berupa solusi parametrik dari sistem persamaan linear.

**e. eliminasiGaussJordan(double matriks[MAKS][MAKS], int baris, int kolom)**

- **Parameter:**  
double matriks[MAKS][MAKS]: Matriks augmented yang akan diproses.  
int baris: Jumlah baris dalam matriks.

int kolom: Jumlah kolom dalam matriks.

- **Proses:** Fungsi ini melakukan **eliminasi Gauss-Jordan** dengan serangkaian operasi baris elementer (menukar, mengalikan, dan menambahkan baris) untuk mengubah matriks augmented menjadi matriks identitas. Setelah setiap langkah, fungsi menampilkan kondisi matriks. Setelah eliminasi selesai, fungsi ini akan memanggil `tampilkanSolusiParametrik` untuk menampilkan hasil akhir dalam bentuk parametrik.
- **Nilai Kembalian:** Fungsi ini tidak mengembalikan nilai, tetapi mengubah matriks input secara langsung dan menampilkan hasil eliminasi.

**f. main() (dalam file main.cpp)**

- **Parameter:** Fungsi main tidak menerima parameter.
- **Proses:** Fungsi main meminta pengguna memasukkan ukuran matriks augmented, lalu menerima input dari elemen-elemen matriks tersebut. Setelah itu, `eliminasiGaussJordan` dipanggil untuk memproses matriks dan menampilkan hasilnya. Setelah satu kali perhitungan, pengguna diberi pilihan untuk mengulangi atau keluar dari program.
- **Nilai Kembalian:** Fungsi main mengembalikan nilai integer (0) sebagai tanda bahwa program telah berjalan dengan sukses.

**4. Kendala dan Solusi**

- a. **Penanganan Variabel Bebas:** Variabel bebas muncul ketika tidak semua variabel dalam sistem persamaan memiliki solusi unik, yang berarti beberapa variabel tidak terikat oleh persamaan tertentu. Dalam kasus seperti ini, saya harus memastikan kalkulator dapat mendeteksi variabel bebas dan menampilkan solusi parametrik untuk variabel-variabel terikat. Dengan menerapkan logika pendeteksian variabel bebas, saya bisa menampilkan solusi parametrik secara otomatis untuk variabel yang terikat dengan variabel bebas.
- b. **Pengubahan Desimal ke Pecahan:** Tantangan lain yang saya hadapi adalah ketika solusi dari sistem persamaan linear tidak selalu berupa bilangan bulat. Sering kali hasil yang diperoleh berupa angka desimal, dan saya merasa penting untuk menampilkan solusi dalam bentuk pecahan agar lebih mudah dipahami. Untuk

mengatasi hal ini, saya menggunakan fungsi ``desimalKePecahan`` dan ``cetakPecahanSolusi`` yang secara otomatis mengonversi desimal menjadi pecahan yang disederhanakan, sehingga hasil yang ditampilkan lebih bersifat matematis dan rapi.

- c. Eliminasi pada Matriks dengan Elemen Nol di Diagonal Utama: Selama implementasi, saya juga menghadapi kendala ketika menemukan elemen nol di diagonal utama matriks, yang menghalangi proses eliminasi. Jika elemen di diagonal utama bernilai nol, eliminasi tidak dapat dilanjutkan, karena sistem persamaan menjadi tidak terselesaikan menggunakan metode ini. Untuk itu, saya menambahkan pengecekan pada diagonal utama sebelum melakukan eliminasi. Jika ditemukan elemen nol, program akan menampilkan pesan kesalahan sehingga masalah ini dapat diantisipasi dan tidak menyebabkan kesalahan lebih lanjut dalam perhitungan.

## 5. Lesson learnt

Selama mengerjakan proyek ini, saya belajar banyak tentang pentingnya memahami struktur matriks dalam menyelesaikan sistem persamaan linear. Operasi baris elementer, seperti menukar baris, mengalikan baris dengan skalar, dan menjumlahkan baris, ternyata sangat krusial dalam memecahkan persamaan dengan metode eliminasi Gauss-Jordan. Pengalaman ini membantu saya mempraktikkan bagaimana teori matematis diterapkan langsung dalam perangkat lunak.

Selain itu, saya juga belajar bagaimana menangani solusi parametrik. Ketika sistem persamaan tidak memiliki solusi unik, variabel bebas sering kali muncul. Saya belajar bagaimana mengidentifikasi variabel-variabel ini dan membangun solusi parametrik yang menghubungkan variabel terikat dengan variabel bebas, sehingga saya bisa menghasilkan solusi yang tepat untuk sistem persamaan dengan solusi tak hingga.

Konversi dari desimal ke pecahan juga menjadi pelajaran penting bagi saya. Sering kali hasil perhitungan berupa desimal yang tidak bulat, dan saya menemukan bahwa dengan memanfaatkan FPB (Faktor Persekutuan Terbesar), pecahan bisa disederhanakan sehingga hasil lebih akurat dan mudah dipahami. Ini memberi saya pemahaman tentang pentingnya menjaga presisi dalam perhitungan numerik.

Selain itu, saya juga merasakan manfaat dari membangun program yang modular. Dengan memecah kode menjadi beberapa file terpisah untuk header, implementasi, dan fungsi utama, saya bisa lebih mudah dalam mengembangkan dan memperbaiki program. Struktur ini tidak hanya membuat kode lebih mudah di-debug, tetapi juga memungkinkan penambahan fitur dengan lebih cepat dan efisien di masa depan. Proyek ini secara keseluruhan membantu saya memahami pentingnya modularitas dan manajemen data numerik dalam pengembangan perangkat lunak.