**Prototipe Aplikasi Perhitungan Matriks Berbasis Java**

Muhammad Rizqi Nur1, Dwi Rolliawati2

Aplikasi Studi Sistem Informasi Fakultas Sains dan Teknologi

Universitas Islam Negeri Sunan Ampel Surabaya

Jl. Ahmad Yani No. 117, Jemur Wonosari, Wonocolo, Jemur Wonosari, Wonocolo

1[rizqinur2010@gmail.com](mailto:rizqinur2010@gmail.com), 2[dwi\_roll@uinsby.ac.id](mailto:dwi_roll@uinsby.ac.id)

# Abstrak

Matriks adalah salah satu materi matematika yang dianggap sulit. Materi matriks dianggap sulit utamanya karena banyaknya perhitungan yang harus dilakukan. Karena alasan itu pula mencari kesalahan perhitungan juga sulit, sehingga membuat proses pembelajaran yang sebagian besarnya adalah pengerjaan latihan soal semakin sulit. Untuk mengatasi itu, dibuatlah suatu aplikasi perhitungan matriks yang dapat menampilkan langkah-langkah perhitungannya. Aplikasi ini memiliki lima fitur utama, yaitu pembantu operasi baris elementer, penghitung operasi matriks, penghitung bentuk-bentuk matriks, penghitung determinan matriks, dan penyelesaian persamaan linear dalam bentuk matriks dengan aturan Cramer. Karena keterbatasan waktu dan sumber daya, aplikasi ini belum diuji sehingga masih dianggap sebuah prototipe.

**Kata kunci**: aplikasi; perhitungan; matriks;

# I. Pendahuluan

Matriks adalah salah satu materi matematika yang dianggap sulit. Materi matriks dianggap sulit utamanya karena banyaknya perhitungan yang harus dilakukan. Karena alasan itu pula mencari kesalahan perhitungan juga sulit, sehingga membuat proses pembelajaran yang sebagian besarnya adalah pengerjaan latihan soal semakin sulit. Karena dalam belajar, kita membuat kesalahan, dan kita belajar agar tidak membuat kesalahan serupa. Untuk itu, kita perlu tahu dimana dan apa kesalahan kita. Namun hal itu tidak mudah dalam pembelajaran materi matriks. Untuk membuat pencarian kesalahan perhitungan lebih mudah, dibuatlah suatu aplikasi untuk melakukan perhitungan matriks sederhana dengan tujuan membantu pencarian kesalahan perhitungan.

# II. Tinjauan Pustaka

## 2.1. Matriks

“Matriks didefinisikan sebagai susunan bilangan yang diatur menurut aturan baris dan kolom dalam suatu susunan berbentuk persegi panjang. Susunan bilangan itu diletakkan di dalam kurung biasa “( )” atau kurung siku “[ ]”. Untuk menamakan matriks, disepakati menggunakan huruf kapital.” (Asari *et al.*, 2015)

“Ordo atau ukuran matriks menyatakan banyaknya baris dan kolom suatu matriks dan dinotasikan dengan *m*×*n* (m baris dan n kolom).” (Asari *et al.*, 2015)

Elemen matriks adalah anggota individual matriks. Elemen matriks *M* pada baris *i* dan kolom *j* dilambangkan dengan notasi *mi, j*. (Asari *et al.*, 2015)

Berdasarkan ordonya, matriks dibedakan menjadi matriks persegi, baris, kolom, mendatar, dan tegak. Matriks persegi adalah matriks yang jumlah baris dan kolomnya sama sehingga berordo *n*×*n*. Matriks kolom adalah matriks yang jumlah kolomnya satu sehingga berordo *n*×1. (Sinaga *et al.*, 2014)

Berdasarkan elemennya, matriks dibedakan menjadi matriks nol, segitiga, simetris, diagonal, skalar, dan identitas. Matriks nol adalah matriks yang semua elemennya nol. Matriks identitas adalah matriks persegi yang semua elemennya nol kecuali elemen pada diagonal utama, yang bernilai satu. Diagonal utama adalah diagonal dari kiri atas ke kanan bawah. (Sinaga *et al.*, 2014)

Operasi matriks dibagi menjadi dua, yaitu operasi matriks dengan skalar dan operasi matriks dengan matriks. Operasi matriks dengan skalar secara resmi hanya perkalian, dimana setiap elemen matriks dikalikan dengan besaran skalar. Operasi matriks dengan matriks ada tiga, yaitu penjumlahan, pengurangan, dan perkalian. (Sinaga *et al.*, 2014)

Penjumlahan dan pengurangan antarmatriks dilakukan dengan menjumlahkan atau mengurangkan tiap elemen pada posisi yang sama. Karena itu, penjumlahan dan pengurangan antarmatriks hanya dapat dilakukan pada matriks-matriks yang berordo sama. Penjumlahan dan pengurangan matriks bersifat seperti penjumlahan dan pengurangan biasa, mereka bersifat asosiatif, dan hanya penjumlahan yang bersifat komutatif. (Sinaga *et al.*, 2014)

Perkalian matriks hanya dapat dilakukan jika jumlah kolom matriks yang di sebelah kiri sama dengan jumlah baris matriks yang di sebelah kanan. Perkalian matriks A berordo *m*×*n* dengan matriks *B* berordo *n*×*o* akan menghasilkan matriks *C* berordo *m*×*o*, dimana setiap elemennya dapat dihitung dengan rumus 1. (Sinaga *et al.*, 2014)

Rumus 1. Elemen hasil perkalian matriks.

Perkalian antarmatriks bersifat asosiatif dan distributif tapi tidak bersifat komutatif, yang berarti hasil dari *A*×*B* bisa berbeda dengan *B*×*A*. (Sinaga *et al.*, 2014)

Determinan matriks adalah suatu bilangan tunggal yang diperoleh dengan mengalikan dan menjumlahkan elemen-elemen matriks dengan cara yang khusus. Determinan suatu matriks *A* dilambangkan dengan *|A|*. (Asari *et al.*, 2015).

Menurut Laplace, determinan suatu matriks dapat dihitung dengan memilih **salah satu** baris atau kolom, kemudian mengalikan tiap elemen baris atau kolom tersebut dengan kofaktornya dan menjumlahkan semua hasil perkalian tersebut. Cara ini disebut juga cara kofaktor. (Asari *et al.*, 2015).

Rumus 2. Determinan Laplace.

Kofaktor suatu elemen dapat dihitung dengan rumus 3. (Asari *et al.*, 2015).

Rumus 3. Kofaktor entri matriks.

Sedangkan *minori,j* dapat dihitung dengan menghitung determinan matriks tersebut dengan membuang baris *i* dan kolom *j*. Dengan demikian, untuk menghitung determinan suatu matriks *n*×*n*, perlu dihitung determinan matriks (*n*-1)×(*n*-1), dan begitu seterusnya hingga ditemui penghitungan determinan matriks 1×1. (Asari *et al.*, 2015).

Determinan matriks 1×1 ialah satu-satunya elemen yang ada di dalamnya (Asari *et al.*, 2015).. Secara praktis, determinan matriks 2×2 dapat dihitung dengan rumus 4 (Asari *et al.*, 2015), sehingga rekursi perhitungan determinan dapat langsung diselesaikan pada matriks 2×2.

Rumus 4. Determinan matriks 2×2.

Bentuk-bentuk matriks meliputi transpose, minor, kofaktor, adjoin, dan invers. Matriks transpose suatu matriks *M* adalah matriks *M* yang baris dan kolomnya ditukar (Sinaga *et al.*, 2014). Matriks minor suatu matriks *M* adalah matriks yang tiap elemennya merupakan minor entri matriks *M* pada posisi yang sesuai. Matriks kofaktor suatu amtriks *M* adalah matriks yang tiap elemennya merupakan kofaktor entri matriks *M* pada posisi yang sesuai. Matriks adjoin suatu matriks *M* adalah matriks transpose dari matriks kofaktor matriks *M*. Invers suatu matriks *M* dilambangkan dengan *M-1*. Salah satu cara mendapatkan invers suatu matriks *M* adalah dengan menggunakan adjoin, yaitu dengan rumus 5. (Asari *et al.*, 2015)

Rumus 5. Invers matriks dengan adjoin matriks.

Dalam perkalian matriks, matriks identitas berfungsi seperti angka satu, sebagaimana tertera pada persamaan 1. (Sinaga *et al.*, 2014)

Persamaan 1. Sifat matriks identitas.

Sedangkan invers suatu matriks berfungsi seperti pembagi dalam perkalian matriks, sebagaimana tertera pada persamaan 2 dan 3 (Asari *et al.*, 2015). Penempatan invers sebagai pembagi tidak bisa asal karena perkalian matriks tidak komutatif. (Sinaga *et al.*, 2014)

Persamaan 2. Sifat matriks invers.

Persamaan 3. Matriks invers sebagai pembagi.

Persamaan linear *n* variabel dapat dislesaikan dengan *n* persamaan. Persamaan tersebut dapat diubah ke dalam bentuk matriks dengan persamaan pada persamaan 4. (Asari *et al.*, 2015)

Persamaan 4. Persamaan linear dalam bentuk matriks.

* *K* adalah matriks *n*×*n* yang berisi koefisien tiap variabel dari tiap persamaan secara urut.
* *V* adalah matriks *n*×1 yang berisi tiap variabel secara urut.
* *H* adalah matriks *n*×1 yang berisi hasil persamaan (konstanta) secara urut.

Persamaan matriks dapat diselesaikan dengan invers dengan menerapkan sifat invers matriks sebagai pembagi untuk mengubah persamaan 4 menjadi persamaan 5. (Asari *et al.*, 2015)

Persamaan 5. Penyelesaian persamaan matriks dengan invers.

Persamaan matriks juga dapat diselesaikan dengan determinan dengan aturan Cramer. Untuk persamaan *K*×*V* = *H*, dengan *Ki* sebagai matriks *K* yang kolom ke-*i*-nya diganti dengan *H*, variabel ke-*i* dapat dihitung dengan rumus 6. (S. and S., 2008)

Rumus 6.

Operasi baris elementer (OBE) adalah operasi-operasi yang diterapkan pada baris suatu matriks. OBE dapat digunakan untuk mencari invers suatu matriks atau menyelesaikan persamaan matriks. OBE disebut juga eliminasi Gauss-Jordan. Operasi dalam OBE ada tiga, yaitu:

1. Perkalian baris dengan suatu konstanta tidak nol
2. Penukaran baris
3. Penjumlahan atau pengurangan suatu baris dengan baris lain

OBE dilakukan dengan menggunakan paling tidak dua matriks dimana salah satunya adalah matriks persegi yang akan menjadi matriks utama. OBE dilakukan dengan melakukan operasi-operasi tersebut kepada matriks utama sedemikian rupa hingga matriks utama tersebut menjadi matriks identitas. Untuk mencari invers matriks, maka matriks utamanya adalah matriks yang dicari inversnya, sedangkan matriks lainnya adalah matriks identitas. Untuk menyelesaikan persamaan matriks, maka matriks utamanya adalah matriks koefisien, sedangkan matriks lainnya adalah matriks hasil.

## 2.2. Aplikasi Serupa

“APLIKASI PEMBELAJARAN DAN PERHITUNGAN MATRIKS BERBASIS MULTIMEDIA MENGGUNAKAN MACROMEDIA FLASH 8” (Hermawan, Harsani and Qur’ania, no date) adalah aplikasi pembelajaran dan perhitungan matriks yang terfokus pada peningkatan pemahaman siswa dengan media pembelajaran interaktif berbasis komputer. Berbeda dengan aplikasi tersebut, aplikasi ini utamanya adalah aplikasi perhitungan matriks. Materi dan latihan soal pada aplikasi ini hanyalah fitur tambahan. Selain itu, aplikasi ini tidak membatasi ordo matriks dan aplikasi ini tidak menyediakan animasi maupun simulasi.

Kaltriks adalah aplikasi penghitung matriks berbasis web yang dikembangkan dalam penelitian yang berjudul “Development of Web-based Matrix Operations Calculation as a Learning Media” (Dirgantara and Marselino, 2017). Berbeda dengan kaltriks, aplikasi ini tidak berbasis web. Pengguna perlu menggunakan komputer untuk menggunakan aplikasi ini. Selain itu, aplikasi ini tidak membatasi ordo matriks dan aplikasi ini tidak menyediakan animasi maupun simulasi.

# III. Metodologi

Penelitian ini dilakukan dengan pendekatan *waterfall*, tapi karena waktu dan sumber daya yang kurang, penelitian ini sampai pada tahap implementasi saja. Penelitian ini dilakukan dalam tiga tahap, yaitu analisis kebutuhan, desain, dan implementasi.

# IV. Hasil dan Pembahasan

## 4.1. Analisis Kebutuhan

**4.1.1 Perangkat dan Dasar**

Aplikasi dibuat dengan bahasa pemrograman *java* 7 dengan menggunakan *library* *Swing* untuk *GUI*-nya. Aplikasi tidak di-*bundle* dengan *java*, sehingga untuk menjalankan aplikasi ini diperlukan *Java Runtime Environment* *(JRE)* 7+.

**4.1.2 Matriks**

**4.1.2.1. Tinjauan Pustaka**

Untuk membuat aplikasi ini, diperlukan pengetahuan yang cukup tentang matriks, sehingga dilakukan peninjauan pustaka untuk memastikan pengetahuan yang cukup sudah dimiliki. Kelengkapan materi dicocokkan dengan materi matriks pada kelas Kalkulus dan Aljabar Linear kecuali materi penerapan matriks. Materi tersebut dikelompokkan menjadi 5 modul utama, yaitu operasi baris elementer, operasi matriks, determinan matriks, bentuk matriks, dan penyelesaian persamaan linear dengan aturan Cramer.

**4.1.2.2. Tipe Data Matriks**

Untuk membuat aplikasi penghitung matriks ini, diperlukan suatu tipe data matriks yang mendukung semua fungsi yang diperlukan dalam lima modul utama yang disebutkan sebelumnya. Seperti tipe data matematika pada umumnya, tipe data ini dibuat *immutable*.

**4.1.2.3. Operasi Matriks**

Tipe data matriks perlu mendukung semua operasi matriks, yaitu:

1. Perkalian matriks dengan skalar
2. Penjumlahan dan pengurangan antarmatriks
3. Perkalian antarmatriks

Semua operasi antarmatriks (2 dan 3) memiliki syarat, sehingga tipe data matriks tersebut harus dapat menentukan mana operasi yang dapat dilakukan dan mana yang tidak. Daripada menampilkan *error*, membatasi pilihan operasi yang tidak memenuhi syarat lebih baik.

Selain operasi-operasi resmi di atas, ada operasi-operasi tambahan yang dibuat hanya untuk memudahkan.

1. Penjumlahan dan pengurangan matriks secara *uniform* dengan skalar
2. Pembagian matriks secara *uniform* dengan skalar.
3. Operasi modulus matriks secara *uniform* dengan skalar.

**4.1.2.4. Determinan Matriks**

Karena aplikasi ini mendukung matriks *n*×*n*, cara Laplace dipilih untuk menghitung determinan. Cara Laplace juga mudah untuk diimplementasikan dengan rekursi, meskipun mungkin kurang efisien.

Untuk mengimplementasikan penghitungan determinan dengan cara Laplace, diperlukan suatu fungsi untuk menentukan kofaktor suatu elemen dari suatu matriks, dan sebagai konsekuensinya, juga diperlukan suatu fungsi untuk menentukan minor suatu elemen dari suatu matriks.

Hanya matriks persegi yang memiliki determinan. Seperti operasi, penghitungan determinan juga dibatasi.

**4.1.2.5. Bentuk-Bentuk Matriks**

Tiap bentuk matriks dibuatkan fungsi masing-masing. Tidak semua bentuk dimiliki semua matriks, jadi bentuk yang dapat dipilih pun dibatasi sesuai matriksnya. Bentuk minor dan kofaktor menggunakan fungsi yang sama dengan yang digunakan untuk penghitungan determinan.

**4.1.2.6. Operasi Baris Elementer**

Tiap operasi dari operasi baris elementer dibuatkan fungsi masing-masing. Untuk memudahkan pengguna, dibuat juga fitur *undo-redo*. Karena tipe data matriks *immutable*, fitur *undo-redo* dibuat untuk *list* yang menampung matriks-matriks tersebut.

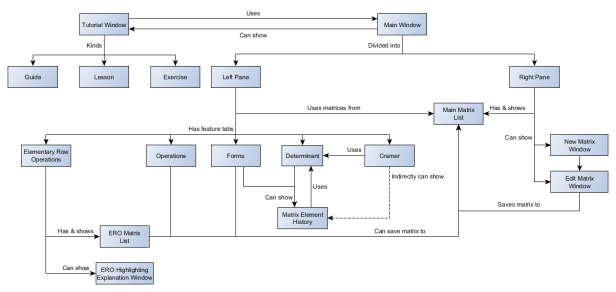
**4.1.2.7. Penyelesaian Persamaan Linear dengan Aturan Cramer**

Penyelesaian Persamaan Linear dengan Aturan Cramer dilakukan dengan mengganti kolom suatu matriks dengan kolom dari matriks kolom. Untuk itu, dibuatkan fungsi khusus. Matriks itu kemudian dihitung determinannya dengan fungsi yang sama dengan yang digunakan fitur determinan. Pilihan matriks untuk ini juga dibatasi.

## 4.2. Desain

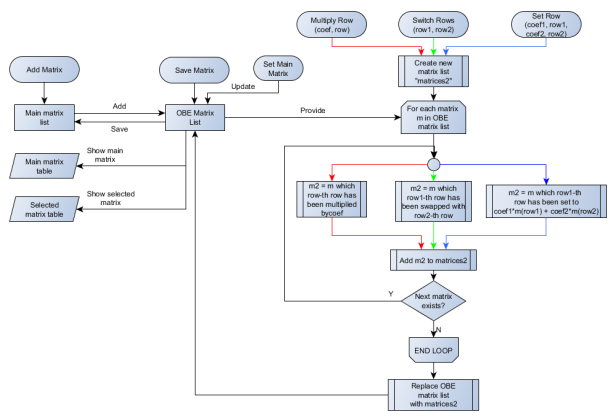
Untuk menggunakan aplikasi ini, pengguna perlu membuat matriks terlebih dahulu. Matriks-matriks yang dibuat pengguna kemudian disimpan dalam *list* utama. Tiap matriks dalam *list* utama dapat diubah atau dihapus sesuai keinginan pengguna. Setiap fitur dari aplikasi ini menggunakan matriks yang bersumber dari *list* utama. Pengguna dapat menyimpan matriks dari fitur-fitur yang menghasilkan matriks kembali ke *list* utama.

Karena aplikasi ini dibuat untuk tujuan pembelajaran, aplikasi ini juga menyediakan *tutorial* penggunaan, materi pembelajaran matris, dan soal latihan matriks. Pengguna dapat memuat contoh pada aplikasi sebagaimana pengguna dapat menggunakannya untuk menyelesaikan contoh soal. Soal latihan dibuat secara acak setiap kali pengguna hendak menampilkan *window* soal latihan. Soal latihan hanya menyediakan pengecekan jawaban, tapi setiap matriks pada soal latihan dapat langsung ditambahkan pada *window* utama dengan menekan tombol, sehingga pengguna dapat dengan mudah menggunakan aplikasi ini untuk menyelesaikannya. Fitur operasi baris elementer memiliki *list-*nya sendiri agar dapat menampung banyak matriks sekaligus. Untuk menjelaskan relasi antar entitas internal utama, dibuatkan diagram arsitektur sistem yang ada pada gambar 1.

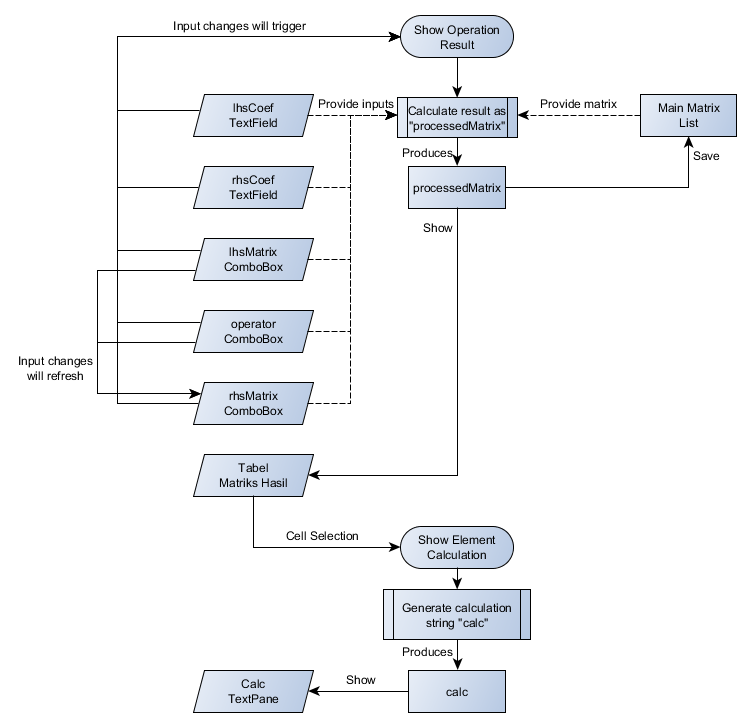


Gambar 1. Diagram arsitektur sistem.

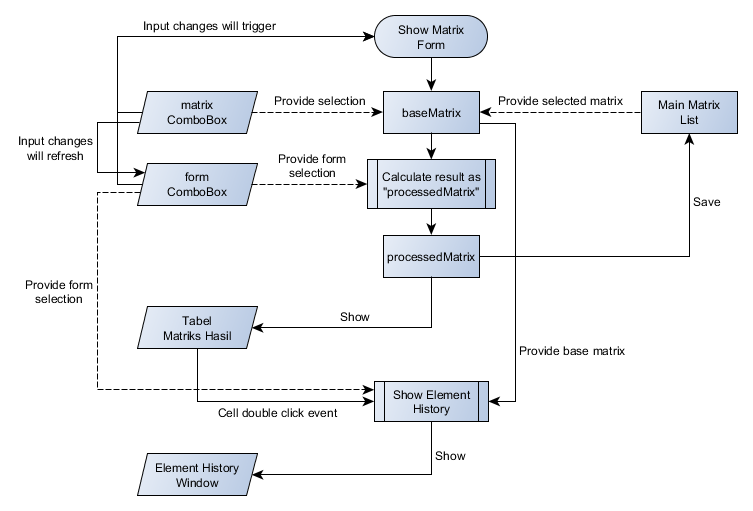
Alur program pada modul-modul utama ada pada gambar 2, 3, 4, 5, dan 6.



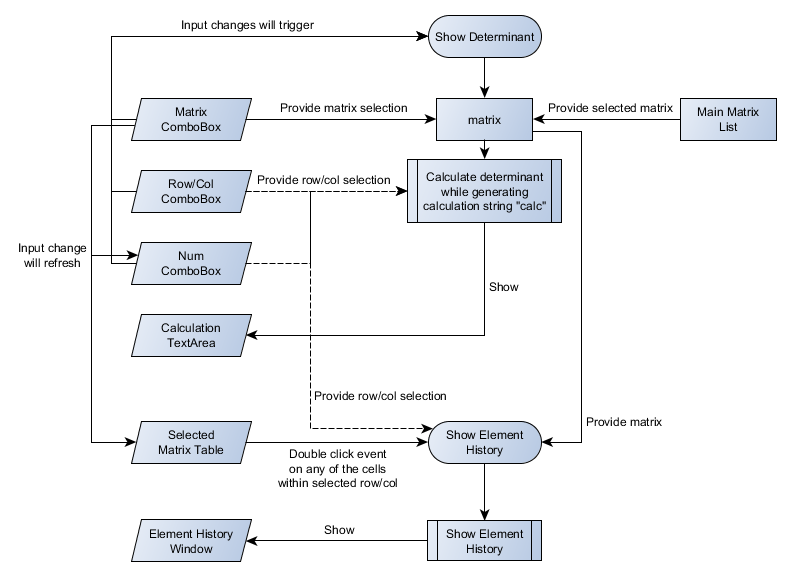
Gambar 2. Diagram modul operasi baris elementer.



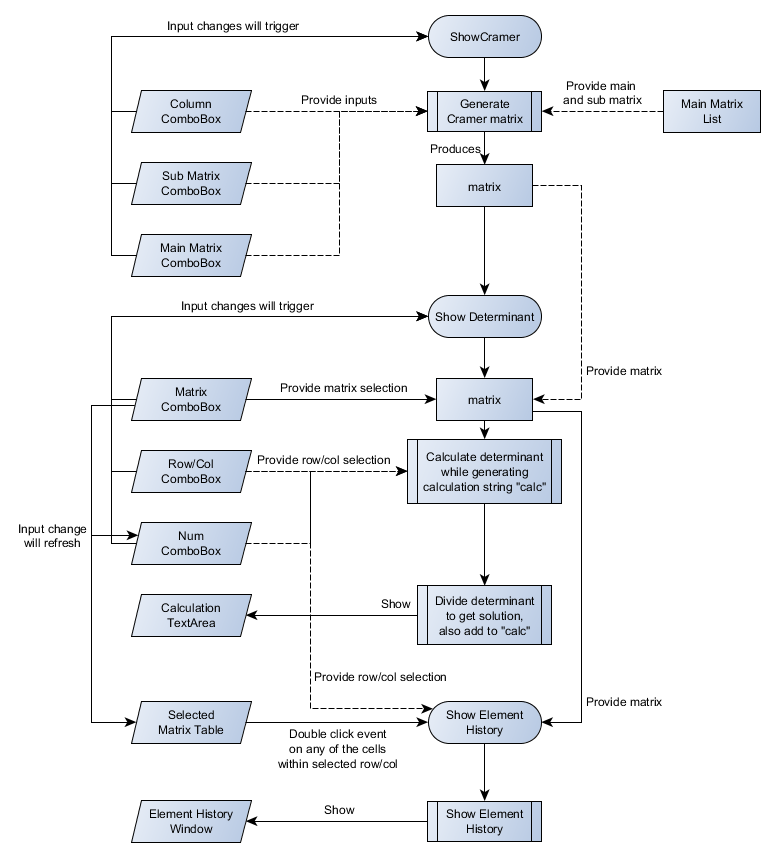
Gambar 3. Diagram modul operasi matriks.



Gambar 4. Diagram modul bentuk matriks.



Gambar 5. Diagram modul determinan matriks.

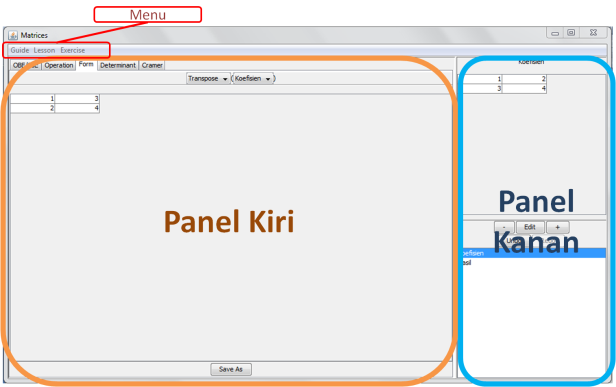


Gambar 6. Diagram modul Cramer.

## 4.3. Implementasi

**4.3.1. *Window* Utama**

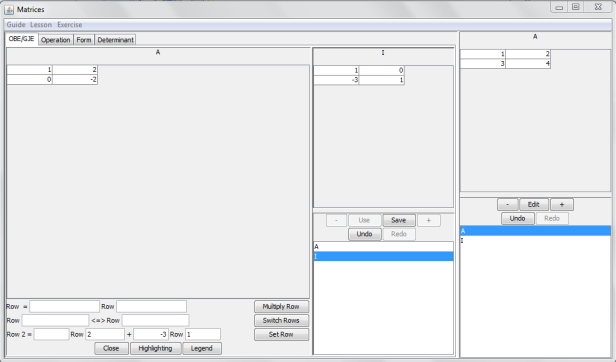
Saat pertama kali membuka aplikasi, akan muncul *window* utama dan *window* panduan (*Guide*) bagian sambutan. Tampilan awal *window* utama kosong. Pengguna harus membuat matriks terlebih dahulu dengan menekan tombol “+”, atau pengguna dapat membuka *window* materi (*Lesson*) atau latihan soal (*Exercise*) dan menambahkan matriks yang terdapat di sana. *Window* panduan, materi, dan latihan soal dapat dibuka sendiri oleh pengguna lewat menu yang ada di bagian atas *window* utama. *Window* utama dibagi menjadi panel kiri dan kanan. Panel kanan berfungsi untuk manajemen matriks, sedangkan panel kiri berisi kelima modul utama yang tiap modulnya baru akan muncul jika matriks yang ada memenuhi syarat tiap modul. Dalam aplikasi ini, matriks direpresentasikan dalam tabel. Manajemen matriks memiliki fitur undo-redo sehingga pengguna tidak perlu khawatir salah menghapus atau mengubah matriks.



Gambar 7. *Window* utama.

**4.3.2. Operasi Baris Elementer**

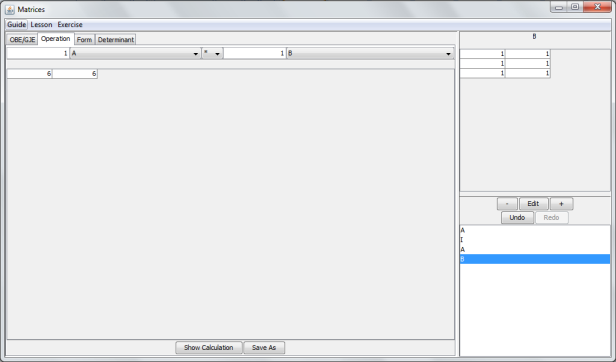
Modul operasi baris elementer memiliki *list­*-nya sendiri. Pengguna harus menambahkan matriks dari panel kanan terlebih dahulu. Setelah itu, pengguna harus memilih matriks utama, kemudian panel utama akan muncul. Pengguna kemudian dapat melakukan operasi baris elementer pada matriks utama. Setiap operasi baris elementer dilakukan pada matriks utama, operasi yang sama akan dilakukan pada setiap matriks pada daftar matriks OBE. Untuk memudahkan pengguna, dibuat juga fitur *undo-redo*. Pengguna kemudian dapat menyimpan matriks pada daftar matriks OBE kembali ke daftar matriks utama.



Gambar 8. Modul OBE.

**4.3.3. Operasi Matriks**

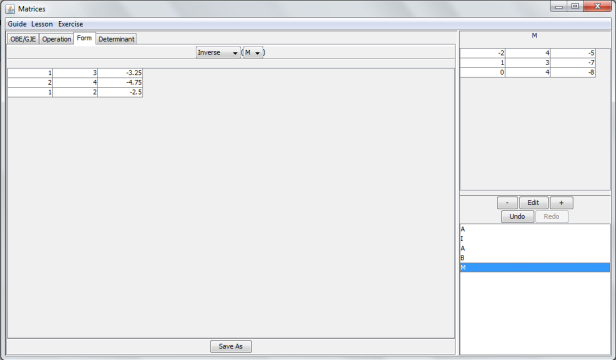
Untuk melakukan operasi matriks, pengguna dapat memilih matriks yang mau dioperasikan dan operatornya, dimulai dari matriks kiri, operator, lalu matriks kanan. Pilihan matriks kanan dibatasi menjadi hanya matriks-matriks yang dapat dioperasikan dengan matriks kiri sesuai dengan operasi yang dipilih. Pengguna juga dapat menentukan koefisien tiap matriks (kosong berarti 1). Untuk memudahkan pengguna, tersedia juga operasi *uniform*, yaitu operasi skalar dengan skalar yang dilakukan pada tiap elemen matriks. Pengguna dapat melihat perhitungan suatu elemen hasil dengan memilih elemen dan menekan tombol “Show Calculation”.



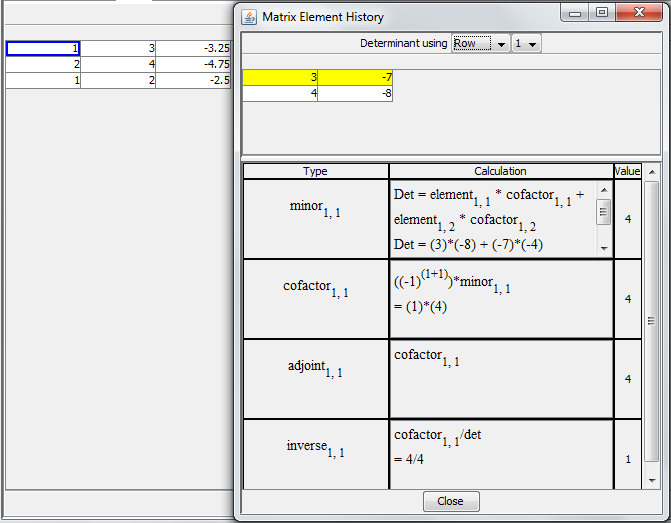
Gambar 9. Modul operasi matriks.

**4.3.4 Bentuk Matriks**

Pengguna dapat melihat bentuk lain dari suatu matriks pada modul ini dengan memilih matriks yang diinginkan lalu memilih bentuk yang diinginkan. Pengguna dapat menyimpan matriks yang dihasilkan kembali ke daftar matriks utama dengan menekan tombol “Save As” di bagian bawah. Pengguna juga dapat melihat langkah-langkah perhitungan tiap elemen matriks dengan melakukan *double click* pada elemen yang diinginkan.



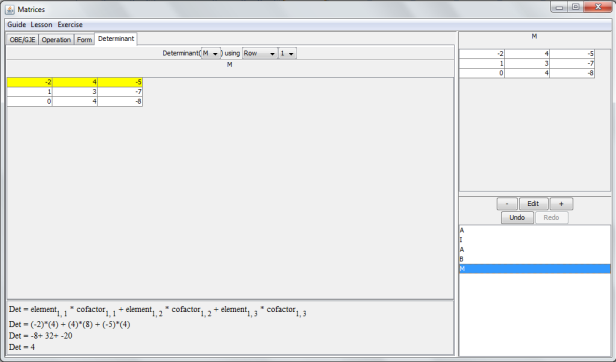
Gambar 10. Modul bentuk matriks.



Gambar 11. Perhitungan elemen matriks.

**4.3.5. Determinan Matriks**

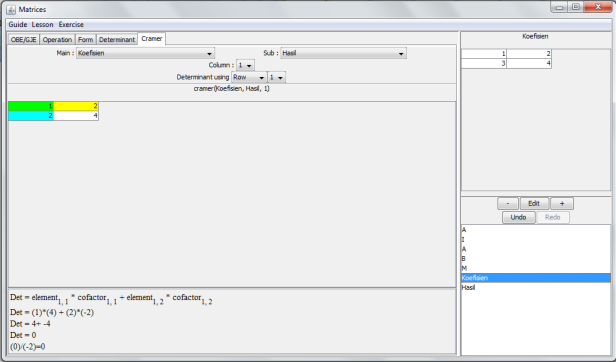
Pengguna dapat melihat perhitungan determinan suatu matriks pada modul ini (dengan cara Laplace) dengan memilih matriks yang diinginkan lalu memilih baris atau kolom untuk perhitungan. Pilihan matriks dibatasi menjadi matriks nxn saja. Perhitungan determinan akan ditampilkan di bagian bawah. Pengguna juga dapat melihat langkah-langkah perhitungan elemen yang digunakan untuk perhitungan determinan dengan melakukan *double click* pada elemen tersebut.



Gambar 12. Modul determinan.

**4.3.6. Aturan Cramer**

Pengguna dapat menyelesaikan persamaan linear dengan cara Cramer pada modul ini dengan memilih matriks utama (*n*×*n*), matriks sekunder (*n*×1), dan kolom matriks utama yang akan diganti dengan matriks sekunder (0 berarti tidak mengganti kolom). Pengguna kemudian dapat melihat perhitungannya di bagian bawah. Modul ini menggunakan modul determinan untuk menampilkan matriks dan perhitungan determinannya. Pengguna juga dapat menampilkan langkah-langkah perhitungan elemen seperti pada modul determinan.



Gambar 13. Modul Cramer.

# V. Penutup

## 5.1. Kesimpulan

Tujuan penelitian ini adalah mengembangkan suatu program perhitungan matriks untuk membantu proses pembelajaran matriks khususnya dalam pengerjaan latihan soal.

Latihan adalah cara yang sangat efektif dalam pembelajaran, karena selain dapat menerapkan materi secara langsung, pelajar dapat menemui dan belajar mengatasi kesalahan-kesalahan dalam menerapkan materi pembelajaran. Namun untuk mencari letak kesalahan dalam latihan soal matriks tidak mudah karena banyaknya perhitungan yang dilakukan. Aplikasi ini menunjukkan langkah-langkah perhitungan matriks untuk memudahkan pencarian letak kesalahan.

Tiap langkah pengembangan disajikan dalam artikel ini. Diagram dan *screenshot* aplikasi juga dilampirkan dalam artikel ini. Akan tetapi, karena keterbatasan waktu dan sumber daya, aplikasi ini belum diuji efektivitasnya, sehingga masih dianggap sebuah prototipe.

## 5.2. Penelitian Selanjutnya

Aplikasi ini masih perlu diuji sehingga pengujiannya direncanakan menjadi penelitian selanjutnya.

## 5.3. Saran

Aplikasi ini menggunakan tipe data *double* untuk merepresentasikan bilangan real sehingga tidak dapat menampilkan hasil berupa pecahan biasa. Padahal, dalam pembelajaran matriks, pecahan biasa paling umum digunakan. Sebaiknya hal ini diperhatikan dalam pengembangan aplikasi serupa.

## 5.4. Referensi

Asari, A. R. *et al.* (2015) *Buku Siswa Matematika SMA / MA Kelas 12*.

Dirgantara, H. B. and Marselino, T. L. (2017) ‘Development of Web-based Matrix Operations Calculation as a Learning Media’, *IJNMT*, IV(December), pp. 105–111.

Hermawan, Y. A., Harsani, P. and Qur’ania, A. (no date) ‘APLIKASI PEMBELAJARAN DAN PERHITUNGAN MATRIKS BERBASIS MULTIMEDIA MENGGUNAKAN MACROMEDIA FLASH 8’.

S., P. E. and S., C. A. H. F. S. (2008) *Matematika Aplikasi untuk SMA dan MA Kelas XII Program Studi IPA*.

Sinaga, B. *et al.* (2014) *Buku Siswa Matematika Kelas X SMA/MA/SMK/MAK*.