

PROTOTYPE APLIKASI PENGHITUNG MATRIKS BERBASIS JAVA

Muhammad Rizqi Nur¹⁾, Dwi Rolliawati²⁾

^{1,2)} Sistem Informasi, Fakultas Sains dan Teknologi
Universitas Islam Negeri Sunan Ampel Surabaya

E-mail : rizqinur2010@gmail.com¹⁾, dwi_roll@uinsby.ac.id²⁾

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 besar adalah pengerjaan latihan soal semakin sulit. Untuk mengatasi itu, dikembangkanlah suatu aplikasi penghitung matriks yang dapat menampilkan langkah-langkah perhitungannya. Pengembangan aplikasi ini menggunakan model prototipe yang memiliki lima fitur utama, yaitu pembantu operasi baris elementer; penghitung operasi matriks, penghitung bentuk-bentuk matriks, penghitung determinan matriks dengan ekspansi Laplace, dan penyelesaian persamaan linear dalam bentuk matriks dengan aturan Cramer. Pengujian aplikasi dilakukan dengan metode black box testing untuk menguji fungsionalitasnya yang menunjukkan bahwa setiap fitur aplikasi berfungsi dengan baik. Akan tetapi, aplikasi ini belum diuji efektivitasnya, sehingga masih dianggap prototipe.

Kata Kunci: aplikasi; perangkat lunak; penghitung; matriks; prototipe; black box testing

Abstract

Matrix is one among math subjects which are considered hard. Matrix is considered hard mainly because there are so many calculations needed to be done. Because of the same reason, finding faults in the calculations is also hard, which makes the learning process that mainly consists of exercises harder. To overcome that, a matrix calculator software which can show the calculation steps was developed. The development produced prototype model which has five main features, which are elementary row operation helper, basic matrix operation calculator, matrix form calculator, matrix determinant calculator using Laplace expansion, and a solver for linear equations using Cramer's rule. The application had been tested with black box testing for its functionality which shows that every feature was working properly. However, this application has not been tested for its effectivity, so it is still considered a prototype.

Keywords: application; software; calculator; matrix; prototype; black box testing

1. PENDAHULUAN

Matriks adalah salah satu materi matematika yang diajarkan di tingkat sekolah menengah atas dan perguruan tinggi. Materi matriks dianggap sulit, utamanya karena banyaknya perhitungan yang harus dilakukan. Dalam penelitian Indah, dkk, disebutkan bahwa kesulitan isi materi matriks dalam mata kuliah aljabar linear sebesar 38.1%, kesulitan kesadaran untuk belajar sebesar 27%, dan kesulitan pemahaman masalah sebesar 52.8% [1]. Survey yang dilakukan terhadap 120 siswa kelas XII IPA SMAN 4 Surabaya menunjukkan bahwa sebanyak 55% siswa masih mendapat nilai rata-rata kurang dari 60 untuk materi matriks [2]. Penelitian lainnya memaparkan bahwa pemahaman konseptual siswa pada materi operasi

matriks masih tergolong rendah [3]. Ketiga penelitian tersebut menguatkan asumsi bahwa materi matriks memang tergolong sulit. Karena itu, untuk mempermudah pembelajaran matriks banyak dikembangkan *tools* untuk mendukung proses pembelajaran tersebut. Di antaranya pengembangan berbasis *e-learning* [2], penggunaan multimedia *flash* untuk aplikasi pembelajaran matriks [4, 5, 6] dilanjutkan dengan pengembangan aplikasi pembelajaran berbasis python yang dibatasi pada operasi matriks inti saja dengan maksimal ordo 3x3 tanpa disertai latihan [7]. Dari sisi perangkat pembantu penghitungan matriks sendiri, telah dikembangkan prototipe penghitungan matriks (KaltriKS) tapi hanya mengakomodasi maksimal ordo 3x3 saja [8]. Dari semua pemaparan tersebut, baik pengembangan

aplikasi pembelajaran maupun penghitungan matriks masih dibutuhkan terus seiring dengan implementasi matriks yang semakin rumit di berbagai bidang sebagaimana dipaparkan pada berbagai penelitian [9, 10, 11]. Untuk itu, penelitian ini bertujuan mengembangkan prototipe aplikasi penghitungan matriks tanpa dibatasi ordo yang disertai dengan tutorial pembelajaran dan evaluasi pembelajaran untuk melengkapi proses belajar mandiri bagi siswa menengah atas maupun mahasiswa di perguruan tinggi. Aplikasi ini dikembangkan berbasis *desktop* dengan menggunakan bahasa pemrograman *Java*, dengan harapan untuk memudahkan penggunaannya, karena tidak perlu terhubung ke internet.

2. TINJAUAN PUSTAKA

2.1 Aplikasi

Aplikasi merupakan subkelas perangkat lunak komputer yang bersifat mandiri untuk menjawab kebutuhan tertentu/terinci [12]. Jogyanto mendefinisikan aplikasi sebagai sistem yang dirancang sedemikian rupa untuk menghasilkan informasi yang terpadu dengan menggunakan sarana komputer sebagai penunjangnya [13].

2.2 Matriks

Menurut J. Supranto, matriks adalah suatu kumpulan dari pada angka-angka (sering disebut elemen-elemen) yang disusun menurut baris dan kolom sehingga berbentuk empat persegi panjang, dimana panjangnya dan lebarnya ditunjukkan oleh banyaknya kolom-kolom dan baris-baris [5]. Ordo atau ukuran matriks menyatakan banyaknya baris dan kolom suatu matriks dan dinotasikan dengan $m \times n$ (m baris dan n kolom) [14]. Elemen matriks adalah anggota individual matriks. Elemen matriks M pada baris i dan kolom j dilambangkan dengan notasi $m_{i,j}$ [14].

Berdasarkan ordonya, matriks dibedakan menjadi matriks persegi, baris, kolom, mendatar, dan tegak. Matriks persegi adalah matriks yang jumlah baris dan kolomnya sama sehingga berordo $n \times n$. Matriks kolom adalah matriks yang jumlah kolomnya satu sehingga berordo $n \times 1$ [15].

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 bernilai nol, kecuali elemen pada diagonal utama yang bernilai satu. Diagonal utama adalah diagonal dari kiri atas ke kanan bawah [15].

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 [15].

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. Penjumlahan bersifat asosiatif dan komutatif, sedangkan pengurangan tidak [15]. 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 $m \times n$ dengan matriks B berordo $n \times o$ akan menghasilkan matriks C berordo $m \times o$, dimana setiap elemennya dapat dihitung dengan persamaan (1) [15].

$$c_{i,j} = \sum_{k=1}^n a_{i,k} \times b_{k,j}, \quad (1)$$

Perkalian antarmatriks bersifat asosiatif dan distributif, tapi tidak bersifat komutatif sehingga hasil dari $A \times B$ bisa berbeda dari $B \times A$ [15]. 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 $|A|$ [14].

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. [14].

$$|M| = \sum(m_{i,j} \times \text{kofaktor}_{i,j}), \quad (2)$$

Kofaktor suatu elemen dapat dihitung dengan rumus (3) [14].

$$\text{kofaktor}_{i,j} = (-1)^{i+j} \times \text{minor}_{i,j}, \quad (3)$$

Sedangkan $\text{minor}_{i,j}$ dapat dihitung dengan menghitung determinan matriks tersebut dengan membuang baris i dan kolom j . Dengan demikian, untuk menghitung determinan suatu matriks $n \times n$, perlu dihitung determinan matriks $(n-1) \times (n-1)$, dan begitu seterusnya hingga ditemui penghitungan determinan matriks 1×1 [14].

Determinan matriks 1×1 ialah satu-satunya elemen yang ada di dalamnya [14]. Secara praktis, determinan matriks 2×2 dapat dihitung dengan rumus (4) [14], sehingga rekursi perhitungan determinan dapat langsung diselesaikan pada matriks 2×2 .

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc, \quad (4)$$

Bentuk-bentuk matriks meliputi *transpose*, *minor*, *kofaktor*, *adjoin*, dan *invers*. Matriks *transpose* suatu matriks M adalah matriks M yang baris dan kolomnya ditukar [15]. Matriks *minor* suatu matriks M adalah matriks yang tiap elemennya merupakan minor entri matriks M pada posisi yang sesuai. Matriks *kofaktor* suatu matriks 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*, sebagaimana persamaan (5) [14].

$$M^{-1} = \frac{1}{|M|} \times \text{adj}(M), \quad (5)$$

Dalam perkalian matriks, matriks identitas berfungsi seperti angka satu, sebagaimana tertera pada persamaan (6) [15],

$$M \times I = I \times M = M, \quad (6)$$

Sedangkan *invers* suatu matriks berfungsi seperti pembagi dalam perkalian matriks, sebagaimana tertera pada persamaan (7) dan (8) [14]. Penempatan *invers* sebagai pembagi tidak bisa asal, karena perkalian matriks tidak komutatif [15].

$$M \times M^{-1} = M^{-1} \times M = I, \quad (7)$$

$$A \times B = C \rightarrow B = A^{-1} \times C, \quad (8)$$

Adapun untuk persamaan linear dalam bentuk matriks dengan n variabel dapat diselesaikan dengan persamaan (9) [14].

$$K \times V = H, \quad (9)$$

K adalah matriks $n \times n$ yang berisi koefisien tiap variabel dari tiap persamaan secara urut.

V adalah matriks $n \times 1$ yang berisi tiap variabel secara urut.

H adalah matriks $n \times 1$ 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 (10) [14].

$$V = K^{-1} \times H, \quad (10)$$

Persamaan matriks juga dapat diselesaikan dengan determinan, yaitu dengan aturan *Cramer*. Untuk persamaan $K \times V = H$, dengan K_i sebagai matriks K yang kolom ke- i -nya diganti dengan H , variabel ke- i dapat dihitung dengan rumus (11) [16].

$$v_{i,1} = \frac{|K_i|}{|K|}, \quad (11)$$

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 koefisien tidak nol
2. Penukaran baris
3. Penjumlahan atau pengurangan suatu baris dengan baris lain setelah dikalikan dengan suatu koefisien tidak nol

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 sehingga matriks utama tersebut menjadi matriks identitas. Untuk mencari *invers* matriks, maka matriks utamanya adalah matriks yang dicari *invers*nya, sedangkan matriks lainnya adalah matriks identitas. Untuk menyelesaikan persamaan matriks, maka matriks utamanya adalah matriks koefisien, sedangkan matriks lainnya adalah matriks hasil.

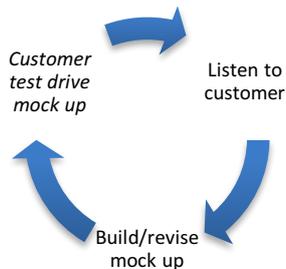
3. METODE PENGEMBANGAN PERANGKAT LUNAK

Metode pengembangan perangkat lunak yang digunakan dalam penelitian ini adalah metode *prototyping*. Metode ini dianggap paling cocok untuk diimplementasikan karena aplikasi masih akan terus dikembangkan untuk perbaikan atau

evaluasi berikutnya. Metode *prototyping* ini memiliki 3 tahapan yaitu:

1. Listen to customer
2. Build/revise mock up
3. Customer test drive mock up

Alur pengembangan metode ini ditunjukkan pada Gambar 1.



Gambar 1. Metode Pengembangan Prototyping[12]

Pada tahap *listen to customer*, dilakukan studi literatur yang berkaitan dengan survei aplikasi serupa, serta literatur untuk operasi penghitungan matriks. Studi literatur ini bertujuan untuk memastikan validitas pengetahuan dalam penghitungan matriks. Kelengkapan materi yang digunakan pada aplikasi ini, 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 (OBE), operasi matriks, determinan matriks, bentuk matriks, dan penyelesaian persamaan linear dengan aturan *Cramer*.

Tahap berikutnya yaitu *build/revise mock up*, pada proses ini dilakukan perancangan/desain arsitektur sistem disertai dengan diagram alir dari setiap fitur yang disediakan pada aplikasi ini. Selanjutnya, aplikasi dibangun 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+*.

Untuk tahapan *customer test drive mock up*, sementara ini aplikasi yang dibangun masih belum diujikan secara masal kepada pengguna. Pengujian dilakukan hanya untuk menilai fungsionalitas aplikasi menggunakan *black box testing*.

4. Hasil dan Pembahasan

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*. Tipe data matriks perlu mendukung semua operasi matriks, yaitu:

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

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. Selain operasi-operasi tersebut, 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.

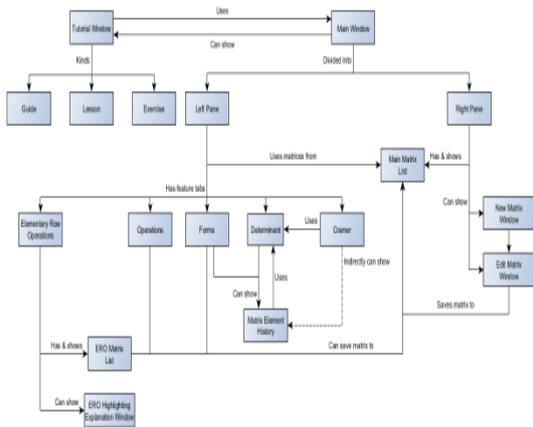
Karena aplikasi ini mendukung matriks $n \times n$, cara *Laplace* dipilih untuk menghitung determinan. Cara *Laplace* juga mudah untuk diimplementasikan dengan rekursi, meskipun mungkin kurang efisien. Untuk memudahkan pengguna, dibuat juga fitur *undo-redo*. Karena tipe data matriks *immutable*, fitur *undo-redo* dibuat untuk *list* yang menampung matriks-matriks tersebut. Proses 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 pada fitur determinan. Pilihan matriks untuk modul ini juga dibatasi.

4.1 Desain Sistem

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.

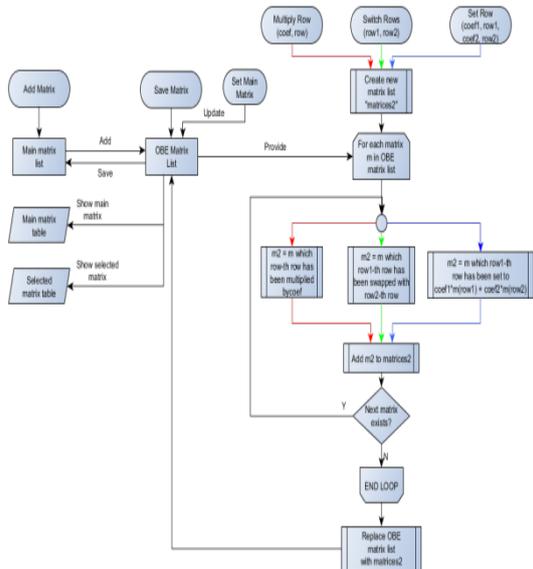
Karena aplikasi ini dibuat untuk tujuan pembelajaran, aplikasi ini juga menyediakan *tutorial* penggunaan, materi pembelajaran matriks, dan soal latihan matriks. Pengguna dapat memuat contoh pada aplikasi sebagaimana pengguna dapat menggunakannya untuk menyelesaikan contoh

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 menampilkan banyak matriks sekaligus. Untuk menjelaskan relasi antar entitas internal utama, dibuatkan diagram arsitektur sistem yang ada pada Gambar 2.

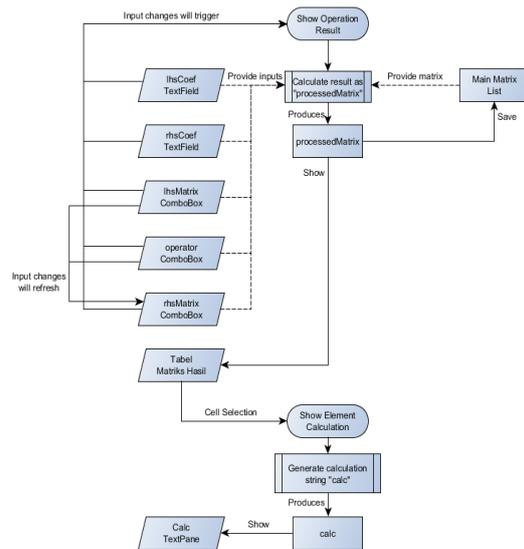


Gambar 2. Diagram arsitektur sistem.

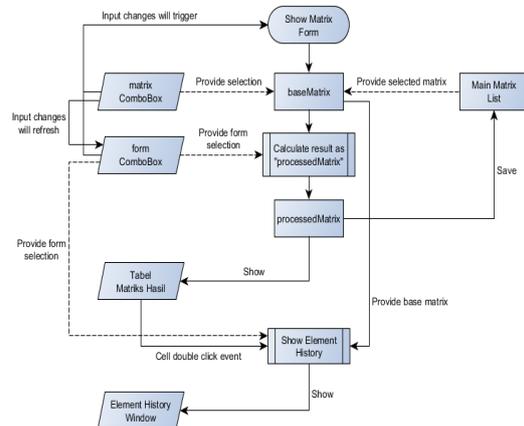
Alur program pada modul-modul utama ada pada Gambar 3, 4, 5, 6 dan 7.



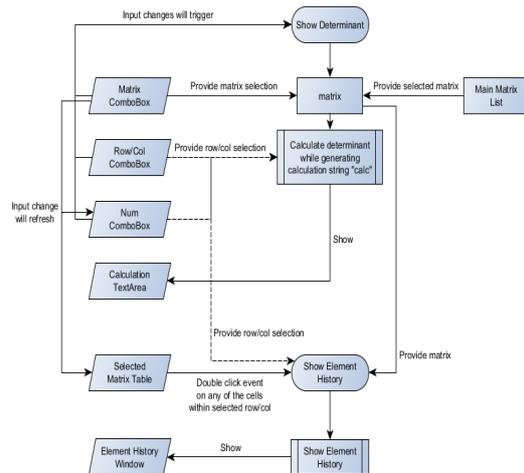
Gambar 3. Diagram modul operasi baris elementer.



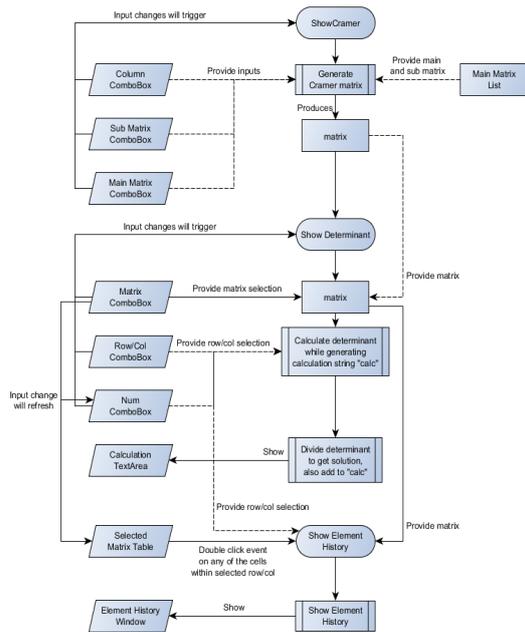
Gambar 4. Diagram modul operasi matriks.



Gambar 5. Diagram modul bentuk matriks.



Gambar 6. Diagram modul determinan matriks.

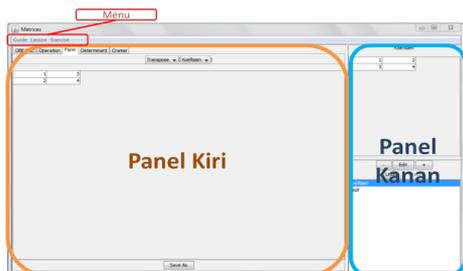


Gambar 7. Diagram modul Cramer.

4.2 Implementasi Sistem

4.2.1. Window Utama

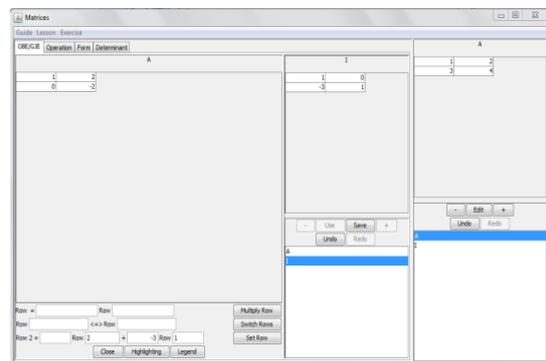
Saat pertama kali membuka aplikasi, akan muncul *window* utama dan *window* panduan (*Guide*) bagian sambutan sebagaimana ditunjukkan Gambar 8. Tampilan awal *window* utama kosong. Pengguna harus membuat matriks terlebih dahulu dengan menekan tombol “+”, atau pengguna dapat membuka *window* materi (*Lesson*) 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 8. Window utama.

4.2.2. Operasi Baris Elementer

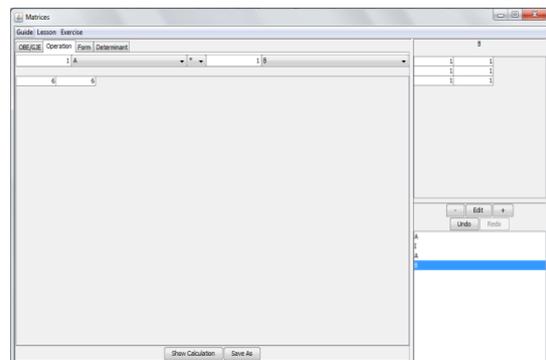
Modul operasi baris elementer memiliki *list*-nya sendiri. Pengguna harus menambahkan matriks dari panel kanan terlebih dahulu. Setelah itu, pengguna harus memilih matriks utama, kemudian panel utama akan muncul. Pengguna kemudian dapat melakukan operasi baris elementer pada matriks utama. Pada setiap kali 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 9. Modul OBE.

4.2.3. Operasi Matriks

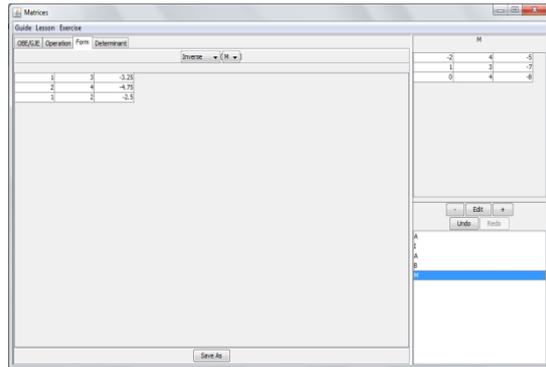
Untuk melakukan operasi matriks, pengguna dapat memilih matriks yang mau dioperasikan dan operatornya, dimulai dari matriks kiri, operator, lalu matriks kanan. Pilihan matriks kanan dibatasi menjadi hanya matriks-matriks yang dapat dioperasikan dengan matriks kiri sesuai dengan operasi yang dipilih. 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 yang menghasilkan suatu elemen pada matriks hasil operasi dengan memilih elemen dan menekan tombol “Show Calculation”.



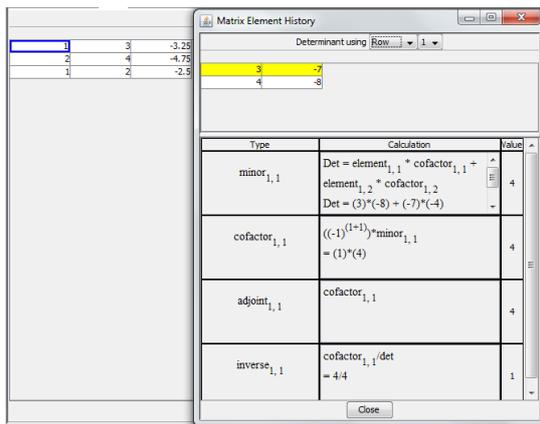
Gambar 10. Modul operasi matriks.

4.2.4 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 11. Modul bentuk matriks.



Gambar 12. Perhitungan elemen matriks.

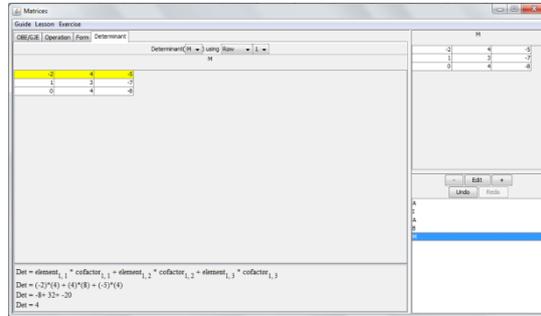
4.2.5. Determinan Matriks

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 $n \times n$ 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.

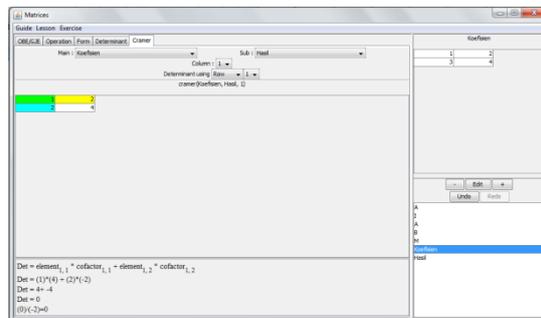
4.2.6. Aturan Cramer

Pengguna dapat menyelesaikan persamaan linear dengan cara Cramer pada modul ini dengan memilih matriks utama ($n \times n$), matriks sekunder ($n \times 1$), 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 determinan.



Gambar 14. Modul Cramer.

4.3 Pengujian

Pengujian fungsionalitas fitur menggunakan *black box testing* menunjukkan bahwa semua fitur berjalan sesuai dengan baik sesuai dengan *requirement*. Hasil *black box testing* terdapat pada tabel 1. Aplikasi ini belum diuji *user experience* dan *usability*-nya, sehingga masih dianggap prototipe.

Tabel 1
Hasil Pengujian

Kelas Uji	Butir Uji	Hasil
Manajemen matriks: pembuatan matriks: preset	Pembuatan matriks <i>Scalar</i>	OK
	Pembuatan matriks <i>Uniform</i>	OK
	Pembuatan matriks <i>Custom</i>	OK
Manajemen matriks: penetapan detail matriks	Input karakter valid	OK
	Input karakter tidak valid	OK
	Penetapan ordo normal	OK
	Penetapan ordo nol atau negatif	OK
	Penetapan ordo dengan karakter ilegal	OK
	Penyisipan baris atau kolom	OK
	Penghapusan baris atau	OK

	kolom normal	
	Percobaan penghapusan baris atau kolom ketika hanya ada satu	OK
Penampilan matriks	Penampilan matriks yang dipilih	OK
Manajemen matriks: perubahan matriks: penyimpanan	Penyimpanan sebagai pembaruan matriks lama	OK
	Penyimpanan sebagai matriks baru	OK
Manajemen matriks: penghapusan matriks	Penghapusan matriks	OK
Manajemen matriks: <i>refresh</i> tab	Matriks persegi panjang	OK
	Matriks persegi singular	OK
	Matriks persegi non-singular	OK
	Matriks 1x1	OK
	Matriks cramer	OK
Manajemen matriks: <i>undo-redo</i>	<i>Undo</i>	OK
	<i>Redo</i>	OK
OBE: Penambahan matriks	Percobaan penambahan ketika matriks utama terpilih	OK
	Penambahan ketika matriks utama tidak terpilih	OK
OBE: Penampilan matriks OBE	Penampilan matriks yang dipilih	OK
Penghapusan matriks OBE	Percobaan penghapusan ketika matriks utama terpilih	OK
	Penghapusan ketika matriks utama tidak terpilih	OK
Pemilihan matriks utama OBE	Tidak ada matriks dengan jumlah baris berbeda	OK
	Ada matriks dengan jumlah baris berbeda	OK
<i>Undo-redo</i> manajemen matriks OBE	<i>Undo</i>	OK
	<i>Redo</i>	OK
OBE: Perkalian baris	Kondisi normal	OK
	Input koefisien kosong	OK
	Koefisien nol atau satu	OK
	Input baris kosong	OK
	Baris tidak ada	OK
	Adanya karakter ilegal	OK
OBE: Penukaran baris	Kedua baris ada	OK
	Salah satu input baris kosong	OK
	Salah satu baris tidak ada	OK
	Adanya karakter ilegal	OK
OBE: Penjumlahan baris	Kondisi normal	OK
	Koefisien kosong	OK
	Salah satu koefisien nol	OK
	Salah satu input baris kosong	OK
	Salah satu baris tidak ada	OK
	Adanya karakter ilegal	OK
OBE: penyimpanan	Penyimpanan matriks	OK
Operasi matriks: input dan hasilnya	Pemilihan matriks kiri ketika ada matriks yang tidak dapat dioperasikan dengan matriks kiri dan operator yang dipilih	OK

	Pemilihan operator: penjumlahan dan pengurangan ketika ada matriks yang tidak bisa dijumlahkan atau dikurangkan dengan matriks kiri yang dipilih	OK	
	Pemilihan operator: perkalian ketika ada matriks yang tidak bisa dikalikan dengan matriks kiri yang dipilih	OK	
	Pemilihan operator: operator tambahan	OK	
	Pemilihan matriks kanan: matriks	OK	
	Pemilihan matriks kanan: <i>uniform scalar</i>	OK	
	Koefisien kosong	OK	
	Koefisien mengandung karakter ilegal	OK	
	Operasi matriks: perhitungan elemen	Perhitungan penjumlahan atau pengurangan antarmatriks	OK
		Perhitungan perkalian antarmatriks	OK
		Perhitungan operasi <i>uniform scalar</i>	OK
Bentuk matriks: pemilihan matriks	Pemilihan matriks persegi panjang	OK	
	Pemilihan matriks persegi singular	OK	
	Pemilihan matriks persegi non-singular	OK	
Bentuk matriks: hasil dan riwayat elemen	Hasil dan riwayat elemen bentuk transpose	OK	
	Hasil dan riwayat elemen bentuk minor	OK	
	Hasil dan riwayat elemen bentuk kofaktor	OK	
	Hasil dan riwayat elemen bentuk invers	OK	
Bentuk matriks: penyimpanan	Penyimpanan matriks	OK	
Determinan matriks	Pemilihan baris atau kolom	OK	
	Penampilan perhitungan	OK	
	Penampilan riwayat elemen determinan nxn; n>1	OK	
	Percobaan penampilan riwayat elemen determinan 1x1	OK	
Aturan Cramer: <i>refresh</i>	Perubahan daftar matriks utama me- <i>refresh</i> pilihan matriks utama	OK	
Aturan Cramer: input	Pemilihan matriks utama ketika ada matriks lain yang tidak bisa menjadi matriks sekunder	OK	
	Pemilihan matriks sekunder	OK	
	Pemilihan kolom yang diganti	OK	
Aturan Cramer: matriks	Penampilan matriks	OK	

Aturan Cramer: determinan	Pemilihan baris atau kolom	OK
	Penampilan perhitungan	OK
	Penampilan riwayat elemen	OK

5. 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. Fungsionalitas aplikasi sudah diuji dengan *black box testing* yang menunjukkan bahwa aplikasi bekerja dengan semestinya. Akan tetapi, aplikasi ini belum diuji efektivitasnya sehingga masih dianggap sebuah prototipe.

5.2 Penelitian Selanjutnya

Aplikasi ini masih perlu diuji kepada pengguna dari sisi *user experience* dan *usability* 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.

DAFTAR PUSTAKA

- [1] I. Nursupriah & M. Sholikhah, "Analisis Kesulitan Mahasiswa Dalam Memahami Mata Kuliah Aljabar Matriks (Studi Kasus Pada Semester IV Tadris Matematika Tahun Akademik 2008/2009 Di STAIN Cirebon) Indah Nursupriah, Marati Sholikhah," pp. 75–85, 2009.
- [2] J.I. Prastyawan, M.J.D. Sunarto, & M. Arifin, "Rancang Bangun Aplikasi Pembelajaran Matriks Berbasis Web Dengan Model Multimedia Learning (Studi Kasus : SMAN 4 Surabaya) 1) Joane Indra Prastyawan 2) M.J. Dewiyani Sunarto 3) Mochammad Arifin," pp. 1–12, 2012.
- [3] H. Lesmana, E. Yusmin, & S. Sayu, "Pendesripsian Pemahaman Konseptual Siswa Menyelesaikan Soal-Soal Operasi Matriks Kelas X SMKN 3 Pontianak," 2015.
- [4] P. Soepomo, "Multimedia pembelajaran diagonalisasi matriks 1," vol. 2, pp. 721–730, 2014.
- [5] J.R. Sihombing, "Aplikasi Pembelajaran Matriks Untuk Sekolah Menengah Atas (SMA) Kelas X Menggunakan Metode Computer Assisted Instruction (CAI)," pp. 23–28, 2016.
- [6] Y. A. Hermawan, P. Harsani, & A. Qur'ania, "Aplikasi Pembelajaran dan Perhitungan Matriks Berbasis Multimedia Menggunakan Macromedia Flash 8."
- [7] A. Syafarirna, "Aplikasi Pembelajaran Matriks Untuk Siswa Kelas Xii Sma Dengan Python 2.7.3 Dan Toolkit Tkinter," 2012.
- [8] H.B. Dirgantara & T.L. Marselino, "Development of Web-based Matrix Operations Calculation as a Learning Media," *INMT*, vol. IV, no. December, pp. 105–111, 2017.
- [9] A. Purnama, "Aplikasi Matriks dalam Pengolahan Gambar," no. December 2015, 2016.
- [10] D. Tinjauan & S. Elemen, "Rancang Bangun Alat Bantu Proses Belajar Mengajar Statika Dan Analisis Struktur Metode Matriks pada Sekolah Kejuruan dan Perguruan Tinggi Teknik Dengan Tinjauan Struktur Elemen Frame (Portal Bidang)," *Wahana Tek. SIPIL Vol.*, vol. 14, pp. 17–28, 2007.
- [11] L. Starkey, "Evaluating learning in the 21st century : a digital age learning matrix," no. May, 2016.
- [12] Presman Roger, *Software Engineering : A*

Practical Approach. 2014.

- [13] J. Hartono, *Pengenalan Komputer*. 2004.
- [14] A.R. Asari, I. Yuwono, M. Muksar, T.D. Chandra, L.M. Lestyanto, L. Anwar, N. Atikah, D. Hasanah, S.H. Nasution, & V. Kusumasari, *Buku Siswa Matematika SMA / MA Kelas 12*, no. 1. 2015.
- [15] B. Sinaga, P.N.J.M. Sinambela, A.K. Sitanggang, T.A. Hutapea, L.P. Sinaga, S. Manullang, M. Simanjorang, & Y.T. Bayuzetra, *Buku Siswa Matematika Kelas X SMA/MA/SMK/MAK*. 2014.
- [16] P.E. S. & C.A.H.F.S. Santosa, *Matematika Aplikasi untuk SMA dan MA Kelas XII Program Studi IPA*. 2008.