

**INTERACTIVE MATRIX DETERMINANT METODE KOFAKTOR
ALJABAR LINIER**



INSTITUT TEKNOLOGI PLN

ANGGOTA : ZAIN AKBAR RIZKIA (202531091)

IQBAL RAIHAN RAFFIANZA (202531111)

FX. OKTABIMO DWIPRIABUDI SUMINTRO (202531085)

LEWIS RADHI OKTASAMIQDAD (202531107)

KELAS : C31020101

DOSEN : RIZQIA CAHYANINGTYAS, ST., M.Kom

**FAKULTAS TELEMATIKA ENERGI
TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI PLN-JAKARTA**

2025/2026

BAB I: LAPORAN TEORI

1.1. Definisi Determinan

Determinan adalah nilai skalar yang dihasilkan fungsi dari entri-entri suatu matriks persegi. Determinan dinyatakan dengan notasi $\det(A)$, $\det A$, atau $|A|$.

Determinan dari matriks 2×2 adalah

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

dan determinan dari matriks 3×3 adalah

$$\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = aei + bfg + cdh - ceg - bdi - afh.$$

1.2. Metode Perhitungan Ekspansi Kofaktor

Ekspansi Laplace mendefinisikan determinan dari matriks A ukuran $n \times n$ secara rekursif sebagai penjumlahan determinan matriks-matriks yang lebih kecil, yang disebut minor. Minor M_{ij} didefinisikan sebagai determinan submatriks berukuran $(n-1) \times (n-1)$ yang dihasilkan dari menghapus baris ke-i dan kolom ke-j matriks A. Untuk sembarang i (ekspansi menurut baris ke-i) akan berlaku hubungan

$$\det(A) = \sum_{j=1}^n (-1)^{i+j} a_{i,j} M_{i,j}$$

BAB II: LAPORAN PRAKTIKUM

2.1. Program Input Biodata

Program "Interactive Matrix Determinant" yang terdapat pada repositori GitHub
<https://github.com/Rizkiacry/interactive-matrix-determinant>

merupakan sebuah aplikasi berbasis teks (Terminal User Interface - TUI) untuk secara interaktif memasukkan elemen-elemen matriks persegi dan melihat hasil perhitungan determinannya secara real-time menggunakan library [termui-cpp](#) untuk UI.

2.1.1. Konsep Dasar Program

Program ini dirancang untuk memvisualisasikan dan menghitung determinan matriks persegi dengan ukuran yang dapat diubah-ubah oleh pengguna (dari 1x1 hingga **MAX_N** x **MAX_N**, di mana **MAX_N** didefinisikan sebagai 10 dalam source code). Pengguna dapat:

1. Menavigasi Matriks: Menggunakan tombol panah, **hjkl**, atau **wasd** untuk berpindah antar elemen matriks.
 2. Menginput Nilai Elemen: Mengetikkan angka (termasuk bilangan negatif) untuk mengubah nilai elemen matriks yang sedang dipilih.
 3. Mengubah Ukuran Matriks: Meningkatkan atau menurunkan ordo matriks dengan menekan tombol **'.'** atau **', '**.
 4. Melihat Hasil Determinan: Nilai determinan diperbarui dan ditampilkan secara otomatis setiap kali ada perubahan pada elemen matriks atau ukuran matriks.
 5. Keluar dari Program: Menekan tombol **'q'** untuk mengakhiri eksekusi program.

2.1.1. Source Code Program

Gambar 1: Source Code Program Interactive Matrix Determinant Metode Kofaktor

```

+--Matrix Determinant Calculator-----+
|Matrix (Size: 3x3)
|[0] 0 0
|0 0 0
|0 0 0
|
|Determinant: 0
|
|Use arrow keys, hjkl, or wasd to navigate.
|Use Enter to confirm input.
|Press '.' to increase matrix size, ',' to decrease.
|Press 'q' to quit.
|
|
|
+-

```

Gambar 2: Interface Awal Program Interactive Matrix Determinant Metode Kofaktor

2.1.2. Penjelasan Source Code

Penulis memberikan pseudocode untuk menjelaskan source code. Penulis juga memberi penjelasan berupa comments pada source code, yang diawali dengan // (dua garis miring).

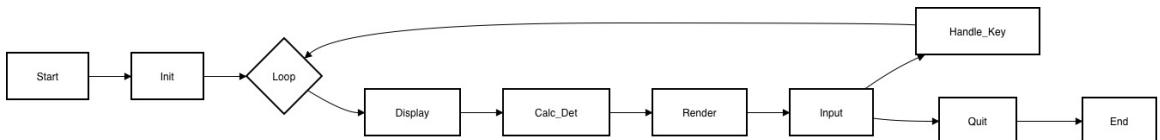
```

1 // Include necessary libraries
2 #include <iostream>
3 #include <string>
4 #include <math.h>
5 #include <windows.h>
6
7 // Use standard namespace
8 using namespace std;
9
10 // Define maximum matrix size
11 #define MAX_N 10
12
13 // Declare global matrix
14 DECLARE INTEGER 2D_ARRAY mat[MAX_N][MAX_N];
15
16 // Declare global variable to store determinant
17 DECLARE INTEGER current_determinant = 0;
18
19 // External declaration for function to get string value
20 EXTERNAL STRING get_string(BOOLEAN running, INTEGER mat[SIZE][SIZE], INTEGER mat_size, INTEGER current_col, INTEGER current_row, INTEGER current_col_size, STRING mat_size_ref, STRING current_col_ref, STRING current_row_ref);
21
22 // External declaration for function to handle key presses
23 EXTERNAL void handle_key_press(BOOLEAN running, INTEGER mat[SIZE][SIZE], INTEGER mat_size, INTEGER mat_size_ref, INTEGER current_col, INTEGER current_row, INTEGER current_col_ref, INTEGER current_row_ref, STRING mat_size_ref);
24
25 // Function to get cofactor of a matrix
26 FUNCTION get_cofactor(2D_ARRAY temp, INTEGER n, INTEGER p, INTEGER q) RETURNS INTEGER
27 {
28     DECLARE INTEGER sign = 1;
29     FOR i FROM 0 TO n-1
30     {
31         IF i == p THEN
32             CONTINUE LOOP;
33         FOR j FROM 0 TO n-1
34         {
35             IF j == q THEN
36                 CONTINUE LOOP;
37             SET temp[i][j] = mat[i][j];
38         }
39     }
40     RETURN temp;
41 }
42
43 // Function to calculate determinant of a matrix
44 FUNCTION determinant(2D_ARRAY mat[SIZE][SIZE], INTEGER n) RETURNS INTEGER
45 {
46     IF n == 1 THEN
47         RETURN mat[0][0];
48     ELSE
49     {
50         DECLARE INTEGER 2D_ARRAY temp[MAX_N][MAX_N];
51         DECLARE INTEGER sign = 1;
52
53         FOR i FROM 0 TO n-1
54         {
55             SET memory_of_temp TO i;
56             SET memory_of_det TO i;
57             SET D TO 0;
58             FOR j FROM 0 TO n-1
59             {
60                 IF mat[i][j] != 0 THEN
61                     SET D = D + (sign * mat[i][j]) * determinant(temp, n - 1);
62                 sign = -sign;
63             }
64         }
65         RETURN D;
66     }
67 }
68
69 // Main Function
70 EXTERNAL void main() RETURNS INTEGER
71 {
72     DECLARE tui.Window window;
73     DECLARE tui.InputEvent event;
74     DECLARE BOOLEAN running = TRUE;
75
76     // Function to build all key presses from the user
77     FUNCTION build_all_key_presses(BOOLEAN running, INTEGER mat_size, INTEGER mat_size_ref, INTEGER current_col, INTEGER current_col_ref, INTEGER current_row, INTEGER current_row_ref, STRING mat_size_ref, STRING current_col_ref, STRING current_row_ref) RETURNS STRING
78     {
79         IF event.key.type IS NOT EQUAL TO KEY_UP THEN
80             RETURN "";
81
82         IF event.key.value IS NOT EQUAL TO '0' THEN
83             SET running = FALSE;
84             RETURN "";
85         END IF;
86
87         IF event.key.value IS EQUAL TO 'q' THEN
88             SET running = FALSE;
89             RETURN "";
90         END IF;
91
92         IF event.key.value IS EQUAL TO '0' OR event.key.value IS EQUAL TO '.' THEN
93             SET current_col_size = MINIMUM OF (n, MAX_N);
94             SET current_col = 0;
95             SET current_row_size = MINIMUM OF (n, MAX_N);
96             SET current_row = 0;
97             SET current_determinant = 0;
98             SET current_input.strt = "Matrix ";
99             SET current_input.strc = "Determinant";
100            CALL determinant(OPMatrix(mat));
101            CALL window.render();
102
103            // Handle user input event
104            DECLARE tui.InputEvent event;
105            IF event.type IS EQUAL TO KEY_UP THEN
106                CALL handleKeyPress(event, running, mat, current_row, current_col, current_input.strt);
107            END IF;
108            IF event.type IS EQUAL TO KEY_DOWN THEN
109                CALL handleKeyPress(event, running, mat, current_row, current_col, current_input.strt);
110            END IF;
111            IF event.type IS EQUAL TO KEY_ENTER THEN
112                CALL handleKeyPress(event, running, mat, current_row, current_col, current_input.strt);
113            END IF;
114            IF event.type IS EQUAL TO KEY_ESC THEN
115                CALL handleKeyPress(event, running, mat, current_row, current_col, current_input.strt);
116            END IF;
117
118            // Function to build the string that gets displayed in the TUI
119            FUNCTION build_display_string(BOOLEAN running, INTEGER mat_size, INTEGER mat_size_ref, INTEGER current_col, INTEGER current_input.strt, STRING mat_size_ref, STRING current_col_ref) RETURNS STRING
120            {
121                DECLARE STRING display.strt = Matrix (size = CONVERT TO STRING n) + CONVERT TO STRING " x " + CONVERT TO STRING n;
122                FOR i FROM 0 TO n-1
123                {
124                    IF i == current_row THEN
125                        IF current_col == current_col_size THEN
126                            IF current_input.strt IS EQUAL TO current_col_size THEN
127                                SET display.strt = display.strt + " | ";
128                            ELSE
129                                SET display.strt = display.strt + CONVERT TO STRING mat[i][current_col];
130                            END IF;
131                            SET display.strt = display.strt + " | ";
132                        ELSE
133                            SET display.strt = display.strt + CONVERT TO STRING mat[i][current_col];
134                            SET display.strt = display.strt + " | ";
135                        END IF;
136                    ELSE
137                        SET display.strt = display.strt + " | ";
138                    END IF;
139                }
140                SET display.strt = display.strt + " | ";
141                SET display.strt = display.strt + "Determinant: " + CONVERT current_determinant TO STRING;
142                SET display.strt = display.strt + "\n\nUse arrow keys, hjkl, or wasd to navigate.\n";
143                SET display.strt = display.strt + "Press '-' to increase matrix size, '+' to decrease.\n";
144                SET display.strt = display.strt + "Press 'q' to quit.\n";
145            }
146            RETURN display.strt;
147        END FUNCTION;
148
149        // Function to handle all key presses from the user
150        FUNCTION handleKeyPress(tui.Event event, BOOLEAN running, INTEGER mat_size, INTEGER mat_size_ref, INTEGER current_col, INTEGER current_col_ref, INTEGER current_row, INTEGER current_row_ref, STRING mat_size_ref, STRING current_col_ref, STRING current_row_ref) RETURNS STRING
151        {
152            IF event.key.type IS EQUAL TO KEY_UP THEN
153                SET current_col_size = MINIMUM OF (n, MAX_N);
154                SET current_col = 0;
155                SET current_row_size = MINIMUM OF (n, MAX_N);
156                SET current_row = 0;
157                SET current_determinant = 0;
158                SET current_input.strt = "Matrix ";
159                SET current_input.strc = "Determinant";
160                CALL determinant(OPMatrix(mat));
161                CALL window.render();
162            END IF;
163
164            IF event.key.type IS EQUAL TO KEY_DOWN THEN
165                SET current_col_size = MINIMUM OF (n, MAX_N);
166                SET current_col = current_col_size;
167                SET current_row_size = MINIMUM OF (n, MAX_N);
168                SET current_row = current_row_size;
169                SET current_determinant = 0;
170                SET current_input.strt = "Matrix ";
171                SET current_input.strc = "Determinant";
172                CALL determinant(OPMatrix(mat));
173                CALL window.render();
174            END IF;
175
176            IF event.key.type IS EQUAL TO KEY_ENTER THEN
177                SET current_col_size = MINIMUM OF (n, MAX_N);
178                SET current_col = current_col_size;
179                SET current_row_size = MINIMUM OF (n, MAX_N);
180                SET current_row = current_row_size;
181                SET current_determinant = 0;
182                SET current_input.strt = "Matrix ";
183                SET current_input.strc = "Determinant";
184                CALL determinant(OPMatrix(mat));
185                CALL window.render();
186            END IF;
187
188            IF event.key.type IS EQUAL TO KEY_ESC THEN
189                SET current_col_size = MINIMUM OF (n, MAX_N);
190                SET current_col = current_col_size;
191                SET current_row_size = MINIMUM OF (n, MAX_N);
192                SET current_row = current_row_size;
193                SET current_determinant = 0;
194                SET current_input.strt = "Matrix ";
195                SET current_input.strc = "Determinant";
196                CALL determinant(OPMatrix(mat));
197                CALL window.render();
198            END IF;
199
200            IF event.key.type IS EQUAL TO KEY_BACKSPACE THEN
201                SET current_input.strt = current_input.strt + CONV

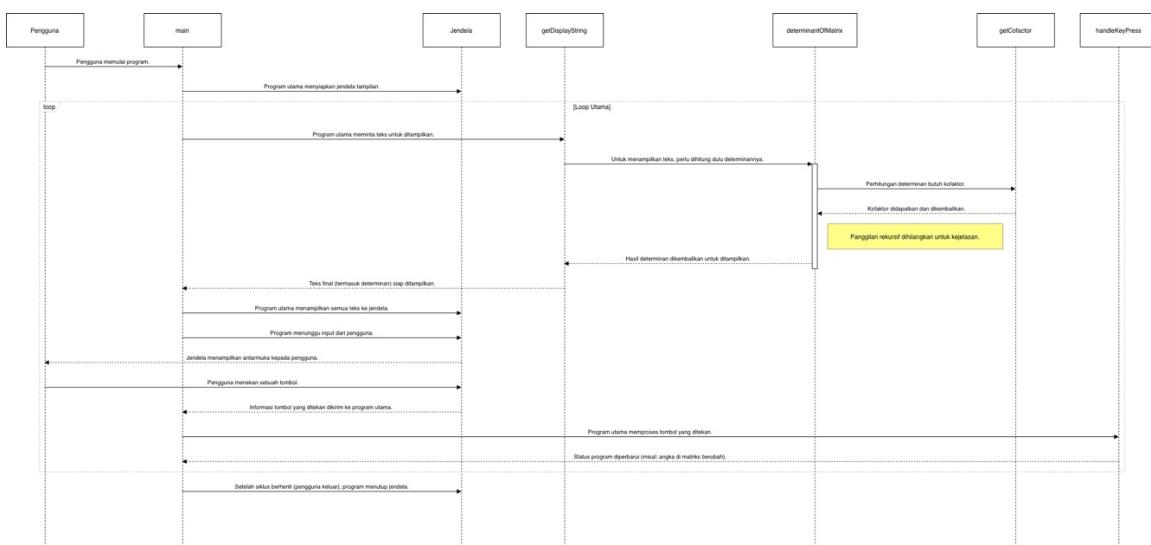
```

Gambar 3: Pseudocode Program Interactive Matrix Determinant Metode Kofaktor

Adapun algoritma dapat dijelaskan melalui flowchart dan sequence diagram



Gambar 4: Flowchart Program Interactive Matrix Determinant Metode Kofaktor



Gambar 5: Sequence Diagram Program Interactive Matrix Determinant Metode Kofaktor

Semua resource telah diunggah di <https://github.com/Rizkiacry/interactive-matrix-determinant>



Gambar 6: spoo.me/aljabar0001

DAFTAR PUSTAKA

claby2. (2020). *termui-cpp* (Initial commit). GitHub. <https://github.com/claby2/termui-cpp>

Determinan. (2017, November 28). <https://id.wikipedia.org/wiki/Determinan>