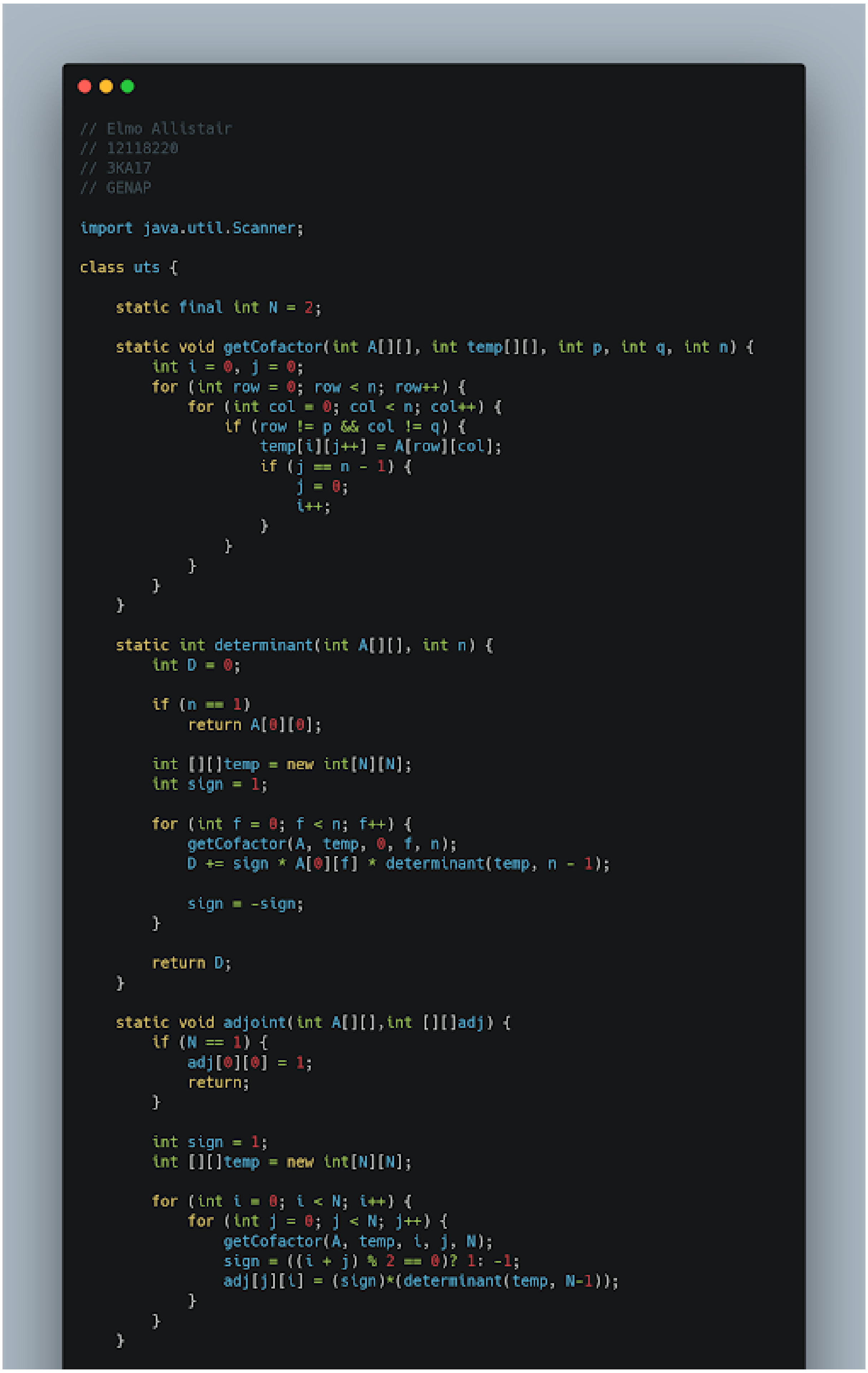
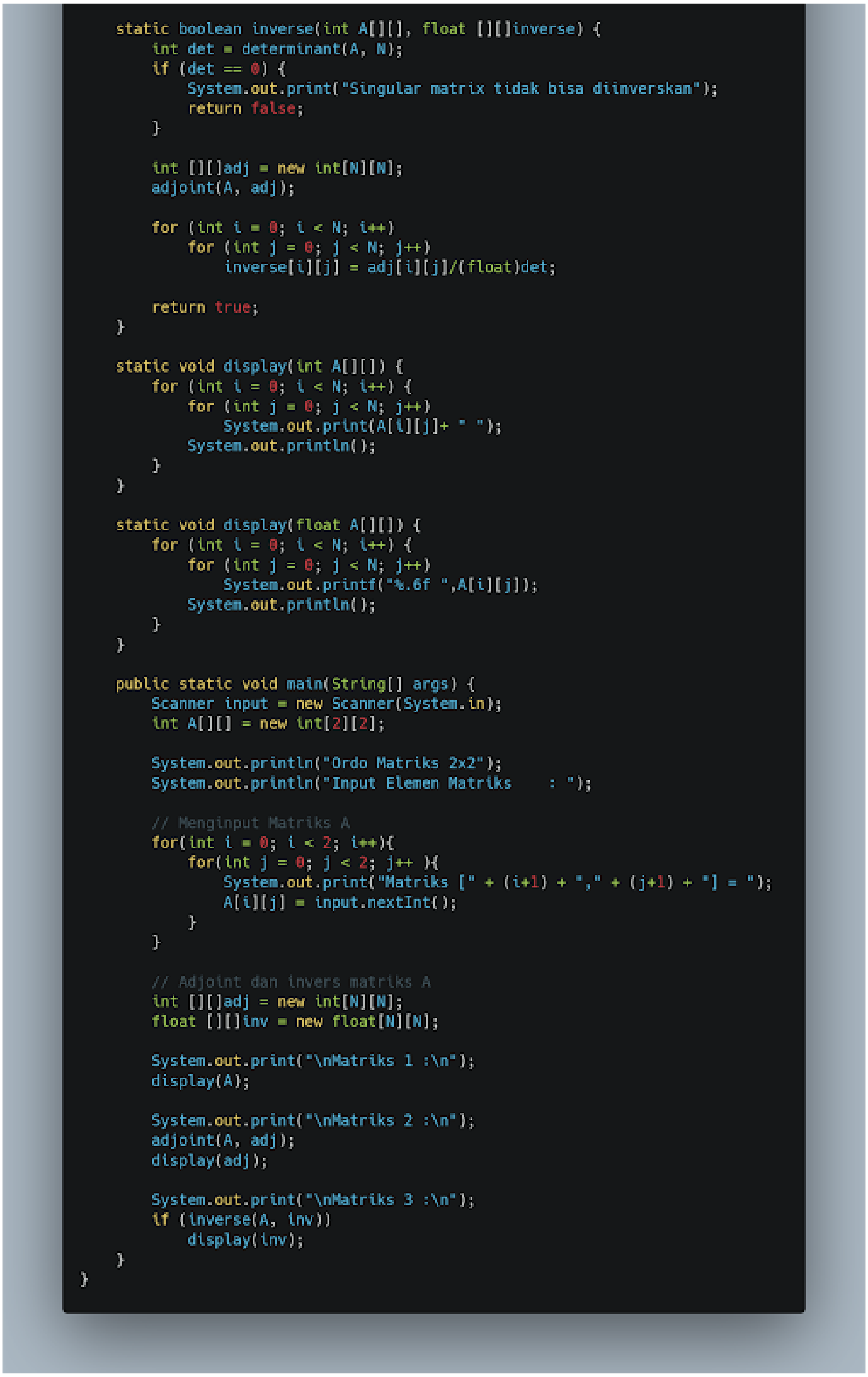
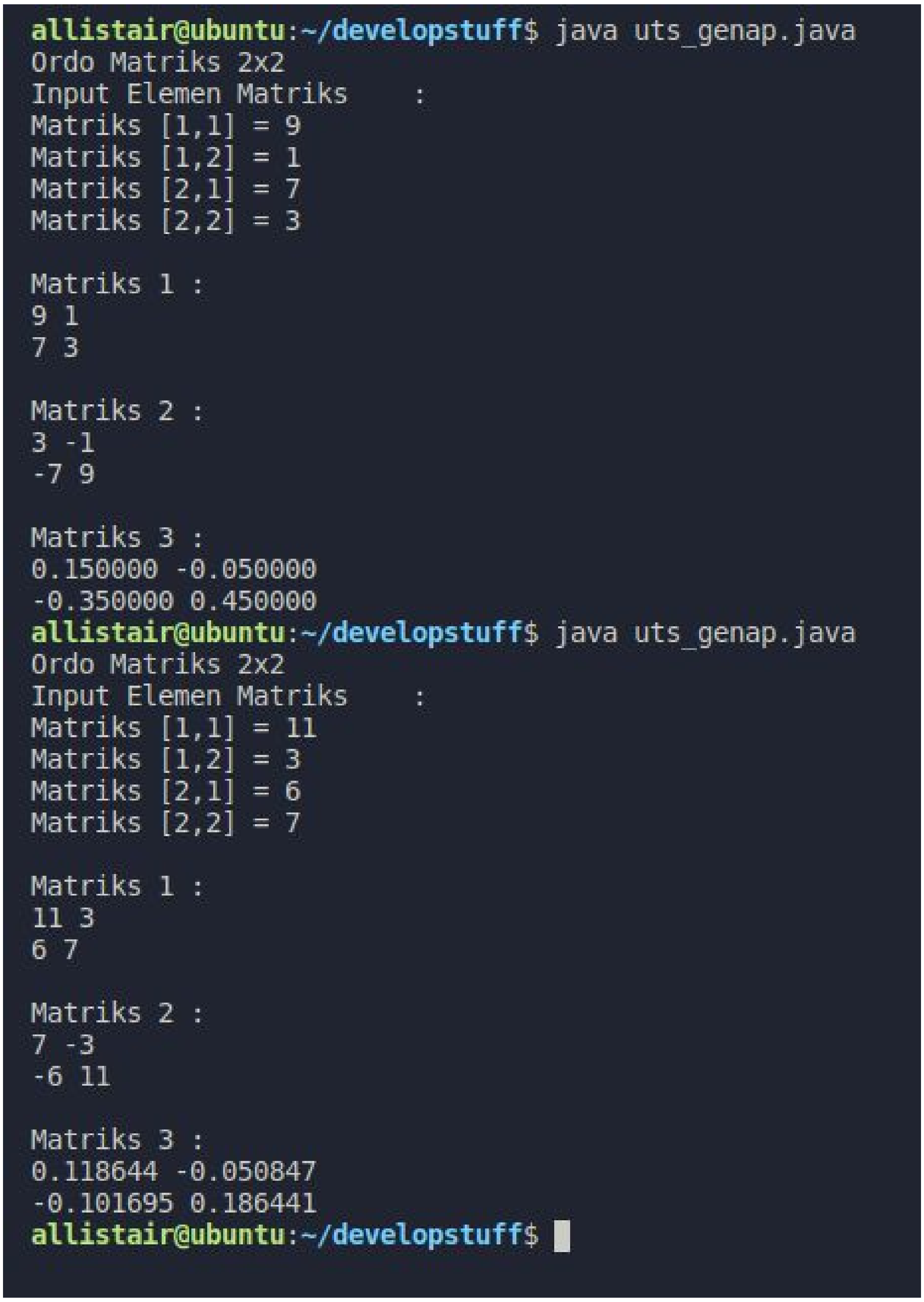


```java
// Elmo Allistair
// 12118220
// 3KA17
// GENAP

import java.util.Scanner;

class uts {

    static final int N = 2;

    static void getCofactor(int A[][], int temp[][], int p, int q, int n) {
        int i = 0, j = 0;
        for (int row = 0; row < n; row++) {
            for (int col = 0; col < n; col++) {
                if (row != p && col != q) {
                    temp[i][j++] = A[row][col];
                    if (j == n - 1) {
                        j = 0;
                        i++;
                    }
                }
            }
        }
    }

    static int determinant(int A[][], int n) {
        int D = 0;

        if (n == 1)
            return A[0][0];

        int [][]temp = new int[N][N];
        int sign = 1;

        for (int f = 0; f < n; f++) {
            getCofactor(A, temp, 0, f, n);
            D += sign * A[0][f] * determinant(temp, n - 1);

            sign = -sign;
        }

        return D;
    }

    static void adjoint(int A[][],int [][]adj) {
        if (N == 1) {
            adj[0][0] = 1;
            return;
        }

        int sign = 1;
        int [][]temp = new int[N][N];

        for (int i = 0; i < N; i++) {
            for (int j = 0; j < N; j++) {
                getCofactor(A, temp, i, j, N);
                sign = ((i + j) % 2 == 0)? 1: -1;
                adj[j][i] = (sign)*(determinant(temp, N-1));
            }
        }
    }
```



```java
    static boolean inverse(int A[][], float [][]inverse) {
        int det = determinant(A, N);
        if (det == 0) {
            System.out.print("Singular matrix tidak bisa diinverskan");
            return false;
        }

        int [][]adj = new int[N][N];
        adjoint(A, adj);

        for (int i = 0; i < N; i++)
            for (int j = 0; j < N; j++)
                inverse[i][j] = adj[i][j]/(float)det;

        return true;
    }

    static void display(int A[][]) {
        for (int i = 0; i < N; i++) {
            for (int j = 0; j < N; j++)
                System.out.print(A[i][j]+ " ");
            System.out.println();
        }
    }

    static void display(float A[][]) {
        for (int i = 0; i < N; i++) {
            for (int j = 0; j < N; j++)
                System.out.printf("%.6f ",A[i][j]);
            System.out.println();
        }
    }

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int A[][] = new int[2][2];

        System.out.println("Ordo Matriks 2x2");
        System.out.println("Input Elemen Matriks    : ");

        // Menginput Matriks A
        for(int i = 0; i < 2; i++){
            for(int j = 0; j < 2; j++ ){
                System.out.print("Matriks [" + (i+1) + "," + (j+1) + "] = ");
                A[i][j] = input.nextInt();
            }
        }

        // Adjoint dan invers matriks A
        int [][]adj = new int[N][N];
        float [][]inv = new float[N][N];

        System.out.print("\nMatriks 1 :\n");
        display(A);

        System.out.print("\nMatriks 2 :\n");
        adjoint(A, adj);
        display(adj);

        System.out.print("\nMatriks 3 :\n");
        if (inverse(A, inv))
            display(inv);
    }
}
```

Output



```
allistair@ubuntu:~/developstuff$ java uts_genap.java
Ordo Matriks 2x2
Input Elemen Matriks    :
Matriks [1,1] = 9
Matriks [1,2] = 1
Matriks [2,1] = 7
Matriks [2,2] = 3

Matriks 1 :
9 1
7 3

Matriks 2 :
3 -1
-7 9

Matriks 3 :
0.150000 -0.050000
-0.350000 0.450000
allistair@ubuntu:~/developstuff$ java uts_genap.java
Ordo Matriks 2x2
Input Elemen Matriks    :
Matriks [1,1] = 11
Matriks [1,2] = 3
Matriks [2,1] = 6
Matriks [2,2] = 7

Matriks 1 :
11 3
6 7

Matriks 2 :
7 -3
-6 11

Matriks 3 :
0.118644 -0.050847
-0.101695 0.186441
allistair@ubuntu:~/developstuff$
```