

Nama : Sufyan Abdur Rofiq
NIM : H1D022004
Shift : B => C
Matkul : Praktikum Pemrograman Mobile

KASUS OPERASI PEMJUMLAHAN DAN PERKALIAN MATRIKS

A. Source Code Program :

```
// Definisi class Matrix untuk melakukan operasi matriks
class Matrix {
    // Properti matriks yang disimpan dalam bentuk List 2D
    List<List<int>>> matrix;

    // Constructor untuk inisialisasi matriks
    Matrix(this.matrix);

    // Fungsi untuk menampilkan isi matriks
    void display() {
        for (var row in matrix) {
            print(row);
        }
    }

    // Fungsi untuk penjumlahan dua matriks
    Matrix add(Matrix other) {
        if (matrix.length != other.matrix.length || matrix[0].length != other.matrix[0].length) {
            throw Exception('Ukuran matriks tidak sama, penjumlahan tidak dapat dilakukan.');
```

```

        (i) => List.generate(matrix[0].length, (j) => matrix[i][j] + other.matrix[i][j]));

    return Matrix(result);
}

// Fungsi untuk perkalian dua matriks
Matrix multiply(Matrix other) {
    if (matrix[0].length != other.matrix.length) {
        throw Exception('Ukuran matriks tidak kompatibel untuk perkalian.');
```

```

    }

    int rows = matrix.length;
    int cols = other.matrix[0].length;
    int n = matrix[0].length;
```

```

    // Matriks hasil perkalian
    List<List<int>>> result = List.generate(rows, (i) => List.generate(cols, (j) => 0));
```

```

    // Melakukan perkalian matriks
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            for (int k = 0; k < n; k++) {
                result[i][j] += matrix[i][k] * other.matrix[k][j];
            }
        }
    }
```

```

    return Matrix(result);
}
}
```

```
// Fungsi utama untuk menjalankan program
void main() {
    // Inisialisasi dua matriks
    List<List<int>> matrix1Data = [
        [1, 2, 3],
        [4, 5, 6],
        [7, 8, 9]
    ];

    List<List<int>> matrix2Data = [
        [9, 8, 7],
        [6, 5, 4],
        [3, 2, 1]
    ];

    // Membuat objek matriks
    Matrix matrix1 = Matrix(matrix1Data);
    Matrix matrix2 = Matrix(matrix2Data);

    // Menampilkan matriks pertama
    print('Matriks Pertama:');
    matrix1.display();

    // Menampilkan matriks kedua
    print("\nMatriks Kedua:");
    matrix2.display();

    // Melakukan penjumlahan matriks
    Matrix sumMatrix = matrix1.add(matrix2);
```

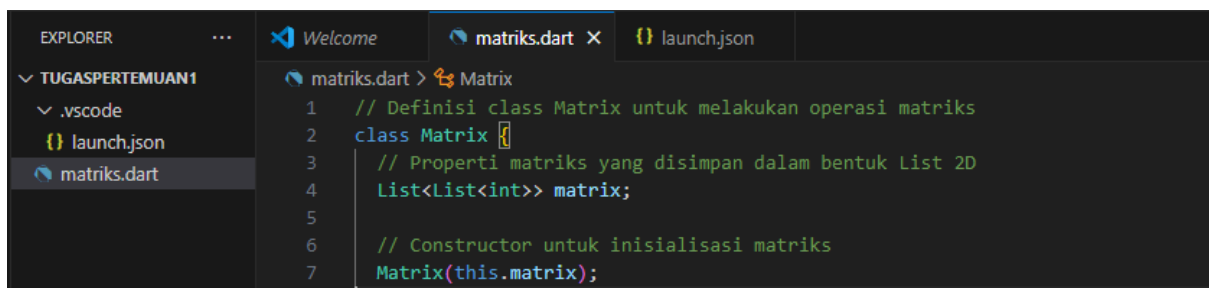
```
print("\nHasil Penjumlahan Matriks:");  
sumMatrix.display();
```

// Melakukan perkalian matriks

```
Matrix multipliedMatrix = matrix1.multiply(matrix2);  
print("\nHasil Perkalian Matriks:");  
multipliedMatrix.display();  
}
```

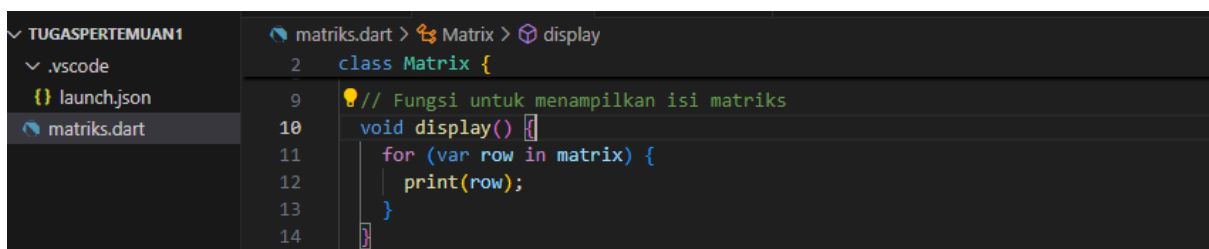
B. Penjelasan Code Program :

1. Definisi class Matrix :



- **class Matrix:** Membuat sebuah class bernama Matrix yang akan menyimpan matriks dalam bentuk List dua dimensi (List<List<int>>).
- **List<List<int>> matrix:** Properti matrix berupa list 2D yang berisi elemen-elemen bertipe integer.
- **Matrix(this.matrix):** Constructor untuk inisialisasi objek Matrix dengan memberikan nilai ke properti matrix.

2. Fungsi display :



- **void display():** Fungsi untuk menampilkan isi dari matriks ke layar.
- Menggunakan loop for untuk mengiterasi setiap baris (row) dalam matriks, kemudian mencetaknya menggunakan print(row). Baris tersebut berupa list yang mencakup elemen-elemen integer dalam matriks.

3. Fungsi add (Penjumlahan Matriks) :

```
matrks.dart > Matrix > add
2 class Matrix {
15
16 // Fungsi untuk penjumlahan dua matriks
17 Matrix add(Matrix other) {
18   if (matrix.length != other.matrix.length || matrix[0].length != other.matrix[0].length) {
19     throw Exception('Ukuran matriks tidak sama, penjumlahan tidak dapat dilakukan.');
```

- **Matrix add(Matrix other):** Fungsi untuk menjumlahkan dua matriks.
- **if:** Mengecek apakah ukuran kedua matriks (dari segi baris dan kolom) sama. Jika tidak, akan dilempar exception dengan pesan "Ukuran matriks tidak sama, penjumlahan tidak dapat dilakukan."
- **List.generate:** Digunakan untuk membuat list dua dimensi (result) dengan ukuran yang sama dengan matriks pertama (matrix), di mana setiap elemen matriks pertama dijumlahkan dengan elemen matriks kedua pada indeks yang sama.
- **return Matrix(result):** Setelah operasi penjumlahan selesai, dikembalikan objek Matrix baru yang berisi hasil penjumlahan.

4. Fungsi multiply (Perkalian Matriks) :

```
matrks.dart > Matrix > add
2 class Matrix {
28 // Fungsi untuk perkalian dua matriks
29 Matrix multiply(Matrix other) {
30   if (matrix[0].length != other.matrix.length) {
31     throw Exception('Ukuran matriks tidak kompatibel untuk perkalian.');
```

```
45       result[i][j] += matrix[i][k] * other.matrix[k][j];
46     }
47   }
48 }
49
50 return Matrix(result);
51 }
52 }
```

- **Matrix multiply(Matrix other):** Fungsi untuk melakukan perkalian antara dua matriks.
- **if:** Mengecek apakah jumlah kolom dari matriks pertama sama dengan jumlah baris dari matriks kedua. Jika tidak, akan dilempar exception dengan pesan "Ukuran matriks tidak kompatibel untuk perkalian."
- **int rows, cols, n:** Mendefinisikan beberapa variabel:

- rows: Jumlah baris dari matriks pertama.
- cols: Jumlah kolom dari matriks kedua.
- n: Jumlah kolom dari matriks pertama (atau jumlah baris dari matriks kedua).
- **List.generate**: Membuat list dua dimensi (result) untuk menyimpan hasil perkalian matriks, di mana semua elemen awalnya diisi dengan nol.
- **Loop bersarang (nested loop)**: Digunakan untuk melakukan perkalian matriks:
 - **i**: Untuk mengiterasi baris dari matriks pertama.
 - **j**: Untuk mengiterasi kolom dari matriks kedua.
 - **k**: Untuk melakukan perkalian elemen-elemen yang relevan dari matriks pertama dan matriks kedua, dan menambahkannya ke hasil pada posisi [i][j].
- **return Matrix(result)**: Setelah operasi perkalian selesai, dikembalikan objek Matrix baru yang berisi hasil perkalian.

5. Fungsi main :

```

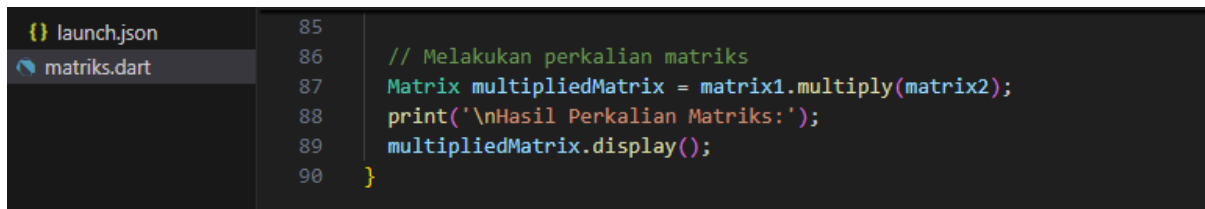
TUGASPERTEMUAN1
  .vscode
  {} launch.json
  matiks.dart
matiks.dart > Matrix > add
54 // Fungsi utama untuk menjalankan program
Run | Debug
55 void main() {
56 // Inisialisasi dua matriks
57 List<List<int>> matrix1Data = [
58   [1, 2, 3],
59   [4, 5, 6],
60   [7, 8, 9]
61 ];
62
63 List<List<int>> matrix2Data = [
64   [9, 8, 7],
65   [6, 5, 4],
66   [3, 2, 1]
67 ];
68

```

```

{} launch.json
matiks.dart
69 // Membuat objek matriks
70 Matrix matrix1 = Matrix(matrix1Data);
71 Matrix matrix2 = Matrix(matrix2Data);
72
73 // Menampilkan matriks pertama
74 print('Matriks Pertama:');
75 matrix1.display();
76
77 // Menampilkan matriks kedua
78 print('\nMatriks Kedua:');
79 matrix2.display();
80
81 // Melakukan penjumlahan matriks
82 Matrix sumMatrix = matrix1.add(matrix2);
83 print('\nHasil Penjumlahan Matriks:');
84 sumMatrix.display();
85

```



```
85
86 // Melakukan perkalian matriks
87 Matrix multipliedMatrix = matrix1.multiply(matrix2);
88 print('\nHasil Perkalian Matriks:');
89 multipliedMatrix.display();
90 }
```

- **main()**: Fungsi utama yang menjalankan program.
- **List<List<int>> matrix1Data**: Mendefinisikan matriks pertama dengan nilai-nilai tertentu.
- **List<List<int>> matrix2Data**: Mendefinisikan matriks kedua dengan nilai-nilai tertentu.
- **Matrix matrix1 = Matrix(matrix1Data)**: Membuat objek matrix1 dari data matrix1Data.
- **Matrix matrix2 = Matrix(matrix2Data)**: Membuat objek matrix2 dari data matrix2Data.
- **print('Matriks Pertama:')**: Mencetak judul "Matriks Pertama".
- **matrix1.display()**: Menampilkan matriks pertama.
- **print('\nMatriks Kedua:')**: Mencetak judul "Matriks Kedua".
- **matrix2.display()**: Menampilkan matriks kedua.
- **Matrix sumMatrix = matrix1.add(matrix2)**: Menjumlahkan matrix1 dan matrix2, kemudian menyimpan hasilnya dalam sumMatrix.
- **sumMatrix.display()**: Menampilkan hasil penjumlahan kedua matriks.
- **Matrix multipliedMatrix = matrix1.multiply(matrix2)**: Mengalikan matrix1 dan matrix2, kemudian menyimpan hasilnya dalam multipliedMatrix.
- **multipliedMatrix.display()**: Menampilkan hasil perkalian kedua matriks.

Kesimpulan:

Kode di atas mendefinisikan class Matrix dengan operasi penjumlahan dan perkalian matriks. Dengan menggunakan fungsi display(), add(), dan multiply(), kita dapat mengoperasikan dua matriks dan menampilkan hasilnya.

6. Output Program :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  Cod

[Running] dart "c:\Materi Kuliah\SEMESTER 5 A & B\Praktikum Pemrograman Mobile C\Tugaspertemuan1\matriks.dart"
Matriks Pertama:
[1, 2, 3]
[4, 5, 6]
[7, 8, 9]

Matriks Kedua:
[9, 8, 7]
[6, 5, 4]
[3, 2, 1]

Hasil Penjumlahan Matriks:
[10, 10, 10]
[10, 10, 10]
[10, 10, 10]

Hasil Perkalian Matriks:
[30, 24, 18]
[84, 69, 54]
[138, 114, 90]

[Done] exited with code=0 in 0.852 seconds

ter (Tugaspertemuan1)  Live Share  Ln 23, Col 52  Spaces: 2  UTF-8  CRLF  {} Dart  Go Live
```

7. Penjelasan Output Program :

a). Matriks Pertama :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  Code

[Running] dart "c:\Materi Kuliah\SEMESTER 5 A & B\Praktikum Pemrograman Mobile C\Tugaspertemuan1\matriks.dart"
Matriks Pertama:
[1, 2, 3]
[4, 5, 6]
[7, 8, 9]
```

Matriks pertama adalah matriks berukuran 3x3 yang terdiri dari tiga baris dan tiga kolom dengan elemen:

- Baris pertama: 1, 2, 3
- Baris kedua: 4, 5, 6
- Baris ketiga: 7, 8, 9

b). Matriks Kedua :

```
Matriks Kedua:
[9, 8, 7]
[6, 5, 4]
[3, 2, 1]
```

Matriks kedua juga berukuran 3x3 dengan elemen:

- Baris pertama: 9, 8, 7
- Baris kedua: 6, 5, 4
- Baris ketiga: 3, 2, 1

c). Hasil Penjumlahan Matriks :


```
Hasil Penjumlahan Matriks:  
[10, 10, 10]  
[10, 10, 10]  
[10, 10, 10]
```

Penjumlahan dilakukan elemen-per-elemen dari dua matriks:

- **Baris pertama:**

- $1 + 9 = 10$
- $2 + 8 = 10$
- $3 + 7 = 10$

- **Baris kedua:**

- $4 + 6 = 10$
- $5 + 5 = 10$
- $6 + 4 = 10$

- **Baris ketiga:**

- $7 + 3 = 10$
- $8 + 2 = 10$
- $9 + 1 = 10$

Hasil penjumlahan adalah matriks dengan semua elemen bernilai 10.

d). Hasil Perkalian Matriks :

```
Hasil Perkalian Matriks:  
[30, 24, 18]  
[84, 69, 54]  
[138, 114, 90]  
  
[Done] exited with code=0 in 0.852 seconds
```

- **Baris 1 dari Matriks Pertama dan Kolom 1 dari Matriks Kedua:**

Untuk menghitung elemen pertama di posisi $[0,0][0, 0][0,0]$, kita melakukan operasi berikut:

$$(1 \times 9) + (2 \times 6) + (3 \times 3) = 9 + 12 + 9 = 30$$

Hasilnya adalah **30**.

- **Baris 1 dari Matriks Pertama dan Kolom 2 dari Matriks Kedua:**

Untuk menghitung elemen kedua di posisi $[0,1][0, 1][0,1]$:

$$(1 \times 8) + (2 \times 5) + (3 \times 2) = 8 + 10 + 6 = 24$$

Hasilnya adalah **24**.

➤ **Baris 1 dari Matriks Pertama dan Kolom 3 dari Matriks Kedua:**

Untuk menghitung elemen ketiga di posisi $[0,2][0,2][0,2]$:

$$(1 \times 7) + (2 \times 4) + (3 \times 1) = 7 + 8 + 3 = 18$$

Hasilnya adalah **18**.

➤ **Baris 2 dari Matriks Pertama dan Kolom 1 dari Matriks Kedua:**

Untuk menghitung elemen pertama di baris kedua, posisi $[1,0][1,0][1,0]$:

$$(4 \times 9) + (5 \times 6) + (6 \times 3) = 36 + 30 + 18 = 84$$

Hasilnya adalah **84**.

➤ **Baris 2 dari Matriks Pertama dan Kolom 2 dari Matriks Kedua:**

Untuk menghitung elemen kedua di posisi $[1,1][1,1][1,1]$:

$$(4 \times 8) + (5 \times 5) + (6 \times 2) = 32 + 25 + 12 = 69$$

Hasilnya adalah **69**.

➤ **Baris 2 dari Matriks Pertama dan Kolom 3 dari Matriks Kedua:**

Untuk menghitung elemen ketiga di posisi $[1,2][1,2][1,2]$:

$$(4 \times 7) + (5 \times 4) + (6 \times 1) = 28 + 20 + 6 = 54$$

Hasilnya adalah **54**.

➤ **Baris 3 dari Matriks Pertama dan Kolom 1 dari Matriks Kedua:**

Untuk menghitung elemen pertama di baris ketiga, posisi $[2,0][2,0][2,0]$:

$$(7 \times 9) + (8 \times 6) + (9 \times 3) = 63 + 48 + 27 = 138$$

Hasilnya adalah **138**.

➤ **Baris 3 dari Matriks Pertama dan Kolom 2 dari Matriks Kedua:**

Untuk menghitung elemen kedua di posisi $[2,1][2,1][2,1]$:

$$(7 \times 8) + (8 \times 5) + (9 \times 2) = 56 + 40 + 18 = 114$$

Hasilnya adalah **114**.

➤ **Baris 3 dari Matriks Pertama dan Kolom 3 dari Matriks Kedua:**

Untuk menghitung elemen ketiga di posisi [2,2][2, 2][2,2]:

$$(7 \times 7) + (8 \times 4) + (9 \times 1) = 49 + 32 + 9 = 90$$

Hasilnya adalah **90**.

Kesimpulan:

- Untuk menghitung setiap elemen dari hasil perkalian matriks, kita mengalikan elemen-elemen dari **baris** matriks pertama dengan elemen-elemen dari **kolom** matriks kedua, kemudian menjumlahkan hasilnya.
- Setiap kombinasi baris dan kolom menghasilkan elemen baru di matriks hasil.
- Hasil akhir dari operasi ini adalah matriks 3x3 yang merepresentasikan perkalian dari kedua matriks.