Nama : Andera Singgih Pratama

NIM : 2211104007

Kelas : S1SE-06-01

1.

Source code

```dart
import 'dart:math';
import 'dart:io'; // Importing dart:io for stdout

void generateMatrix(int A, int B) {
  List<List<int>> matrix =
      List.generate(A, (_) => List.generate(B, (_) => Random().nextInt(10)));

  print('Matriks $A x $B:');
  for  var row in matrix) {
    print(row.join(' '));
  }

  print('\nHasil transpose:');
  for  int i = 0; i < B; i++) {
    for  int j = 0; j < A; j++) {
      // Print each element followed by a space
      stdout.write('${matrix[j][i]}  );
    }
    print(''); // Move to the next line after each row
  }
}

void main() {
  generateMatrix(3, 2); // Example call
}
```

Output

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
Connecting to VM Service at ws://127.0.0.1:50054/rRUzQldp5eA=/ws
Connected to the VM Service.
Matriks 3 x 2:
3 1
6 5
6 4

Hasil transpose:
3 6 6
1 5 4

Exited.
```

Program Dart ini berfungsi untuk menghasilkan dan menampilkan matriks dua dimensi, serta menampilkan hasil transposisinya.

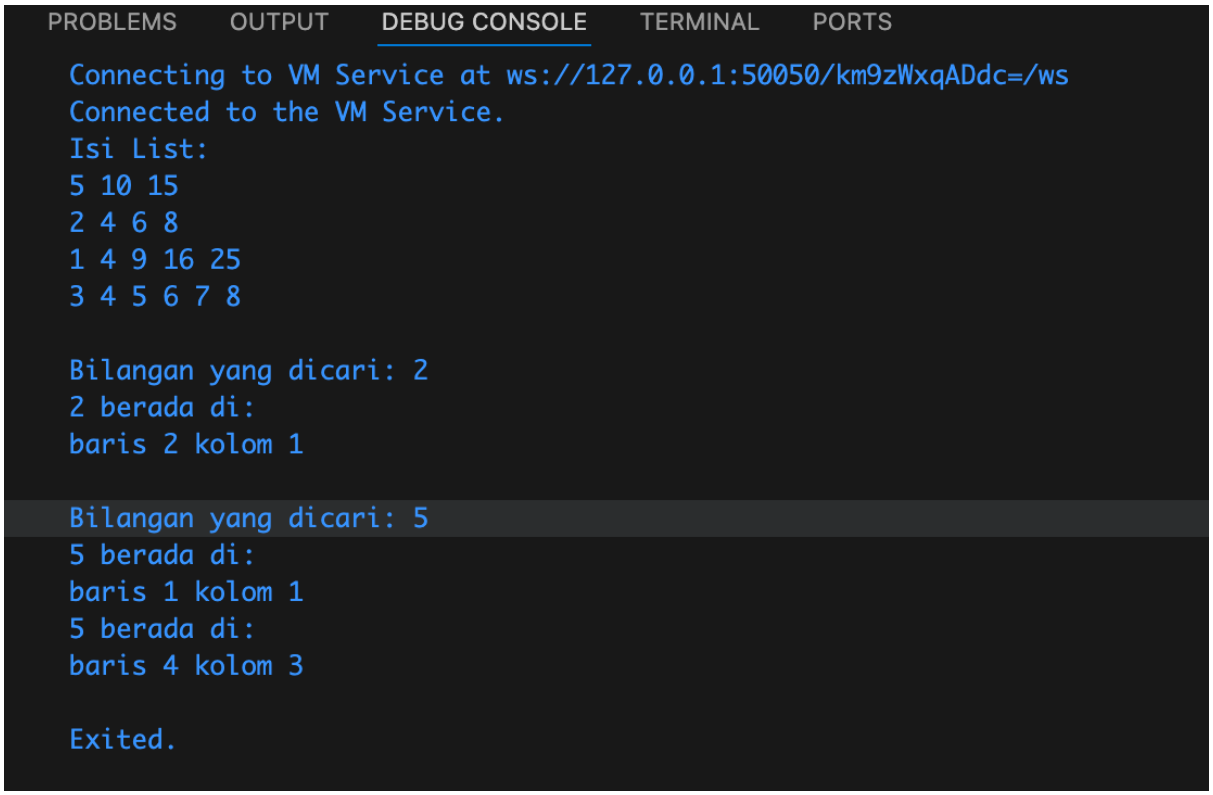
1. **Import Library**: Program mengimpor pustaka dart:math untuk menghasilkan angka acak dan dart:io untuk menggunakan fungsi input/output.
2. **Fungsi generateMatrix**:
    - Menerima dua parameter, yaitu jumlah baris (A) dan jumlah kolom (B).
    - Membuat matriks dengan ukuran A x B, diisi dengan angka acak antara 0 hingga 9.
    - Menampilkan matriks yang telah dibuat dengan memisahkan elemen pada setiap baris menggunakan spasi.
3. **Transposisi Matriks**:
    - Setelah menampilkan matriks, program melakukan transposisi, yaitu menukar baris dan kolom.
    - Hasil transpose ditampilkan di mana setiap elemen ditampilkan berdasarkan posisi baru.
4. **Fungsi main**:
    - Merupakan titik awal program yang memanggil fungsi generateMatrix dengan parameter 3 untuk baris dan 2 untuk kolom, sehingga menghasilkan matriks berukuran 3x2.

2.

Source code

```
void main() {
  // Create the 2D list
  List<List<int>> matrix = createMatrix();

  // Display the list
  printMatrix(matrix);

  // Search for a number (you can change this to test with other values)
  int numberToSearch = 2; // Example number
  searchInMatrix(matrix, numberToSearch);

  numberToSearch = 5; // Another example number
  searchInMatrix(matrix, numberToSearch);
}

List<List<int>> createMatrix() {
  List<List<int>> matrix =
                          [];
  // Row 1: 3 numbers that are multiples of 5 starting from 5
  matrix.add([for  int i = 1; i <= 3; i++) 5 * i]);
                  (
  // Row 2: 4 even numbers starting from 2
  matrix.add([for  int i = 1; i <= 4; i++) 2 * i]);
                  (
  // Row 3: 5 squares of natural numbers starting from 1
  matrix.add([for  int i = 1; i <= 5; i++) i * i]);
                  (
  // Row 4: 6 natural numbers starting from 3
  matrix.add([for  int i = 0; i < 6; i++) 3 + i]);
                  (
  return matrix;
}

void printMatrix(List<List<int>> matrix) {
  print('Isi List:');
  for  var row in matrix) {
    print(row.join(' '));
  }
}

void searchInMatrix(List<List<int>> matrix, int number) {
  print('\nBilangan yang dicari: $number');

  bool found = false;

  for  int i = 0; i < matrix.length; i++) {
    for  int j = 0; j < matrix[i].length; j++) {
      if  matrix[i][j] == number) {
        found = true;
        print('$number berada d   );
        print('baris ${i'+ 1} kolom ${j + 1}'); // Using 1-based index

}   }
  }

  if  !found) {
    print('$number tidak ditemukan.');
  }
}
```

Output



Program ini bertujuan untuk membuat, menampilkan, dan mencari angka dalam matriks dua dimensi.

1. **Fungsi main**:
    - Ini adalah titik awal program.
    - Memanggil fungsi untuk membuat matriks dan menyimpannya dalam variabel.
    - Menampilkan isi matriks.
    - Melakukan pencarian angka dalam matriks menggunakan dua contoh angka.
2. **Fungsi createMatrix**:
    - Membuat matriks dengan 4 baris.
    - Baris pertama berisi 3 angka kelipatan 5 (5, 10, 15).
    - Baris kedua berisi 4 angka genap (2, 4, 6, 8).
    - Baris ketiga berisi 5 angka kuadrat dari 1 hingga 5 (1, 4, 9, 16, 25).
    - Baris keempat berisi 6 angka asli mulai dari 3 (3, 4, 5, 6, 7, 8).
3. **Fungsi printMatrix**:
    - Menampilkan isi matriks dengan format yang rapi, memisahkan angka dalam setiap baris dengan spasi.

4. **Fungsi searchInMatrix**:
    - Mencari angka tertentu dalam matriks.
    - Menampilkan lokasi angka jika ditemukan, menggunakan indeks berbasis 1 untuk baris dan kolom.
    - Jika angka tidak ditemukan, mencetak pesan yang sesuai.

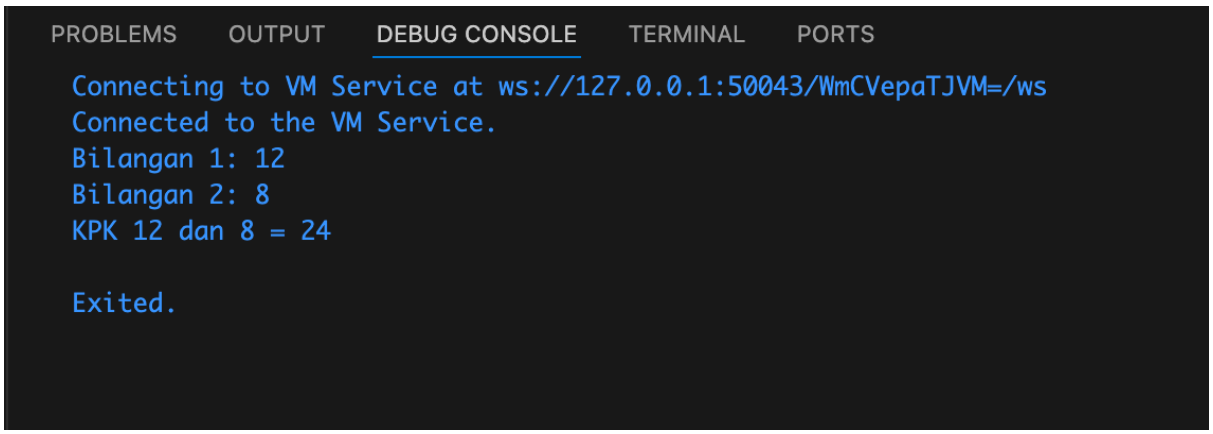
Secara keseluruhan, program ini menunjukkan cara menangani matriks dalam Dart, termasuk penciptaan, penampilan, dan pencarian elemen.

3.

Source code

```dart
void main() {
  int num1 = 12; // Example number 1
  int num2 = 8; // Example number 2

  print('Bilangan 1: $num1');
  print('Bilangan 2: $num2');

  int lcm = calculateLCM(num1, num2);
  print('KPK $num1 dan $num2 = $lcm');
}

int calculateGCD(int a, int b) {
  while  b != 0) {
    int(temp = b;
    b = a % b;
    a = temp;
  }
  return a;
}

int calculateLCM(int a, int b) {
  return  a * b) ~/ calculateGCD(a, b); // LCM = (a * b) / GCD(a, b)
}       (
```

Output



Program ini digunakan untuk menghitung dan menampilkan Kelipatan Persekutuan Terkecil (KPK) dari dua bilangan bulat.

1. **Fungsi `main`**:

- Mendeklarasikan dua angka, `num1` dan `num2`, dengan nilai 12 dan 8.

- Menampilkan kedua angka tersebut di layar.

- Menghitung KPK dari kedua angka dengan memanggil fungsi `calculateLCM`.

- Menampilkan hasil KPK di layar.

2. **Fungsi `calculateGCD`**:

- Menghitung Bilangan Persekutuan Terbesar (GCD) dari dua bilangan.

- Menggunakan algoritma Euclid, di mana proses pembagian dilakukan berulang hingga sisa pembagian menjadi nol.

- Mengembalikan nilai GCD.

3. **Fungsi `calculateLCM`**:

- Menghitung KPK dengan menggunakan rumus:

\[

\text{KPK}(a, b) = \frac{a \times b}{\text{GCD}(a, b)}

\]

- Mengalikan kedua bilangan dan membagi hasilnya dengan GCD untuk mendapatkan KPK.

Secara keseluruhan, program ini menunjukkan cara untuk menghitung KPK dari dua bilangan dengan memanfaatkan GCD sebagai langkah perantara.