

## Larik Berdimensi

### A. Pengenalan

Larik 1 Dimensi adalah struktur data yang terdiri dari kumpulan elemen yang disusun secara linear dalam satu baris. Setiap elemen dalam larik 1D diidentifikasi menggunakan indeks yang dimulai dari 0 hingga panjang larik dikurangi 1.

Larik 2 Dimensi adalah struktur data yang terdiri dari kumpulan elemen yang disusun dalam baris dan kolom. Setiap elemen dalam larik 2D diidentifikasi menggunakan dua indeks, yaitu indeks baris dan indeks kolom.

Larik 3 Dimensi adalah struktur data yang terdiri dari kumpulan elemen yang disusun dalam baris, kolom, dan kedalaman. Setiap elemen dalam larik 3D diidentifikasi menggunakan tiga indeks, yaitu indeks baris, indeks kolom, dan indeks kedalaman. Semua elemen dalam larik pasti memiliki tipe data yang sama.

### B. Kelebihan juga Kekurangan serta Fungsi

Larik 1 dimensi memiliki beberapa kelebihan yang membuatnya menjadi struktur data yang sederhana dan mudah digunakan. Larik 1 dimensi sangat efisien dalam penyimpanan dan akses elemen, sehingga operasi seperti pencarian atau pengubahan nilai dapat dilakukan dengan cepat. Larik 1 dimensi umumnya digunakan untuk menyimpan data yang memiliki urutan linear, seperti deret bilangan, daftar nama, atau data yang diambil secara sekuensial. Namun, larik 1 dimensi memiliki keterbatasan dalam merepresentasikan data dengan struktur yang lebih kompleks. Misalnya, sulit untuk mengatasi data dengan dimensi yang lebih tinggi atau hubungan yang lebih kompleks antara elemen-elemennya.

Larik 2 dimensi memberikan kelebihan tambahan dengan memperluas dimensi menjadi baris dan kolom. Ini memungkinkan representasi data yang lebih kompleks, seperti tabel, matriks, gambar, atau peta. Larik 2 dimensi sangat berguna dalam manipulasi data yang melibatkan perhitungan baris dan kolom, seperti pengolahan gambar atau analisis data spasial. Namun, larik 2 dimensi juga memiliki kekurangan, seperti membutuhkan penanganan yang lebih kompleks dalam akses elemen dengan memerlukan dua indeks. Selain itu, jumlah elemen dalam larik 2 dimensi harus seragam di setiap baris dan kolom, sehingga tidak efektif untuk menyimpan data yang memiliki jumlah atau jenis elemen yang bervariasi.

Larik 3 dimensi memperluas lagi dimensi dengan menambahkan kedalaman. Hal ini memungkinkan representasi data yang lebih kompleks, seperti citra 3D, model geometri, atau data medis yang memiliki komponen tiga dimensi. Larik 3 dimensi memberikan fleksibilitas lebih dalam menyimpan dan memanipulasi data yang memiliki struktur yang lebih kompleks. Namun, kelemahan larik 3 dimensi terletak pada kompleksitas akses dan manipulasi data yang memerlukan tiga indeks. Selain itu, larik 3 dimensi juga cenderung mengkonsumsi lebih banyak ruang memori dan sumber daya komputasi daripada larik dengan dimensi yang lebih rendah.

Larik 1 dimensi, 2 dimensi, dan 3 dimensi digunakan untuk berbagai tujuan. Mereka digunakan untuk menyimpan dan mengolah data dalam bentuk struktur yang terorganisir, seperti matriks, tabel, atau model geometri. Larik multidimensi juga digunakan dalam perhitungan matematis dan manipulasi data yang melibatkan dimensi dan hubungan antar elemen. Implementasi larik multidimensi dapat ditemukan dalam berbagai bidang seperti grafika komputer, pemrosesan citra, simulasi fisika, analisis data, dan banyak lagi.

### C. Contoh Implementasi

```
import java.util.Scanner;

public class MatrixAddition {

    public static void main(String[] args) {
        int[][] matrixA = new int[3][3];
        int[][] matrixB = new int[3][3];
        int[][] matrixC = new int[3][3];

        // Meminta input dari pengguna untuk matriks A
        System.out.println("Masukkan elemen-elemen matriks A (3x3):");
        matrixA = getInputMatrix();

        // Meminta input dari pengguna untuk matriks B
        System.out.println("Masukkan elemen-elemen matriks B (3x3):");
        matrixB = getInputMatrix();

        // Menjumlahkan matriks A dan B
        for (int i = 0; i < matrixA.length; i++) {
            for (int j = 0; j < matrixA[i].length; j++) {
                matrixC[i][j] = matrixA[i][j] + matrixB[i][j];
            }
        }

        // Menampilkan hasil penjumlahan
        System.out.println("Hasil penjumlahan matriks A dan B:");
        displayMatrix(matrixC);
    }

    // Fungsi untuk meminta input matriks dari pengguna
    public static int[][] getInputMatrix() {
        int[][] matrix = new int[3][3];
        Scanner scanner = new Scanner(System.in);

        for (int i = 0; i < matrix.length; i++) {
            for (int j = 0; j < matrix[i].length; j++) {
                System.out.print("Masukkan elemen matriks [" + i + "][" + j + "]:");
                matrix[i][j] = scanner.nextInt();
            }
        }

        return matrix;
    }

    // Fungsi untuk menampilkan matriks
    public static void displayMatrix(int[][] matrix) {
        for (int i = 0; i < matrix.length; i++) {
            for (int j = 0; j < matrix[i].length; j++) {
                System.out.print(matrix[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

D. Studi Kasus

Boleh lakukan modifikasi pada program, yaitu perhitungan matriks menggunakan menu atau apapun itu, atau membuat aplikasi baru operasi matematika yang juga menggunakan larik berdimensi