

Nama : Birgita Cleosa Chatriona Agriani

NIM : A11.2023.15315

Mata Kuliah : PBO

## Penjelasan Soal

### Nomor 1 (FAKTORIAL)

```
1 package faktorial;
2
3 import java.util.Scanner;
4
5 class Faktorial {
6     public int hitungFaktorial(int n) {
7         if (n == 0 || n == 1) {
8             return 1;
9         }
10        return n * hitungFaktorial(n - 1);
11    }
12 }
```

Fungsi : Menghitung faktorial dari suatu bilangan menggunakan rekursi.

```
14 public class FaktorialImplementasi {
15     public static void main(String[] args) {
16         Scanner scanner = new Scanner(System.in);
17         Faktorial faktorial = new Faktorial();
18         char choice;
19     }
```

Fungsi : Membuat objek Scanner untuk membaca input dari pengguna, membuat objek Faktorial agar bisa menggunakan metode hitungFaktorial()

```
16     do {
17         System.out.print("Masukkan angka untuk menghitung faktorial: ");
18         int number = scanner.nextInt();
19
20         if (number < 0) {
21             System.out.println("Faktorial tidak terdefinisi untuk bilangan negatif.");
22         } else {
23             int result = faktorial.hitungFaktorial(number);
24             System.out.println("Faktorial dari " + number + " adalah " + result);
25         }
26
27         System.out.print("Apakah anda ingin melanjutkan? (y/n): ");
28         choice = scanner.next().charAt(0);
29     } while (choice == 'y' || choice == 'Y');
30
31     scanner.close();
32 }
```

### NOMOR 2 (ARRAY)

```
1 public class ArrayListDemo {
2     public static void main(String[] args) {
3         Scanner scanner = new Scanner(System.in);
4
5         System.out.print("Masukkan jumlah data: ");
6         int jumlahData = scanner.nextInt();
7     }
```

- Public class ArrayListDemo untuk nama kelas utama
- Untuk meminta pengguna memasukkan jumlah elemen yang dimasukkan dalam ArrayList

```
13 ArrayList<Integer> nilai1 = new ArrayList<>();
14 ArrayList<Integer> nilai2 = new ArrayList<>();
15 ArrayList<Integer> jumlah = new ArrayList<>();
16 ArrayList<Integer> kurang = new ArrayList<>();
17 ArrayList<Integer> kali = new ArrayList<>();
18 ArrayList<Integer> bagi = new ArrayList<>();
19 }
```

- Nilai1 & nilai2 untuk menyimpan bilangan dari input pengguna
- Jumlah untuk menyimpan hasil penjumlahan nilai1 dan nilai2
- Kurang untuk menyimpan hasil pengurangan nilai1 dan nilai2
- Kali untuk menyimpan hasil perkalian nilai1 dan nilai2
- Bagi untuk menyimpan hasil pembagian nilai1 dan nilai2

```

23     System.out.println("Masukkan nilai untuk arraylist 1");
24     for (int i = 0; i < jumlahData; i++) {
25         System.out.print("Nilai ke-" + i + " = ");
26         nilai1.add(scanner.nextInt());
27     }
28
29     System.out.println("Masukkan nilai untuk arraylist 2");
30     for (int i = 0; i < jumlahData; i++) {
31         System.out.print("Nilai ke-" + i + " = ");
32         nilai2.add(scanner.nextInt());
33     }

```

- Mengisi nilai1 dan nilai2 dengan angka yang dimasukkan oleh pengguna

```

34     for (int i = 0; i < jumlahData; i++) {
35         jumlah.add(nilai1.get(i) + nilai2.get(i));
36         kurang.add(nilai1.get(i) - nilai2.get(i));
37         kali.add(nilai1.get(i) * nilai2.get(i));
38         if (nilai2.get(i) != 0) {
39             bagi.add((double) nilai1.get(i) / nilai2.get(i));
40         } else {
41             bagi.add(Double.NaN); // menandai pembagian oleh nol
42         }
43     }

```

- Menjumlahkan, mengurangi, dan mengalihkan elemen-elemen nilai1 dan nilai2
- Membagi nilai1 dengan nilai2 tetapi jika nilai2 == 0, hasilnya NaN (Not a Number)

```

44     System.out.println("Hasil operasi:");
45     for (int i = 0; i < jumlahData; i++) {
46         System.out.print("Index ke-" + i + " : jumlah: " + jumlah.get(i) +
47             ", kurang: " + kurang.get(i) +
48             ", kali: " + kali.get(i) +
49             ", bagi: " + (nilai2.get(i) != 0 ? bagi.get(i) : "Tidak bisa dibagi oleh 0"));
50     }
51     scanner.close();

```

- Menampilkan hasil operasi matematika untuk setiap indeks

## TUGAS SOAL 1

```

1     System.out.println("Jumlah Data: ");
2     int jumlahData = scanner.nextInt();
3     int[] dataArray = new int[jumlahData];
4
5     for (int i = 0; i < jumlahData; i++) {
6         System.out.print("Data ke-" + i + " = ");
7         dataArray[i] = scanner.nextInt();
8     }
9
10    System.out.println("Masukan nilai:");
11    for (int i = 0; i < jumlahData; i++) {
12        System.out.print("Masukan nilai ke-" + i + " = ");
13        dataArray[i] = scanner.nextInt();
14    }
15
16    System.out.println("Hasil operasi:");
17    scanner.close();

```

- Membuat array dataArray dengan ukuran sesuai input pengguna
- Looping digunakan untuk mengisi array dengan nilai yang dimasukkan oleh pengguna
- Untuk menampilkan permintaan input di setiap indeks array

## TUGAS SOAL 2

```

18 // Input data
19 for (int i = 0; i < jumlah; i++) {
20     System.out.print("Masukkan nama mahasiswa ke-" + (i + 1) + ": ");
21     String nama = input.nextLine();
22     System.out.print("Masukkan nilai mahasiswa ke-" + (i + 1) + ": ");
23     int nilai = input.nextInt();
24     input.nextLine(); // konsumsi newline
25
26     daftarMahasiswa[i] = new Mahasiswa(nama, nilai);
27 }
28
29 // Tampilkan daftar nilai
30 daftarNilai(daftarMahasiswa);
31 input.close();
32
33 // Method untuk menampilkan daftar mahasiswa dan nilainya
34 static void daftarNilai(Mahasiswa[] daftar) {
35     System.out.println("===== Daftar Nilai Mahasiswa =====");
36     for (Mahasiswa mhs : daftar) {
37         mhs.tampilkan();
38     }
39 }

```

- Looping for mengulang sebanyak jumlah mahasiswa
- String nama untuk membaca input nama mahasiswa
- Int nilai untuk membaca nilai mahasiswa
- Memanggil method daftarNilai() untuk menampilkan daftar mahasiswa.
- input.close(); → Menutup Scanner untuk menghindari kebocoran memori.
- static void daftarNilai(Mahasiswa[] daftar) → Method untuk mencetak semua data mahasiswa.
- for (Mahasiswa mhs : daftar) → Looping for-each untuk menampilkan data setiap mahasiswa dalam array.
- mhs.tampilkan(); → Memanggil method tampilkan() dari objek Mahasiswa untuk menampilkan data.

## TUGAS SOAL 3

```

1 // fungsi untuk mencetak array
2 static void printArray(int[] arr) {
3     for (int num : arr) {
4         System.out.print(num + " ");
5     }
6     System.out.println();
7 }
8
9 // bubble sort
10 static void bubbleSort(int[] arr) {
11     int n = arr.length;
12     for (int i = 0; i < n - 1; i++) {
13         for (int j = 0; j < n - i - 1; j++) {
14             if (arr[j] > arr[j + 1]) {
15                 int temp = arr[j];
16                 arr[j] = arr[j + 1];
17                 arr[j + 1] = temp;
18             }
19         }
20         printArray(arr);
21     }
22 }

```

- Method ini digunakan untuk mencetak isi array.
- Dipanggil setelah setiap perubahan array dalam sorting.
- Bubble Sort membandingkan elemen bersebelahan dan menukar jika salah urutan.
- Loop luar mengontrol jumlah iterasi.
- Loop dalam melakukan pertukaran elemen.
- Setiap perubahan array langsung ditampilkan (printArray(arr);).

```

30 // Quick Sort
31 static void quickSort(int[] arr, int low, int high) {
32     if (low < high) {
33         int pi = partition(arr, low, high);
34         quickSort(arr, low, pi - 1);
35         quickSort(arr, pi + 1, high);
36     }
37 }
38
39 static int partition(int[] arr, int low, int high) {
40     int pivot = arr[high];
41     int i = (low - 1);
42     for (int j = low; j < high; j++) {
43         if (arr[j] < pivot) {
44             i++;
45             int temp = arr[j];
46             arr[j] = arr[i];
47             arr[i] = temp;
48         }
49     }
50     int temp = arr[i + 1];
51     arr[i + 1] = arr[high];
52     arr[high] = temp;
53     return i + 1;
54 }

```

- Quick Sort menggunakan pivot untuk membagi array menjadi dua bagian.
- Memanggil partition() untuk menempatkan pivot pada posisi yang benar.
- Rekursi dilakukan pada subarray kiri dan kanan.
- Fungsi partition() menempatkan pivot ke posisi yang benar.

```

30 // Insertion Sort
31 static void insertionSort(int[] arr) {
32     int n = arr.length;
33     for (int i = 1; i < n; i++) {
34         int key = arr[i];
35         int j = i - 1;
36         while (j >= 0 && arr[j] > key) {
37             arr[j + 1] = arr[j];
38             j--;
39         }
40         arr[j + 1] = key;
41         printArray(arr);
42     }
43 }

```

- Insertion Sort menyisipkan elemen pada posisi yang benar.
- Loop luar berjalan dari indeks 1 ke n.
- Loop dalam menggeser elemen yang lebih besar ke kanan.
- Menampilkan array setiap perubahan.

```

30 // Selection Sort
31 static void selectionSort(int[] arr) {
32     int n = arr.length;
33     for (int i = 0; i < n - 1; i++) {
34         int minIndex = i;
35         for (int j = i + 1; j < n; j++) {
36             if (arr[j] < arr[minIndex]) {
37                 minIndex = j;
38             }
39         }
40         int temp = arr[minIndex];
41         arr[minIndex] = arr[i];
42         arr[i] = temp;
43         printArray(arr);
44     }
45 }

```

- Selection Sort mencari nilai terkecil dalam sisa array.
- Elemen terkecil ditukar dengan elemen pertama di subarray.
- Menampilkan array setiap perubahan.

```

30 // Merge Sort
31 static void mergeSort(int[] arr, int l, int r) {
32     if (l < r) {
33         int m = (l + r) / 2;
34         mergeSort(arr, l, m);
35         mergeSort(arr, m + 1, r);
36         merge(arr, l, m, r);
37         printArray(arr);
38     }
39 }

```

- Merge Sort membagi array menjadi dua bagian, mengurutkan, lalu menggabungkan.
- merge() menggabungkan dua bagian yang telah diurutkan.

```

100 static void merge(int[] arr, int l, int m, int r) {
101     int n1 = m - l + 1;
102     int n2 = r - m;
103     int[] L = new int[n1];
104     int[] R = new int[n2];
105     for (int i = l; i < m; i++) L[i] = arr[i + l];
106     for (int i = m; i < r; i++) R[i] = arr[i + 1 + l];
107     int i = 0, j = 0, k = l;
108     while (i < n1 && j < n2) {
109         if (L[i] <= R[j]) arr[k++] = L[i++];
110         else arr[k++] = R[j++];
111     }
112     while (i < n1) arr[k++] = L[i++];
113     while (j < n2) arr[k++] = R[j++];
114 }
115
116

```

- Fungsi merge() menggabungkan dua subarray yang sudah terurut.

```

117 public static void main(String[] args) {
118     Scanner scanner = new Scanner(System.in);
119     System.out.println("Masukkan angka (dipisahkan spasi): ");
120     String[] input = scanner.nextLine().split(" ");
121     int[] arr = new int[input.length];
122     for (int i = 0; i < input.length; i++) {
123         arr[i] = Integer.parseInt(input[i]);
124     }
125 }

```

- Membaca input angka dari pengguna dalam bentuk string.
- Mengubah input string ke array integer.

```

126 System.out.println("Pilih metode sorting:");
127 System.out.println("1. Bubble sort");
128 System.out.println("2. Quick sort");
129 System.out.println("3. Insertion sort");
130 System.out.println("4. Selection sort");
131 System.out.println("5. Merge sort");
132 System.out.print("Pilihan: ");
133 int choice = scanner.nextInt();
134

```

- Menampilkan pilihan metode sorting.
- Membaca pilihan pengguna.

```

135 System.out.println("Aturases sorting:");
136 switch (choice) {
137     case 1 -> bubbleSort(arr);
138     case 2 -> quickSort(arr, 0, arr.length - 1);
139     case 3 -> insertionSort(arr);
140     case 4 -> selectionSort(arr);
141     case 5 -> mergeSort(arr, 0, arr.length - 1);
142     default -> System.out.println("Pilihan tidak valid");
143 }
144
145 System.out.println("Hasil akhir:");
146 printArray(arr);
147 scanner.close();
148 }

```

- Menjalankan sorting sesuai pilihan pengguna
- Menampilkan hasil akhir sorting
- Menutup scanner untuk menghindari kebocoran memori

## TUGAS SOAL 4

```

1 // Fungsi untuk mengisi matriks
2 public static int[][] inputMatriks(int rows, int cols, Scanner scanner, String name) {
3     int[][] matriks = new int[rows][cols];
4     System.out.println("Masukkan elemen untuk matriks " + name + ":");
5     for (int i = 0; i < rows; i++) {
6         for (int j = 0; j < cols; j++) {
7             System.out.print("Masukkan [" + i + "][ " + j + "]: ");
8             matriks[i][j] = scanner.nextInt();
9         }
10    }
11    return matriks;
12 }

```

- Fungsi ini menerima jumlah baris (rows) dan jumlah kolom (cols) sebagai parameter.
- Scanner digunakan untuk membaca input angka dari pengguna.
- Loop bersarang (for) digunakan untuk mengisi elemen matriks satu per satu.
- Matriks dikembalikan dalam bentuk array 2D (int[][]).

```

19 // Fungsi untuk menampilkan matriks
20
21 public static void displayMatrix(int[][] matrix) {
22     for (int[] row : matrix) {
23         for (int elem : row) {
24             System.out.print(elem + " ");
25         }
26         System.out.println();
27     }
28 }

```

- Loop pertama (for) membaca setiap baris dalam matriks.
- Loop kedua (for) mencetak setiap elemen dalam baris.
- Setiap baris ditampilkan dalam bentuk grid.

```

29 // Fungsi untuk penjumlahan matriks
30 public static int[][] addMatrix(int[][] A, int[][] B) {
31     int rowA = A.length;
32     int colA = A[0].length;
33     int rowB = B.length;
34     int colB = B[0].length;
35     int[][] result = new int[rowA][colA];
36     for (int i = 0; i < rowA; i++) {
37         for (int j = 0; j < colA; j++) {
38             result[i][j] = A[i][j] + B[i][j];
39         }
40     }
41     return result;
42 }

```

- Fungsi ini melakukan penjumlahan elemen per elemen antara matriks A dan B.
- Loop bersarang digunakan untuk menjumlahkan setiap elemen.
- Hasilnya dikembalikan dalam bentuk matriks baru (result).

```

43 // Fungsi untuk perkalian matriks
44 public static int[][] multiplyMatrix(int[][] A, int[][] B) {
45     int rowA = A.length;
46     int colA = A[0].length;
47     int rowB = B.length;
48     int colB = B[0].length;
49     int[][] result = new int[rowA][colB];
50     for (int i = 0; i < rowA; i++) {
51         for (int j = 0; j < colB; j++) {
52             for (int k = 0; k < colA; k++) {
53                 result[i][j] += A[i][k] * B[k][j];
54             }
55         }
56     }
57     return result;
58 }

```

- Perkalian matriks hanya valid jika jumlah kolom matriks pertama = jumlah baris matriks kedua.
- Loop bersarang tiga tingkat digunakan : Loop i (baris pertama), loop j (kolom kedua), loop k (Mengalihkan elemen sesuai aturan perkalian matriks)

```

59 public static void main(String[] args) {
60     Scanner scanner = new Scanner(System.in);
61
62     System.out.print("Masukkan jumlah baris matriks A: ");
63     int rowA = scanner.nextInt();
64     System.out.print("Masukkan jumlah kolom matriks A: ");
65     int colA = scanner.nextInt();
66
67     int[][] matrixA = inputMatrix(rowA, colA, scanner, "A");
68     int[][] matrixB = inputMatrix(rowA, colA, scanner, "B");
69
70     System.out.println("Pilih operasi:");
71     System.out.println("1. Penjumlahan matriks");
72     System.out.println("2. Perkalian matriks");
73     System.out.print("Pilihan: ");
74     int pilihan = scanner.nextInt();
75 }

```

- Pengguna memasukkan jumlah baris & kolom yang sama untuk kedua matriks.
- Memanggil inputMatrix() untuk Matriks A & B.
- Pengguna memilih penjumlahan (1) atau perkalian (2).
- Jika pilihan = 1, maka matriks akan dijumlahkan.
- Jika pilihan = 2, maka matriks akan dikalikan.
- Jika input salah, program menampilkan pesan error.