

Laporan Pemrograman Berorientasi Objek Dengan Bahasa Pemrograman Java

Dosen Pengampu : IMANUEL HARKESPAN, M.Kom.

Tanggal Pertemuan : 26 Maret 2025

Nama : Ferdian Nur Fariza
NIM : A11.2023.15074
Kelas : A11.4403
Mata Kuliah : Pemrograman Berorientasi Objek

Topik :

Tugas Praktikum 5 (Array)

Tabel Deskripsi :

Pertemuan	Pembahasan Laporan	Link Dokumentasi Github
5	Latihan 1 dan Latihan 2	Link Here
	Sorting <ol style="list-style-type: none">Bubble SortInsertion SortSelection SortMerge Sort	
	Matrix	
	Queue	

Latihan 1 dan Latihan 2

Latihan 1

Program ini dapat menentukan panjang array yang diinginkan kemudian mengisi semua index array tersebut. Kemudian program akan menampilkan array yang berhasil diinput oleh user.

```
Latihan_1.java praktek_java\week_5\array M

3  import java.util.Scanner;
4  public class Latihan_1 {
    Run main | Debug main | Run | Debug
5      public static void main(String[] args){
6          Scanner scanner=new Scanner(System.in);
7          System.out.print("Masukkan jumlah data: ");
8          int data= scanner.nextInt();
9          int[] array = new int[data];
```

Karena melibatkan input dari user maka program memerlukan library Scanner. Kemudian membuat objek Scanner dengan nama scanner dengan parameter System.in. Kemudian kita memanggil objek scanner untuk menerima input integer dengan:

```
int data = scanner.nextInt();
```

Dikarenakan array adalah objek dalam Java maka kita perlu mendeklarasikan sebagai objek baru dimana parameternya adalah variabel data(input pengguna).

```
Int[] arraay = new in[data];
```

```
11      //input
12      for (int i=0 ; i<data ; i++) {
13          System.out.print("Input data ke [" + (i+1) + "] : ");
14          array[i]= scanner.nextInt();
15      }
16
17      //output
18      for (int i=0 ; i<data ; i++) {
19          System.out.println("Data ke - "+ (i+1) + ": " + array[i]);
20      }
```

Kemudian menggunakan perulangan for untuk menerima input dan menampilkan hasil dari array dari user dengan iterasi berdasarkan input user sebelumnya, yaitu variabel data. Berikut output dari program Latihan_1.java :

```
PS C:\Users\riza\Java_OOP\praktek_java\week_5\array> java Latihan_1
Masukkan jumlah data: 3
Input data ke [1] : 12
Input data ke [2] : 89
Input data ke [3] : 76
Data ke - 1: 12
Data ke - 2: 89
Data ke - 3: 76
```

Latihan 2

Program Latihan_2.java ini adalah modifikasi dari program Latihan_1 dimana kita hanya dapat memberikan nilai array secara 1 dimensi. Program ini menerima input dan memberikan output dari array 2 dimensi. Perbedaannya adalah deklarasi variabel array yang memiliki 2 dimensi yaitu:

```
14
15 int[][] array = new int[totalSiswa][totalMatkul];
16
```

Kemudian untuk memberikan nilai pada array dua dimensi ini, membutuhkan nested loop dengan [x][y] dimana x sebagai index siswa dan y sebagai nilai dari beberapa matkul

```
17     for (int i = 0 ; i < totalSiswa ; i++){
18         for (int j= 0 ; j < totalMatkul ; j++){
19             System.out.print("Siswa ke-" + (i+1) +", ");
20             System.out.print("Masukkan nilai ke-" + (j+1) +": ");
21             array [i][j] = scanner.nextInt();
22         }
23     }
```

```

28  public static void daftarNilai(int totalSiswa, int totalMatkul, int[][] array){
29      for (int i = 0 ; i < totalSiswa ; i++){
30          System.out.println("Siswa ke-" + (i+1));
31          for (int j= 0 ; j < totalMatkul ; j++){
32              System.out.println("|--->Nilai ke-" + (j+1) + ": " + array[i][j]);
33          }
34          System.out.println("");
35      }
36  };

```

Kemudian membuat method `daftarNilai` dengan parameter siswa, matkul dan array 2 dimensinya yang bertujuan untuk menampilkan hasil dari input user. Berikut screenshot dari output programnya:

```

PS C:\Users\riza\Java_OOP\praktek_java\week_5\array> java Latihan_2
Total siswa: 2
Total daftar nilai: 2
Siswa ke-1, Masukkan nilai ke-1: 90
Siswa ke-1, Masukkan nilai ke-2: 94
Siswa ke-2, Masukkan nilai ke-1: 89
Siswa ke-2, Masukkan nilai ke-2: 79
-----hasil nilai-----
Siswa ke-1
|--->Nilai ke-1: 90
|--->Nilai ke-2: 94

Siswa ke-2
|--->Nilai ke-1: 89
|--->Nilai ke-2: 79

```

Sorting

Bubble Sort

Bubble sort adalah sorting sederhana dengan membandingkan 2 elemen yang berdekatan, proses ini dilakukan hingga tidak ada lagi pertukaran atau swaping 2 elemen yang berdekatan tersebut.

Program ini menggunakan library random java, yang akan membuat angka random dari 1-20. Kemudian algoritma sorting nya sebagai berikut:

```
17         boolean swaping = true;
18
19         while (swaping) {
20             swaping = false;
21             for (int i = 0; i < angka.length - 1; i++) {
22                 if (angka[i] > angka[i + 1]) {
23                     swaping = true;
24                     int temp = angka[i];
25                     angka[i] = angka[i + 1];
26                     angka[i + 1] = temp;
27                 }
28             }
29         }
```

Dengan mendeklarasikan variabel swaping sebagai true, perulangan while akan terjadi karena swaping bernilai true, dan didalam while diberhentikan dengan false agar while tidak terus menerus di eksekusi.

Di dalam perulangan for terdapat metode swapingnya, swappingnya adalah true jika angka pada kiri atau index lebih kecil bernilai lebih besar dibandingkan index selanjutnya (i + 1). Dan jika tidak terjadi swaping lagi maka swaping adalah false sehingga perulangan while berhenti dilakukan.

Nantinya akan menggunakan method printArray dengan perulangan for untuk menampilkan hasilnya. Berikut adalah outputnya

```

PS C:\Users\riza\Java_OOP\praktek_java\week_5\sorting> java Bubble
Random Array:
 3 7 9 16 16 7 8 8 1 5
Sorting Array:
 3 7 9 16 7 8 8 1 5 16
 3 7 9 7 8 8 1 5 16 16
 3 7 7 8 8 1 5 9 16 16
 3 7 7 8 1 5 8 9 16 16
 3 7 7 1 5 8 8 9 16 16
 3 7 1 5 7 8 8 9 16 16
 3 1 5 7 7 8 8 9 16 16
 1 3 5 7 7 8 8 9 16 16
 1 3 5 7 7 8 8 9 16 16
Hasil Sorting:
 1 3 5 7 7 8 8 9 16 16

```

Merge Sort

Merge sort adalah algoritma sorting dengan metode membagi sebuah array menjadi dua array yaitu array kanan dan array kiri dan nantinya pada array kiri dan kanan dilakukan sorting dengan cara membandingkan index pertama dan selanjutnya.

Ketika array kiri dan array kanan sudah dilakukan sorting maka nantinya akan dilakukan merging atau penggabungan kedua array tersebut.

```

19
20 > public static void mergeSort(int[] angka){ ...
52 > public static void merge(int[] angka, int[] kiri,int[] kanan){ ...
81 }

```

Dalam file Merge.java terdapat method mergeSort 1 parameter dan merge dengan 3 parameter variabel. mergeSort berfungsi untuk sorting pada array kanan atau kiri, sedangkan method merge untuk menyatukan kedua array tersebut yaitu array angka, array kanan dan kiri.

```

Merge.java praktek_java\week_5\sorting 1
6 public class Merge {
51 }
52 public static void merge(int[] angka, int[] kiri,int[] kanan){
53     int panjangKiri = kiri.length;
54     int panjangKanan = kanan.length;
55     int i = 0, j=0, k=0;
56     while (i < panjangKiri && j < panjangKanan ){
57
58         //proses penggabungan di array angka dengan,
59         //1. kita bandingkan array index 0 di kiri dan kanan, jika di kiri lebih kecil dari kanan, maka k
60         if(kiri[i] <= kanan[j]){
61             angka[k] = kiri[i];
62             i++;
63         }
64         else {
65             angka[k] = kanan[j];
66             j++;
67         }
68         k++;
69     }
70     while(i<panjangKiri){
71         angka[k]=kiri[i];
72         i++;
73         k++;
74     }
75     while(j<panjangKanan){
76         angka[k]=kanan[j];
77         j++;
78         k++;

```

Pada method mergeSort ini terdapat rekursif method mergeSort itu sendiri, dan memanggil method merge. Berikut hasil iterasinya:

```

PS C:\Users\riza\Java_OOP\praktek_java\week_5\sorting> java Merge
[47, 6, 10, 8, 30, 18, 3, 23, 17, 48]
Array: [47][6]--->[6, 47]
Array: [8][30]--->[8, 30]
Array: [10][8, 30]--->[8, 10, 30]
Array: [6, 47][8, 10, 30]--->[6, 8, 10, 30, 47]
Array: [18][3]--->[3, 18]
Array: [17][48]--->[17, 48]
Array: [23][17, 48]--->[17, 23, 48]
Array: [3, 18][17, 23, 48]--->[3, 17, 18, 23, 48]
Array: [6, 8, 10, 30, 47][3, 17, 18, 23, 48]--->[3, 6, 8, 10, 17, 18, 23, 30, 47, 48]
PS C:\Users\riza\Java_OOP\praktek_java\week_5\sorting>

```

Insertion Sort

Cara kerja insertion sort adalah dengan membandingkan nilai current yang seiring iterasi bertambah nilainya dan nantinya akan dibandingkan dengan nilai sebelumnya hingga index 0.

```
24 private static void insertionSort(int[] angka) {  
25     for (int i = 1; i < angka.length; i++) {  
26  
27         int nilaiCurrent = angka[i];  
28  
29         int j = i - 1;  
30         while (j ≥ 0 && angka[j] > nilaiCurrent ) {  
31             angka[j + 1] = angka[j];  
32             j--;  
33         }  
34         angka[j + 1] = nilaiCurrent ;  
35     }  
36 }  
37 private static void cetakHasil(int[] angka) {  
38     for (int i = 0; i < angka.length; i++) {  
39         System.out.println(angka[i]);  
40     }  
41 }  
42 }
```

Ini adalah method insertionSort dengan parameter array variabel integer, pada main terdapat random da n arrays. Ketika sudah membuat objek random maka selanjutnya adalah dilakukan sorting dengan memanggil method insertionSort ini. Berikut hasil iterasinya:

Selection Sort

```
20     public static void selectionSort(int[] angka){
21         int panjang = angka.length;
22         for (int i= 0 ; i < panjang - 1 ; i++){
23             int currentMin = angka[i];
24             int indexMin = i;
25             for (int j = i +1; j< panjang; j++){
26                 if(angka[j] < min){
27                     currentMin = angka[j];
28                     indexMin = j;
29                 }
30             }
31             swap(angka, i, indexMin);
32             System.out.println(Arrays.toString(angka));
33         }
34     }
35     public static void swap(int[] angka, int a, int b){
36         int temp = angka[a];
37         angka[a]= angka[b];
38         angka[b] = temp;
39     }
```

Selection sort dilakukan dengan membandingkan nilai current dalam perulangan for variabel i dengan angka selanjutnya atau j++.

```
public static void swap(int[] angka, int a, int b){
    int temp = angka[a];
    angka[a]= angka[b];
    angka[b] = temp;
}
```

Terdapat method swap, agar index a dan index b dapat bertukar posisi pada array. Berikut hasilnya:

```
PS C:\Users\riza\Java_OOP\praktek_java\week_5\sorting> java Selection
[33, 37, 24, 3, 33, 14, 15, 22, 48, 39]
[3, 37, 24, 33, 33, 14, 15, 22, 48, 39]
[3, 14, 24, 33, 33, 37, 15, 22, 48, 39]
[3, 14, 15, 33, 33, 37, 24, 22, 48, 39]
[3, 14, 15, 22, 33, 37, 24, 33, 48, 39]
[3, 14, 15, 22, 24, 37, 33, 33, 48, 39]
[3, 14, 15, 22, 24, 33, 37, 33, 48, 39]
[3, 14, 15, 22, 24, 33, 33, 37, 48, 39]
[3, 14, 15, 22, 24, 33, 33, 37, 48, 39]
[3, 14, 15, 22, 24, 33, 33, 37, 39, 48]
PS C:\Users\riza\Java_OOP\praktek_java\week_5\sorting>
```

Matrix

Program matrix dengan menggunakan arraylist

Struktur

Representasi baris matriks adalah dengan `ArrayList<integer>` kemudian setiap integer adalah elemen matriksnya, berikut adalah method untuk membaca setiap matriksnya.

```
8  public static ArrayList<ArrayList<Integer>> bacaMatrix(Scanner sc, int baris, int kolom, String namaMatriks) {
9      ArrayList<ArrayList<Integer>> matrix = new ArrayList<>();
10     System.out.println("Masukkan elemen-elemen " + namaMatriks + ":");
11     for (int i = 0; i < baris; i++) {
12         ArrayList<Integer> dataBaris = new ArrayList<>();
13         for (int j = 0; j < kolom; j++) {
14             System.out.print("[ " + i + ", " + j + " ] = ");
15             int val = sc.nextInt();
16             dataBaris.add(val);
17         }
18         matrix.add(dataBaris);
19     }
20     return matrix;
21 }

24 public static void printMatrix(ArrayList<ArrayList<Integer>> matrix, String namaMatriks) {
25     System.out.println("Matriks " + namaMatriks + ":");
26     for (ArrayList<Integer> row : matrix) {
27         for (Integer val : row) {
28             System.out.print(val + "\t");
29         }
30         System.out.println();
31     }
32 }
```

Method utama

Method utama dalam program ini adalah `bacaMatrix` dengan parameter `Scanner sc`, integer `baris`, integer `kolom` dan `String namaMatrix`, fungsinya adalah untuk membaca input matriks dari user. Method-method nya sebagai berikut:

1. `bacaMatrix`
2. `printMatrix`
3. `addMatrix`
4. `subtractMatrix`
5. `divideMatrix`
6. `multiplyMatrix`

Keenam matrix ini adalah method yang membentuk program di main. Dengan hasil sebagai berikut:

```
PS C:\Users\riza\Java_OOP\praktek_java\week_5\matrix> java Matrix
===== Program Operasi Matriks dengan ArrayList =====

Menu:
1. Penjumlahan Matriks
2. Pengurangan Matriks
3. Perkalian Matriks (Aljabar)
4. Pembagian Matriks (element-wise)
5. Transpose Matriks
6. Keluar
Pilih operasi [1-6]: 1
Masukkan jumlah baris matriks A: 2
Masukkan jumlah kolom matriks A: 1
Masukkan elemen-elemen Matriks A:
[0,0] = 5
[1,0] = 6
Masukkan jumlah baris matriks B: 2
Masukkan jumlah kolom matriks B: 1
Masukkan elemen-elemen Matriks B:
[0,0] = 3
[1,0] = 5
Matriks A:
5
6
Matriks B:
3
5
Matriks A + B:
8
11
```

Queue

Program queue ini disusun untuk FIFO yaitu First In First Out,

Struktur Utama

1. insert() gunanya untuk menambahkan elemen di akhir
2. remove() untuk menghapus elemen dari depan
3. peek() melihat elemen terdepan tanpa menghapus
4. isEmpty() mengecek apakah queue kosong
5. isFull() mengecek apakah queue sudah penuh
6. size() untuk mengembalikan nilai jumlah elemen saat ini
7. displayQueue() untuk menampilkan queue dari depan ke akhir

dengan perulangan while program akan terus dilakukan hingga user menginput n

```
128      System.out.print("\n Mau lanjutkan? (y/n): ");
129      char cont = scanner.next().charAt(0);
130      if (cont == 'n' || cont == 'N') {
131          break;
132      }
133  }
```

Kemudian setelah menerima input pilhan dari user, program menggunakan switch case untuk menerima kondisi dari user

```
94      switch (choice) {
95 >          case 1: ...
101 >          case 2: ...
108 >          case 3: ...
114 >          case 4: ...
117 >          case 5: ...
120 >          case 6: ...
124          default:
125              System.out.println("Pilihan tidak tepat, coba lagi!");
126      }
```

Pilihan 1 sampai 6 adalah method utama dari struktur utama.

Berikut adalah hasilnya:

```
PS C:\Users\riza\Java_OOP\praktek_java\week_5\queue> java QueueImplement
```

```
Queue Operations
```

1. Insert
2. Remove
3. Peek
4. Check Empty
5. Check Full
6. Size

```
Your Choice? 1
```

```
Enter value to insert: 45
```

```
Queue = 45
```

```
Do you want to continue (Type y or n)? y
```

```
Queue Operations
```

1. Insert
2. Remove
3. Peek
4. Check Empty
5. Check Full
6. Size

```
Your Choice? 2
```

```
Removed: 45
```

```
Queue is empty!
```

```
Do you want to continue (Type y or n)? _
```