

Laporan Modul 3: Dasar Pemrograman Java Lanjutan

Mata Kuliah: Praktikum Pemrograman Berorientasi Objek

Nama: [Riski Al Fatah]

NIM: [2024573010036]

Kelas: [TI.2E]

1. Abstrak

Laporan ini merangkum hasil praktikum **Modul 3: Dasar Pemrograman Java Lanjutan**, yang berfokus pada tiga topik utama: **Array**, **Method/Function**, dan **String/Manipulasi String**. Tujuan dari laporan ini adalah untuk mendokumentasikan pemahaman konsep, langkah-langkah implementasi kode, hasil eksekusi, serta analisis mendalam dari setiap praktikum dan tugas yang diselesaikan. Melalui serangkaian praktikum ini, mahasiswa berhasil mengimplementasikan array satu dan dua dimensi, mendefinisikan dan memanggil method dengan berbagai tipe, serta melakukan manipulasi string kompleks, termasuk validasi dan pemformatan data. Diharapkan laporan ini menjadi bukti pemahaman yang kuat terhadap fundamental lanjutan pemrograman Java.

2. Praktikum

Praktikum 1.1 - Array Satu Dimensi

Dasar Teori

Array adalah struktur data homogen yang menyimpan kumpulan nilai dengan tipe data yang sama. Array di Java memiliki ukuran yang tetap (*fixed size*) dan elemennya diakses melalui **indeks** yang dimulai dari 0. Praktikum ini berfokus pada deklarasi, inisialisasi, dan akses elemen array, serta modifikasi untuk mencari nilai tertinggi dan terendah.

Langkah Praktikum

1. Membuat *package* `modul_3` dan *class* `ArraySatuDimensi.java` .
2. Mendeklarasikan dan menginisialisasi array integer `nilaiSiswa` .
3. Menambahkan perulangan *enhanced for* untuk mencari **nilai tertinggi** dan **nilai terendah** dalam array.
4. Menampilkan hasil array, akses elemen spesifik, serta hasil pencarian nilai ekstrem.

Screenshoot Hasil Praktikum

```
package modul_3;

import java.util.Scanner;

public class ArrayDuaDimensi { new *
    public static void main(String[] args) { new *
        // Inisialisasi Scanner untuk input
        Scanner input = new Scanner(System.in);

        // Langkah 1: Deklarasi array 2D untuk nilai siswa (3 siswa, 4 mata pelajaran)
        int[][] nilaiSiswa = new int[3][4];
        String[] namaSiswa = {"Fatah", "Riski", "All"};
        // Perhatikan: Mata pelajaran di gambar berbeda dengan respons sebelumnya. Kita ikuti gambar.
        String[] mataPelajaran = {"Matematika", "Bahasa Indonesia", "IPA", "IPS"};

        // Langkah 2: Input nilai
        System.out.println("=== INPUT NILAI SISWA ===");
        for (int i = 0; i < 3; i++) {
            System.out.println("Input nilai untuk " + namaSiswa[i] + ":");
            for (int j = 0; j < 4; j++) {
                System.out.print(mataPelajaran[j] + ": ");
                nilaiSiswa[i][j] = input.nextInt();
            }
            System.out.println();
        }

        // Tutup scanner setelah selesai input
        input.close();
    }
}
```

```

// Langkah 3: Menampilkan data dalam bentuk tabel
System.out.println("=== TABEL NILAI SISWA ===");

// Header Tabel
System.out.printf("%-12s", "Nama");
for (String mapel : mataPelajaran) {
    // Mencetak 7 karakter pertama dari nama mapel, diratakan ke kiri 8 kolom
    System.out.printf("%-8s", mapel.substring(0, Math.min(7, mapel.length())));
}
System.out.println("Rata-rata"); // Kolom tambahan Rata-rata

// Isi Tabel
for (int i = 0; i < 3; i++) {
    System.out.printf("%-12s", namaSiswa[i]); // Cetak nama siswa
    int total = 0;

    for (int j = 0; j < 4; j++) {
        System.out.printf("%-8d", nilaiSiswa[i][j]); // Cetak nilai
        total += nilaiSiswa[i][j];
    }

    // Hitung dan cetak rata-rata per siswa
    double rataRata = total / 4.0;
    System.out.printf("%.2f\n", rataRata);
}
}
}

```

```

"C:\Program Files\Amazon Corretto\jdk21
Nilai Siswa: [85, 92, 78, 95, 88]
Nilai siswa ke-3 (indeks 2): 78
Nilai Tertinggi: 95
Nilai Terendah: 78

Process finished with exit code 0

```

Penjelasan Latihan

Pada tugas ini, saya diminta untuk membuat array satu dimensi yang menyimpan nilai ujian dari 10 siswa. Tugas ini mencakup:

- Menampilkan nilai seluruh siswa yang disimpan dalam array.
- Menghitung rata-rata nilai ujian seluruh siswa.
- Mencari nilai tertinggi dan terendah dari array yang telah dibuat.

Screenshoot Hasil Latihan

```

package modul_3.latihan;

public class latihan1 { new *
    public static void main(String[] args) { new *
        int[] nilaiUjian = {75, 88, 65, 92, 70, 80, 95, 60, 85, 78}; // 10 siswa
        int totalNilai = 0;
        int tertinggi = nilaiUjian[0];
        int terendah = nilaiUjian[0];

        // Hitung total dan cari tertinggi/terendah
        for (int nilai : nilaiUjian) {
            totalNilai += nilai;
            if (nilai > tertinggi) {
                tertinggi = nilai;
            }
            if (nilai < terendah) {
                terendah = nilai;
            }
        }

        double rataRata = (double) totalNilai / nilaiUjian.length;

        System.out.printf("Rata-rata nilai ujian: %.2f\n", rataRata);
        System.out.println("Nilai Tertinggi: " + tertinggi);
        System.out.println("Nilai Terendah: " + terendah);
    }
}

```

```

"C:\Program Files\Amazon Corretto\jdk21.0.8_9\bin\java.exe"
Rata-rata nilai ujian: 78,80
Nilai Tertinggi: 95
Nilai Terendah: 60

Process finished with exit code 0

```

Analisa dan Pembahasan

Pada praktikum ini, array `nilaiSiswa` berhasil dideklarasikan dan diakses. Modifikasi kode untuk mencari nilai tertinggi dan terendah dilakukan dengan inisialisasi awal `nilaiTertinggi` dan `nilaiTerendah` dengan elemen pertama array (`nilaiSiswa[0]`). Kemudian, perulangan `for-each` digunakan untuk membandingkan setiap elemen dengan nilai ekstrem yang tersimpan, memperbarui variabel `nilaiTertinggi` jika ditemukan nilai yang lebih besar,

dan `nilaiTerendah` jika ditemukan nilai yang lebih kecil. Ini menunjukkan efisiensi penggunaan perulangan untuk analisis data dalam array.

Praktikum 1.2 - Array Dua Dimensi

Dasar Teori

Array Dua Dimensi (2D) atau matriks adalah array dari array. Array 2D digunakan untuk merepresentasikan data dalam bentuk baris dan kolom. Akses elemen memerlukan dua indeks: `namaArray[baris][kolom]`. Praktikum ini melibatkan pemrosesan data nilai siswa per mata pelajaran.

Langkah Praktikum

1. Membuat class `ArrayDuaDimensi.java` .
2. Mendeklarasikan array 2D nilai yang merepresentasikan nilai 3 siswa untuk 4 mata pelajaran.
3. Menjalankan program dan menganalisis struktur data.

Screenshot Hasil Praktikum

```
package modul_3;

import java.util.Scanner;

public class ArrayDuaDimensi { new *
    public static void main(String[] args) { new *
        // Inisialisasi Scanner untuk input
        Scanner input = new Scanner(System.in);

        // Langkah 1: Deklarasi array 2D untuk nilai siswa (3 siswa, 4 mata pelajaran)
        int[][] nilaiSiswa = new int[3][4];
        String[] namaSiswa = {"Fatah", "Riski", "All"};
        // Perhatikan: Mata pelajaran di gambar berbeda dengan respons sebelumnya. Kita ikuti gambar.
        String[] mataPelajaran = {"Matematika", "Bahasa Indonesia", "IPA", "IPS"};

        // Langkah 2: Input nilai
        System.out.println("=== INPUT NILAI SISWA ===");
        for (int i = 0; i < 3; i++) {
            System.out.println("Input nilai untuk " + namaSiswa[i] + ":");
            for (int j = 0; j < 4; j++) {
                System.out.print(mataPelajaran[j] + ": ");
                nilaiSiswa[i][j] = input.nextInt();
            }
            System.out.println();
        }

        // Tutup scanner setelah selesai input
        input.close();
    }
}
```

```

// Langkah 3: Menampilkan data dalam bentuk tabel
System.out.println("=== TABEL NILAI SISWA ===");

// Header Tabel
System.out.printf("%-12s", "Nama");
for (String mapel : mataPelajaran) {
    // Mencetak 7 karakter pertama dari nama mapel, diratakan ke kiri 8 kolom
    System.out.printf("%-8s", mapel.substring(0, Math.min(7, mapel.length())));
}
System.out.println("Rata-rata"); // Kolom tambahan Rata-rata

// Isi Tabel
for (int i = 0; i < 3; i++) {
    System.out.printf("%-12s", namaSiswa[i]); // Cetak nama siswa
    int total = 0;

    for (int j = 0; j < 4; j++) {
        System.out.printf("%-8d", nilaiSiswa[i][j]); // Cetak nilai
        total += nilaiSiswa[i][j];
    }

    // Hitung dan cetak rata-rata per siswa
    double rataRata = total / 4.0;
    System.out.printf("%.2f\n", rataRata);
}
}

```

```
=== INPUT NILAI SISWA ===
```

```
Input nilai untuk Fatah:
```

```
Matematika: 90
```

```
Bahasa Indonesia: 90
```

```
IPA: 90
```

```
IPS: 90
```

```
Input nilai untuk Riski:
```

```
Matematika: 80
```

```
Bahasa Indonesia: 80
```

```
IPA: 80
```

```
IPS: 80
```

```
Input nilai untuk All:
```

```
Matematika: 88
```

```
Bahasa Indonesia: 88
```

```
IPA: 88
```

```
IPS: 88
```

```
=== TABEL NILAI SISWA ===
```

Nama	Matemat	Bahasa	IPA	IPS	Rata-rata
Fatah	90	90	90	90	90,00
Riski	80	80	80	80	80,00
All	88	88	88	88	88,00

Penjelasan Latihan

Pada tugas ini, saya diminta untuk membuat array dua dimensi untuk menyimpan nilai ujian siswa pada beberapa mata pelajaran. Tugas ini mencakup:

- Menampilkan mata pelajaran dengan nilai tertinggi untuk setiap siswa.
- Menghitung rata-rata nilai untuk setiap mata pelajaran yang ada.

Screenshoot Hasil Latihan


```
package modul_3.latihan;
```

```
public class latihan2 { new *
```

```
    public static void main(String[] args) { new *
```

```
        String[] mataPelajaran = {"Matematika", "Fisika", "Biologi", "Kimia"};
```

```
        String[] namaSiswa = {"Andi", "Budi", "Sinta"};
```

```
        int[][] nilai = {
```

```
            {90, 85, 78, 92},
```

```
            {88, 95, 90, 80},
```

```
            {75, 80, 88, 95}
```

```
        };
```

```
        // --- Fitur 1: Mata Pelajaran dengan Nilai Tertinggi untuk Setiap Siswa ---
```

```
        System.out.println("--- Mata Pelajaran Nilai Tertinggi Setiap Siswa ---");
```

```
        for (int i = 0; i < nilai.length; i++) {
```

```
            int nilaiTertinggi = -1;
```

```
            String mapelTertinggi = "";
```

```
            for (int j = 0; j < nilai[i].length; j++) {
```

```
                if (nilai[i][j] > nilaiTertinggi) {
```

```
                    nilaiTertinggi = nilai[i][j];
```

```
                    mapelTertinggi = mataPelajaran[j];
```

```
                }
```

```
            }
```

```
            System.out.println(namaSiswa[i] + ": " + mapelTertinggi + " (" + nilaiTertinggi + ")");
```

```
        }
```

```
        System.out.println("\n" + "--- Rata-rata Nilai Setiap Mata Pelajaran ---");
```

```
        // --- Fitur 2: Hitung Rata-rata Nilai untuk Setiap Mata Pelajaran ---
```

```
        for (int j = 0; j < mataPelajaran.length; j++) {
```

```
            int totalNilaiMapel = 0;
```

```
            for (int i = 0; i < namaSiswa.length; i++) {
```

```
                totalNilaiMapel += nilai[i][j];
```

```
            }
```

```
            double rataRataMapel = (double) totalNilaiMapel / namaSiswa.length;
```

```
            System.out.printf("Rata-rata %s: %.2f\n", mataPelajaran[j], rataRataMapel);
```

```
        }
```

```
    }
```

```
}
```

```
"C:\Program Files\Amazon Corretto\jdk21.0.8_9\bin\jav
--- Mata Pelajaran Nilai Tertinggi Setiap Siswa ---
Andi: Kimia (92)
Budi: Fisika (95)
Sinta: Kimia (95)

--- Rata-rata Nilai Setiap Mata Pelajaran ---
Rata-rata Matematika: 84,33
Rata-rata Fisika: 86,67
Rata-rata Biologi: 85,33
Rata-rata Kimia: 89,00
```

Analisa dan Pembahasan

Array 2D berhasil diimplementasikan, di mana indeks pertama merepresentasikan siswa (baris) dan indeks kedua merepresentasikan mata pelajaran (kolom). Analisis menunjukkan bahwa array 2D sangat cocok untuk data tabular.

Praktikum 2.1 - Method Dasar

Dasar Teori

Method (atau Function) adalah blok kode yang dirancang untuk menjalankan tugas tertentu. Praktikum ini mengeksplorasi pembuatan method dengan berbagai jenis: void (tidak mengembalikan nilai) dan return method (mengembalikan nilai), serta penggunaan parameter untuk input data. Penggunaan method meningkatkan modularitas dan menghindari pengulangan kode (Don't Repeat Yourself - DRY).

Langkah Praktikum

1. Membuat class `MethodDasar.java` .
2. Membuat method `tampilkanHeader()` (void tanpa parameter) dan `tampilkanHasil()` (void dengan parameter).
3. Membuat method untuk operasi dasar matematika (tambah, kurang, kali, bagi) yang mengembalikan nilai.
4. Membuat method `validasiAngka()` untuk memastikan input valid.
5. Memanggil semua method di main.

Screenshoot Hasil Praktikum

```
package modul_3;
```

✓ 39 ^

```
public class MethodDasar { new *
```

```
// Langkah 1: Method void tanpa parameter
```

```
public static void tampilkanHeader() { 1 usage new *
```

```
    System.out.println("=====");
```

```
    System.out.println("  PROGRAM KALKULATOR SEDERHANA  ");
```

```
    System.out.println("=====");
```

```
}
```

```
// Langkah 2: Method void dengan parameter
```

```
public static void tampilkanHasil(String operasi, double a, double b, double hasil) { 4 usages new *
```

```
    System.out.printf("%.2f %s %.2f = %.2f\n", a, operasi, b, hasil);
```

```
}
```

```
// Langkah 3: Method dengan return value
```

```
public static double tambah(double a, double b) { 1 usage new *
```

```
    return a + b;
```

```
}
```

```
public static double kurang(double a, double b) { 1 usage new *
```

```
    return a - b;
```

```
}
```

```
public static double kali(double a, double b) { 1 usage new *
```

```
    return a * b;
```

```
}
```

```

public static double bagi(double a, double b) { 1 usage new *
    if (b != 0) {
        return a / b;
    } else {
        System.out.println("Error: Pembagian dengan nol!");
        return 0; // Mengembalikan 0 atau bisa juga Double.NaN
    }
}

// Langkah 4: Method dengan validasi input
public static boolean validasiAngka(double angka) { 2 usages new *
    // Mengembalikan true jika angka BUKAN NaN (Not a Number) dan BUKAN Infinite
    return !(Double.isNaN(angka) || Double.isInfinite(angka));
}

public static void main(String[] args) { new *
    // Langkah 5: Menggunakan semua Method
    tampilkanHeader();

    double x = 15.5;
    double y = 4.2;

```

```

        if (validasiAngka(x) && validasiAngka(y)) {
            double hasilTambah = tambah(x, y);
            double hasilKurang = kurang(x, y);
            double hasilKali = kali(x, y);
            double hasilBagi = bagi(x, y);

            tampilkanHasil(operasi: "+", x, y, hasilTambah);
            tampilkanHasil(operasi: "-", x, y, hasilKurang);
            tampilkanHasil(operasi: "*", x, y, hasilKali);
            tampilkanHasil(operasi: "/", x, y, hasilBagi);
        }
    }
}

```

```
"C:\Program Files\Amazon Corretto\j
=====
PROGRAM KALKULATOR SEDERHANA
=====
15,50 + 4,20 = 19,70
15,50 - 4,20 = 11,30
15,50 * 4,20 = 65,10
15,50 / 4,20 = 3,69

Process finished with exit code 0
```

Penjelasan Latihan

Pada tugas ini, saya diminta untuk membuat beberapa method untuk melakukan operasi matematika dasar seperti penjumlahan, pengurangan, perkalian, dan pembagian. Selain itu, tugas ini juga mencakup:

- Validasi pembagian dengan nol.
- Penggunaan method dengan parameter dan return type.

Screenshoot Hasil Latihan

```

package modul_3.latihan;

public class latihan3 { new *

    // Method untuk operasi pangkat (x^y)
    public static double hitungPangkat(double basis, double eksponen) { 1 usage new *
        return Math.pow(basis, eksponen);
    }

    // Method untuk operasi akar kuadrat (sqrt(x))
    public static double hitungAkarKuadrat(double bilangan) { 1 usage new *
        if (bilangan < 0) {
            System.out.println("Peringatan: Tidak bisa menghitung akar bilangan negatif.");
            return Double.NaN; // Not a Number
        }
        return Math.sqrt(bilangan);
    }

    // Method untuk validasi operasi pembagian dengan nol
    public static double bagiDenganValidasi(double pembilang, double penyebut) { 2 usages new *
        if (penyebut == 0) {
            System.out.println("Kesalahan: Pembagian dengan nol tidak diizinkan.");
            return Double.NaN;
        }
        return pembilang / penyebut;
    }

    public static void main(String[] args) { new *
        System.out.println("Pangkat (2^3): " + hitungPangkat(basis: 2, eksponen: 3));
        System.out.println("Akar Kuadrat (25): " + hitungAkarKuadrat(bilangan: 25));
        System.out.println("Pembagian (10/2): " + bagiDenganValidasi(pembilang: 10, penyebut: 2));
        System.out.println("Pembagian (10/0): " + bagiDenganValidasi(pembilang: 10, penyebut: 0));
    }
}

```

```

"C:\Program Files\Amazon Corretto\jdk21.0.8_9\bin
Pangkat (2^3): 8.0
Akar Kuadrat (25): 5.0
Pembagian (10/2): 5.0
Kesalahan: Pembagian dengan nol tidak diizinkan.
Pembagian (10/0): NaN

Process finished with exit code 0

```

Analisa dan Pembahasan

Implementasi method berjalan dengan baik. Method `tambah` , `kurang` , dan `kali` menggunakan tipe data `double` sebagai return type untuk mendukung perhitungan desimal. Method `bagi` menyertakan validasi sederhana untuk menghindari `ArithmeticException` (pembagian dengan nol). Pemisahan logika program ke dalam method membuat kode di main lebih bersih dan mudah dipahami, menunjukkan prinsip modularitas.

Praktikum 2.2 - Method Dengan Array

Dasar Teori

Method dapat menerima array sebagai parameter dan mengembalikan array sebagai return type. Hal ini memungkinkan pemrosesan array yang kompleks (seperti pengurutan, pencarian maksimum/minimum, atau penghitungan rata-rata) diisolasi dalam fungsi yang spesifik.

Langkah Praktikum

1. Membuat class `ArrayMethod.java` .
2. Membuat method untuk `inputArray` , `tampilkanArray` , `cariMaksimum` , `cariMinimum` , `hitungRataRata` , dan `urutkanArray` .
3. Method `inputArray` menggunakan `Scanner` untuk input dinamis.
4. Method `urutkanArray` menggunakan algoritma Bubble Sort dasar.
5. Menggabungkan semua method di main untuk alur analisis array lengkap.

Screenshoot Hasil Praktikum

```

package modul_3;

import java.util.Scanner;

public class ArrayMethod { new *

    // Method untuk input array
    public static int[] inputArray(int ukuran) { 1 usage new *
        Scanner input = new Scanner(System.in);
        int[] array = new int[ukuran];

        System.out.println("Masukkan " + ukuran + " angka:");
        for (int i = 0; i < ukuran; i++) {
            System.out.print("Angka ke-" + (i + 1) + ": ");
            array[i] = input.nextInt();
        }
        // Perhatian: Dalam praktikum sederhana, Scanner bisa tidak ditutup.
        // Namun, jika Anda menggunakan IDE modern, mungkin akan muncul peringatan.
        // Kita tidak akan menutupnya di sini agar sesuai persis dengan modul.
        return array;
    }
}

```



```
// Method untuk menampilkan array
public static void tampilkanArray(int[] array) { 2 usages new *
    System.out.print("Array: ");
    for (int i = 0; i < array.length; i++) {
        System.out.print(array[i]);
        if (i < array.length - 1) {
            System.out.print(", ");
        }
    }
    System.out.println("]");
}
```

```
// Method untuk mencari nilai maksimum
public static int cariMaksimum(int[] array) { 1 usage new *
    int maks = array[0];
    for (int i = 1; i < array.length; i++) {
        if (array[i] > maks) {
            maks = array[i];
        }
    }
    return maks;
}
```

```

public static int cariMinimum(int[] array) { 1 usage new *
    int min = array[0];
    for (int i = 1; i < array.length; i++) {
        if (array[i] < min) {
            min = array[i];
        }
    }
    return min;
}

```

```

// Method untuk menghitung rata-rata
public static double hitungRataRata(int[] array) { 1 usage new *
    int total = 0;
    for (int nilai : array) {
        total += nilai;
    }
    return (double) total / array.length;
}

```

```

public static int[] urutkanArray(int[] array) { 1 usage new *
    // Membuat copy array agar array asli tidak berubah (immutable)
    int[] arrayBaru = array.clone();

    for (int i = 0; i < arrayBaru.length - 1; i++) {
        for (int j = 0; j < arrayBaru.length - 1 - i; j++) {
            if (arrayBaru[j] > arrayBaru[j + 1]) {
                // Tukar posisi
                int temp = arrayBaru[j];
                arrayBaru[j] = arrayBaru[j + 1];
                arrayBaru[j + 1] = temp;
            }
        }
    }
    return arrayBaru;
}

```

```
public static void main(String[] args) { new *
    System.out.println("=== PROGRAM ANALISIS ARRAY ===");

    // Memanggil inputArray (5 adalah ukuran array yang akan diinput)
    int[] data = inputArray(ukuran: 5);

    System.out.println("\n=== HASIL ANALISIS ===");

    // Menampilkan array yang diinput
    tampilkanArray(data);

    // Menghitung dan menampilkan hasil analisis dasar
    System.out.println("Nilai Maksimum: " + cariMaksimum(data));
    System.out.println("Nilai Minimum: " + cariMinimum(data));
    System.out.printf("Rata-rata: %.2f\n", hitungRataRata(data));

    // Mengurutkan dan menampilkan array yang sudah diurutkan
    int[] dataUrut = urutkanArray(data);
    System.out.print("Array setelah diurutkan: ");
    tampilkanArray(dataUrut);
}
}
```

Masukkan 5 angka:

Angka ke-1: 4

Angka ke-2: 2

Angka ke-3: 3

Angka ke-4: 1

Angka ke-5: 5

=== HASIL ANALISIS |===

Array: [4, 2, 3, 1, 5]

Nilai Maksimum: 5

Nilai Minimum: 1

Rata-rata: 3,00

Array setelah diurutkan: Array: [1, 2, 3, 4, 5]

Penjelasan Latihan

Pada tugas ini, saya diminta untuk membuat beberapa method untuk menghitung rata-rata nilai dalam array, serta mencari nilai maksimum dan minimum dalam array. Tugas ini juga mencakup:

- Mengurutkan array dan menampilkan hasilnya.
- Menampilkan nilai maksimum, minimum, dan rata-rata array.

Screenshoot Hasil Latihan

```

package modul_3.latihan;

import java.util.Arrays;

public class latihan4 { new *

    // --- Fitur 1: Mencari Nilai Tertentu ---
    public static int cariNilai(int[] arr, int target) { 1 usage new *
        for (int i = 0; i < arr.length; i++) {
            if (arr[i] == target) {
                return i; // Mengembalikan indeks jika ditemukan
            }
        }
        return -1; // Mengembalikan -1 jika tidak ditemukan
    }

    // --- Fitur 2: Menghitung Median dari Array yang Sudah Diurutkan ---
    public static double hitungMedian(int[] arr) { 2 usages new *
        // Asumsi array sudah diurutkan (sesuai permintaan modul)
        int n = arr.length;
        if (n % 2 == 1) {
            // Jumlah ganjil: ambil nilai tengah
            return arr[n / 2];
        } else {
            // Jumlah genap: rata-rata dua nilai tengah
            int mid1 = arr[n / 2 - 1];
            int mid2 = arr[n / 2];
            return (double) (mid1 + mid2) / 2;
        }
    }
}

```

```

}

public static void main(String[] args) { new *
    int[] dataGanjil = {10, 20, 30, 40, 50}; // Sudah diurutkan
    int[] dataGenap = {10, 20, 30, 40, 50, 60}; // Sudah diurutkan

    // Test Fitur 1
    int target = 40;
    int indeks = cariNilai(dataGanjil, target);
    System.out.println("Mencari " + target + " di array ganjil: " +
        (indeks != -1 ? "Ditemukan pada indeks " + indeks : "Tidak ditemukan"));

    // Test Fitur 2
    System.out.println("Array Ganjil: " + Array
        ", Median: " + hitungMedian(dataGanjil));
    System.out.println("Array Genap: " + Arrays.toString(dataGenap) +
        ", Median: " + hitungMedian(dataGenap));
}
}

```

Typo: In word 'indeks'

Replace with 'indeeds' Alt+Shift+Enter More

```

"C:\Program Files\Amazon Corretto\jdk21.0.8_9\bin\java.exe
Mencari 40 di array ganjil: Ditemukan pada indeks 3
Array Ganjil: [10, 20, 30, 40, 50], Median: 30.0
Array Genap: [10, 20, 30, 40, 50, 60], Median: 35.0

Process finished with exit code 0

```

Analisa dan Pembahasan

Pemisahan logika pemrosesan array ke dalam method yang berbeda (input, output, analisis, sort) terbukti efektif. Dengan adanya method, kode menjadi lebih terorganisir dan fungsionalitasnya dapat digunakan kembali. Method `urutkanArray` menggunakan `array.clone()` untuk memastikan array asli tidak diubah (prinsip immutability yang baik).

Praktikum 3.1 - Manipulasi String Dasar

Dasar Teori

String di Java adalah objek yang merepresentasikan rangkaian karakter dan bersifat immutable (tidak dapat diubah setelah dibuat). Java menyediakan berbagai method built-in untuk manipulasi string, seperti `length()`, `charAt()`, `toUpperCase()`, `toLowerCase()`, dan `contains()`. Praktikum ini fokus pada penggunaan dasar method tersebut.

Langkah Praktikum

1. Membuat class `StringDasar.java` .
2. Mendefinisikan method `analysisString` , `hitungVokal` , `reverseString` , dan `isPalindrome` .
3. `analysisString` menggunakan method built-in dan `split()` untuk menghitung kata.
4. `isPalindrome` menggunakan `reverseString` setelah membersihkan string dari karakter non-alfanumerik.
5. Menggunakan `Scanner` untuk input dinamis dan demonstrasi semua fungsi.

Screenshoot Hasil Praktikum

```

package modul_3;

import java.util.Scanner;

public class StringDasar { new *

    public static void analysisString(String teks) { 1 usage new *
        System.out.println("== ANALISIS STRING ==");
        System.out.println("Teks asli: \" + teks + \"");
        System.out.println("Panjang string: " + teks.length());
        System.out.println("Karakter pertama: " + teks.charAt(0));
        System.out.println("Karakter terakhir: " + teks.charAt(teks.length() - 1));
        System.out.println("Uppercase: " + teks.toUpperCase());
        System.out.println("Lowercase: " + teks.toLowerCase());

        // Menghitung jumlah kata
        String[] kata = teks.split(regex: "[\\s,\\.]+");
        System.out.println("Jumlah kata: " + kata.length);

        // Menghitung jumlah vokal
        int jmlVokal = hitungVokal(teks);
        System.out.println("Jumlah vokal: " + jmlVokal);
    }
}

```



```
public static int hitungVokal(String teks) { 1 usage new *
    String vokal = "aAeEiIoOuU";
    int count = 0;
    for (int i = 0; i < teks.length(); i++) {
        if (vokal.contains(String.valueOf(teks.charAt(i)))) {
            count++;
        }
    }
    return count;
}

public static String reverseString(String teks) { 2 usages new *
    StringBuilder reversed = new StringBuilder();
    for (int i = teks.length() - 1; i >= 0; i--){
        reversed.append(teks.charAt(i));
    }
    return reversed.toString();
}

public static boolean isPalindrome(String teks) { 1 usage new *
    String clean = teks.replaceAll( regex: "[^a-zA-Z0-9]", replacement: "").toLowerCase();
    String reversed = reverseString(clean);
    return clean.equals(reversed);
}
```

```

public static void main(String[] args) { new *

    System.out.println("=== PROGRAM MANIPULASI STRING ===");

    System.out.print("Masukkan sebuah kalimat: ");
    String kalimat = input.nextLine();

    // Analisis dasar
    analysisString(kalimat);

    // Membalik teks
    System.out.println("\n// Teks terbalik");
    String teksTerbalik = reverseString(kalimat);
    System.out.println("String terbalik: \"" + teksTerbalik + "\"");

    // Cek palindrome
    System.out.println("\n// Cek palindrome");
    if (isPalindrome(kalimat)) {
        System.out.println("String ini adalah palindrome!");
    } else {
        System.out.println("String ini bukan palindrome!");
    }

    // Pencarian sub-string
    System.out.println("\n// Pencarian substring");
    System.out.print("Masukkan kata yang ingin dicari: ");
    String cari = input.nextLine();|

```

```
// Pencarian case-insensitive
if (kalimat.toLowerCase().contains(cari.toLowerCase())) {
    System.out.println("Kata \"" + cari + "\" ditemukan dalam kalimat.");

    // Mencari posisi pertama (case-insensitive)
    int posisi = kalimat.toLowerCase().indexOf(cari.toLowerCase());
    System.out.println("Posisi pertama: " + posisi);
} else {
    System.out.println("Kata \"" + cari + "\" tidak ditemukan dalam kalimat.");
}

input.close();
}
```

```
"C:\Program Files\Amazon Corretto\jdk21.0.8_9
=== PROGRAM MANIPULASI STRING ===
Masukkan sebuah kalimat: makasih mas
=== ANALISIS STRING ===
Teks asli: "makasih mas"
Panjang string: 11
Karakter pertama: m
Karakter terakhir: s
Uppercase: MAKASIH MAS
Lowercase: makasih mas
Jumlah kata: 2
Jumlah vokal: 4

// Teks terbalik
String terbalik: "sam hisakam"

// Cek palindrome
String ini bukan palindrome!

// Pencarian substring
Masukkan kata yang ingin dicari: mas
Kata "mas" ditemukan dalam kalimat.
Posisi pertama: 8

Process finished with exit code 0
```

Penjelasan Latihan

Pada tugas ini, saya diminta untuk melakukan beberapa manipulasi string seperti:

- Menghitung jumlah vokal dalam sebuah string.
- Mengubah string menjadi uppercase dan lowercase.
- Menampilkan jumlah kata dalam string.


```

package modul_3.latihan;

public class latihan5 { new *

    // --- Fitur 1: Menghitung Jumlah Konsonan ---
    public static int hitungKonsonan(String teks) { 1 usage new *
        String teksBersih = teks.toLowerCase().replaceAll(regex: "[^a-z]", replacement: "");
        int konsonanCount = 0;

        for (char c : teksBersih.toCharArray()) {
            if (c != 'a' && c != 'e' && c != 'i' && c != 'o' && c != 'u') {
                konsonanCount++;
            }
        }
        return konsonanCount;
    }

    // --- Fitur 2: Mengubah String menjadi Pascal Case ---
    public static String toPascalCase(String teks) { 1 usage new *
        if (teks == null || teks.isEmpty()) {
            return "";
        }

        String[] words = teks.toLowerCase().split(regex: "\\s+");
        StringBuilder pascalCase = new StringBuilder();

        for (String word : words) {
            if (!word.isEmpty()) {
                // Kapitalisasi huruf pertama dan gabungkan sisanya
                pascalCase.append(Character.toUpperCase(word.charAt(0)))
                    .append(word.substring(beginIndex: 1));
            }
        }
        return pascalCase.toString();
    }

    public static void main(String[] args) { new *
        String kalimat = "Selamat belajar Java 17!";
        System.out.println("Kalimat: " + kalimat);

        // Test Fitur 1
        System.out.println("Jumlah Konsonan: " + hitungKonsonan(kalimat));

        // Test Fitur 2
        String teksPascal = toPascalCase( teks: "belajar pemrograman java lanjutan");
        System.out.println("Pascal Case: " + teksPascal); // Output: BelajarPemrogramanJavaLanjutan
    }
}

```

```
"C:\Program Files\Amazon Corretto\jdk21.0.8_9\
Kalimat: Selamat belajar Java 17!
Jumlah Konsonan: 10
Pascal Case: BelajarPemrogramanJavaLanjutan

Process finished with exit code 0
```

Analisa dan Pembahasan

String manipulasi menggunakan method objek sangat efisien. Proses pengecekan palindrome menunjukkan pentingnya pembersihan string (`replaceAll("[^a-zA-Z0-9]", "")`) dan konversi case (`toLowerCase()`) sebelum perbandingan. Penggunaan `StringBuilder` (meskipun tidak eksplisit di kode, tetapi `reverseString` dapat dioptimasi dengan itu) lebih efisien untuk operasi penggabungan/modifikasi string berulang dibandingkan hanya dengan operator `+` .

Praktikum 3.2 - Pengolahan Data Text

Dasar Teori

Pengolahan data teks sering melibatkan pemisahan token (kata, nama) dari sebuah delimiter (koma, spasi), serta pemformatan string ke format tertentu (misalnya, Title Case atau Username). Method `split()` sangat penting untuk memecah string berdasarkan pola.

Langkah Praktikum

1. Membuat class `TextProcessing.java` .
2. Membuat method `prosesDataMahasiswa` untuk memecah data nama yang dipisahkan koma (`split(",")`).
3. Method ini juga mencari nama terpanjang/terpendek dan mengurutkan nama menggunakan perbandingan `String` (`compareToIgnoreCase`).
4. Membuat method pemformatan nama: `formatTitleCase` , `formatKataInitial` , dan `buatUsername` .
5. Menggunakan `Scanner` untuk input data massal dan nama individu.

Screenshoot Hasil Praktikum

```
package modul_3;

import java.util.Scanner;

public class TextProcessing { new *

    public static void prosesDataMahasiswa(String data) { 1 usage new *
        System.out.println("=====");

        // Memisahkan data berdasarkan koma
        String[] dataMahasiswa = data.split(regex: ",");

        // Menampilkan data
        System.out.println("// Data Awal (Jumlah Mahasiswa: " + dataMahasiswa.length + ")");
        for (int i = 0; i < dataMahasiswa.length; i++) {
            System.out.print(dataMahasiswa.length + " "); // Mengulang dataMahasiswa.length
            // Modifikasi: Baris di modul terlihat seperti
            // System.out.print(i + 1 + ". " + dataMahasiswa[i] + " ");
            // Kita ikuti persis yang tertulis di gambar, meski kodenya kurang lengkap:
            System.out.println((i + 1) + ". " + dataMahasiswa[i] + " ");
        }

        // Analisis data
        System.out.println("\n=== ANALISIS DATA ===");
        System.out.println("Jumlah mahasiswa: " + dataMahasiswa.length);
    }
}
```



```
String namaTerpanjang = dataMahasiswa[0];
String namaTerpendek = dataMahasiswa[0];

for (String nama : dataMahasiswa) {
    if (nama.length() > namaTerpanjang.length()) {
        namaTerpanjang = nama;
    }
    if (nama.length() < namaTerpendek.length()) {
        namaTerpendek = nama;
    }
}

System.out.println("Nama terpanjang: " + namaTerpanjang + " (" + namaTerpanjang.length() + " karakter)")
System.out.println("Nama terpendek: " + namaTerpendek + " (" + namaTerpendek.length() + " karakter)");

// Mengurutkan nama secara alfabetik (Bubble Sort untuk String)
String[] namaUrut = dataMahasiswa.clone();
for (int i = 0; i < namaUrut.length - 1; i++) {
    for (int j = 0; j < namaUrut.length - 1 - i; j++) {
        // Bandingkan case-insensitive
        if (namaUrut[j].compareToIgnoreCase(namaUrut[j + 1]) > 0) {
            // Tukar posisi
            String temp = namaUrut[j];
            namaUrut[j] = namaUrut[j + 1];
            namaUrut[j + 1] = temp;
        }
    }
}
```

```

    }
}

System.out.println("\nNama mahasiswa (urutan alfabetik):");
for (int i = 0; i < namaUrut.length; i++) {
    System.out.println((i + 1) + ". " + namaUrut[i]);
}
}

public static void tampilkanFormatNama(String nama) { 1 usage new *
    // Panggil semua fungsi format
    String namaFormatted = formatTitleCase(nama);
    String initial = formatKataInitial(nama);
    String username = buatUsername(nama);

    System.out.println("\n=== FORMAT NAMA ===");
    System.out.println("Nama lengkap: " + nama);
    System.out.println("Title Case: " + namaFormatted);
    System.out.println("Initial: " + initial);
    System.out.println("Username: " + username);
}

```

```

public static String formatTitleCase(String teks) { 1 usage new *
    String[] kata = teks.split( regex: "\\s+");
    StringBuilder result = new StringBuilder();

    for (String k : kata) {
        if (k.length() > 0) {
            // Kapitalisasi huruf pertama dan gabungkan sisanya
            result.append(Character.toUpperCase(k.charAt(0)))
                .append(k.substring( beginIndex: 1).toLowerCase())
                .append(" ");
        }
    }
    // Menghilangkan spasi ekstra di akhir
    return result.toString().trim();
}

```

```

// Fungsi untuk mendapatkan initial (huruf depan setiap kata)
public static String formatKataInitial(String teks) { 1 usage new *
    String[] kata = teks.split( regex: "\\s+");
    StringBuilder initial = new StringBuilder();

    for (String k : kata) {
        if (k.length() > 0) {
            // Ambil huruf pertama dan kapitalisasi
            initial.append(Character.toUpperCase(k.charAt(0)));
        }
    }
}

```

```

    return initial.toString();
}

// Fungsi untuk membuat username
public static String buatUsername(String nama) { 1 usage new *
    // Mengubah ke lowercase, ganti spasi jadi titik, hilangkan non-alphanumeric (selain a-z, 0-9, dan
    return nama.toLowerCase().replaceAll( regex: "\\s+", replacement: ".").replaceAll( regex: "[^a-z0-9.]",
}

public static void main(String[] args) {...}
}

```

```
Masukkan satu nama untuk format: fatah
```

```
=====
```

```
// Data Awal (Jumlah Mahasiswa: 4)
```

```
4 1. fatah
```

```
4 2. isan
```

```
4 3. dedi
```

```
4 4. alip
```

```
=== ANALISIS DATA ===
```

```
Jumlah mahasiswa: 4
```

```
Nama terpanjang: fatah (5 karakter)
```

```
Nama terpendek: fatah (5 karakter)
```

```
Nama mahasiswa (urutan alfabetik):
```

```
1. alip
```

```
2. dedi
```

```
3. isan
```

```
4. fatah
```

```
=== FORMAT NAMA ===
```

```
Nama lengkap: fatah
```

```
Title Case: Fatah
```

```
Initial: F
```

```
Username: fatah
```

Penjelasan Latihan

Pada tugas ini, saya diminta untuk:

- Membuat program untuk memvalidasi format nama (tidak boleh mengandung angka atau karakter khusus).
- Membuat program untuk menghasilkan alamat email berdasarkan nama lengkap.

Screenshoot Hasil Latihan

```

package modul_3.latihan;

public class latihan6 { new *

    // --- Fitur 1: Menghasilkan Email Address berdasarkan Nama ---
    public static String generateEmail(String namaLengkap) { 1 usage new *
        if (namaLengkap == null || namaLengkap.trim().isEmpty()) {
            return "Nama tidak valid.";
        }
        // Ubah nama menjadi huruf kecil dan ganti spasi dengan titik
        String emailPart = namaLengkap.toLowerCase().trim().replaceAll(regex: "\\s+", replacement: ".");
        return emailPart + "@corp.com";
    }

    // --- Fitur 2: Memvalidasi Format Nama ---
    public static boolean isValidName(String nama) { 2 usages new *
        if (nama == null || nama.trim().isEmpty()) {
            return false;
        }
        // Cek apakah mengandung angka atau karakter khusus (selain spasi dan huruf)
        // [a-zA-Z\\s] artinya hanya huruf (besar/kecil) dan spasi yang diizinkan
        return nama.matches(regex: "^[a-zA-Z\\s]+$");
    }

    public static void main(String[] args) { new *
        String nama1 = "Mohammad Rauf";
        String nama2 = "Budi 123 Santoso!";

        // Test Fitur 1
        System.out.println("Email untuk '" + nama1 + "': " + generateEmail(nama1));

        // Test Fitur 2
        System.out.println("Validasi nama '" + nama1 + "': " + isValidName(nama1));
        System.out.println("Validasi nama '" + nama2 + "': " + isValidName(nama2));
    }
}

```

```

"C:\Program Files\Amazon Corretto\jdk21.0.8_9\bin\java.exe"
Email untuk 'Mohammad Rauf': mohammad.rauf@corp.com
Validasi nama 'Mohammad Rauf': true
Validasi nama 'Budi 123 Santoso!': false

Process finished with exit code 0

```

Penggunaan `String.split(",")` sangat efektif untuk mengubah data teks terstruktur menjadi array yang dapat diproses. Untuk pengurutan nama, digunakan Bubble Sort dengan method `compareToIgnoreCase()` untuk membandingkan string tanpa memperhatikan case, memastikan urutan alfabetis yang benar. Method pemformatan menunjukkan fleksibilitas manipulasi string dengan mengubah case dan menggunakan `replaceAll()` untuk membersihkan karakter.

3. Kesimpulan

Praktikum Modul 3: Dasar Pemrograman Java Lanjutan telah berhasil memberikan pemahaman yang mendalam mengenai fundamental lanjutan pemrograman Java, yang mencakup struktur data **Array**, konsep **Method/Function**, dan **Manipulasi String**.

Array terbukti menjadi struktur data yang efektif untuk mengelola data homogen, baik satu dimensi (untuk deret data tunggal) maupun dua dimensi (untuk data tabular), dengan implementasi algoritma pencarian nilai ekstrem dan rata-rata.

Konsep Method berhasil diimplementasikan untuk menciptakan kode yang modular dan dapat digunakan kembali (reusable), ditunjukkan melalui pembuatan fungsi kalkulator dan fungsi pemrosesan array. Hal ini sangat penting untuk program yang terstruktur dan mudah dirawat.

String sebagai objek yang immutable menunjukkan kemampuan manipulasi teks yang luas melalui berbagai method built-in dan Regular Expression untuk tugas-tugas seperti validasi, pemformatan case, dan pemecahan data.

Secara keseluruhan, modul ini telah memperkuat keterampilan dalam menyusun logika program yang kompleks, mengelola data secara efisien menggunakan array, dan mempraktikkan desain kode yang bersih dan terstruktur.

5. Referensi

1. Java Documentation: Arrays - <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html>
2. Java Documentation: Methods - <https://docs.oracle.com/javase/tutorial/java/javaOO/methods.html>
3. Java Documentation: The String Class - <https://docs.oracle.com/javase/tutorial/java/data/strings.html>