

## LAPORAN SEMENTARA

```

public class ArrayDemo {
    public static void main(String[] args) {
        int[] anArray; // declare an array of integers

        anArray = new int[10]; // create an array of integers

        // assign a value to each array element and print
        for (int i = 0; i < anArray.length; i++) {
            anArray[i] = i;
            System.out.print(anArray[i] + " ");
        }
        System.out.println();
    }
}

```

```

modul3.ArrayDemo > main for (int i = 0; i < anArray.length; i++) >
out - Run (ArrayDemo) X

--- compiler:3.13.0:compile (default-compile) @ workshoppenrolanjut ---
Nothing to compile - all classes are up to date.

--- exec:3.1.0:exec (default-cli) @ workshoppenrolanjut ---
0 1 2 3 4 5 6 7 8 9

```

1.

```

BUILD SUCCESS

public class ArrayOfStringDemo {
    public static void main(String[] args) {
        String[] anArray = { "String One", "String Two", "String Three" };

        for (int i = 0; i < anArray.length; i++) {
            System.out.println(anArray[i].toLowerCase());
        }
    }
}

```

```

modul3.ArrayOfStringDemo >
out - Run (ArrayOfStringDemo) X

Nothing to compile - all classes are up to date.

--- exec:3.1.0:exec (default-cli) @ workshoppenrolanjut ---
string one
string two
string three

```

2.

```

BUILD SUCCESS

public class TestObjectArray {
    public static void main(String[] args) {
        Object[] a = new Object[4];

        a[0] = new String("Hello, World!");
        a[1] = Long.valueOf(44); // gunakan wrapper modern
        a[2] = Float.valueOf(1.5f);

        for (int i = 0; i < a.length; i++) {
            System.out.println("a[" + i + "] = " + a[i]);
        }
    }
}

```

```

modul3.TestObjectArray > main
out - Run (TestObjectArray) X

--- exec:3.1.0:exec (default-cli) @ workshoppenrolanjut ---
a[0] = Hello, World!
a[1] = 44
a[2] = 1.5
a[3] = null

```

3.

```

BUILD SUCCESS

public class ArrayOfArraysDemo {
    public static void main(String[] args) {
        String[][] cartoons = {
            { "Flintstones", "Fred", "Wilma", "Pebbles", "Dino" },
            { "Rubble", "Barney", "Betty", "Sam Sam" },
            { "Jetsons", "George", "Jane", "Elroy", "Judy", "Rosie", "Astro" },
            { "Scooby Doo Gang", "Scooby Doo", "Shaggy", "Velma", "Fred", "Daphne" }
        };

        for (int i = 0; i < cartoons.length; i++) {
            System.out.print(cartoons[i][0] + " ");
            for (int j = 1; j < cartoons[i].length; j++) {
                System.out.print(cartoons[i][j] + " ");
            }
            System.out.println();
        }
    }
}

```

```

modul3.ArrayOfArraysDemo > main
out - Run (ArrayOfArraysDemo) X

--- exec:3.1.0:exec (default-cli) @ workshoppenrolanjut ---
Flintstones: Fred Wilma Pebbles Dino
Rubble: Barney Betty Sam Sam
Jetsons: George Jane Elroy Judy Rosie Astro
Scooby Doo Gang: Scooby Doo Shaggy Velma Fred Daphne

```

4.

**Analisa:** Latihan ini mendemonstrasikan dasar penggunaan array, yaitu deklarasi, alokasi memori, pengisian nilai berdasarkan indeks, serta pencetakan dengan perulangan. Percobaan ini menguatkan konsep dasar array satu dimensi dalam Java.

**Analisa:** Percobaan ini menggunakan array bertipe objek String yang kemudian diproses dengan method `toLowerCase()`. Latihan ini menunjukkan bahwa array tidak terbatas pada tipe data primitif, melainkan dapat menyimpan objek, dan bahwa method bawaan objek dapat diterapkan pada setiap elemen array.

**Analisa:** Pada latihan ini digunakan array bertipe Object yang diisi dengan beragam data seperti String, Long, dan Float. Percobaan ini menunjukkan fleksibilitas array bertipe Object serta menegaskan sifat polimorfisme di Java, di mana berbagai jenis objek dapat disimpan dalam satu array.

**Analisa:** Latihan ini menggunakan array dua dimensi dengan initializer untuk menyimpan data karakter kartun. Percobaan ini mengajarkan cara membangun array multi-dimensi menggunakan initializer dan bagaimana menampilkan data dengan nested loop.

```
public class ArrayOfArraysDemo2 {
    public static void main(String[] args) {
        int[][] aMatrix = new int[4][5] // array 2D dengan 4 baris

        // populate matrix
        for (int i = 0; i < aMatrix.length; i++) {
            aMatrix[i] = new int[5]; // tiap baris punya 5 kolom
            for (int j = 0; j < aMatrix[i].length; j++) {
                aMatrix[i][j] = i + j; // isi dengan i+j
            }
        }

        // print matrix
        for (int i = 0; i < aMatrix.length; i++) {
            for (int j = 0; j < aMatrix[i].length; j++) {
                System.out.print(aMatrix[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

modul3.ArrayOfArraysDemo2 > main >

put - Run (ArrayOfArraysDemo2) X

```
--- exec:3.1.0:exec (default-cli) @ workshoppemrolanjut ---
0 1 2 3 4
1 2 3 4 5
2 3 4 5 6
3 4 5 6 7
```

5.

```
public class ArrayCopyDemo {
    public static void main(String[] args) {
        char[] copyFrom = { 'd', 'e', 'c', 'a', 'f', 'e', 'e',
                           'i', 'n', 'a', 't', 'e', 'd' };

        char[] copyTo = new char[7];

        // salin 7 elemen mulai dari index ke-2 (yaitu 'c') ke copyTo mulai index
        System.arraycopy(copyFrom, 2, copyTo, 0, 7);

        // ubah copyTo menjadi String lalu cetak
        System.out.println(new String(copyTo));
    }
}
```

modul3.ArrayCopyDemo > main >

put - Run (ArrayCopyDemo) X

```
--- exec:3.1.0:exec (default-cli) @ workshoppemrolanjut ---
caffeine
```

6.

```
public class ArrayCopyDemo2 {
    public static void main(String[] args) {
        char[] copyFrom = { 'd', 'e', 'c', 'a', 'f', 'e', 'e',
                           'i', 'n', 'a', 't', 'e', 'd' };

        char[] copyTo = new char[7];
        char[] coba = new char[20];

        // ini bukan menyalin isi array, tapi hanya mengalihkan referensi
        coba = copyFrom;
        coba[0] = 't'; // mengubah index 0 juga mengubah isi copyFrom

        // salin 7 elemen dari index 2 copyFrom ke copyTo
        System.arraycopy(copyFrom, 2, copyTo, 0, 7);

        // tampilkan hasil
        System.out.println(new String(copyTo)); // hasil array copy
        System.out.println(new String(coba));   // hasil array yang sama dengan copyFrom
    }
}
```

modul3.ArrayCopyDemo2 >

put - Run (ArrayCopyDemo2) X

```
--- exec:3.1.0:exec (default-cli) @ workshoppemrolanjut ---
caffeine
teacaffeinated
```

7.

**Analisa:** Latihan ini menekankan pembuatan array dua dimensi secara eksplisit menggunakan perulangan. Hal ini memperlihatkan bahwa array dua dimensi di Java sebenarnya merupakan array dari array, sehingga sub-array harus dibuat secara manual jika tidak menggunakan initializer.

**Analisa:** Percobaan ini memperkenalkan penggunaan method `System.arraycopy()` untuk menyalin sebagian elemen array ke array lain. Hal ini mengajarkan cara penyalinan data yang lebih efisien dibandingkan perulangan manual, sekaligus mengenalkan fungsi built-in dalam Java.

**Analisa:** Latihan ini menunjukkan perbedaan antara assignment langsung pada array (shallow copy) dengan `System.arraycopy()` (deep copy parsial). Percobaan ini menekankan pentingnya memahami perbedaan teknik penyalinan array agar tidak terjadi kesalahan manipulasi data.

```
public static void main(String[] args) {
    int[][] arrayOfInts = {
        { 32, 87, 3, 589 },
        { 12, 1076, 2000, 8 },
        { 622, 127, 77, 955 }
    };

    boolean found = false;

    // gunakan label untuk keluar dari nested loop jika ketemu
    search:
    for (int i = 0; i < arrayOfInts.length; i++) {
        for (int j = 0; j < arrayOfInts[i].length; j++) {
            if (arrayOfInts[i][j] == 12) {
                System.out.println("Found 12 at " + i + ", " + j);
                found = true;
                break search;
            }
        }
    }

    if (!found) {
        System.out.println("12 not found in the array.");
    }
}
```

put - Run (Soal1) X

```
--- exec:3.1.0:exec (default-cli) @ workshoppemrolanjut ---
Found 12 at 1, 0
-----
BUILD SUCCESS
```

8.

```
public class Soal2 {
    public static void main(String[] args) {
        char[] matKul = {
            'p','e','m','r','o','g','r','a','m','a','n',
            'b','e','r','b','a','s','i','s',
            'o','b','y','e','k'
        };

        int count = 0;

        for (int i = 0; i < matKul.length; i++) {
            if (matKul[i] == 'a') {
                count++;
            }
        }

        System.out.println("Jumlah huruf 'a' = " + count);
    }
}
```

modul3.Sol2 > main >

put - Run (Soal2) X

```
--- compiler:3.13.0:compile (default-compile) @ workshoppemrolanjut ---
Nothing to compile - all classes are up to date.

--- exec:3.1.0:exec (default-cli) @ workshoppemrolanjut ---
Jumlah huruf 'a' = 3
-----
BUILD SUCCESS
```

9.

**Analisa:** Soal ini berfokus pada pencarian posisi angka tertentu dalam array dua dimensi. Pencarian dilakukan dengan perulangan bertingkat untuk menelusuri baris dan kolom, lalu dihentikan dengan perintah break ketika data ditemukan. Hal ini menekankan pentingnya logika looping dan kontrol alur program saat bekerja dengan array dua dimensi.

**Analisa:** Soal ini mengarahkan pada perhitungan jumlah huruf 'a' dalam array karakter. Dengan perulangan satu dimensi, setiap elemen diperiksa satu per satu, lalu jumlah kemunculan huruf dihitung. Percobaan ini memperkuat pemahaman akses elemen array serta logika perhitungan sederhana.

**Analisa:**

Berdasarkan percobaan yang telah dilakukan, setiap bagian memiliki peran penting dalam membangun pemahaman mengenai penggunaan array di Java. Soal pertama menekankan pencarian data dalam array dua dimensi dengan logika looping dan kontrol alur program, sedangkan soal kedua menekankan pada perhitungan jumlah karakter dalam array satu dimensi. Latihan pertama hingga ketiga memperkenalkan dasar penggunaan array, mulai dari array bertipe primitif hingga array bertipe objek, serta bagaimana objek yang berbeda dapat disimpan dalam array bertipe Object. Selanjutnya, latihan keempat dan kelima memperluas pemahaman ke arah array multi-dimensi, baik dengan initializer maupun dengan pembuatan sub-array secara eksplisit. Latihan keenam dan ketujuh kemudian berfokus pada teknik penyalinan array, membandingkan efisiensi penggunaan method `System.arraycopy()` dengan penyalinan manual maupun assignment langsung yang menimbulkan shallow copy. Secara keseluruhan, rangkaian percobaan ini memberikan gambaran lengkap mengenai konsep dasar array, variasi implementasi, serta tantangan yang muncul dalam manipulasi data menggunakan array.

**Kesimpulan Keseluruhan**

Dari semua percobaan dapat disimpulkan bahwa array adalah struktur data penting dalam Java yang dapat menyimpan sekumpulan data dengan tipe yang sama. Array dapat berupa satu dimensi maupun multi-dimensi, dan mampu menyimpan baik tipe data primitif maupun objek. Percobaan ini juga menekankan pentingnya memahami perbedaan shallow copy dan deep copy agar pengolahan data lebih aman. Dengan menguasai array, mahasiswa dapat mengimplementasikan struktur data ini secara efektif dalam program sederhana hingga kompleks.