

**LAPORAN PRATIKUM STRUKTUR DATA
PENGUNAAN ARRAYLIST DI JAVA**



**MATA KULIAH STRUKTUR DATA
DOSEN PENGAMPU : Dr. Wahyudi, S.T, M.T**

**OLEH:
AUFAN TAUFIQURRAHMAN
NIM 2411532011**

**FAKULTAS TEKNOLOGI INFORMASI
DEPARTEMEN INFORMATIKA
UNIVERSITAS ANDALAS**

2025

A. Pendahuluan

Dalam pemrograman Java, struktur data ArrayList merupakan salah satu implementasi dari interface List yang memungkinkan penyimpanan elemen secara dinamis. Berbeda dengan array biasa, ArrayList dapat bertambah dan berkurang ukurannya sesuai kebutuhan tanpa harus mendeklarasikan ukuran tetap di awal. Praktikum ini bertujuan untuk memahami cara kerja ArrayList dalam berbagai operasi dasar seperti penambahan, penghapusan, penyisipan, pengambilan elemen, serta pengelolaan data dengan input dari pengguna.

B. Tujuan

- Memahami konsep dasar penggunaan ArrayList di Java.
- Menerapkan operasi dasar ArrayList seperti add, remove, get, dan contains.
- Mengimplementasikan interaksi ArrayList dengan input pengguna melalui Scanner.

C. Prosedur praktikum

1. Class ArrayList1.java

class ini membuat ArrayList<Integer>, menambahkan elemen 1–5, kemudian menghapus elemen di indeks ke-3 dan menampilkannya kembali.

- Buat file dengan nama ArrayList1.java.
- Ketik kode sesuai contoh.

```
package pekan1;

import java.util.ArrayList;

public class ArrayList1 {

    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        int n = 5;
        ArrayList<Integer> arrli = new ArrayList<Integer>(n);
        for (int i = 1; i <= n; i++) {
            arrli.add(i);
        }
        System.out.println(arrli);
        arrli.remove(index:3);
        System.out.println(arrli);
        for (int i = 0; i < arrli.size(); i++) {
            System.out.print(arrli.get(i) + " ");
        }
    }
}
```

menambahkan elemen ke dalam list menggunakan method add(), lalu menghapus elemen pada **indeks ke-3** (elemen keempat). Ini penting karena remove() berbasis indeks dalam konteks ini.

- Jalankan program, amati output awal dan setelah remove(3)

```
[1, 2, 3, 4, 5]
[1, 2, 3, 5]
1 2 3 5
```

2. Class ArrayListGetElemen.java

class ini menambahkan beberapa elemen ke dalam ArrayList, lalu menampilkan elemen pada indeks tertentu menggunakan method get().

- Buat file ArrayListGetElemen.java.

Ketik kode program.

```
package pekan1;

import java.util.*;

public class ArrayListGetElemen {

    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        ArrayList<Integer> list = new ArrayList<Integer>();
        list.add(e:9);
        list.add(e:5);
        list.add(e:6);
        System.out.println(list);
        Integer n = list.get(index:1);
        System.out.println("Pada indeks ke 1 angkanya adalah:" + n);
    }
}
```

Mengambil elemen dari list berdasarkan **indeks** dengan syntax `get(index)`. Pada contoh ini, mengambil elemen ketiga (karena indeks dimulai dari 0).

- Jalankan program, perhatikan indeks yang diakses.

```
[9, 5, 6]
Pada indeks ke 1 angkanya adalah:5
```

3. Class ArrayListSisip.java

Menambahkan elemen dan menyisipkan elemen baru ke dalam indeks tertentu menggunakan `add(index, value)`.

- Buat file ArrayListSisip.java
- Ketik program sesuai instruksi.

```
package pekan1;

import java.util.ArrayList;

public class ArrayListSisip {

    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        ArrayList<Integer> list = new ArrayList<Integer>();
        list.add(e:1);
        list.add(e:2);
        list.add(e:4);
        System.out.println(list);
        list.add(index:2, element:3);
        System.out.println(list);
    }
}
```

menyisipkan elemen "A" ke **indeks ke-0**, mendorong elemen lain ke kanan, lalu menyisipkan elemen "D" di tengah list, menunjukkan bahwa ArrayList bisa menyisipkan bukan hanya di akhir. `add(index, element)` sebagai cara menyisipkan elemen pada posisi tertentu.

- Jalankan dan amati perubahan isi list sebelum dan sesudah penyisipan.

```
[1, 2, 4]
[1, 2, 3, 4]
```

4. Class DaftarSiswa.java

Simulasi program daftar siswa yang interaktif, memungkinkan pengguna menambahkan, menampilkan, mencari, dan menghapus nama siswa dari ArrayList.

- Buat file DaftarSiswa.java.
- Ketikkan program

```

    }
    } while (pilihan != 5);
}

private static void tambahNamaSiswa(Scanner scanner) {
    System.out.println(x: "Masukkan nama siswa: ");
    String nama = scanner.nextLine();
    daftarNamaSiswa.add(nama);
    System.out.println(x: "Nama siswa berhasil ditambahkan.");
}

private static void tampilkanDaftarNamaSiswa() {
    if (daftarNamaSiswa.isEmpty()) {
        System.out.println(x: "Tidak ada siswa dalam daftar");
    } else {
        System.out.println(x: "Daftar Nama Siswa: ");
        for (String nama : daftarNamaSiswa) {
            System.out.println(nama);
        }
    }
}

private static void hapusNamaSiswa(Scanner scanner) {
    System.out.println(x: "Masukkan nama siswa yang ingin dihapus: ");
    String nama = scanner.nextLine();
    if (daftarNamaSiswa.remove(nama)) {
        System.out.println(x: "Nama siswa berhasil dihapus");
    } else {
        System.out.println(x: "Nama siswa not found");
    }
}

package pekan1;

import java.util.*;

public class DaftarSiswa {

    private static ArrayList<String> daftarNamaSiswa = new ArrayList<>();

    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int pilihan;
        do {
            System.out.println(x: "\nMenu:");
            System.out.println(x: "1. Tambah Nama siswa");
            System.out.println(x: "2. Tampilkan Daftar Nama siswa");
            System.out.println(x: "3. Hapus Nama Siswa");
            System.out.println(x: "4. Cari Nama Siswa");
            System.out.println(x: "5. Keluar");
            System.out.println(x: "pilih menu");
            pilihan = scanner.nextInt();
            scanner.nextLine();
            switch (pilihan) {
                case 1:
                    tambahNamaSiswa(scanner);
                    break;
                case 2:
                    tampilkanDaftarNamaSiswa();
                    break;
                case 3:
                    hapusNamaSiswa(scanner);
                    break;
                case 4:
                    cariNamaSiswa(scanner);
                    break;
            }
        } while (pilihan != 5);
    }

    private static void cariNamaSiswa(Scanner scanner) {
        System.out.println(x: "Masukkan nama siswa yang ingin dicari");
        String nama = scanner.nextLine();
        if (daftarNamaSiswa.contains(nama)) {
            System.out.println("Nama siswa ditemukan: " + nama);
        } else {
            System.out.println(x: "Nama siswa tidak ditemukan");
        }
    }
}

```

- Inisialisasi ArrayList

```
private static ArrayList<String> daftarNamaSiswa = new ArrayList<>();
```

- Menambahkan nama dari input pengguna.

```
daftarNamaSiswa.add(nama);
```

- Pengkondisian untuk menampilkan seluruh isi arraylist

```
if (daftarNamaSiswa.isEmpty()) {
    System.out.println(x:"Tidak ada siswa dalam daftar");
} else {
    System.out.println(x:"Daftar Nama Siswa: ");
    for (String nama : daftarNamaSiswa) {
        System.out.println(nama);
    }
}
```

- Mengambil input pengguna dan memasukkannya ke penkondisian untuk menghapus element dalam arraylist

```
System.out.println(x:"Masukkan nama siswa yang ingin dihapus: ");
String nama = scanner.nextLine();
if (daftarNamaSiswa.remove(nama)) {
    System.out.println(x:"Nama siswa berhasil dihapus");
} else {
    System.out.println(x:"Nama siswa not found");
}
```

- Mencari elemen yang cocok berdasarkan input pengguna dan menampilkan output jika ada dan tidak

```
System.out.println(x:"Masukkan nama siswa yang ingin dicari");
String nama = scanner.nextLine();
if (daftarNamaSiswa.contains(nama)) {
    System.out.println("Nama siswa ditemukan: " + nama);
} else {
    System.out.println(x:"Nama siswa tidak ditemukan");
}
```

- Jalankan program di terminal dan uji masing-masing menu.

```
Menu:
1. Tambah Nama siswa
2. Tampilkan Daftar Nama siswa
3. Hapus Nama Siswa
4. Cari Nama Siswa
5. Keluar
pilih menu
1
Masukkan nama siswa:
udin
Nama siswa berhasil ditambahkan.
```

Menu:

1. Tambah Nama siswa
2. Tampilkan Daftar Nama siswa
3. Hapus Nama Siswa
4. Cari Nama Siswa
5. Keluar

pilih menu

2

Daftar Nama Siswa:

udin

Menu:

1. Tambah Nama siswa
2. Tampilkan Daftar Nama siswa
3. Hapus Nama Siswa
4. Cari Nama Siswa
5. Keluar

pilih menu

3

Masukkan nama siswa yang ingin dihapus:

udin

Nama siswa berhasil dihapus

Menu:

1. Tambah Nama siswa
2. Tampilkan Daftar Nama siswa
3. Hapus Nama Siswa
4. Cari Nama Siswa
5. Keluar

pilih menu

4

Masukkan nama siswa yang ingin dicari

udin

Nama siswa tidak ditemukan

Menu:

1. Tambah Nama siswa
2. Tampilkan Daftar Nama siswa
3. Hapus Nama Siswa
4. Cari Nama Siswa
5. Keluar

pilih menu

5

Keluar dari program

D. Kesimpulan

Melalui praktikum ini, mahasiswa mampu memahami dan mengimplementasikan ArrayList dalam berbagai situasi. Penggunaan ArrayList memungkinkan pengelolaan data dinamis yang lebih fleksibel dibanding array biasa. Operasi-operasi seperti penambahan, penghapusan, penyisipan, serta pencarian menjadi lebih mudah dilakukan berkat metode bawaan yang tersedia.