

LAPORAN PRAKTIKUM STRUKTUR DATA
ARRAYLIST



Oleh :

AHMAD ZAHRAN

NIM:

2411532004

MATA KULIAH STRUKTUR DATA
DOSEN PENGAMPU: WAHYUDI, S.T., M.T.
FAKULTAS TEKNOLOGI INFORMASI
DEPARTEMEN INFORMATIKA
UNIVERSITAS ANDALAS

2025

A. Pendahuluan

ArrayList di Java adalah bagian dari Framework Collections Java dan Merupakan kelas dari paket `java.util` ArrayList ini dinamis pada java. Keuntungan utama dari ArrayList adalah kita tidak perlu menyebutkan ukurannya seperti pada pendeklarasian array biasanya. Beberapa metode ArrayList yang berguna seperti `add()`, `remove()`, `get()`, dan `size()` untuk mengelola dan mengakses data secara efisien. ArrayList biasanya digunakan ketika pengguna membutuhkan daftar objek yang fleksibel yang dapat berubah ukurannya selama runtime program.

B. Tujuan

1. Mengetahui cara menggunakan Arraylist pada Java.
2. Mengimplementasikan ArrayList untuk membuat program manajemen Mahasiswa sederhana

C. Langkah-Langkah

1. ArrayList1

- 1) Buatlah
- 2) Setelah itu, lakukan import ArrayList.

```
package pekan1;

import java.util.ArrayList;
```

3) `public class` ArrayList1 {

- 4) Deklarasikan Arraylist arrli dengan tipe data integer dengan kapasitas awal yaitu integer n.

- 5)

```
int n = 5;
ArrayList<Integer> arrli = new ArrayList<Integer>(n);
```

 5 ke dalam ArrayList arrli. Setelah loop, arrli berisi: [1, 2, 3, 4, 5].

```
for (int i = 1; i <= n; i++) {
    arrli.add(i);
}
```

- 6) Print arrli.

```
System.out.println(arrli);
```

- 7) Hapus index arraylist ketiga.

```
arrli.remove(3);
```

- 8) Print kembali arraylist arrli.

```
System.out.println(arrli);
```

- 9) Ketikkan for loop yang akan mencetak setiap elemen dalam list dengan spasi.

```
for (int i = 0; i < arrli.size(); i++) {
    System.out.print(arrli.get(i) + " ");
}

}
```

10) Berikut adalah output dari program ini:

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

2. ArrayListGetElement

- 1) Buatlah kelas baru dengan nama ArrayListGetElement dan juga kelas main.
- 2) Setelah itu, lakukan import ArrayList.

```
package pekan1;

import java.util.ArrayList;

public class ArrayListGetElement {

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

- 3) Deklarasikan ArrayList list dengan tipe data integer.

```
        ArrayList<Integer> list = new ArrayList<Integer>();
```

- 4) Tambahkan tiga integer ke list, yaitu 9, 5, dan 6 dengan mengetikkan list.add();. Arraylist list akan berbentuk seperti ini: [9, 5, 6].

```
        list.add(9);
        list.add(5);
        list.add(6);
```

- 5) Print arraylist list.

```
        System.out.println(list);
```

- 6) Get elemen pada index ke 1 dengan mengetikkan Integer n = list.get(1). Jadi, n akan bernilai 5.
- 7) Print element yang didapatkan setelah get element.

```
        Integer n = list.get(1);
        System.out.println("Pada indeks ke 1 angkanya adalah:" + n);

    }

}
```

- 8) Output program adalah sebagai berikut:

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

3. ArrayListSisip

- 1) Buat kelas baru dengan nama ArrayListSisip dan juga kelas main.
- 2) Import ArrayList dengan import java.util.ArrayList;.

```
package pekan1;

import java.util.ArrayList;

public class ArrayListSisip {

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

- 3) Deklarasikan Arraylist list dengan tipe data integer.
- 4) Tambahkan integer 1, 2, 4 pada list sehingga ArrayList akan berbentuk: [1, 2, 4].

```
ArrayList<Integer> list = new ArrayList<Integer>();
list.add(1);
list.add(2);
list.add(4);
```

- 5) Print list.

```
System.out.println(list);
```

- 6) Untuk menyisipkan elemen pada arraylist, ketikkan list.add(index, nilai). Sisipkan integer 3 pada index 2. Setelah disisipkan, arraylist menjadi: [1, 2, 3, 4].

```
list.add(2, 3);
```

- 7) Print list.

```
System.out.println(list);
```

- 8) Berikut adalah output dari program:

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

4. DaftarSiswa

Program Java ini adalah program sederhana yang memungkinkan pengguna mengelola daftar nama siswa menggunakan ArrayList.

- 1) Buat kelas baru dengan nama DaftarSiswa. DaftarNamaSiswa akan menjadi list static yang akan digunakan bersama di semua method.
- 2) Buatlah main method. Lalu, import scanner agar pengguna bisa menginputkan data. Setelah itu, import juga arraylist.

```
package pekan1;

import java.util.ArrayList;

public class DaftarSiswa {
    private static ArrayList<String> daftarNamaSiswa = new ArrayList<>();
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int pilihan;
```

- 3) Inisialisasi variabel pilihan dan scanner.

```
Scanner scanner = new Scanner(System.in);  
int pilihan;
```

- 4) Ketikkan loop do-while. Loop ini akan menampilkan menu berulang kali sampai pengguna memilih opsi 5 yaitu keluar. Pada loop do-while ini, pengguna diminta untuk menginputkan pilihan pada menu.
- 5) Print text untuk tampilan menu. Pengguna diminta untuk menginputkan nomor dari 1 sampai 5 dan program akan mengeksekusi tindakan yang sesuai dengan pernyataan switch.

```
do {  
    System.out.println("\nMenu");  
    System.out.println("1. Tambah Nama Siswa");  
    System.out.println("2. Tampilkan Daftar Nama Siswa");  
    System.out.println("3. Hapus Nama Siswa");  
    System.out.println("4. Cari Nama Siswa");  
    System.out.println("5. Keluar");  
    System.out.print("Pilih Menu");  
    pilihan = scanner.nextInt();  
    scanner.nextLine(); //Consume newLine  
    switch (pilihan) {  
        case 1:  
            tambahNamaSiswa(scanner);  
            break;  
        case 2:  
            tampilkanDaftarNamaSiswa();  
            break;  
        case 3:  
            hapusNamaSiswa(scanner);  
            break;  
        case 4:  
            cariNamaSiswa(scanner);  
            break;  
        case 5:  
            System.out.println("Keluar dari program. ");  
            break;  
        default:  
            System.out.println("Pilihan tidak valid. ");  
    }  
} while (pilihan != 5);
```

- 6) Opsi 1, yaitu tambah nama siswa. Jika pengguna memilih opsi 1, program akan memanggil method `tambahNamaSiswa()`; yang akan menambahkan siswa pada list.

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

- 7) Opsi 2, yaitu menampilkan daftar nama siswa. Jika pengguna memilih opsi 2, program akan memanggil method `tampilkanDaftarNamaSiswa()`; yang akan menampilkan daftar nama siswa. Jika daftar kosong, program akan menge-print pesan. Digunakan kondisional `if else` dan juga `for each` loop untuk menampilkan daftar nama siswa.

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

- 8) Opsi 3, yaitu menghapus nama siswa. Jika pengguna memilih opsi 3, program akan memanggil method `hapusNamaSiswa()`. Method ini akan meminta pengguna untuk memasukkan nama siswa yang akan dihapus dari daftar. Jika terdapat nama siswa tersebut, program akan menghapus nama siswa tersebut. Namun, jika nama siswa tidak ditemukan, program akan menge-print pesan siswa tidak ada.

```
private static void hapusNamaSiswa(Scanner scanner) {
    System.out.println("Masukkan nama siswa yang akan dihapus");
    String nama = scanner.nextLine();
    if (daftarNamaSiswa.remove(nama)) {
        System.out.println("Nama siswa berhasil dihapus. ");
    } else {
        System.out.println("Nama siswa tidak ditemukan. ");
    }
}
```


- 9) Opsi 4, yaitu mencari nama siswa. Jika pengguna memilih opsi 4, program akan mengecek apakah nama siswa ada pada daftar. Jika nama siswa terdapat pada daftar, program akan menampilkan nama siswa. Namun, jika nama siswa tidak terdapat pada daftar, program akan menampilkan pesan siswa tidak ditemukan.

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

- 10) Opsi 5, keluar dari program. Jika pengguna memilih opsi 5, program akan menampilkan pesan keluar dan program akan berhenti (loop berhenti).
11) Berikut tampilan dari program ini:

```
Menu  
1. Tambah Nama Siswa  
2. Tampilkan Daftar Nama Siswa  
3. Hapus Nama Siswa  
4. Cari Nama Siswa  
5. Keluar  
Pilih Menu|
```

5. Mahasiswa

Kelas Java ini mendefinisikan model sederhana untuk merepresentasikan Mahasiswa.

- 1) Deklarasikan kelas baru dengan nama Mahasiswa.

```
package pekan1;  
  
public class Mahasiswa {
```

- 2) Inisialisasi variabel – variabel berikut, yaitu nim, nama, dan prodi sebagai tipe data string.

```
String nim;  
String nama;  
String prodi;
```

- 3) Buatlah konstruktor yang akan berjalan setiap kali objek mahasiswa baru dibuat. Konstruktor ini akan menerima nim, nama, dan prodi sebagai parameter. Penggunaan this. Untuk merujuk ke variabel-variabel turunan dan memberikan nilai pada variabel-variabel tersebut.


```

    Mahasiswa(String nim, String nama, String prodi) {
        this.nim = nim;
        this.nama = nama;
        this.prodi = prodi;
    }

```

- 4) Definisikan ulang method toString() dengan @Override bawaan dari kelas object. Method ini akan mengontrol apa yang dicetak ketika pengguna menetak objek.

```

@Override
public String toString() {
    return "NIM: " + nim + ", Nama: " + nama + ", Prodi: " + prodi;
}

```

6. MahasiswaMain

Program MahasiswaMain adalah sebuah sistem manajemen siswa berbasis konsol. Program ini memungkinkan pengguna untuk menambah, melihat, menghapus, dan mencari data mahasiswa menggunakan ArrayList dari objek Mahasiswa.

- 1) Buatlah kelas baru dengan nama MahasiswaMain.
- 2) Import arraylist dan juga scanner.

```

package pekan1;

import java.util.ArrayList;

public class MahasiswaMain {
    public static void main(String[] args) {

```

- 3) Pada main method, deklarasikan arraylist mahasiswa dan juga scanner. Inisialisasi juga variabel choice.

```

        ArrayList<Mahasiswa> mahasiswaList = new ArrayList<>();
        Scanner scanner = new Scanner(System.in);
        int choice;

```

- 4) Ketikkan do-while loop. Do-while loop akan menampilkan menu sampai pengguna memilih untuk keluar. Loop ini menggunakan switch-case untuk menangani opsi yang berbeda.

```

do {
    System.out.println("Menu");
    System.out.println("1. Tambah Mahasiswa");
    System.out.println("2. Tampilkan Semua Mahasiswa");
    System.out.println("3. Hapus Mahasiswa Berdasarkan NIM");
    System.out.println("4. Cari Mahasiswa Berdasarkan NIM");
    System.out.println("5. Keluar");
    System.out.print("Pilih Menu");
    choice = scanner.nextInt();
    scanner.nextLine(); // Consume newline
    switch (choice) {

```

- 5) Opsi 1, yaitu menambahkan mahasiswa. Program ini akan meminta data mahasiswa kepada pengguna. Lalu, program ini akan membuat objek mahasiswa baru dan menambahkannya ke mahasiswaList.

```
case 1:
    System.out.print("Masukkan NIM: ");
    String nim = scanner.nextLine();
    System.out.println("Masukkan Nama: ");
    String nama = scanner.nextLine();
    System.out.println("Masukkan Prodi: ");
    String prodi = scanner.nextLine();
    mahasiswaList.add(new Mahasiswa(nim, nama, prodi));
    break;
```

- 6) Opsi 2, yaitu menampilkan daftar nama mahasiswa. Program akan menampilkan daftar nama siswa menggunakan for each loop.

```
case 2:
    System.out.println("Data Mahasiswa:");
    for (Mahasiswa mhs : mahasiswaList) {
        System.out.println(mhs);
    }
    break;
```

- 7) Opsi 3, menghapus mahasiswa berdasarkan nim. Program akan menggunakan ekspresi lambda untuk menghapus siswa sesuai dengan nim yang diinputkan oleh pengguna.

```
case 3:
    System.out.println("Masukkan NIM yang akan dihapus: ");
    String nimHapus = scanner.nextLine();
    mahasiswaList.removeIf(mhs -> mhs.nim.equals(nimHapus));
    break;
```

- 8) Opsi 4, mencari mahasiswa berdasarkan nim. Program akan menggunakan loop for untuk mencari nim yang cocok dengan inputan pengguna.

```
case 4:
    System.out.println("Masukkan NIM yang dicari: ");
    String nimCari = scanner.nextLine();
    for (Mahasiswa mhs : mahasiswaList) {
        if (mhs.nim.equals(nimCari)) {
            System.out.println(mhs);
            break;
        } else {
            System.out.println("nim tidak ada");
        }
    }
    break;
```

9) Opsi 5, keluar dari program dan menutup scanner.

```
        case 5:
            System.out.println("Keluar dari program.");
            break;
        default:
            System.out.println("Pilihan tidak valid.");
        }
    } while (choice != 5);

    scanner.close();
}
```

10) Berikut adalah output dari program:

```
Menu
1. Tambah Mahasiswa
2. Tampilkan Semua Mahasiswa
3. Hapus Mahasiswa Berdasarkan NIM
4. Cari Mahasiswa Berdasarkan NIM
5. Keluar
Pilih Menu
```

D. Kesimpulan

ArrayList adalah sebuah struktur data dinamis di Java yang digunakan untuk menyimpan dan mengelola sebuah daftar objek yang dapat bertambah atau berkurang sesuai kebutuhan. Pada proyek sebelumnya, struktur ini digunakan untuk mengelola kumpulan data seperti nama mahasiswa (String) dan objek mahasiswa (Mahasiswa). ArrayList memungkinkan program untuk secara efisien menambah, menghapus, menampilkan, dan mencari informasi mahasiswa melalui input pengguna, sehingga ideal untuk membangun sistem manajemen data yang sederhana dan interaktif tanpa perlu menentukan jumlah entri.