

LAPORAN PRAKTIKUM STRUKTUR DATA
PEKAN 1: PENGGUNAAN ARRAYLIST UNTUK MANIPULASI
DATA DINAMIS DALAM BAHASA JAVA



OLEH

Abdullah Al Ramadhani (2411533016)

DOSEN PENGAMPU

DR. Wahyudi, S.T, M.T

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

2025

A. Pendahuluan

Dalam pemrograman Java, struktur data memiliki fungsi yang sangat penting untuk pemrosesan dan manajemen data. ArrayList merupakan salah satu struktur data yang umum diimplementasikan. ArrayList adalah komponen dari framework Collection dalam Java yang memfasilitasi penyimpanan data dalam format list yang fleksibel, dengan kapasitas yang dapat bertambah atau berkurang secara otomatis.

Tidak seperti array biasa yang memiliki dimensi tetap, ArrayList memberikan fleksibilitas bagi pengembang untuk melakukan penambahan, penghapusan, dan pengaksesan elemen. ArrayList juga dilengkapi dengan berbagai metode internal seperti add(), remove(), get(), dan contains() yang mengoptimalkan proses pengelolaan data

B. Tujuan

Tujuan dari praktikum ini adalah sebagai berikut:

1. Untuk memahami konsep dasar ArrayList dalam bahasa pemrograman Java.
2. Untuk mengetahui perbedaan antara ArrayList dan array biasa.
3. Untuk mempraktikkan cara menambahkan, menghapus, menampilkan, dan mencari elemen dalam ArrayList.

C. Langkah – langkah Pengerjaan

A. Program ArrayList1

1. Buatlah sebuah package baru dengan nama “**pekan1**”, lalu tambahkan sebuah class di dalamnya yang diberi nama ArrayList1.
2. Setelah itu, lakukan import terhadap kelas ArrayList yang berada dalam package java.util. Kelas ArrayList ini berfungsi untuk menyimpan data yang sifatnya fleksibel, artinya jumlah data bisa bertambah atau berkurang tergantung kebutuhan program.

```
1 package Pekan1;  
2 import java.util.ArrayList;
```

3. Buatlah sebuah class bernama ArrayList1 sebagai tempat menuliskan kode program. Dalam Java, setiap program minimal terdiri dari satu class atau lebih yang berfungsi sebagai wadah untuk mengelola kode. Untuk menandai awal eksekusi program, tambahkan method main, karena method inilah yang akan dijalankan pertama kali saat program dijalankan. Semua kode yang ingin dieksekusi harus ditempatkan di dalam method ini.

4. Deklarasikan sekaligus beri nilai awal untuk variabel $n = 5$. Variabel ini nantinya akan dipakai sebagai batas perulangan. Nilai dari n digunakan untuk menentukan berapa banyak elemen yang ingin kita tambahkan ke dalam ArrayList. Dengan n diatur ke angka 5, berarti kita akan memasukkan 5 data ke dalam ArrayList.

```
4 public class ArrayList1 {  
5  
6     public static void main(String[] args) {  
7         //Size of the ArrayList  
8         int n=5;
```

5. Buat sebuah objek ArrayList dengan kapasitas awal sebanyak n elemen. Di tahap ini, kita akan membuat objek ArrayList yang mampu menyimpan data bertipe Integer. Kapasitas awal dari ArrayList ini disesuaikan dengan nilai n (yaitu 5), yang artinya akan disiapkan ruang untuk menyimpan 5 data pertama. Meskipun begitu, kapasitasnya bisa bertambah secara otomatis kalau nanti diperlukan.
6. Tambahkan angka 1 sampai 5 ke dalam ArrayList menggunakan perulangan for. Dengan memakai perulangan for, kita bisa menambahkan elemen angka dari 1 sampai 5 ke dalam ArrayList. Gunakan method `add()` untuk menambahkan data secara urut ke dalam ArrayList.

```
ArrayList<Integer> arrli = new ArrayList<Integer>(n);  
for (int i = 1; i <= n; i++)  
    arrli.add(i);
```

7. Tampilkan isi dari ArrayList ke layar menggunakan `System.out.println()` supaya kita bisa melihat semua elemen yang ada di dalamnya. Ini berguna untuk memastikan bahwa data yang kita tambahkan sudah masuk dengan benar.
8. Setelah itu, kita akan coba menghapus elemen yang ada di indeks ke-3 (atau elemen ke-4 kalau dilihat dari urutan aslinya, karena indeks di Java mulai dari 0) dengan menggunakan method `remove()`.
9. Terakhir, kita tampilkan lagi isi ArrayList setelah proses penghapusan tadi untuk melihat hasil akhirnya.

```
System.out.println(arrli);  
arrli.remove(3);  
System.out.println(arrli);
```

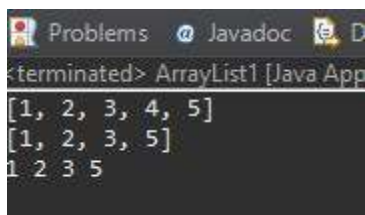
10. Gunakan perulangan for bersama dengan method `get()` untuk mengambil dan menampilkan setiap elemen dalam `ArrayList` satu per satu. Dengan cara ini, kita juga bisa melihat bagaimana cara mengakses elemen berdasarkan indeksnya.

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

11. Kode Program yang lengkap

```
1 package Pekan1;  
2 import java.util.ArrayList;  
3  
4 public class ArrayList1 {  
5  
6     public static void main(String[] args) {  
7         //Size of the ArrayList  
8         int n=5;  
9         ArrayList<Integer> arrli = new ArrayList<Integer>(n);  
10        for (int i = 1; i <= n; i++)  
11            arrli.add(i);  
12        System.out.println(arrli);  
13        arrli.remove(3);  
14        System.out.println(arrli);  
15        for (int i = 0; i < arrli.size(); i++)  
16            System.out.print(arrli.get(i) + " ");  
17  
18    }  
19  
20 }  
21
```

12. Silakan jalankan programnya, dan nanti akan muncul output seperti contoh yang telah disiapkan.



```
Problems Javadoc D  
<terminated> ArrayList1 [Java App  
[1, 2, 3, 4, 5]  
[1, 2, 3, 5]  
1 2 3 5
```

B. Program ArrayListGetElemen

1. Buatlah sebuah class baru dengan nama “ArrayListGetElemen”.
2. Lakukan import terhadap library ArrayList dari Java, lalu buat class dengan nama ArrayListGetElemen.

```
package Pekan1;  
import java.util.ArrayList;  
  
public class ArrayListGetElemen {
```

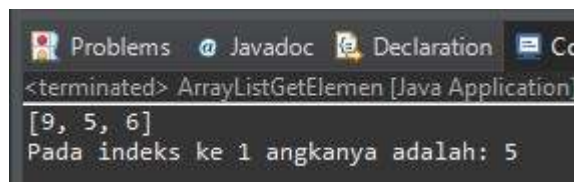
3. Deklarasikan dan inisialisasi sebuah objek ArrayList bertipe Integer yang nantinya akan digunakan untuk menyimpan data bertipe angka (Integer).
4. Tambahkan beberapa elemen ke dalam ArrayList menggunakan method add(). Setelah itu, tampilkan semua isi dari ArrayList yang sudah diisi tadi.

```
    public static void main(String[] args) {  
        ArrayList<Integer> list = new  
            ArrayList<Integer>();  
        list.add(9);  
        list.add(5);  
        list.add(6);  
        System.out.println(list);  
    }
```

5. Berikutnya, kita akan mencoba mengambil elemen yang ada di indeks pertama (atau urutan ke-2 secara posisi dalam daftar) menggunakan method get(). Nilai yang diambil akan disimpan dalam sebuah variabel bernama n, kemudian ditampilkan ke layar.

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

6. Jalankan programnya, dan kamu akan melihat output seperti yang diharapkan.



```
<terminated> ArrayListGetElemen [Java Application]  
[9, 5, 6]  
Pada indeks ke 1 angkanya adalah: 5
```

C. Program ArrayListSisip

1. Buat sebuah class baru dengan nama “ArrayListSisip”.
2. Lakukan import terhadap library ArrayList dari Java seperti biasa, lalu buat kelas dengan nama ArrayListSisip.

```
1 package Pekan1;  
2 import java.util.ArrayList;  
3  
4 public class ArrayListSisip {  
5
```

3. Di dalam method main, buat sebuah objek ArrayList dengan tipe data Integer yang akan digunakan untuk menyimpan angka-angka. Kemudian, tambahkan elemen 1, 2, dan 4 ke dalam ArrayList menggunakan method add().

```
public static void main(String[] args) {  
    ArrayList<Integer> list = new  
        ArrayList<Integer>();  
    list.add(1);  
    list.add(2);  
    list.add(4);
```

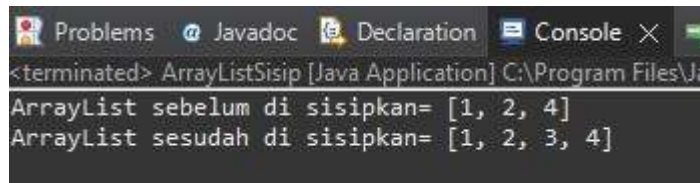
4. Tampilkan isi dari list yang sudah diisi tersebut sebelum elemen baru disisipkan, agar kita bisa melihat perbedaannya nanti.

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

5. Setelah itu, kita akan menyisipkan elemen baru ke dalam ArrayList tadi. Kali ini kita akan menambahkan angka 3 di indeks ke-2. Nantinya, elemen yang ada di posisi tersebut dan setelahnya akan bergeser untuk memberi tempat pada angka 3. Kemudian cetak kembali isi ArrayList setelah penyisipan dilakukan.

```
System.out.println(list);  
list.add(2, 3);  
System.out.println(list);  
}
```

6. Jalankan program dan perhatikan perbedaan isi ArrayList sebelum dan sesudah elemen disisipkan.



```
<terminated> ArrayListSisip [Java Application] C:\Program Files\Ja
ArrayList sebelum di sisipkan= [1, 2, 4]
ArrayList sesudah di sisipkan= [1, 2, 3, 4]
```

D. Program Daftar Siswa

1. Buatlah sebuah class baru dengan nama DaftarSiswa, seperti yang telah dilakukan di tugas-tugas sebelumnya.
2. Pertama-tama, kita perlu import dua library penting: ArrayList untuk menyimpan data secara fleksibel, dan Scanner untuk membaca input dari keyboard. ArrayList memungkinkan kita menambahkan atau menghapus elemen tanpa batasan jumlah, sementara Scanner digunakan agar program bisa menerima input dari pengguna.

```
1 package Pekan1;
2 import java.util.ArrayList;
```

3. Deklarasikan kelas DaftarSiswa, yang akan menjadi wadah utama seluruh program. Di dalam kelas ini, kita buat sebuah ArrayList<String> dengan nama daftarNamaSiswa untuk menyimpan nama-nama siswa. Karena variabel ini menggunakan keyword static, maka bisa langsung diakses oleh method main tanpa harus membuat objek dari kelas tersebut.

```
private static ArrayList<String> daftarNamaSiswa = new ArrayList<>();
public static void main(String[] args) {
```

4. Buatlah method main, dan tambahkan objek Scanner untuk menerima input dari pengguna.

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


5. Gunakan perulangan do-while agar menu bisa ditampilkan secara berulang. Program akan terus meminta input dari pengguna sampai mereka memilih opsi keluar (pilihan nomor 5). Kita juga perlu menggunakan `scanner.nextLine()` setelah membaca angka agar tidak terjadi masalah saat membaca input berupa teks.

```
int pilihan;
do {
    System.out.println("\nMenu");
    System.out.println("1. Tambahkan 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();
    switch (pilihan) {
```

6. Buat case 1 untuk menambahkan nama siswa. Program akan meminta pengguna memasukkan nama, lalu nama tersebut langsung dimasukkan ke dalam ArrayList menggunakan method `add()`.

```
case 1 :
    tambahNamaSiswa(scanner);
    break;
```

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

7. Di case 2, tampilkan semua nama siswa yang telah dimasukkan. Gunakan perulangan for-each agar setiap nama dalam ArrayList bisa ditampilkan satu per satu dengan mudah.

```
case 2 :
    tampilkanDaftarNamaSiswa();
    break;
```

```
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. Tambahkan case 3 untuk menghapus nama siswa. Pengguna akan diminta memasukkan nama yang ingin dihapus, dan kita gunakan method `remove()` untuk menghapusnya dari daftar.

```
case 3 :  
    hapusNamaSiswa(scanner);  
    break;
```

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

9. Buat juga case 4 untuk fitur pencarian nama siswa. Gunakan method `contains()` untuk memeriksa apakah nama yang dimasukkan ada dalam daftar. Jika ada, tampilkan pesan “Ditemukan”.

```
case 4 :  
    cariNamaSiswa(scanner);  
    break;
```

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

10. Terakhir, jika pengguna memilih menu keluar (case 5), tampilkan pesan “Keluar dari program” dan hentikan perulangan agar program berhenti.

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

11. Saat dicoba, gunakan menu 1 untuk menambahkan 3 nama siswa. Lalu, pilih menu 2 untuk menampilkan semua nama yang sudah dimasukkan. Coba hapus salah satu nama dengan menu 3, dan cek lagi dengan menu 2 apakah sudah terhapus. Terakhir, gunakan menu 4 untuk mencari nama tertentu dalam daftar.

```
Menu
1. Tambahkan nama siswa
2. Tampilkan daftar nama siswa
3. Hapus nama siswa
4. Cari nama siswa
5. Keluar
Pilih Menu: 1
Masukkan Nama Siswa: Dhanu
Nama siswa berhasil ditambahkan.

Menu
1. Tambahkan nama siswa
2. Tampilkan daftar nama siswa
3. Hapus nama siswa
4. Cari nama siswa
5. Keluar
Pilih Menu: 1
Masukkan Nama Siswa: Teguh
Nama siswa berhasil ditambahkan.

Menu
1. Tambahkan nama siswa
2. Tampilkan daftar nama siswa
3. Hapus nama siswa
4. Cari nama siswa
5. Keluar
Pilih Menu: 1
Masukkan Nama Siswa: Fadil
Nama siswa berhasil ditambahkan.
```

```
Menu
1. Tambahkan nama siswa
2. Tampilkan daftar nama siswa
3. Hapus nama siswa
4. Cari nama siswa
5. Keluar
Pilih Menu: 2
Daftar Nama Siswa:
Dhani
Teguh
Fadil
```

```
Menu
1. Tambahkan nama siswa
2. Tampilkan daftar nama siswa
3. Hapus nama siswa
4. Cari nama siswa
5. Keluar
Pilih Menu: 3
Masukkan nama siswa yang akan di hapus: Fadil
Nama siswa berhasil dihapus.

Menu
1. Tambahkan nama siswa
2. Tampilkan daftar nama siswa
3. Hapus nama siswa
4. Cari nama siswa
5. Keluar
Pilih Menu: 2
Daftar Nama Siswa:
Dhani
Teguh
```

```
Menu
1. Tambahkan 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 di cari: Dhani
Nama siswa ditemukan: Dhani
```

E. Mahasiswa

1. Buat sebuah class baru dengan nama "Mahasiswa". Class ini akan berfungsi sebagai cetakan (blueprint) untuk membuat objek-objek mahasiswa.
2. Tambahkan atribut data untuk masing-masing mahasiswa, yaitu nim (nomor induk mahasiswa), nama (nama lengkap), dan prodi (program studi). Ketiga atribut ini bersifat non-static, artinya setiap objek Mahasiswa yang dibuat akan memiliki data masing-masing.
3. Buatlah sebuah konstruktor untuk menginisialisasi data mahasiswa saat objek dibuat. Gunakan keyword this untuk membedakan antara parameter yang dikirim ke konstruktor (misalnya nim, nama, prodi) dengan atribut yang ada dalam class. Jadi,

nilai-nilai yang dikirim saat membuat objek akan langsung dimasukkan ke atribut objek tersebut.

```
ArrayList1.java  ArrayListGetElemen.java  ArrayListSisip.java  DaftarS
1 package Pekan1;
2
3 public class Mahasiswa {
4     String nim;
5     String nama;
6     String prodi;
7     Mahasiswa (String nim, String nama, String prodi) {
8         this.nim = nim;
9         this.nama = nama;
10        this.prodi = prodi;|
11    }
```

4. Tambahkan method toString() agar saat objek Mahasiswa dicetak (misalnya dengan System.out.println(mhs)), program akan otomatis menampilkan isi data lengkap dari objek tersebut. Method ini merupakan override dari method bawaan class Object.

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

F. MahasiswaMain

1. Buat class baru dengan nama “MahasiswaMain”. Kelas ini akan menjadi program utama tempat semua logika aplikasi dijalankan.
2. Import library yang diperlukan, yaitu ArrayList untuk menyimpan data mahasiswa dalam bentuk list, dan Scanner untuk menerima input dari pengguna.

```
1 package Pekan1;
2 import java.util.ArrayList;|
3
4
5 public class MahasiswaMain {
6
```

3. Buat ArrayList untuk menyimpan semua objek Mahasiswa yang ditambahkan. ArrayList ini bersifat dinamis, sehingga memudahkan untuk menambah atau menghapus elemen.

```
public static void main(String[] args) {  
    ArrayList<Mahasiswa> mahasiswaList = new ArrayList<>();
```

4. Buat objek Scanner untuk membaca input dari pengguna dan deklarasikan variabel untuk pilihan menu.

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

5. Gunakan do-while loop untuk menampilkan menu interaktif. Menu akan terus ditampilkan selama pengguna belum memilih untuk keluar (yaitu choice != 5). Gunakan nextLine() untuk menangani masalah newline setelah input angka, supaya input berikutnya terbaca dengan benar.

```
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(); // Mengonsumsi newline
```

6. Pada case 1, buat fitur untuk menambah mahasiswa baru berdasarkan input dari pengguna. Objek Mahasiswa akan dibuat dengan menggunakan konstruktor dan langsung dimasukkan ke dalam mahasiswaList.

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

7. Pada case 2, buat fitur untuk menampilkan seluruh data mahasiswa. Karena method toString() sudah di-override, setiap objek mahasiswa yang ditampilkan akan tampil dalam format yang rapi.

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

8. Pada case 3, buat fitur untuk menghapus mahasiswa berdasarkan NIM. Program akan meminta pengguna untuk memasukkan NIM yang ingin dihapus, kemudian data mahasiswa tersebut akan dihapus dari mahasiswaList.

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

9. Pada case 4, buat fitur untuk mencari mahasiswa berdasarkan NIM. Jika mahasiswa dengan NIM tersebut ditemukan, program akan menampilkan data mahasiswa tersebut.

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


10. Pada case 5, buat opsi untuk keluar dari program. Program akan berhenti dan perulangan akan dihentikan. Jangan lupa untuk menutup scanner setelah program selesai.

```
        case 5:
            System.out.println("Keluar dari program.");
            break;

        default:
            System.out.println("Pilihan tidak valid.");
    }
} while (choice != 5);

scanner.close();
}
```

11. Jalankan program dan coba semua menu yang sudah dibuat.

(gambar belum ada karna belum coding sendiri)

D. KESIMPULAN

Melalui praktikum ini, saya jadi lebih paham bagaimana cara mengimplementasikan struktur data **ArrayList** di dalam bahasa pemrograman **Java**. Dibandingkan dengan array biasa, penggunaan **ArrayList** jauh lebih fleksibel, terutama saat ingin menambahkan, menghapus, mencari, atau menampilkan data secara dinamis.

Praktikum ini bukan hanya membantu saya memahami konsep dasar **ArrayList**, tapi juga melatih kemampuan logika dan pemrograman dalam menyusun kode yang efisien dan terorganisir dengan baik. Setelah menyelesaikan praktikum ini, saya merasa lebih siap untuk membuat aplikasi yang lebih kompleks dengan memanfaatkan struktur data yang tepat.