

LAPORAN PRAKTIKUM
STRUKTUR DATA



DISUSUN OLEH :

Sasya Zamora

2411533014

DOSEN PENGAMPU :

Dr. Wahyudi, S.T.M.T

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS

I. TUJUAN

1. Mahasiswa mampu mengetahui konsep ArrayList pada Java
2. Mahasiswa mampu mengetahui konsep macam-macam ArrayList pada Java
3. Mahasiswa mampu membuat program menggunakan ArrayList

II. PEMBAHASAN

Salah satu struktur data Java yang paling populer adalah ArrayList, yang memungkinkan Anda menyimpan dan mengubah koleksi objek secara dinamis. Karena Java ArrayList bersifat dinamis, ukurannya bertambah dan menyusut seiring dengan penambahan atau penghilangan elemen. Perlu diingat bahwa kelas ArrayList tidak tersedia secara asli karena merupakan bagian dari Java Collections Framework. Tidak seperti array, array perlu diimpor dari perpustakaan java.util.

Berikut adalah beberapa fitur dan karakteristik utama dari ArrayList:

1. Dinamis: Ukurannya bisa bertambah atau berkurang secara dinamis sesuai dengan penambahan atau penghapusan elemen.
2. Indeks Berbasis Nol: Elemen dalam ArrayList diakses menggunakan indeks, yang dimulai dari nol.
3. Mendukung Duplikasi: ArrayList bisa menyimpan elemen duplikat.
4. Mempertahankan Urutan: Elemen dalam ArrayList disimpan dalam urutan penyisipan.
5. Tidak Sinkron: ArrayList tidak thread-safe secara default. Jika perlu thread-safety, bisa menggunakan `Collections.synchronizedList(new ArrayList<>());`

Macam-Macam ArrayList Berdasarkan Penggunaan

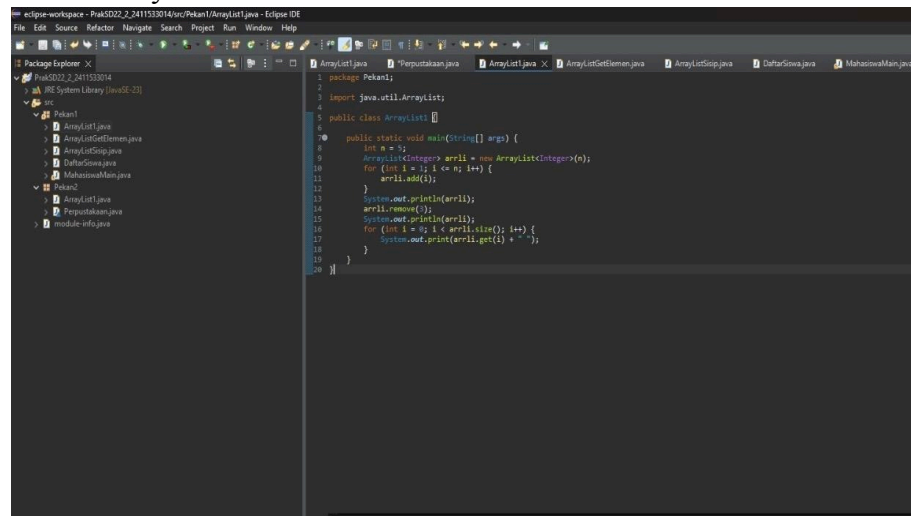
1. ArrayList dari Tipe Data Primitive Wrapper:
 - `ArrayList<Integer>`: Untuk menyimpan angka bulat.
 - `ArrayList<Double>`: Untuk menyimpan angka desimal.
 - `ArrayList<Character>`: Untuk menyimpan karakter.
2. ArrayList dari Objek Kelas:
 - Anda bisa membuat ArrayList yang menyimpan objek dari kelas yang Anda buat sendiri.
 - Contoh: `ArrayList<Person>` dimana `Person` adalah kelas yang Anda definisikan.
3. ArrayList Multi-Dimensional:
 - Anda bisa membuat ArrayList yang menyimpan ArrayList lain.
 - Contoh: `ArrayList<ArrayList<String>>` untuk menyimpan tabel data atau daftar dua dimensi.

ArrayList adalah alat yang sangat fleksibel dan serbaguna untuk menyimpan dan mengelola kumpulan data di Java. Dengan memahami dasar-dasar dan macam-macam penggunaannya, Anda bisa memanfaatkannya dengan lebih efektif dalam pengembangan aplikasi Java.

III. LANGKAH PRAKTIKUM

1. Buka Eclipse IDE
2. Buat Java Project baru dengan nama PrakSD22
3. Klik kanan pada project lalu klik New, Package
4. Lalu buat nama Package Pekan1
5. Buat class pada pekan1 dengan nama ArrayList1
6. Buatlah kodingan seperti di bawah ini

7. Class ArrayList1



import java.util.ArrayList;

Baris ini mengimpor kelas ArrayList dari paket java.util. ArrayList adalah struktur data dinamis yang dapat menyimpan sejumlah elemen dan ukurannya dapat berubah selama runtime.

```
public class ArrayList1 {
```

Ini mendeklarasikan kelas publik bernama ArrayList1. Kata kunci public menunjukkan bahwa kelas ini dapat diakses dari kelas lain.

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

```
        int n = 5;
```

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

Metode main adalah titik awal eksekusi program. Sebuah variabel integer n diinisialisasi dengan nilai 5, yang akan digunakan sebagai kapasitas awal ArrayList. Kemudian, sebuah ArrayList bernama arrli dibuat dengan tipe data Integer dan kapasitas awal n. Perlu diperhatikan bahwa kapasitas awal hanya merupakan saran; ArrayList masih bisa tumbuh melebihi kapasitas ini.

```
        for (int i = 1; i <= n; i++) {
```

```
            arrli.add(i);
```

```
        }
```

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

Loop for ini menambahkan angka dari 1 hingga n (5) ke dalam ArrayList. `arlli.add(i)` menambahkan integer `i` ke akhir ArrayList. `System.out.println(arlli)` mencetak isi ArrayList ke konsol.

```
arlli.remove(3); // Menghapus elemen pada indeks 3 (ingat, indeks dimulai dari 0)
```

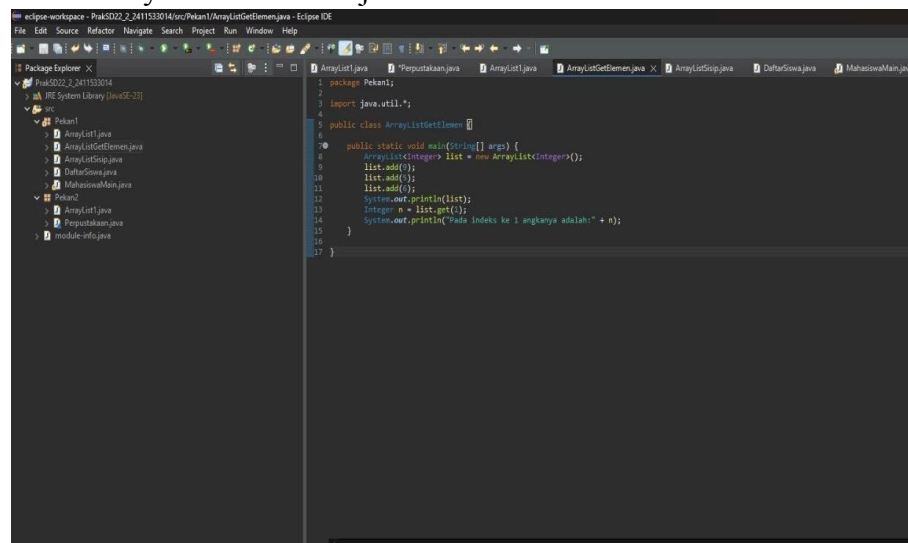
```
System.out.println(arlli);
```

`arlli.remove(3)` menghapus elemen pada indeks 3 (yang merupakan angka 4). Perhatikan bahwa penghapusan elemen akan menggeser indeks elemen-elemen berikutnya.

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

Loop for ini mengiterasi melalui ArrayList menggunakan metode `arlli.size()` untuk mendapatkan jumlah elemen dan `arlli.get(i)` untuk mengakses elemen pada indeks `i`. Elemen-elemen dicetak ke konsol, dipisahkan oleh spasi.

8. Class ArrayListGetElemen.java



```
package Pekan1;  
import java.util.*;
```

Kode ini mendeklarasikan paket tempat kelas ini berada (Pekan1) dan mengimpor semua kelas dari paket `java.util`, yang mencakup ArrayList.

```
public class ArrayListGetElemen {  
    public static void main(String[] args) {  
        ArrayList<Integer> list = new ArrayList<Integer>();
```

Kelas ArrayListGetElemen didefinisikan, dan di dalam metode main, sebuah ArrayList bernama list yang berisi objek Integer dibuat.

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

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

```
list.add(7);
```

Tiga angka integer, 5, 6, dan 7, ditambahkan ke dalam ArrayList list.

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

Baris ini mencetak isi ArrayList ke konsol. Outputnya akan berupa [5, 6, 7].

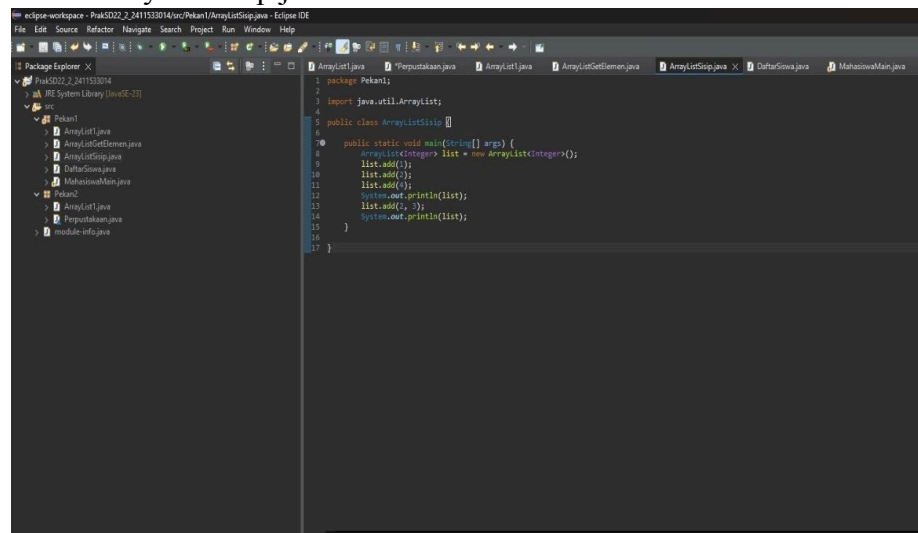
```
Integer n = list.get(1);
```

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

```
}
```

list.get(1) mengakses elemen pada indeks 1 (ingat, indeks dimulai dari 0), yang merupakan angka 6. Angka ini kemudian disimpan dalam variabel n (tipe Integer) dan dicetak ke konsol. Outputnya akan berupa Pada indeks ke 1 angkanya adalah: 6.

9. Class ArrayListSisip.java



```
package Pekan1;
```

```
import java.util.ArrayList;
```

- package Pekan1;: Menyatakan bahwa kode ini termasuk dalam paket bernama "Pekan1". Penggunaan paket membantu mengorganisir kode dalam proyek yang lebih besar dan menghindari konflik nama kelas.

- import java.util.ArrayList;: Mengimpor kelas ArrayList dari paket java.util. Kelas ArrayList adalah bagian dari Java Collections Framework dan menyediakan implementasi dinamis dari array.

```
public class ArrayListSisip {
```

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

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

- public class ArrayListSisip: Mendeklarasikan kelas bernama ArrayListSisip. public berarti kelas ini dapat diakses dari kelas lain dalam proyek.

- `public static void main(String[] args):` Metode `main` adalah titik awal eksekusi program Java.

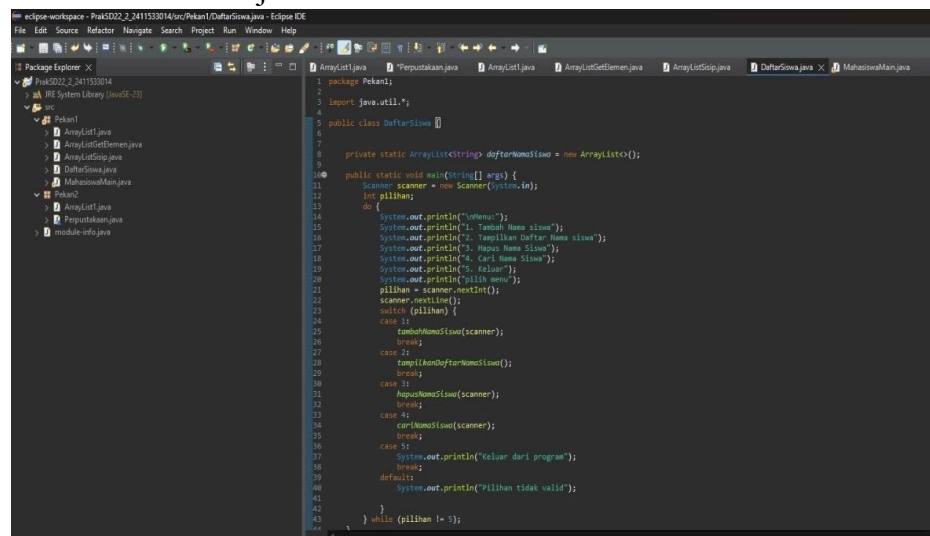
- `ArrayList<Integer> list = new ArrayList<Integer>();`: Membuat objek `ArrayList` bernama `list`. `<Integer>` menentukan bahwa `ArrayList` ini akan menyimpan objek tipe `Integer` (bilangan bulat). `new ArrayList<Integer>()` mengalokasikan memori untuk `ArrayList` baru.

```
list.add(1);
list.add(2);
list.add(3);
System.out.println(list);
}
```

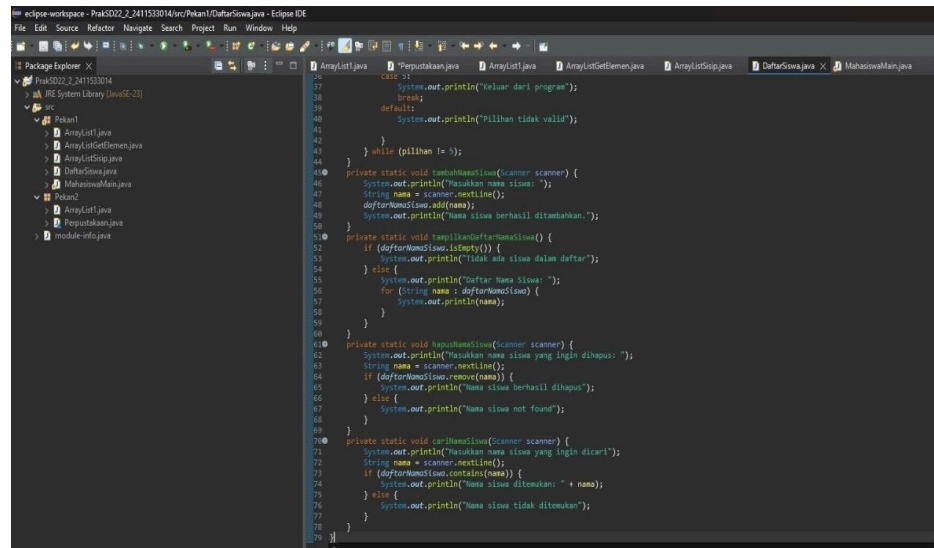
- `list.add(1);`, `list.add(2);`, `list.add(3);`: Tiga baris ini menambahkan angka 1, 2, dan 3 ke dalam `ArrayList list`. Elemen ditambahkan ke akhir `ArrayList`.

- `System.out.println(list);`: Mencetak isi `ArrayList list` ke konsol. Outputnya akan berupa `[1, 2, 3]`.

10. Class `DaftarSiswa.java`



```
1 package Pekani;
2
3 import java.util.*;
4
5 public class DaftarSiswa {
6
7     private static ArrayList<String> daftarNamaSiswa = new ArrayList<>();
8
9     public static void main(String[] args) {
10         Scanner scanner = new Scanner(System.in);
11         int pilihan;
12         do {
13             System.out.println("\nMenu:");
14             System.out.println("1. Tambah Nama siswa");
15             System.out.println("2. Tampilkan Daftar Nama siswa");
16             System.out.println("3. Hapus Nama Siswa");
17             System.out.println("4. Cari Nama Siswa");
18             System.out.println("5. Keluar");
19             System.out.println("Pilih menu:");
20             pilihan = scanner.nextInt();
21             scanner.nextLine();
22             switch (pilihan) {
23                 case 1:
24                     tambahNamaSiswa(scanner);
25                     break;
26                 case 2:
27                     tampilkanDaftarNamaSiswa();
28                     break;
29                 case 3:
30                     hapusNamaSiswa(scanner);
31                     break;
32                 case 4:
33                     cariNamaSiswa(scanner);
34                     break;
35                 case 5:
36                     System.out.println("Keluar dari program");
37                     break;
38                 default:
39                     System.out.println("Pilihan tidak valid");
40             }
41         } while (pilihan != 5);
42     }
43 }
```



```
package Pekan1;
import java.util.*;
```

```
public class DaftarSiswa {
    private static ArrayList<String> daftarNamaSiswa = new ArrayList<>();
```

- package Pekan1;: Menentukan paket tempat kelas ini berada.

- import java.util.*;: Mengimpor semua kelas dari paket java.util, termasuk ArrayList dan Scanner.

- private static ArrayList<String> daftarNamaSiswa = new ArrayList<>();: Mendeklarasikan variabel statis daftarNamaSiswa bertipe ArrayList<String> untuk menyimpan daftar nama siswa. private membatasi akses langsung dari luar kelas, static berarti variabel ini milik kelas itu sendiri, bukan instance kelas. new ArrayList<>() membuat ArrayList kosong.

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    int pilihan;
    do {
        // Menampilkan menu
        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("Pilihan menu: ");
        pilihan = scanner.nextInt();
        scanner.nextLine(); // membersihkan buffer

        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);
scanner.close();
}

```

- Scanner scanner = new Scanner(System.in);: Membuat objek Scanner untuk membaca input dari pengguna.

- do-while loop: Loop ini menampilkan menu pilihan kepada pengguna hingga pengguna memilih opsi 5 (keluar).

- switch statement: Memproses pilihan pengguna dan memanggil metode yang sesuai.

```

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

```

Metode ini meminta input nama siswa dari pengguna menggunakan scanner.nextLine(), lalu menambahkan nama tersebut ke dalam ArrayList daftarNamaSiswa menggunakan daftarNamaSiswa.add().

```

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);
        }
    }
}

```


Metode ini memeriksa apakah ArrayList kosong. Jika kosong, pesan "Tidak ada siswa dalam daftar" ditampilkan. Jika tidak kosong, isi ArrayList (daftar nama siswa) dicetak ke konsol.

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

Metode ini meminta input nama siswa yang akan dihapus. `daftarNamaSiswa.remove(nama)` mencoba menghapus nama dari ArrayList. Metode `remove()` mengembalikan true jika nama ditemukan dan dihapus, dan false jika tidak ditemukan. Pesan yang sesuai ditampilkan berdasarkan hasil operasi penghapusan.

```
private static void cariNamaSiswa(Scanner scanner) {  
    System.out.print("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("Nama siswa tidak ditemukan");  
    }  
}
```

Metode ini meminta input nama siswa yang akan dicari. `daftarNamaSiswa.contains(nama)` memeriksa apakah nama tersebut ada di dalam ArrayList. Pesan yang sesuai ditampilkan berdasarkan hasil pencarian.

11. Class MahasiswaMain.java

```

1 package Pekan1;
2
3 import java.util.ArrayList;
4 import java.util.Scanner;
5 import java.util.*;
6
7 public class MahasiswaMain {
8     private Object nim;
9
10    public static void main(String[] args) {
11        ArrayList<Mahasiswa> mahasiswaList = new ArrayList<>();
12        Scanner scanner = new Scanner(System.in);
13        int choice;
14        do {
15            System.out.println("Menu: ");
16            System.out.println("1. Tambah Mahasiswa");
17            System.out.println("2. Tampilkan Semua Mahasiswa");
18            System.out.println("3. Hapus Mahasiswa Berdasarkan NIM");
19            System.out.println("4. Cari Mahasiswa Berdasarkan NIM");
20            System.out.println("5. Keluar");
21            System.out.println("Pilih Menu: ");
22            choice = scanner.nextInt();
23            scanner.nextLine(); //Consume head_line
24            switch (choice) {
25                case 1:
26                    System.out.println("Masukan NIM: ");
27                    String nim = scanner.nextLine();
28                    System.out.println("Masukan Nama: ");
29                    String nama = scanner.nextLine();
30                    System.out.println("Masukan Prodi: ");
31                    String prodi = scanner.nextLine();
32                    mahasiswaList.add(new MahasiswaMain());
33                    break;
34                case 2:
35                    System.out.println("Data Mahasiswa: ");
36                    for (MahasiswaMain mhs : mahasiswaList) {
37                        System.out.println(mhs);
38                    }
39                    break;
40                case 3:
41                    System.out.println("Masukan NIM yang akan di hapus: ");
42                    String nimhapus = scanner.nextLine();
43                    mahasiswaList.removeIf(mhs -> mhs.nim.equals(nimhapus));
44                    break;
45            }
46        } while (choice != 5);
47        scanner.close();
48    }
49 }

```

```

25
26    }
27
28    System.out.println("Masukan NIM: ");
29    String nim = scanner.nextLine();
30    System.out.println("Masukan Nama: ");
31    String nama = scanner.nextLine();
32    System.out.println("Masukan Prodi: ");
33    String prodi = scanner.nextLine();
34    mahasiswaList.add(new MahasiswaMain());
35    break;
36    case 2:
37        System.out.println("Data Mahasiswa: ");
38        for (MahasiswaMain mhs : mahasiswaList) {
39            System.out.println(mhs);
40        }
41        break;
42    case 3:
43        System.out.println("Masukan NIM yang akan di hapus: ");
44        String nimhapus = scanner.nextLine();
45        mahasiswaList.removeIf(mhs -> mhs.nim.equals(nimhapus));
46        break;
47    case 4:
48        System.out.println("Masukan NIM yang di cari: ");
49        String nimCari = scanner.nextLine();
50        for (MahasiswaMain mhs : mahasiswaList) {
51            if (mhs.nim.equals(nimCari)) {
52                System.out.println(mhs);
53            } else {
54                System.out.println("nim tidak ada");
55            }
56        }
57        break;
58    case 5:
59        System.out.println("Keluar dari program.");
60        break;
61    default:
62        System.out.println("Pilihan tidak valid.");
63    }
64    while (choice != 5);
65    scanner.close();
66    }
67 }
68 }

```

package Pekan1;
import java.util.ArrayList;
import java.util.Scanner;
Kode ini mendeklarasikan paket (Pekan1) dan mengimpor kelas yang dibutuhkan: ArrayList untuk menyimpan data mahasiswa dan Scanner untuk menerima input dari pengguna.

Kelas Mahasiswa (Asumsikan):
Kode ini diasumsikan memiliki kelas Mahasiswa terpisah (tidak ditampilkan dalam gambar) dengan atribut seperti nim, nama, dan prodi. Kelas ini digunakan untuk merepresentasikan data setiap mahasiswa.

```

public class MahasiswaMain {
    private static ArrayList<Mahasiswa> mahasiswaList = new ArrayList<>();

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

```

```

int choice;
do {
    // Menampilkan menu
    System.out.println("\nMenu:");
    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("Pilihan Menu: ");
    choice = scanner.nextInt();
    scanner.nextLine(); // Consume newline

    switch (choice) {
        // ... (case 1 sampai 5) ...
    }
} while (choice != 5);
scanner.close();
}

// ... (Metode tambahMahasiswa, tampilkanMahasiswa, hapusMahasiswa,
cariMahasiswa) ...
}
• private static ArrayList<Mahasiswa> mahasiswaList = new ArrayList<>();
Mendeklarasikan ArrayList bernama mahasiswaList untuk menyimpan
objek-objek Mahasiswa.

```

- do-while loop: Loop ini menampilkan menu dan memproses pilihan pengguna sampai pengguna memilih untuk keluar (opsi 5).
- switch statement: Memilih fungsi yang akan dijalankan berdasarkan pilihan pengguna.

Kode ini akan berisi beberapa metode untuk setiap pilihan menu:

- tambahMahasiswa(Scanner scanner): Meminta data mahasiswa (NIM, nama, prodi) dari pengguna dan menambahkannya ke mahasiswaList.
- tampilkanMahasiswa(): Menampilkan seluruh data mahasiswa dalam mahasiswaList.
- hapusMahasiswa(Scanner scanner): Meminta NIM mahasiswa yang akan dihapus dan menghapusnya dari mahasiswaList.
- cariMahasiswa(Scanner scanner): Meminta NIM mahasiswa yang akan dicari dan menampilkan data mahasiswa tersebut jika ditemukan.

Contoh Implementasi Metode (tambahMahasiswa):

```

private static void tambahMahasiswa(Scanner scanner) {
    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)); // Membutuhkan
    konstruktor di kelas Mahasiswa
    System.out.println("Mahasiswa berhasil ditambahkan.");
}

```

case 1:

```

System.out.println("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;

```

Kasus ini meminta input NIM, nama, dan prodi dari pengguna, lalu membuat objek Mahasiswa baru dengan data tersebut dan menambahkannya ke dalam mahasiswaList. Asumsi di sini adalah kelas Mahasiswa memiliki konstruktor yang menerima NIM, nama, dan prodi sebagai parameter.

case 2:

```

System.out.println("Data Mahasiswa:");
for (Mahasiswa mhs : mahasiswaList) {
    System.out.println(mhs); // Membutuhkan toString() method di kelas
    Mahasiswa
}
break;

```

Kasus ini menampilkan data semua mahasiswa dalam mahasiswaList. Asumsi di sini adalah kelas Mahasiswa telah meng-override method toString() untuk menampilkan data mahasiswa dengan format yang mudah dibaca.

case 3:

```

System.out.println("Masukkan NIM yang akan di hapus: ");
String nimHapus = scanner.nextLine();
mahasiswaList.removeIf(mhs -> mhs.getNim().equals(nimHapus)); //
    Membutuhkan getter untuk NIM di kelas Mahasiswa
break;

```

Kasus ini meminta input NIM mahasiswa yang akan dihapus. mahasiswaList.removeIf() menggunakan lambda expression untuk menghapus mahasiswa yang NIM-nya cocok dengan nimHapus. Asumsi di sini adalah kelas Mahasiswa memiliki method getNim() untuk mengakses nilai NIM.

case 4:

```
System.out.println("Masukkan NIM yang di cari: ");
String nimCari = scanner.nextLine();
for (Mahasiswa mhs : mahasiswaList) {
    if (mhs.getNim().equals(nimCari)) {
        System.out.println(mhs); // Membutuhkan toString() method di kelas
        Mahasiswa
    } else {
        System.out.println("nim tidak ada");
    }
}
break;
```

Kasus ini meminta input NIM mahasiswa yang akan dicari. Loop for-each akan mengiterasi mahasiswaList dan menampilkan data mahasiswa jika NIM-nya cocok. Jika tidak ditemukan, pesan "nim tidak ada" akan ditampilkan.

case 5:

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

default:

```
System.out.println("Pilihan tidak valid.");
```

Kasus 5 mengakhiri program, sedangkan default menangani pilihan yang tidak valid.

IV. KESIMPULAN

Pratikum ini telah berhasil menunjukkan pemahaman tentang penggunaan ArrayList dalam bahasa pemrograman Java. Melalui serangkaian percobaan, kami berhasil mengimplementasikan berbagai metode ArrayList seperti penambahan, penghapusan, pengurutan, dan pencarian elemen. Hasil percobaan menunjukkan efektivitas ArrayList dalam menangani koleksi data dinamis. Meskipun demikian, terdapat beberapa kendala yang dijumpai, seperti kompleksitas dalam implementasi beberapa metode tertentu. Ke depannya, diharapkan pemahaman dan kemampuan dalam mengaplikasikan ArrayList akan semakin terasah.