

LAPORAN PRAKTIKUM
STRUKTUR DATA
ARRAYLIST



OLEH:
DEVINA AMANDA PUTRI
(2411533009)

DOSEN PENGAMPU:
DR. WAHYUDI, S.T, M.T

FAKULTAS TEKNOLOGI INFORMASI
DEPARTEMEN INFORMATIKA
UNIVERSITAS ANDALAS

2024

A. PENDAHULUAN

Java ArrayList merupakan implementasi dari interface List dalam framework Collection Java. ArrayList memberi kelas dinamis, sehingga dapat menampung data tanpa harus mengetahui ukuran awalnya. Meskipun lebih lambat daripada array tetapi ArrayList dapat membantu dalam program yang membutuhkan banyak manipulasi dalam array. Untuk membuat sebuah kelas ArrayList, perlu dibuat objek terlebih dahulu menggunakan konstruktor yang memungkinkan pembuatan daftar array. Kelas Java ArrayList dapat ditemukan pada paket java.util, sehingga saat menggunakan ArrayList dalam program perlu meng-import java.util.ArrayList agar program dapat berjalan.

Terdapat method pada ArrayList yang dibutuhkan untuk menjalankan suatu program, berikut beberapa method yang paling sering digunakan, seperti:

1. add (index, object, element), digunakan untuk menyisipkan elemen pada posisi tertentu dalam list
2. remove(), digunakan untuk menghapus data tertentu
3. for atau forEach(), digunakan untuk menampilkan semua data
4. get(index), digunakan untuk mengembalikan elemen pada posisi yang ditentukan dalam daftar
5. clear(), digunakan untuk menghapus semua elemen dari daftar
6. isEmpty(), digunakan untuk memeriksa apakah sebuah elemen kosong atau berisi.
7. size(), digunakan untuk memeriksa jumlah data

Kelebihan Java ArrayList dapat terlihat dari ukuran ArrayList yang dinamis, dimana kita dapat menambah atau menghapus elemen sesuai kebutuhan. ArrayList juga menyediakan akses cepat ke elemen dan memiliki koleksi yang teratur, ArrayList mempertahankan urutan elemen sehingga memungkinkan untuk mengakses elemen sesuai urutan penambahannya. Selain itu, ArrayList juga mendukung nilai null, berguna saat ketiadaan nilai diperlukan. Namun ArrayList lebih lambat dari array untuk operasi menyisipkan elemen dan membutuhkan lebih banyak memori.

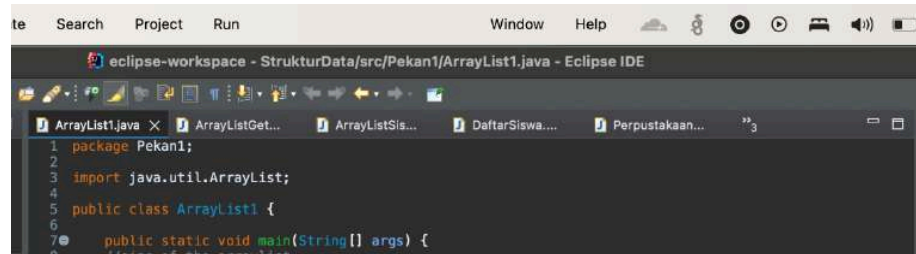
B. TUJUAN

1. Memahami konsep struktur data ArrayList dalam Java
2. Mengimplementasikan operasi dasar seperti add, remove, get dan contains
3. Menerapkan struktur data ArrayList dalam beberapa kasus

C. LANGKAH-LANGKAH Pengerjaan

a. Class ArrayList1.java

1. Setelah masuk ke Eclipse dan membuat folder baru dengan nama project “StrukturData”, buat class pertama dengan nama ArrayList1



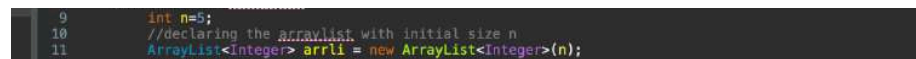
```
1 package Pekan1;
2
3 import java.util.ArrayList;
4
5 public class ArrayList1 {
6
7     public static void main(String[] args) {
8         //size of the arraylist
```

2. Buat sebuah variabel n bertipe integer yang bernilai 5. Nilai ini akan digunakan untuk menentukan jumlah elemen yang ingin dimasukkan ke dalam ArrayList




```
7     public static void main(String[] args) {
8         //size of the arraylist
9         int n=5;
10        //declaring the arraylist with initial size n
```

3. Buat objek ArrayList ke dalam program



```
9         int n=5;
10        //declaring the arraylist with initial size n
11        ArrayList<Integer> arrli = new ArrayList<Integer>(n);
12        //for (int i = 0; i < n; i++)
```

4. Melakukan perulangan dari 1 sampai 5. Di setiap iterasi, nilai i ditambahkan ke dalam arrli menggunakan method add(), maka hasilnya arrli akan berisi [1,2,3,4,5]



```
11        ArrayList<Integer> arrli = new ArrayList<Integer>(n);
12        for (int i = 1; i <= n; i++)
13            arrli.add(i);
14        //for (int i = 0; i < n; i++)
```

5. Untuk mencetak isi ArrayList dan menampilkan hasil dari pengisian sebelumnya menggunakan System.out.println(arrli); dan Untuk menghapus elemen pada index ke-3 arrli.remove(3);, karena index dimulai dari 0, maka elemen ke-4 akan terhapus.



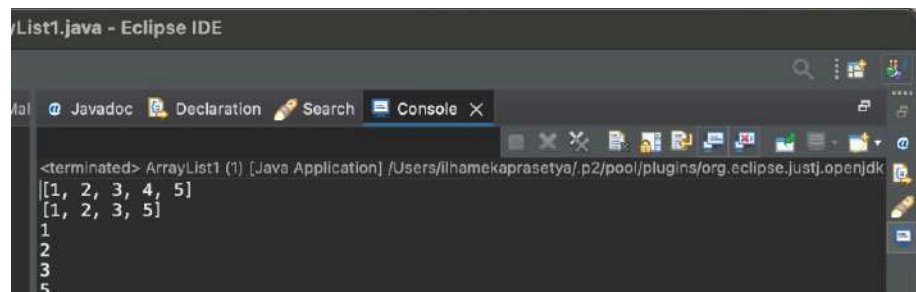
```
13            arrli.add(i);
14        System.out.println(arrli);
15        arrli.remove(3);
16        //for (int i = 0; i < n; i++)
```

6. Mencetak isi arrli setelah index ke-3 dihapus dengan System.out.println(arrli); dan loop mencetak setiap elemen ArrayList secara terpisah dengan fungsi get(i) untuk mengambil elemen pada index ke-1



```
16        System.out.println(arrli);
17        for (int i = 0; i < arrli.size(); i++)
18            System.out.println(arrli.get(i) + " ");
19    }
20 }
21 }
22 }
```

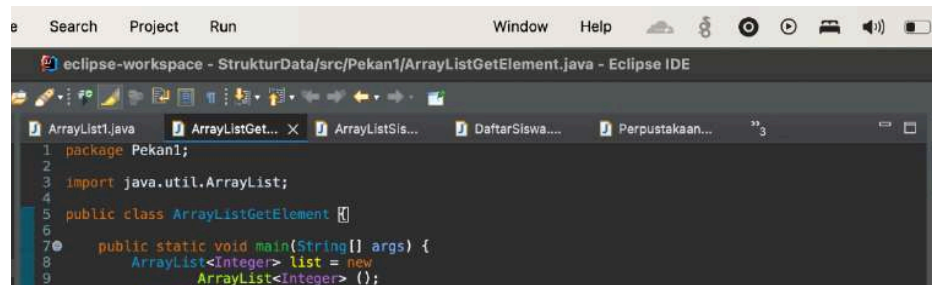
maka outputnya:



```
<terminated> ArrayList (1) [Java Application] /Users/Ilhamekaprasetya/.p2/pool/plugins/org.eclipse.justj.openjdk
[1, 2, 3, 4, 5]
[1, 2, 3, 5]
1
2
3
5
```

b. Class ArrayListGetElement.java

1. Buat class baru pada package yang sama, beri nama class ArrayListGetElement dan buat objek ArrayList ke dalam program bertipe Integer dan bernama list



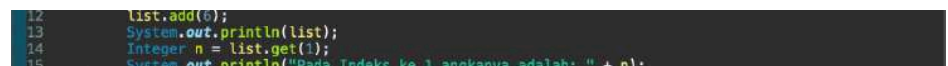
```
package Pekan1;
import java.util.ArrayList;
public class ArrayListGetElement {
    public static void main(String[] args) {
        ArrayList<Integer> list = new
        ArrayList<Integer> ();
```

2. Tambahkan 3 angka menggunakan list.add() ke dalam list, elemen disisipkan secara berurutan di index 0,1,2. Lalu cetak isi list



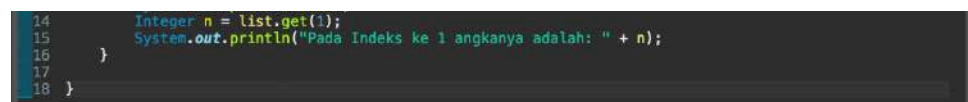
```
list.add(9);
list.add(5);
list.add(6);
System.out.println(list);
```

3. Mengambil nilai pada index ke-1, yaitu angka 5 dan menyimpannya ke dalam variabel n



```
list.add(6);
System.out.println(list);
Integer n = list.get(1);
System.out.println("Pada Indeks ke 1 angkanya adalah: " + n);
```

4. Tampilkan angka yang tersimpan dalam n dan memberi informasi posisi dan nilainya



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

Maka outputnya:



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

c. Class ArrayListSisip.java

1. Buat class baru pada package yang sama, beri nama class ArrayListGetSisip dan buat objek ArrayList ke dalam program bertipe Integer dan bernama list

```
1 package Pekan1;
2
3 import java.util.ArrayList;
4
5 public class ArrayListSisip {
6
7     public static void main(String[] args) {
8         ArrayList<Integer> list = new
9         ArrayList<Integer>();
```

2. Menambahkan angka 1,2,4 ke dalam list menggunakan list.add(), sehingga list berisi [1,2,4] lalu cetak isi list

```
10 list.add(1);
11 list.add(2);
12 list.add(4);
13 System.out.println(list);
```

3. Sisipkan angka 2 dan 3 menggunakan list.add(2,3); angka 3 disisipkan ke index 2 yang sebelumnya di index 2 adalah angka 4 sekarang bergeser ke index 3. Maka isi list menjadi [1,2,3,4]. Dan cetak untuk menampilkan isi list setelah disisipkan elemen 3

```
14 list.add(2,3);
15 System.out.println(list);
16
17 }
18
19 }
```

Maka outputnya:

```
<terminated> ArrayListSisip [Java Application] /Users/jinamkaprasetya/p2/p001/plugins/org.eclipse.just.openjdk
[1, 2, 4]
[1, 2, 3, 4]
```

d. Class DaftarSiswa.java

1. Buat class baru pada package yang sama, beri nama class DaftarSiswa dan buat objek ArrayList global untuk menyimpan daftar nama siswa ke dalam program bertipe String. Lalu tambahkan objek Scanner untuk membaca input dari user

```
1 package Pekan1;
2
3 import java.util.ArrayList;
4
5
6 public class DaftarSiswa {
7     private static ArrayList<String> daftarNamaSiswa = new ArrayList<>();
8     public static void main(String[] args) {
9         Scanner scanner = new Scanner(System.in);
```

2. Buat menu interaktif menggunakan perulangan Do-While agar opsi terus ditampilkan sampai user memilih menu keluar, menggunakan switch-case untuk menjalankan berbagai pilihan menu seperti tambah, tampil, hapus, cari, dan keluar

```
9 scanner scanner = new Scanner (System.in);
10 int pilihan;
11 do {
12     System.out.println("\nMenu");
13     System.out.println("1. Tambah Nama Siswa");
14     System.out.println("2. Tampilkan Daftar Nama Siswa");
15     System.out.println("3. Hapus Nama Siswa");
16     System.out.println("4. Cari Nama Siswa");
17     System.out.println("5. Keluar");
18     pilihan = scanner.nextInt();
19     switch (pilihan) {
20         case 1:
21             tambahNamaSiswa(scanner);
22             break;
23         case 2:
24             tampilkanDaftarNamaSiswa();
25             break;
26         case 3:
27             hapusNamaSiswa(scanner);
28             break;
29         case 4:
30             cariNamaSiswa(scanner);
31             break;
32         case 5:
33             System.out.println("Keluar dari program");
34             break;
35         default:
36             System.out.println("Pilihan tidak valid");
37     }
38 } while (pilihan != 5);
39 }
```

3. Method pertama tambahNamaSiswa, menggunakan scanner dan daftarNamaSiswa.add() untuk membaca nama dari input user dan menambahkannya ke dalam list

```
39 }
40 private static void tambahNamaSiswa (Scanner scanner) {
41     System.out.println("Masukkan nama siswa: ");
42     String nama = scanner.nextLine();
43     daftarNamaSiswa.add(nama);
44     System.out.println("Nama siswa berhasil ditambahkan");
45 }
46 private static void tampilkanDaftarNamaSiswa() {
```

4. Method kedua, tampilkanDaftarNamaSiswa, menggunakan perulangan if-else lalu lakukan iterasi dan cetak seluruh nama siswa yang ada di dalam list

```
46 private static void tampilkanDaftarNamaSiswa() {
47     if (daftarNamaSiswa.isEmpty()) {
48         System.out.println("Tidak ada siswa dalam daftar");
49     } else {
50         System.out.println("Daftar Nama Siswa");
51         for (String nama : daftarNamaSiswa) {
52             System.out.println(nama);
53         }
54     }
55 }
```

5. Method ketiga, hapusNamaSiswa menggunakan perulangan if-else dan remove.(nama) untuk mencoba menghapus nama dari list dan memberi pemberitahuan apakah nama berhasil dihapus dari list

```

56 private static void hapusNamaSiswa (Scanner scanner) {
57     System.out.println("Masukkan nama siswa yang akan dihapus");
58     String nama = scanner.nextLine();
59     if (daftarNamaSiswa.remove(nama)) {
60         System.out.println("Nama siswa berhasil dihapus.");
61     } else {
62         System.out.println("Nama siswa tidak ditemukan.");
63     }
64 }

```

6. Method keempat, cariNamaSiswa masih menggunakan perulangan if-else diikuti .contains(nama) untuk mengecek apakah nama yang dimasukkan ada di list dan mencetak hasilnya

```

65 private static void cariNamaSiswa (Scanner scanner) {
66     System.out.println("Masukkan nama siswa yang dicari: ");
67     String nama = scanner.nextLine();
68     if (daftarNamaSiswa.contains(nama)) {
69         System.out.println("Nama siswa ditemukan: " + nama);
70     } else {
71         System.out.println("Nama siswa tidak ditemukan");
72     }
73 }
74 }

```

maka outputnya:

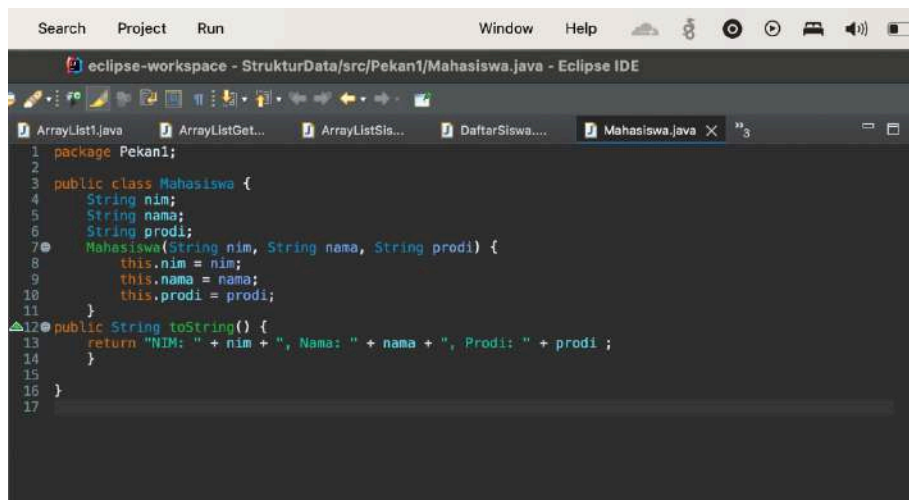
```

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

```

e. Class Mahasiswa.java

1. Buat class baru pada package yang sama, beri nama class Mahasiswa
2. Buat String dengan tiga atribut utama mahasiswa, yaitu: nim, nama dan prodi. Tambahkan konstruktor untuk menginisialisasikan objek Mahasiswa dengan parameter yang diberikan
3. Override dari method toString() untuk mencetak objek mahasiswa dengan format yang lebih mudah dibaca



```

eclipse-workspace - StrukturData/src/Pekan1/Mahasiswa.java - Eclipse IDE
package Pekan1;

public class Mahasiswa {
    String nim;
    String nama;
    String prodi;

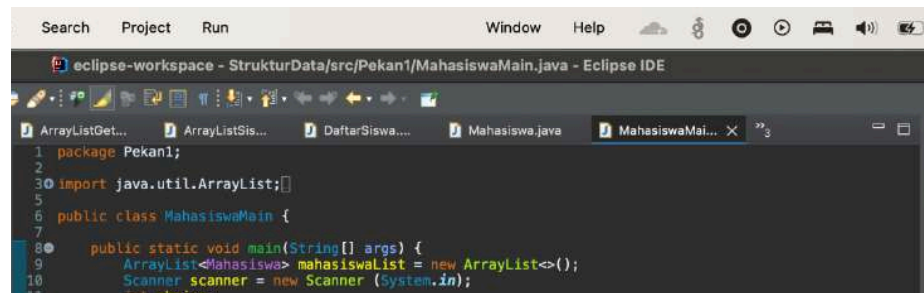
    Mahasiswa(String nim, String nama, String prodi) {
        this.nim = nim;
        this.nama = nama;
        this.prodi = prodi;
    }

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

```

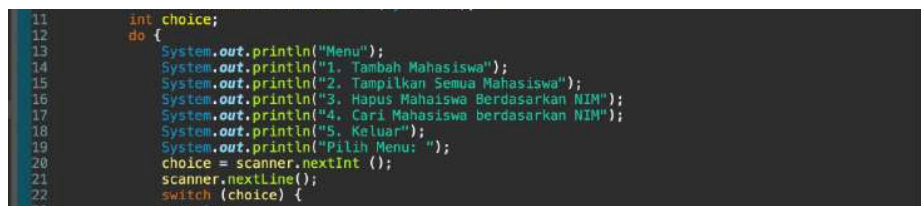

f. Class MahasiswaMain.java

1. Buat class baru pada package yang sama, beri nama class MahasiswaMain dan buat objek ArrayList global untuk menyimpan objek Mahasiswa, tambahkan objek Scanner untuk membaca input dari user



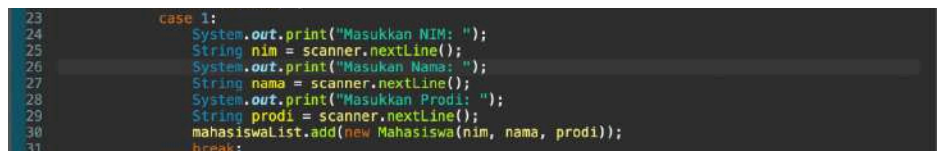
```
1 package Pekan1;
2
3 import java.util.ArrayList;
4
5 public class MahasiswaMain {
6
7     public static void main(String[] args) {
8         ArrayList<Mahasiswa> mahasiswaList = new ArrayList<>();
9         Scanner scanner = new Scanner(System.in);
10     }
```

2. Membuat menu interaktif untuk pengelolaan data mahasiswa, menggunakan switch-case untuk menjalankan berbagai pilihan menu dan tambahkan variabel choice untuk menyimpan pilihan menu dari user



```
11     int choice;
12     do {
13         System.out.println("Menu");
14         System.out.println("1. Tambah Mahasiswa");
15         System.out.println("2. Tampilkan Semua Mahasiswa");
16         System.out.println("3. Hapus Mahasiswa Berdasarkan NIM");
17         System.out.println("4. Cari Mahasiswa berdasarkan NIM");
18         System.out.println("5. Keluar");
19         System.out.println("Pilih Menu: ");
20         choice = scanner.nextInt();
21         scanner.nextLine();
22         switch (choice) {
```

3. Buat Case 1, yaitu Tambah Mahasiswa. String nim = scanner.nextLine();, String nama = scanner.nextLine();, String prodi = scanner.nextLine();, digunakan untuk membaca input nim, nama dan prodi yang dimasukkan oleh user. Lalu membuat objek Mahasiswa baru berdasarkan input user, lalu menambahkannya ke ArrayList dengan mahasiswa.add(new mahasiswa(nim, nama, prodi));



```
23         case 1:
24             System.out.print("Masukkan NIM: ");
25             String nim = scanner.nextLine();
26             System.out.print("Masukkan Nama: ");
27             String nama = scanner.nextLine();
28             System.out.print("Masukkan Prodi: ");
29             String prodi = scanner.nextLine();
30             mahasiswaList.add(new Mahasiswa(nim, nama, prodi));
31             break;
```

4. Buat Case 2, yaitu Tampilkan Semua Mahasiswa dengan perulangan for-each untuk menampilkan semua data di dalam list. Karena class Mahasiswa sudah override di method toString(), maka dapat dipanggil menggunakan System.out.print(mhs); dan

akan otomatis memanggil dan muncul dengan format NIM:..
Nama:... Prodi;...

```
32 case 2:
33     System.out.print("Data Mahasiswa: ");
34     for (Mahasiswa mhs : mahasiswaList) {
35         System.out.println(mhs);
36     }
37     break;
```

5. Buat Case 3, yaitu Hapus Mahasiswa Berdasarkan NIM. Ketikkan `System.out.print("Masukkan NIM yang akan dihapus: ");`

`String nimHapus = scanner.nextLine();` untuk membaca input NIM yang ingin dihapus. Setiap objek dalam `mhs` akan di cek apakah nim nya sesuai dengan nim yang di inputkan, jika sesuai maka akan dihapus dari list dengan `removeIf()`.

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

6. Buat Case 4, yaitu Cari Mahasiswa Berdasarkan NIM menggunakan scanner agar dapat membaca input NIM yang dicari lalu menggunakan for untuk mengecek satu persatu nim dari mahasiswa yang cocok dengan yang di cari, jika cocok maka data mahasiswa akan ditampilkan dengan perulangan if-else dan dihentikan dengan `break;`.

```
43 case 4:
44     System.out.print("Masukkan NIM yang dicari: ");
45     String nimCari = scanner.nextLine();
46     for (Mahasiswa mhs : mahasiswaList) {
47         if (mhs.nim.equals(nimCari)) {
48             System.out.println("nim tidak ada");
49             break;
50         } else {
51             System.out.println("nim tidak ada");
52         }
53     }
54     break;
```

7. Case 5, Keluar dari menu. Pesan “Keluar dari program” akan muncul saat user memilih angka 5, maka `choice` akan kembali ke default dan bisa menutup scanner dengan `scanner.close();`

```
55 case 5:
56     System.out.println("Keluar dari program.");
57     break;
58 default:
59     System.out.println("Pilihan tidak valid.");
60 }
61 while (choice != 5);
62 scanner.close();
63 }
64 }
65 }
```

outputnya:

```
MahasiswaManajemen [Java Application] /Users/imanekaprasetya/.p2/pool/plugins/org.eclipse.jdt.op
Menu
1. Tambah Mahasiswa
2. Tampilkan Semua Mahasiswa
3. Hapus Mahasiswa Berdasarkan NIM
4. Cari Mahasiswa berdasarkan NIM
5. Keluar
Pilih Menu:
1
Masukkan NIM: 2411533009
Masukkan Nama: devina amanda
Masukkan Prodi: informatika
Menu
1. Tambah Mahasiswa
2. Tampilkan Semua Mahasiswa
3. Hapus Mahasiswa Berdasarkan NIM
4. Cari Mahasiswa berdasarkan NIM
5. Keluar
Pilih Menu:
4
Masukkan NIM yang dicari: 2411533009
nim tidak ada
Menu
1. Tambah Mahasiswa
2. Tampilkan Semua Mahasiswa
```

D. KESIMPULAN

Dari praktikum ini dapat disimpulkan bahwa ArrayList dalam Java sebagai struktur data dinamis sangat berguna untuk pengolahan data. Hasil praktikum membuktikan bahwa ArrayList memberikan fleksibilitas dengan kemampuannya untuk menyesuaikan ukuran secara otomatis berbeda dengan array standar yang bersifat statis. Telah dilakukan pengujian berbagai operasi seperti penambahan elemen, penghapusan elemen, pengambilan data dan pencarian yang semua dapat diimplementasikan dengan sintaks yang sederhana namun tetap powerful.