

LAPORAN PRAKTIKUM STRUKTUR DATA  
PROGRAM ARRAYLIST JAVA



Oleh :

DERIEL CHAERAHMAN

NIM 2411533007

DOSEN PENGAMPU :

DR. WAHYUDI, S.T, M.T

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

2025

## A. Pendahuluan

Praktikum ini dilakukan untuk membuat program yang dapat menyimpan data menggunakan **ArrayList**. Mengimplementasikannya dengan program java yang sudah dipelajari sebelumnya seperti if statement, looping, scanner class dan lainnya untuk menerima input yang nantinya disimpan pada ArrayList dalam pemrograman Java.

### 1. ArrayList

Merupakan bagian dari Java class yang digunakan untuk menyimpan data berupa list objek berbentuk array yang ukurannya dapat berubah secara dinamis. Mengimport dari package java.util.ArrayList sebagai bagian dari API Collection java. Memiliki banyak method untuk mengakses dan memodifikasi elemen yang tersimpan di dalam ArrayList. Elemen dalam ArrayList disimpan di lokasi memori bersebelahan.

Method :

- add() : untuk menambahkan data ke dalam ArrayList
- get() : untuk mengambil data yang tersimpan pada index tertentu
- set() : untuk menimpa data pada indeks tertentu
- remove() : untuk menghapus data pada indeks tertentu
- size() : untuk mengetahui panjang ArrayList
- isEmpty() : untuk memeriksa apakah ArrayList kosong atau tidak
- indexOf() : untuk mengetahui indeks dari suatu nilai
- contains() : untuk memeriksa apakah suatu nilai ada dalam ArrayList

## B. Tujuan

Tujuan dari dilakukannya praktikum ini adalah :

1. Memahami dan mengaplikasikan ArrayList dalam program java.
2. Implementasi method ArrayList pada program.
3. Dapat mengaplikasikan statement (if, while), class (Scanner), dll. yang dipelajari sebelumnya ke dalam program ArrayList
4. Membuat program ArrayList dengan metode OOP (Object Oriented Programming).

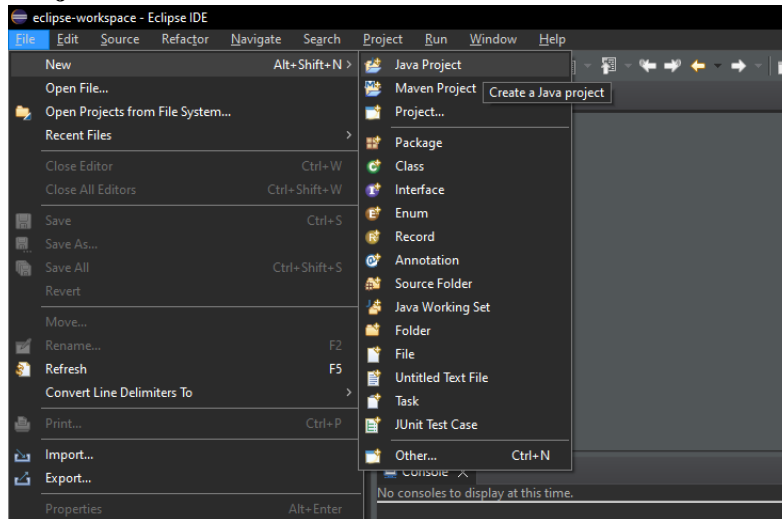
## C. Langkah kerja praktikum

### a. Alat dan Bahan

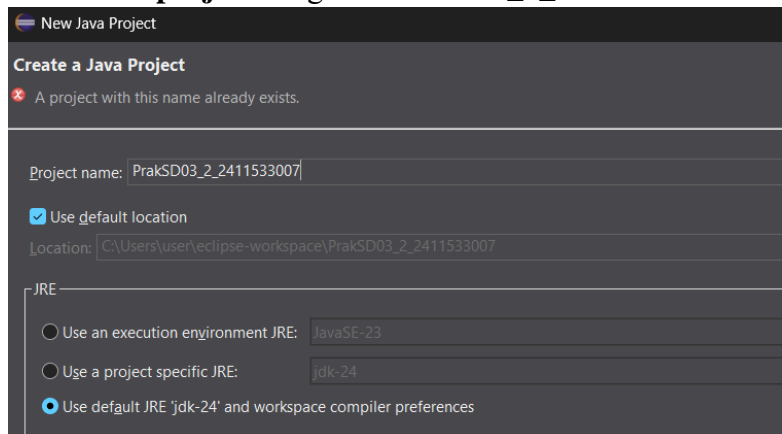
1. Perangkat computer atau laptop
2. Jaringan internet
3. IDE (Integreted Development Environment) direkomendasikan Eclipse IDE
4. Java JDK (Java Development Kit)

## b. New Java Project & Package

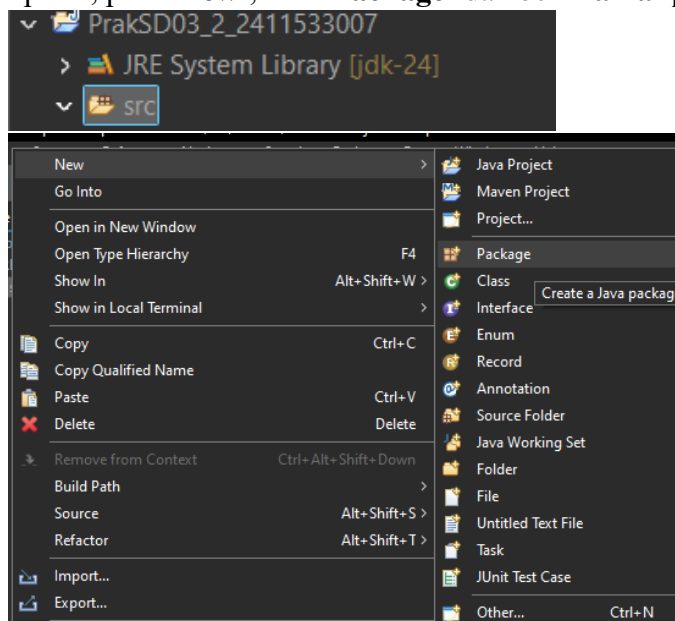
1. Buat Java Project baru, klik pada **tab 'File'** di pojok kiri atas, lalu klik '**Java Project**'.



2. tulis **nama project** dengan “PrakSD03\_2\_2411533007”.



3. Setelahnya buat new package. **Buka java project** yang telah dibuat sebelumnya, lalu **klik kanan** pada folder '**src**', setelahnya akan muncul list option, pilih '**new**', lalu '**Package**' dan beri **nama 'pekan1'**, lalu seleso.



c. Program ArrayList1

1. **Buat** terlebih dahulu **new class** dengan klik kanan pada package pekan1 dan beri nama '**ArrayList1**', centang method public static void.
2. **Import** ArrayList class dari paket **Java.util.ArrayList** terlebih dahulu untuk menggunakan fungsi/method dari classnya.

```
1 package pekan1;
2
3 import java.util.ArrayList;
4
5 public class ArrayList1 {
```

3. **Deklarasi variabel** bertipe **integer** dengan nama '**n**', dan beri **value 5**. **Deklarasikan** juga **Arraylist** bertipe **integer** dengan nama '**arrli**', inisialisasi **sizenya** menggunakan **variabel 'n'**.

```
7 public static void main(String[] args) {
8     //size of the ArrayList
9     int n = 5;
10    //Declaring the ArrayList with initial size n
11    ArrayList<Integer> arrli = new ArrayList<Integer>(n);
12    //Appending new elements at the end of the list
```

4. Buat **for loop** dengan **inisialisasi** **i = 1**, **kondisi** **i** dengan variabel '**n**' menggunakan operator relasional kecil sama dengan/ **<=**. **Updation** '**i**' dengan nilai satu. Jika **for loop** bernilai **True**, maka blok kode akan dijalankan, yaitu **menambahkan angka 1** sampai 5 ke dalam ArrayList menggunakan **method** **arrli.add(i)**. Note: '**i**' merupakan iterasi dalam loop.

```
13    for (int i = 1; i<=n; i++) {
14        arrli.add(i);
15    }
```

5. **Menampilkan** isi dari **ArrayList** melalui **sysout** menggunakan nama dari ArrayList yaitu '**arrli**'.

```
16    //Printing elements
17    System.out.println(arrli);
```

6. Implementasi **method remove** untuk menghilangkan data yang tersimpan pada index ke-3.

```
18    //Remove element at index 3
19    arrli.remove(3);
```

7. **Menampilkan** isi dari **ArrayList** melalui **sysout** untuk **melihat perubahan** yang terjadi setelah menggunakan **method remove** pada ArrayList.

```
20    //Displaying the ArrayList
21    //After deletion
22    System.out.println(arrli);
```

8. Buat **for loop** kedua dengan **inisialisasi** **i = 0**, **kondisi** **i** dengan panjang ArrayList **method size()** menggunakan operator relasional kecil sama dengan/ **<=**. **Updation** '**i**' dengan nilai satu. Jika **for loop** bernilai **True**, maka blok kode **sysout** akan dijalankan yang menampilkan ArrayList menggunakan **method get()**.

```
23    //Printing elements one by one
24    for (int i = 0; i < arrli.size(); i++) {
25        System.out.print(arrli.get(i) + " ");
26    }
27 }
28 }
```

d. Program ArrayListGetElement

1. **Buat new class** dengan klik kanan pada package pekan1 dan beri nama 'ArrayListGetElement', centang method public static void.
2. **Import** ArrayList class dari paket **Java.util.ArrayList** terlebih dahulu untuk menggunakan fungsi/method dari classnya.

```
1 package pekan1;
2
3 import java.util.ArrayList;
4
5 public class ArrayListGetElement {
6
```

3. **Deklarasi Arraylist** bertipe **integer** dengan nama 'list', tanpa inisialisasi sizenya.

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

4. Gunakan **method add()** dari kelas ArrayList untuk menambahkan data ke dalam ArraList.

```
9         //add the number
10        list.add(9);
11        list.add(5);
12        list.add(6);
```

5. **Tampilkan data** yang tersimpan di dalam ArrayList saat ini menggunakan sysout dengan memanggil nama dari ArrayList.

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

6. Buat **variabel integer** dengan nama 'n', dan isi valuenya menggunakan **method get(1)** untuk **mendapatkan data** spesifik yang **tersimpan** di dalam ArrayList, yaitu mendapatkan data pada lokasi **index ke-1**.

```
14        //get method
15        Integer n = list.get(1);
16        System.out.println("Pada index ke-1 angkanya adalah : " + n);
17    }
18 }
```

7. Terakhir, **tampilkan data** yang tersimpan pada **variabel n** menggunakan sysout, untuk menampilkan data yang tersimpan pada index ke-1.

```
16        System.out.println("Pada index ke-1 angkanya adalah : " + n);
17    }
18 }
```

e. Program ArrayListSisip

1. **Buat new class** dengan klik kanan pada package pekan1 dan beri nama 'ArrayListSisip', centang method public static void.
2. **Import** ArrayList class dari paket **Java.util.ArrayList** terlebih dahulu untuk menggunakan fungsi/method dari classnya.

```
1 package pekan1;
2
3 import java.util.ArrayList;
4
5 public class ArrayListSisip {
6
```

3. **Deklarasi Arraylist** bertipe **integer** dengan nama 'list', tanpa inisialisasi sizenya.

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

- Gunakan **method add()** dari kelas ArrayList untuk menambahkan data ke dalam ArrayList.

```
9      list.add(1);
10     list.add(2);
11     list.add(4);
```

- Tampilkan data yang tersimpan di dalam ArrayList saat ini menggunakan sysout dengan memanggil nama dari ArrayList yaitu 'list'.

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

- Gunakan **method add(2, 3)**, yang berfungsi untuk menambahkan int 3 ke dalam index ke-2.

```
13     //insert missing element 3
14     list.add(2, 3);
```

- Tampilkan semua data yang tersimpan pada ArrayList dengan sysout yang menggunakan nama dari ArrayList yaitu 'list'.

```
15     System.out.println(list);
16 }
17
18 }
```

f. Program DaftarSiswa

- Buat new class dengan klik kanan pada package pekan1 dan beri nama 'DaftarSiswa', centang method public static void.
- Import ArrayList class dari paket **java.util.ArrayList** terlebih dahulu untuk menggunakan fungsi/method dari classnya. Import Scanner dari package **java.util.Scanner** untuk fungsi menerima input. Deklarasi **private static ArrayList** bertipe **String** dengan nama 'daftarNamaSiswa'. Bersifat **private** karena variabel hanya bisa diakses dalam kelas DaftarSiswa (enkapsulasi), dan **static** karena variabel bersifat shared (dipakai oleh semua method static dalam kelas).

```
1 package pekan1;
2
3 import java.util.Scanner;
4 import java.util.ArrayList;
5
6 public class DaftarSiswa {
7     private static ArrayList <String> daftarNamaSiswa = new ArrayList <>();
```

- Deklarasi scanner class bernama **console** untuk membuat objek scan yang dapat menerima user input. Buat variabel integer bernama 'pilihan'.

```
8     public static void main(String[] args) {
9         Scanner console = new Scanner (System.in);
10        int pilihan;
```

- Do-while loop, dimana **do** loop akan langsung menjalankan sysout untuk menampilkan menu tersedia dari main program. Variabel 'pilihan' yang dibuat sebelumnya menerima input user, dan console.nextLine() untuk membersihkan buffer input (mencegah bug akibat \n yang tertinggal). Bagian ini berfungsi sebagai Entry Point Program/menu.

```

11         do {
12             System.out.println("\nmenu :");
13             System.out.println("1. Tambah nama siswa");
14             System.out.println("2. Tampilan daftar siswa");
15             System.out.println("3. Hapus nama siswa");
16             System.out.println("4. Cari nama siswa");
17             System.out.println("5. Keluar");
18             System.out.println("Masukan pilihan anda :");
19             pilihan = console.nextInt();
20             console.nextLine(); //newline

```

5. Switch-Case untuk Menangani Menu, terdiri dari 5 case. Setiap case memanggil method sesuai dengan case nya. Switch case ditentukan melalui kondisi inputan dari user pada variabel 'pilihan'. Jika input user tidak sesuai dengan case yang disediakan (yaitu 1-5 cases), maka program default akan dijalankan.

```

21         switch (pilihan) {
22             case 1 :
23                 tambahNamaSiswa (console);
24                 break;
25             case 2 :
26                 tampilkanDaftarNamaSiswa ();
27                 break;
28             case 3 :
29                 hapusNamaSiswa (console);
30                 break;
31             case 4 :
32                 cariNamaSiswa (console);
33                 break;
34             case 5 :
35                 System.out.println("Keluar dari program.");
36                 break;
37             default :
38                 System.out.println("Pilihan tidak valid.");
39         }

```

6. Do While-loop akan berhenti jika inputan user bernilai 5, yang akan menghasilkan False pada kondisi while loop.

```

40     } while (pilihan != 5);
41 }

```

7. Method tambahNamaSiswa sesuai dengan case 1 yang berfungsi untuk menambahkan data ke dalam ArrayList dengan cara meminta input kepada user. Bersifat private karena hanya dipanggil di dalam kelas ini (tidak bisa di akses dari kelas luar karena bukan public) dan static bisa diakses tanpa instansiasi objek. Meminta input user melalui objek scanner, lalu method add() yang memanggil vairabel 'nama' untuk memasukan input user ke dalam arraylist, dan terakhir menampilkan pemberitahuan (sysout) kepada user bahwa proses berhasil dilakukan.

```

43●     private static void tambahNamaSiswa (Scanner console) {
44         System.out.println("Masukan nama siswa :");
45         String nama = console.nextLine();
46         daftarNamaSiswa.add(nama);
47         System.out.println("Nama siswa " + nama + " berhasil ditambahkan.");
48     }

```

8. Method tampilkanDaftarNamaSiswa sesuai dengan case 2 yang berfungsi untuk menampilkan isi dari arraylist. Bersifat private karena hanya dipanggil di dalam kelas ini (tidak bisa di akses dari kelas luar karena bukan public) dan static bisa diakses tanpa instansiasi objek. Memakai if-else statement dengan implementasi method isEmpty() untuk memeriksa isi dari daftar jika kosong, jika True menampilkan sysout "tidak ada siswa

dalam daftar” dan jika False akan menampilkan sysout “daftar nama siswa” dalam arraylist dengan menggunakan for loop.

```
49● private static void tampilkanDaftarNamaSiswa () {  
50     if (daftarNamaSiswa.isEmpty()) {  
51         System.out.println("Tidak ada siswa dalam daftar.");  
52     } else {  
53         System.out.println("Daftar Nama Siswa :");  
54         for (String nama : daftarNamaSiswa) {  
55             System.out.println(nama);  
56         }  
57     }  
58     System.out.println();  
59     System.out.println(daftarNamaSiswa);  
60 }
```

9. Method `hapusNamaSiswa` sesuai dengan case 3 yang berfungsi untuk menghapus data dari ArrayList dengan cara meminta input dari user. Bersifat private karena hanya dipanggil di dalam kelas ini (tidak bisa di akses dari kelas luar karena bukan public) dan static bisa diakses tanpa instansiasi objek. Menampilkan sysout yang meminta user untuk menginputkan, yang nantinya ditampung oleh variable `nama`, if statement dengan implementasi method `remove()` untuk menghapus data dari arraylist, jika True maka akan muncul pesan konfirmasi dan jika False akan menampilkan pesan pemberitahuan lanjutan.

```
61● private static void hapusNamaSiswa (Scanner console) {  
62     System.out.println("Silahkan input nama siswa yang ingin dihapus :");  
63     System.out.println(daftarNamaSiswa);  
64     String nama = console.nextLine();  
65     if (daftarNamaSiswa.remove(nama)) {  
66         System.out.println("Nama siswa berhasil dihapus.");  
67     } else {  
68         System.out.println("Nama siswa tidak dapat ditemukan.");  
69     }  
70     System.out.println(daftarNamaSiswa);  
71 }
```

10. Method `cariNamaSiswa` sesuai dengan case 4 yang berfungsi untuk menghapus data dari ArrayList dengan cara meminta input dari user. Bersifat private karena hanya dipanggil di dalam kelas ini (tidak bisa di akses dari kelas luar karena bukan public) dan static bisa diakses tanpa instansiasi objek. Menampilkan sysout yang meminta user untuk menginputkan, yang nantinya ditampung oleh variable `nama`, if statement dengan implementasi method `contains()` yang memeriksa jika data tersebut tersimpan dalam arraylist, jika True maka akan muncul pesan konfirmasi bahwa data terdapat dalam arraylist dan jika False akan menampilkan pesan konfirmasi bahwa data tidak ditemukan.

```
72● private static void cariNamaSiswa (Scanner console) {  
73     System.out.println("Masukan nama siswa yang ingin dicari :");  
74     String nama = console.nextLine();  
75     if (daftarNamaSiswa.contains(nama)) {  
76         System.out.println("Nama siswa berhasil ditemukan : " + nama + ".");  
77     } else {  
78         System.out.println("Nama siswa tidak ditemukan");  
79     }  
80 }  
81 }
```



g. Program Mahasiswa

1. **Buat new class** dengan klik kanan pada package pekan1 dan beri nama 'Mahasiswa', centang method public static void.
2. Deklarasi Atribut (Variabel Instance), bertipe String untuk menyimpan data nim, nama, dan prodi, tanpa menggunakan modifier, sehingga hanya dapat diakses oleh kelas yang sama berada dalam package (pekan 1).

```
1 package pekan1;
2
3 public class Mahasiswa {
4     String nim;
5     String nama;
6     String prodi;
```

3. Membuat Konstruktor, berfungsi untuk inisialisasi objek 'Mahasiswa' dengan nilai NIM, nama, dan prodi saat pembuatan objek. Keyword 'this.' untuk membedakan variabel instance (this.nim) dengan parameter konstruktor (nim).

```
7     Mahasiswa(String nim, String nama, String prodi) {
8         this.nim = nim;
9         this.nama = nama;
10        this.prodi = prodi;
11    }
```

4. Merupakan method toString(), @Override menandakan bahwa method ini meng-override method toString() dari kelas induk (Object). Bersifat public yang artinya dapat diakses dari luar kelas. Berfungsi mengembalikan representasi string dari objek Mahasiswa dalam format: "NIM: 123, Nama: xxxx, Prodi: Informatika".

```
12
13    @Override
14    public String toString() {
15        return "NIM: " + nim + ", Nama: " + nama + ", Prodi: " + prodi;
16    }
17 }
```

h. Program MahasiswaMain

1. **Buat new class** dengan klik kanan pada package pekan1 dan beri nama 'MahasiswaMain', centang method public static void.
2. **Import** ArrayList class dari paket **java.util.ArrayList** terlebih dahulu untuk menggunakan fungsi/method dari classnya. **Import** Scanner dari package **java.util.Scanner** untuk fungsi menerima input.

```
1 package pekan1;
2
3 import java.util.ArrayList;
4 import java.util.Scanner;
5
6 public class MahasiswaMain {
```

3. **Deklarasi ArrayList** yang menyimpan objek Mahasiswa (NIM, nama, prodi) dari kelas mahasiswa.java. Deklarasi **scanner** dengan nama scanner, dan deklarasi **variabel** integer bernama 'choice'.

```
8     public static void main(String[] args) {
9         ArrayList<Mahasiswa> mahasiswaList = new ArrayList<>();
10        Scanner scanner = new Scanner (System.in);
11        int choice;
```

4. Do-while loop, dimana **do** loop akan langsung menjalankan **sysout** untuk menampilkan menu tersedia dari main program. Variabel 'choice'

yang dibuat sebelumnya **menerima input** user, dan `console.nextLine()` untuk membersihkan buffer input (mencegah bug akibat \n yang tertinggal). Bagain ini berfungsi sebagai Entry Point Program/menu.

```

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(); //Consume newline

```

5. Struktur switch-case untuk Menu, kondisi dari swicth menggunakan variabel 'choice' yang sebelumnya berfungsi untuk menampung input. Case 1 tambahMahasiswa, memberikan display (sysout) kepada user untuk memasukan data, terdapat 3 variabel String yang berfungsi menerima input yaitu variabel nim, nama, dan prodi. Menggunakan method add(), lalu membuat objek Mahasiswa baru dan menambahkannya ke ArrayList, diakhiri dengan *break*.

```

22     switch (choice) {
23     case 1 :
24         System.out.println("Masukan NIM : ");
25         String nim = scanner.nextLine();
26         System.out.println("Masukan Nama : ");
27         String nama = scanner.nextLine();
28         System.out.println("Masukan Prodi : ");
29         String prodi = scanner.nextLine();
30         mahasiswaList.add(new Mahasiswa (nim, nama, prodi));
31         break;

```

6. Case 2 tampilkanSemuaMahasiswa, menampilkan semua data mahasiswa yang tersimpan dalam ArrayList, menggunakan for loop untuk menampilkan setiap data yang tersimpan dalam arraylist mahasiswa agar tersusun rapi.

```

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

```

7. Case 3 hapusMahasiswaBerdasarkanNIM, menampilkan sysout yang meminta user untuk memasukan input, variabel String 'nimHapus' sebagai penampung input user, implementasi method removeIf().

```

40     case 3 :
41         System.out.println("Masukkan NIM yang akan dihapus : ");
42         String nimHapus = scanner.nextLine();
43         mahasiswaList.removeIf (mhs -> mhs.nim.equals(nimHapus));
44         break;
45

```

8. Case 4 cariMahasiswaBerdasarkanNIM, menampilkan sysout yang meminta user untuk memasukan input, variabel String 'nimCari' sebagai penampung input user, implementasi method equals() pada if statement.

```

46         case 4 :
47             System.out.println("Masukan NIM yang dicari : ");
48             String nimCari = scanner.nextLine();
49             for (Mahasiswa mhs : mahasiswaList) {
50                 if (mhs.nim.equals(nimCari)) {
51                     System.out.println(mhs);
52                     break;
53                 } else {
54                     System.out.println("NIM tidak ada");
55                 }
56             }
57             break;

```

9. Case 5 keluar, berfungsi untuk menghentikan do-while loop

```

59         case 5 :
60             System.out.println("Keluar dari program.");
61             break;

```

10. Default Menangani input selain 1-5. Scanner.close() berfungsi untuk menghentikan proses inputan, untuk menghindari memory leak.

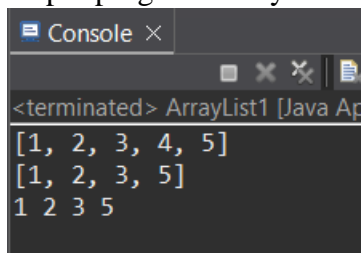
```

63         default :
64             System.out.println("Pilihan tidak valid.");
65         }
66     } while (choice != 5);
67     scanner.close();
68 }
69 }
70 }

```

#### D. Hasil Pratikum

1. Output program ArrayList1.

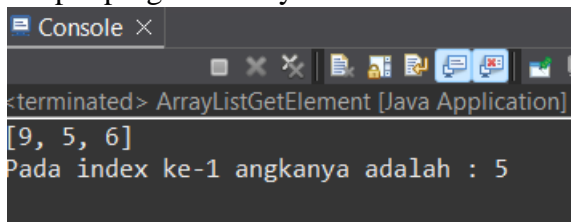


```

<terminated> ArrayList1 [Java Ap
[1, 2, 3, 4, 5]
[1, 2, 3, 5]
1 2 3 5

```

2. Output program ArrayListGetElement.

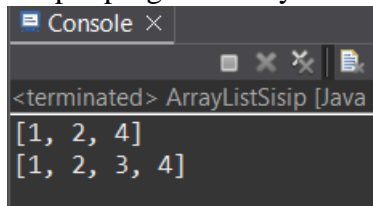


```

<terminated> ArrayListGetElement [Java Application] 0
[9, 5, 6]
Pada index ke-1 angkanya adalah : 5

```

3. Output program ArrayListSisip.



```

<terminated> ArrayListSisip [Java .
[1, 2, 4]
[1, 2, 3, 4]

```

4. Output program DaftarSiswa.

```
Console x
DaftarSiswa (1) [Java Application] C:\Program Files\Ja

menu :
1. Tambah nama siswa
2. Tampilan daftar siswa
3. Hapus nama siswa
4. Cari nama siswa
5. Keluar
Masukan pilihan anda :
1
Masukan nama siswa :
budi
Nama siswa budi berhasil ditambahkan.

menu :
1. Tambah nama siswa
2. Tampilan daftar siswa
3. Hapus nama siswa
4. Cari nama siswa
5. Keluar
Masukan pilihan anda :
2
Daftar Nama Siswa :
budi

[budi]

menu :
1. Tambah nama siswa
2. Tampilan daftar siswa
3. Hapus nama siswa
4. Cari nama siswa
5. Keluar
Masukan pilihan anda :
```

5. Ouput program MahasiswaMain.

```
Console x
MahasiswaMain [Java Application] C:\Program Files\Java\jdk

Menu :
1. Tambah Mahasiswa
2. Tampilkan Semua Mahasiswa
3. Hapus Mahasiswa Berdasarkan NIM
4. Cari Mahasiswa Berdasarkan NIM
5. Keluar
Pilih menu :
1
Masukan NIM :
2411
Masukan Nama :
budi
Masukan Prodi :
informatika
Menu :
1. Tambah Mahasiswa
2. Tampilkan Semua Mahasiswa
3. Hapus Mahasiswa Berdasarkan NIM
4. Cari Mahasiswa Berdasarkan NIM
5. Keluar
Pilih menu :
2
Data Mahasiswa :
NIM: 2411, Nama: budi, Prodi: informatika
Menu :
1. Tambah Mahasiswa
2. Tampilkan Semua Mahasiswa
3. Hapus Mahasiswa Berdasarkan NIM
4. Cari Mahasiswa Berdasarkan NIM
5. Keluar
Pilih menu :
```

## **E. Kesimpulan**

Setelah melakukan praktikum ini kita dapat memahami dan mengaplikasikan ArrayList serta Scanner Class untuk membuat program yang lebih kompleks sehingga seperti membuat program interaktif seperti program daftarSiswa dan MahasiswaMain. Menggunakan ArrayList untuk membuat program penyimpanan data. Mengaplikasikan Scanner class untuk menerima input user dan memasukannya ke dalam ArrayList dengan penerapan berbagai method ArrayList seperti method add(). Membuat program OPP pada mahasiswa.java dengan mahasiswaMain.java.