**LAPORAN PRAKTIKUM 1**

**STRUKTUR DATA**

**A logo of a university

AI-generated content may be incorrect.**

Dosen Pengampu:  
Dr. Wahyudi, M.Kom.

Disusun Oleh:  
Muhammad Luthfi Kautsar Rizata – 2311532020

**DEPARTEMEN INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS ANDALAS**

**2025**

1. A screen shot of a computer code

   AI-generated content may be incorrect.**Class Arraylist1**

Program di atas merupakan implementasi dasar dari penggunaan ArrayList di dalam bahasa pemrograman Java. Program ini berada dalam package bernama pekan1 dan memiliki satu kelas publik bernama Arraylist1 yang di dalamnya terdapat metode main() sebagai titik awal eksekusi program. Pertama, program mendeklarasikan sebuah variabel n bertipe int dengan nilai 5. Nilai ini kemudian digunakan untuk menginisialisasi sebuah objek ArrayList bertipe Integer bernama arrli dengan kapasitas awal sebanyak n.

Selanjutnya, program menggunakan perulangan for dari i = 1 hingga i <= n untuk menambahkan elemen-elemen bertipe Integer ke dalam arrli, yaitu angka dari 1 hingga 5. Setelah elemen-elemen tersebut berhasil ditambahkan, program mencetak isi dari arrli ke layar, sehingga akan menampilkan [1, 2, 3, 4, 5]. Setelah itu, program menghapus elemen pada indeks ke-3 (indeks dimulai dari 0) menggunakan metode remove(). Karena indeks ke-3 pada saat itu adalah elemen 4, maka elemen 4 akan dihapus dari arrli, dan struktur data tersebut menjadi [1, 2, 3, 5]. Isi dari arrli setelah penghapusan kemudian dicetak kembali.

Terakhir, program menggunakan perulangan for untuk mencetak setiap elemen dari arrli satu per satu tanpa tanda kurung atau koma, hanya dipisahkan oleh spasi. Proses ini dilakukan dengan menggunakan metode get() untuk mengakses elemen pada indeks tertentu. Output akhir dari program ini, setelah penghapusan dan pencetakan ulang, adalah: 1 2 3 5. Program ini menunjukkan penggunaan dasar ArrayList dalam menyimpan, mengakses, dan memodifikasi data secara dinamis dalam Java.

1. **A screen shot of a computer program

   AI-generated content may be incorrect.Class ArrayListGetElement**

Program di atas merupakan contoh sederhana penggunaan metode get() pada ArrayList dalam bahasa Java. Program ini berada dalam package pekan1 dan terdiri dari satu kelas publik bernama ArrayListGetElemen yang berisi metode main() sebagai titik awal eksekusi. Di dalam metode main(), pertama-tama dibuat sebuah objek ArrayList bertipe Integer dengan nama list. Kemudian, tiga buah elemen bertipe Integer ditambahkan ke dalam list secara berurutan, yaitu angka 9, 5, dan 6.

Setelah ketiga elemen tersebut dimasukkan, program mencetak isi dari list ke layar menggunakan System.out.println(list), yang akan menampilkan [9, 5, 6]. Selanjutnya, program menggunakan metode get(1) untuk mengambil elemen pada indeks ke-1 dari list. Karena indeks dalam Java dimulai dari 0, maka elemen pada indeks ke-1 adalah angka 5. Nilai tersebut kemudian disimpan ke dalam variabel n bertipe Integer.

Akhirnya, program mencetak informasi mengenai elemen yang diambil dari indeks ke-1 dengan mencetak kalimat: "Pada Indeks ke 1 angkanya adalah:5". Program ini memperlihatkan bagaimana cara mengambil elemen tertentu dari sebuah ArrayList dengan menggunakan metode get(), yang sangat berguna ketika kita ingin mengakses data pada posisi tertentu dalam struktur data dinamis tersebut.

1. **Class ArrayListSisip**

A computer screen shot of a program code

AI-generated content may be incorrect.

Program di atas merupakan contoh penggunaan metode add() dengan dua argumen pada struktur data ArrayList dalam bahasa Java untuk menyisipkan elemen pada posisi tertentu. Program ini berada dalam package pekan1 dan terdiri dari satu kelas publik bernama ArrayListSisip. Di dalam metode main(), program mendeklarasikan dan menginisialisasi sebuah objek ArrayList bertipe Integer dengan nama list. Selanjutnya, tiga elemen bertipe bilangan bulat, yaitu 1, 2, dan 4, ditambahkan ke dalam list menggunakan metode add(). Hasil dari penambahan ini adalah list berisi [1, 2, 4], yang kemudian dicetak ke layar.

Langkah berikutnya adalah proses penyisipan elemen dengan menggunakan metode add(index, element). Program menyisipkan angka 3 pada indeks ke-2 menggunakan perintah list.add(2, 3). Karena indeks dalam Java dimulai dari 0, maka elemen pada indeks ke-2 awalnya adalah angka 4. Setelah penyisipan, elemen 4 akan bergeser ke indeks berikutnya, dan struktur list menjadi [1, 2, 3, 4]. Hasil akhir ini kemudian dicetak kembali ke layar menggunakan System.out.println(list). Program ini menunjukkan bagaimana elemen baru dapat disisipkan ke posisi tertentu dalam ArrayList tanpa harus menulis logika manual untuk menggeser elemen, karena operasi ini telah ditangani secara otomatis oleh metode add() tersebut.

1. A computer screen with text

   AI-generated content may be incorrect.A screen shot of a computer program

   AI-generated content may be incorrect.**Class DaftarSiswa**

Program di atas merupakan aplikasi berbasis teks sederhana dalam bahasa Java yang memungkinkan pengguna untuk mengelola daftar nama siswa menggunakan struktur data ArrayList. Program ini berada dalam package pekan1 dan memiliki kelas publik bernama DaftarSiswa, yang menyimpan ArrayList<String> bernama daftarNamaSiswa secara statis untuk menyimpan daftar nama. Program menggunakan input dari pengguna melalui objek Scanner yang membaca data dari konsol.

Program bekerja dalam sebuah perulangan do-while yang menampilkan menu utama kepada pengguna. Menu ini mencakup lima opsi: menambahkan nama siswa, menampilkan daftar nama siswa, menghapus nama siswa, mencari nama siswa, dan keluar dari program. Pengguna diminta memilih salah satu menu dengan memasukkan angka dari 1 hingga 5. Berdasarkan pilihan tersebut, program akan mengeksekusi metode yang sesuai dengan menggunakan struktur switch-case.

Jika pengguna memilih menu 1, program akan memanggil metode tambahNamaSiswa() yang meminta pengguna memasukkan nama siswa, kemudian menambahkannya ke dalam ArrayList. Pada menu 2, metode tampilkanDaftarNamaSiswa() akan dijalankan untuk mencetak seluruh nama siswa yang telah tersimpan. Jika daftar kosong, akan muncul pesan bahwa tidak ada siswa dalam daftar.

Menu 3 memungkinkan pengguna menghapus nama tertentu dari daftar dengan memanggil metode hapusNamaSiswa(). Pengguna diminta memasukkan nama, lalu program akan mencoba menghapusnya dari ArrayList. Jika berhasil, akan muncul pesan konfirmasi, jika tidak, akan muncul pemberitahuan bahwa nama tidak ditemukan. Menu 4 digunakan untuk mencari apakah sebuah nama siswa terdapat dalam daftar atau tidak, menggunakan metode contains() pada ArrayList. Jika ditemukan, nama ditampilkan; jika tidak, muncul pesan bahwa nama tidak ditemukan.

Menu ke-5 digunakan untuk keluar dari program. Program ini menunjukkan pemanfaatan dasar ArrayList untuk menyimpan dan memanipulasi data string secara dinamis, serta penggunaan Scanner, switch-case, dan beberapa metode pendukung dalam membuat antarmuka interaktif berbasis teks yang sederhana namun fungsional

1. **A computer screen shot of a code

   AI-generated content may be incorrect.Class Mahasiswa**

Program di atas mendefinisikan sebuah kelas bernama Mahasiswa yang berada dalam package pekan1. Kelas ini merepresentasikan entitas mahasiswa dengan tiga atribut utama, yaitu nim (Nomor Induk Mahasiswa), nama, dan prodi (program studi), semuanya bertipe String. Ketiga atribut ini tidak menggunakan modifier akses private atau public, sehingga memiliki tingkat akses default (package-private), yang berarti hanya dapat diakses oleh kelas lain dalam package yang sama.

Konstruktor dalam kelas Mahasiswa memiliki tiga parameter yang sesuai dengan atribut-atribut tersebut. Ketika objek Mahasiswa dibuat menggunakan konstruktor ini, nilai dari parameter akan digunakan untuk menginisialisasi atribut nim, nama, dan prodi melalui keyword this, yang merujuk pada objek saat ini.

Selain itu, kelas ini juga mengoverride metode toString() dari kelas induk Object. Metode ini bertujuan untuk memberikan representasi string yang lebih bermakna ketika objek Mahasiswa dicetak atau dikonversi ke bentuk teks. Dalam implementasinya, metode ini akan mengembalikan string yang berisi informasi lengkap tentang NIM, nama, dan program studi mahasiswa, dalam format yang rapi dan informatif. Dengan adanya override ini, saat objek dicetak menggunakan System.out.println(), informasi yang muncul akan langsung berupa deskripsi atribut-atribut mahasiswa, bukan referensi memori default yang biasanya ditampilkan oleh Object.toString().

1. **Class MahasiswaMain**

A screen shot of a computer program

AI-generated content may be incorrect.

A computer screen shot of code

AI-generated content may be incorrect.

Program di atas merupakan implementasi aplikasi berbasis teks untuk mengelola data mahasiswa menggunakan konsep pemrograman berorientasi objek (OOP) dalam bahasa Java. Program ini berada dalam package pekan1 dan menggunakan dua kelas utama: kelas Mahasiswa (yang telah didefinisikan sebelumnya untuk merepresentasikan objek mahasiswa dengan atribut nim, nama, dan prodi) dan kelas MahasiswaMain sebagai program utama yang mengelola daftar mahasiswa.

Di dalam metode main(), sebuah objek ArrayList<Mahasiswa> digunakan untuk menyimpan daftar mahasiswa secara dinamis. Objek Scanner digunakan untuk membaca input dari pengguna. Program berjalan dalam loop do-while yang akan terus menampilkan menu hingga pengguna memilih opsi untuk keluar. Terdapat lima menu yang disediakan:

1. **Tambah Mahasiswa**  
   Pengguna diminta memasukkan data mahasiswa berupa NIM, nama, dan prodi. Data tersebut kemudian digunakan untuk membuat objek Mahasiswa baru, yang langsung ditambahkan ke dalam ArrayList.
2. **Tampilkan Semua Mahasiswa**  
   Program akan mencetak seluruh data mahasiswa yang tersimpan di dalam list. Karena metode toString() telah dioverride di kelas Mahasiswa, setiap objek akan ditampilkan dalam format informasi lengkap.
3. **Hapus Mahasiswa Berdasarkan NIM**  
   Pengguna diminta untuk memasukkan NIM mahasiswa yang akan dihapus. Program menggunakan metode removeIf() yang memanfaatkan lambda expression untuk menghapus objek berdasarkan pencocokan nilai NIM. Jika ditemukan dan berhasil dihapus, maka akan muncul pesan sukses, jika tidak, akan muncul pesan bahwa data tidak ditemukan.
4. **Cari Mahasiswa Berdasarkan NIM**  
   Program mencari mahasiswa dengan NIM tertentu dengan melakukan iterasi satu per satu terhadap seluruh elemen dalam list. Jika ditemukan, informasi mahasiswa akan ditampilkan. Jika tidak, akan muncul pemberitahuan bahwa data tidak ditemukan.
5. **Keluar dari Program**  
   Program akan berhenti dan menampilkan pesan bahwa pengguna telah keluar.

Secara keseluruhan, program ini menunjukkan bagaimana penggunaan ArrayList, Scanner, OOP, dan fitur seperti lambda expression dapat digunakan secara terpadu untuk membangun aplikasi CRUD (Create, Read, Update, Delete) sederhana yang interaktif dan mudah dipahami.