

UJIAN TENGAH SEMESTER

PRAKTIKUM

PEMROGRAMAN DAN ALGORITMA



DISUSUN OLEH

2024105658 – Aulia Dzakiyyah Marzuqah
2024105628 – Marshanda Ramadani
2024105630 – Mey Frida Zebua
2024105619 – Reisha Sholihah
2024105660 – Sarah Nur'Aini

FAKULTAS ILMU KOMPUTER DAN DESAIN
UNIVERSITAS KALBIS
2024

- Fungsi Tambah Data Mahasiswa

KODE:

```
static void addStudent(Scanner scanner) {
    List<String> students = readStudents();
    if (students.size() >= MAX_STUDENTS) {
        System.out.println("\nError: Kapasitas Sudah Penuh");
        return;
    }

    System.out.print("\nNIM: ");
    String nim = scanner.nextLine();
    if (nimExists(nim, students)) {
        System.out.println("Error: NIM Sudah Terdaftar");
        return;
    }

    System.out.print("Nama: ");
    String name = scanner.nextLine();
    students.add(nim + " " + name);
    writeStudents(students);
    System.out.println("Sukses Menambahkan Data");
}
```

KETERANGAN:

`List<String> students = readStudents();`: Membaca data mahasiswa yang tersimpan di file dan menaruhnya dalam list.

`if (students.size() >= MAX_STUDENTS)`: Mengecek apakah jumlah mahasiswa sudah mencapai batas maksimum (100). Jika ya, program menampilkan pesan error dan menghentikan proses.

`String nim = scanner.nextLine();`: Meminta pengguna untuk memasukkan NIM.

`if (nimExists(nim, students))`: Mengecek apakah NIM sudah ada di daftar mahasiswa. Jika NIM sudah terdaftar, program menampilkan pesan error.

`String name = scanner.nextLine();`: Meminta pengguna untuk memasukkan nama mahasiswa.

`String name = scanner.nextLine();`: Menambahkan data NIM dan nama mahasiswa baru ke list mahasiswa.

`writeStudents(students);`: Menyimpan perubahan data ke file.

`System.out.println("Sukses Menambahkan Data");`: Menampilkan pesan bahwa data berhasil ditambahkan.

- Fungsi Lihat Data Mahasiswa

KODE:

```

static void viewStudents() {
    List<String> students = readStudents();
    if (students.isEmpty()) {
        System.out.println("\nTidak ada data mahasiswa");
    } else {
        System.out.println("\nDATA MAHASISWA");
        for (String student : students) {
            System.out.println(student);
        }
    }
}

```

KETERANGAN:

`static void viewStudents()`: Membaca data mahasiswa dari file.

`if (students.isEmpty())`: Mengecek apakah daftar mahasiswa kosong. Jika ya, menampilkan pesan bahwa tidak ada data mahasiswa.

`for (String student : students)`: Melakukan iterasi untuk menampilkan setiap mahasiswa dari daftar yang sudah dibaca.

Program menampilkan NIM dan nama mahasiswa secara berurutan, atau pesan bahwa tidak ada data jika list kosong.

- Fungsi Perbarui Data Mahasiswa

KODE:

```

static void updateStudent(Scanner scanner) {
    List<String> students = readStudents();
    System.out.print("\nNIM: ");
    String nim = scanner.nextLine();

    int index = findStudentIndex(nim, students);
    if (index == -1) {
        System.out.println("Error: NIM Tidak Ditemukan");
        return;
    }

    System.out.print("Nama Baru: ");
    String name = scanner.nextLine();
    students.set(index, nim + " " + name);
    writeStudents(students);
    System.out.println("Sukses Memperbarui Data");
}

```

KETERANGAN:

`List<String> students = readStudents();`: Membaca data mahasiswa yang ada.

`String nim = scanner.nextLine();`: Meminta pengguna untuk memasukkan NIM yang ingin diperbarui.

`int index = findStudentIndex(nim, students);`: Mencari indeks dari mahasiswa berdasarkan NIM.

`if (index == -1)`: Mengecek apakah NIM tidak ditemukan. Jika tidak ditemukan, program menampilkan pesan error.

`students.set(index, nim + " " + name);`: Memperbarui data mahasiswa yang sesuai dengan NIM.

`writeStudents(students);`: Menyimpan perubahan ke file.

`System.out.println("Sukses Memperbarui Data");`: Menampilkan pesan bahwa data berhasil diperbarui.

- Fungsi Hapus Data Mahasiswa

KODE:

```
static void deleteStudent(Scanner scanner) {
    List<String> students = readStudents();
    System.out.print("\nNIM: ");
    String nim = scanner.nextLine();

    int index = findStudentIndex(nim, students);
    if (index == -1) {
        System.out.println("Error: NIM Tidak Ditemukan");
        return;
    }

    students.remove(index);
    writeStudents(students);
    System.out.println("Sukses Menghapus Data");
}
```

KETERANGAN:

`List<String> students = readStudents();`: Membaca data mahasiswa dari file.

`String nim = scanner.nextLine();`: Meminta pengguna memasukkan NIM mahasiswa yang akan dihapus.

`int index = findStudentIndex(nim, students);`: Mencari indeks mahasiswa berdasarkan NIM.

`if (index == -1)`: Mengecek apakah NIM tidak ditemukan. Jika tidak ditemukan, menampilkan pesan error.

`students.remove(index);`: Menghapus mahasiswa dari list berdasarkan indeks yang ditemukan.

`writeStudents(students);`: Menyimpan perubahan ke file setelah penghapusan.

`System.out.println("Sukses Menghapus Data");`: Menampilkan pesan bahwa data mahasiswa berhasil dihapus.

- Fungsi Membaca Data Mahasiswa (Helper Function)

KODE:

```
static List<String> readStudents() {
    List<String> students = new ArrayList<>();
    try (BufferedReader reader = new BufferedReader(new FileReader(FILE_NAME)))
    {
        String line;
        while ((line = reader.readLine()) != null) {
            students.add(line);
        }
    } catch (IOException e) {
        System.out.println("\nKesalahan membaca file: " + e.getMessage());
    }
    return students;
}
```

KETERANGAN:

`List<String> students = new ArrayList<>();`: Membuat list untuk menampung data mahasiswa.

`BufferedReader reader = new BufferedReader(new FileReader(FILE_NAME));`: Membaca file berisi data mahasiswa.

`while ((line = reader.readLine()) != null)`: Membaca setiap baris dari file, menambahkan baris tersebut ke list.

Jika terjadi kesalahan saat membaca file, program akan menangkap exception dan menampilkan pesan error.

- Fungsi Menulis Data Mahasiswa (Helper Function)

KODE:

```
static void writeStudents(List<String> students) {
    try (BufferedWriter writer = new BufferedWriter(new FileWriter(FILE_NAME)))
    {
        for (String student : students) {
            writer.write(student);
            writer.newLine();
        }
    } catch (IOException e) {
        System.out.println("\nKesalahan menulis file: " + e.getMessage());
    }
}
```

KETERANGAN:

```
BufferedWriter writer = new BufferedWriter(new FileWriter(FILE_NAME));
```

Membuka file untuk menulis ulang data mahasiswa.

`for (String student : students)`: Mengiterasi setiap mahasiswa dalam list dan menuliskannya ke file.

`writer.newLine();`: Menambahkan baris baru setelah setiap data mahasiswa.

Jika terjadi kesalahan saat menulis file, program akan menampilkan pesan error.

- Fungsi Mencari Indeks Mahasiswa (Helper Function)

KODE:

```
static int findStudentIndex(String nim, List<String> students) {  
    for (int i = 0; i < students.size(); i++) {  
        if (students.get(i).startsWith(nim + " ")) {  
            return i;  
        }  
    }  
    return -1;  
}
```

KETERANGAN:

`for (int i = 0; i < students.size(); i++)`: Mengiterasi list mahasiswa untuk mencari NIM yang sesuai.

`if (students.get(i).startsWith(nim + " "))`: Mengecek apakah NIM pada posisi tersebut cocok dengan NIM yang dicari.

Jika cocok, mengembalikan indeks mahasiswa, jika tidak, mengembalikan -1 yang menandakan NIM tidak ditemukan.

- Fungsi Cek Apakah NIM Ada (Helper Function)

KODE:

```
static boolean nimExists(String nim, List<String> students) {  
    return students.stream().anyMatch(student -> student.startsWith(nim + "  
"));  
}
```

KETERANGAN:

`students.stream().anyMatch(...)`: Mengecek apakah ada mahasiswa di dalam list yang NIM-nya cocok dengan NIM yang diberikan.

Mengembalikan true jika NIM ada, atau false jika tidak ditemukan.