Nama : Evi Susilowati

NIM : 18090022

Kelas : 4D

Aplikasi Konsol

Data Mahasiswa

Aplikasi Data Mahasiswa merupakan aplikasi konsol / berbasis teks. aplikasi tersebut digunakan untuk melakukan perekaman data mahasiswa, konsep aplikasi ini sederhana, dimana akan selalu menampilkan menu secara terus menerus sampai pengguna menginginkan keluar dari aplikasi. Menu atau fitur yang terdapat pada aplikasi ini adalah operasi yang biasa terjadi pada perekaman data, berupa tambah data, ubah data, hapus data, dan tampil seluruh data.

Desain Aplikasi

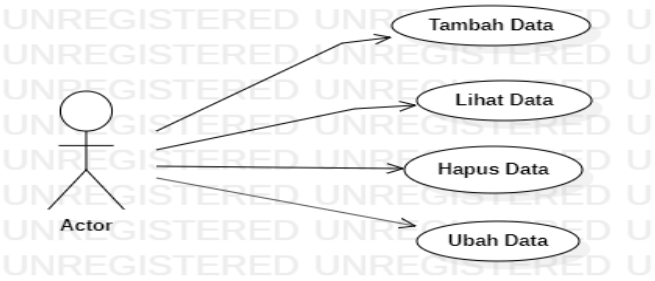
Gambar dibawah ini merupakan tampilan dari aplikasi ketika di Run.

```
---------------------------
APLIKASI  DATA  MAHASISWA
---------------------------
1. Tambah Data
2. Ubah Data
3. Hapus Data
4. Lihat Data
5. Keluar
Masukan Pilihan Anda ---> 
```

Di sini kita mempunyai kelas Data yang nantinya dapat membentuk obejek-objek berupa individu tiap mahasiswa, maka kelas ini memiliki ciri-ciri seperti : NIM, Nama, Jurusan dan Alamat.

```
public class Data {
    private String nim;
    private String nama;
    private String jurusan;
    private String alamat;
```

Agar lebih jelas, mari kita lihat Diagram Use-Case dari aplikasi tersebut.



Dari Diagram *use-case* dan tampilan aplikasi di atas, selanjutnya kita akan mengembangkannya menjadi diagram *Activity*, diagram *Activity* ini akan berbentuk skenario dari masing-masing *case* yang sudah ditentukan sebelumnya.

Diagram Activity Menu Utama

Aplikasi ini akan menampilkan menu utama saat pertama kali kita run.

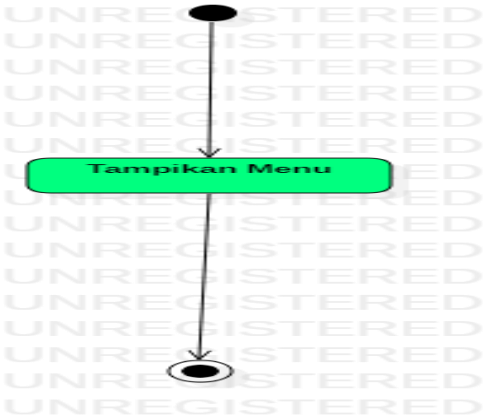
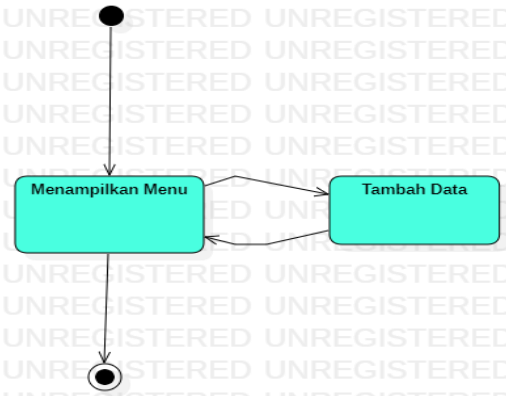


Diagram di atas menunjukkan bahwa begitu aplikasi dijalankan, maka akan menampilkan Menu Utama, setelah itu pengguna dapat langsung keluar dari aplikasi atau melakukan kegiatan lain seperti yang terdapat dalam diagram *activity* yang lain.

Diagram *activity* Penambahan Data

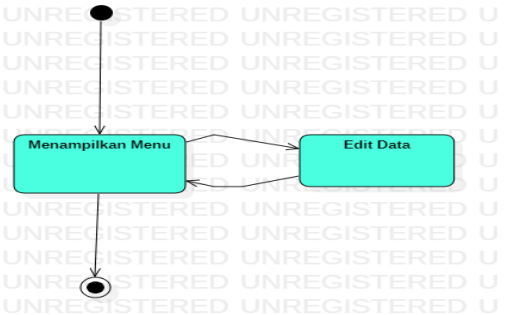
Saat pengguna akan menambahkan data mahasiswa, maka aktivitas yang terjadi adalah seperti berikut :



Saat pengguna akan menambahkan data, maka aplikasi akan menampilkan menu terlebih dahulu, kemudian menampilkan formulir penambahan data, untuk mengakhiri, maka pengguna akan dibawa ke Menu Utama terlebih dahulu, kemudian pengguna dapat keluar atau melakukan operasi yang lain.

Diagram *activity* Pengubahan Data

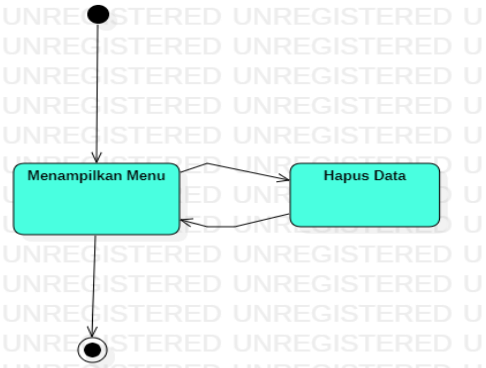
Saat pengguna akan mengubah data, maka aktivitas yang terjadi adalah seperti berikut



Saat pengguna akan mengubah data, sama seperti penambahan data, nantinya akan dimunculkan Menu Utama terlebih dahulu, lalu saat pengguna memilih menu pengubahan data, maka tampilkan formulir pengubahan data, setelah proses pengubahan data selesai, pengguna akan dibawa kembali ke Menu Utama, kemudian mengakhiri aplikasi atau melakukan operasi lain seperti pada Diagram *Activity* yang lain.

Diagram *Activity* Penghapusan Data

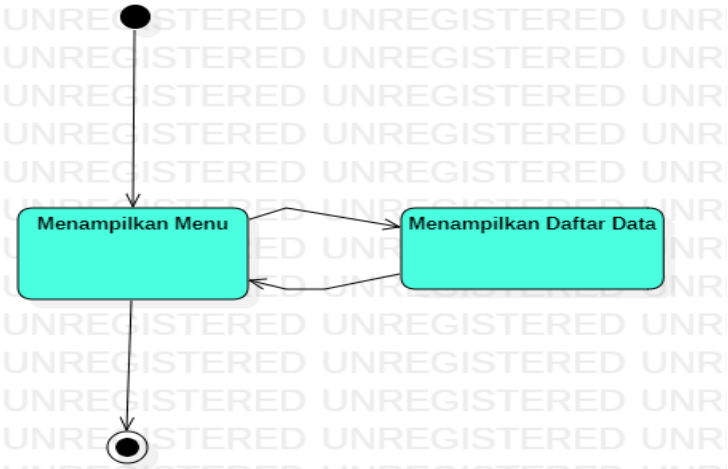
Saat pengguna menghapus data, maka aktivitas yang terjadi adalah seperti berikut :



Seperti aktivitas yang lain, sebelum penghapusan data, pengguna akan dibawa ke Menu Utama, setelah itu menampilkan formulir penghapusan data, setelah proses penghapusan selesai, pengguna akan dibawa kembali ke Menu Utama, lalu melanjutkan proses lainnya.

Diagram *Activity* Tampilkan Daftar Data

Saat pengguna ingin menampilkan data, maka aktivitas yang terjadi adalah seperti berikut :



Aktivitas ini pun diawali dari Menu Utama, setelah itu pengguna akan dibawa ke fitur Tampilkan Data, setelah itu akan kembali lagi ke Menu Utama, kemudian pengguna dapat mengakhiri aplikasi atau melakukan aktivitas yang lain.

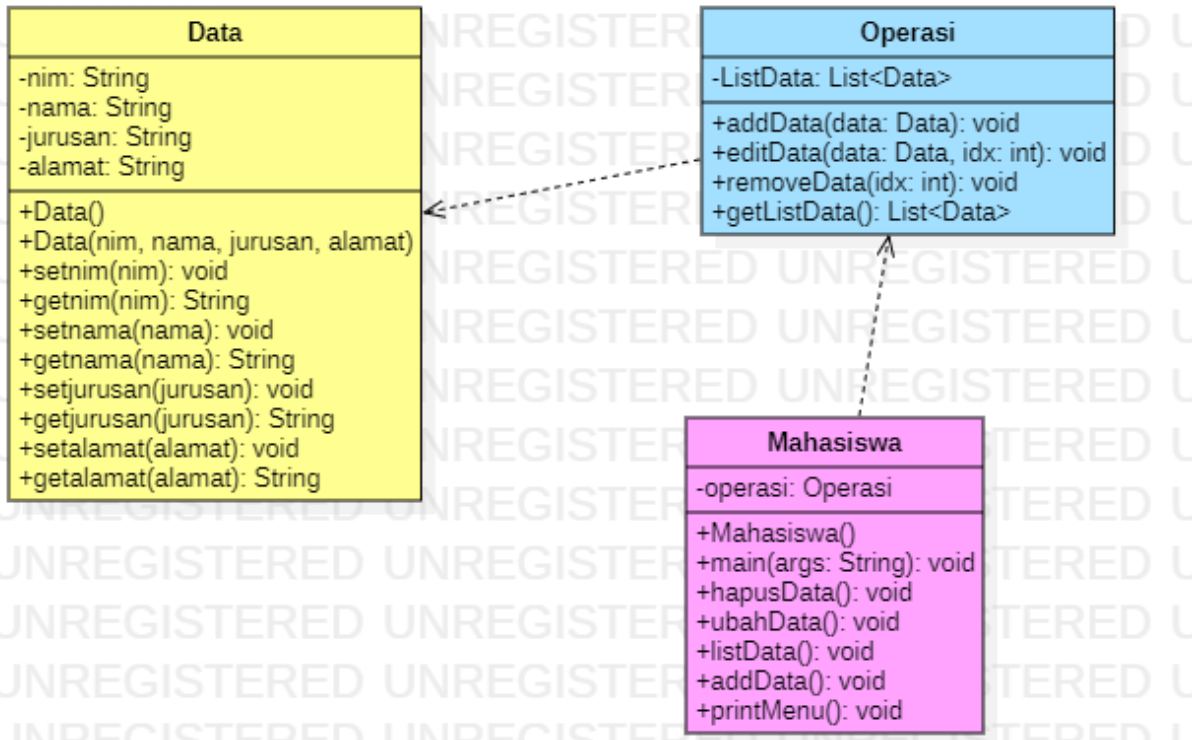
Dari konsep dan informasi yang terdapat pada diagram *use-case*, diagram *activity*, dan tampilan Menu Aplikasi, maka dibentuklah kelas-kelas seperti berikut :
• **Kelas Data** yang nantinya menjadi kelas yang difungsikan untuk merekam informasi data mahasiswa berupa nim, nama, jurusan, dan alamat.
• **Kelas Operasi** yang nantinya digunakan untuk melakukan operasi data untuk tiap menu

yang ditampilkan

• **Kelas Mahasiswa** yang digunakan untuk melakukan pengaturan tampilan aplikasi.

Desain diagram kelas akan terlihat seperti berikut :

Desain diagram kelas



Dari diagram kelas tersebut, kita akan mencoba membuat skenario teknis untuk tiap *activity* yang telah kita desain dalam bentuk diagram *sequence* seperti berikut:

Diagram *Sequence* Menu Utama

Diagram ini akan terlihat seperti gambar berikut :

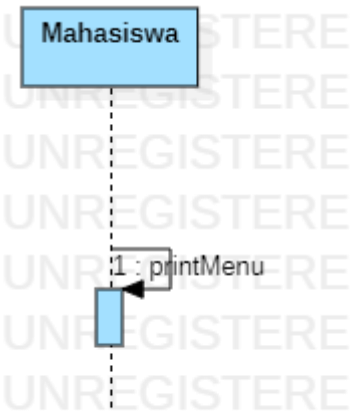


Diagram ini menunjukkan di dalam kelas Mahasiswa nantinya akan dipanggil *method* printMenu() untuk menampilkan menu utama.

Diagram *Sequence* Penambahan Data

Diagram ini akan terlihat seperti gambar berikut :

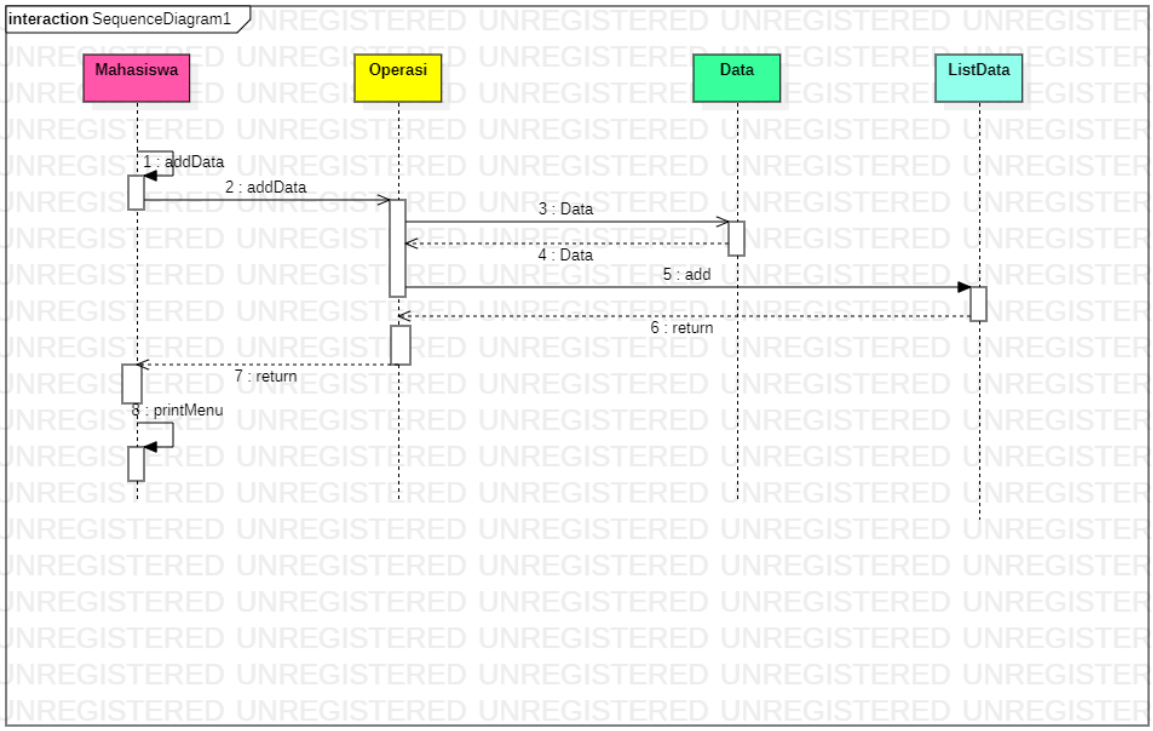


Diagram ini menggambarkan pada saat pengguna memilih menu penambahan data, *method* addData() dipanggil untuk menampilkan formulir penambahan data, setelah itu proses penambahan data dilakukan dengan memanggil *method* addData() milik kelas Operasi, yang

di dalamnya akan membentuk instan dari kelas Data, setelah itu akan disimpan dalam sebuah List dengan tipe data Data melalui *method* add(). Setelah seluruh proses tersebut selesai, kelas Mahasiswa kembali memanggil *method* printMenu() untuk menampilkan Menu Utama.

Diagram *Sequence* Pengubahan Data

Diagram ini akan terlihat seperti berikut :

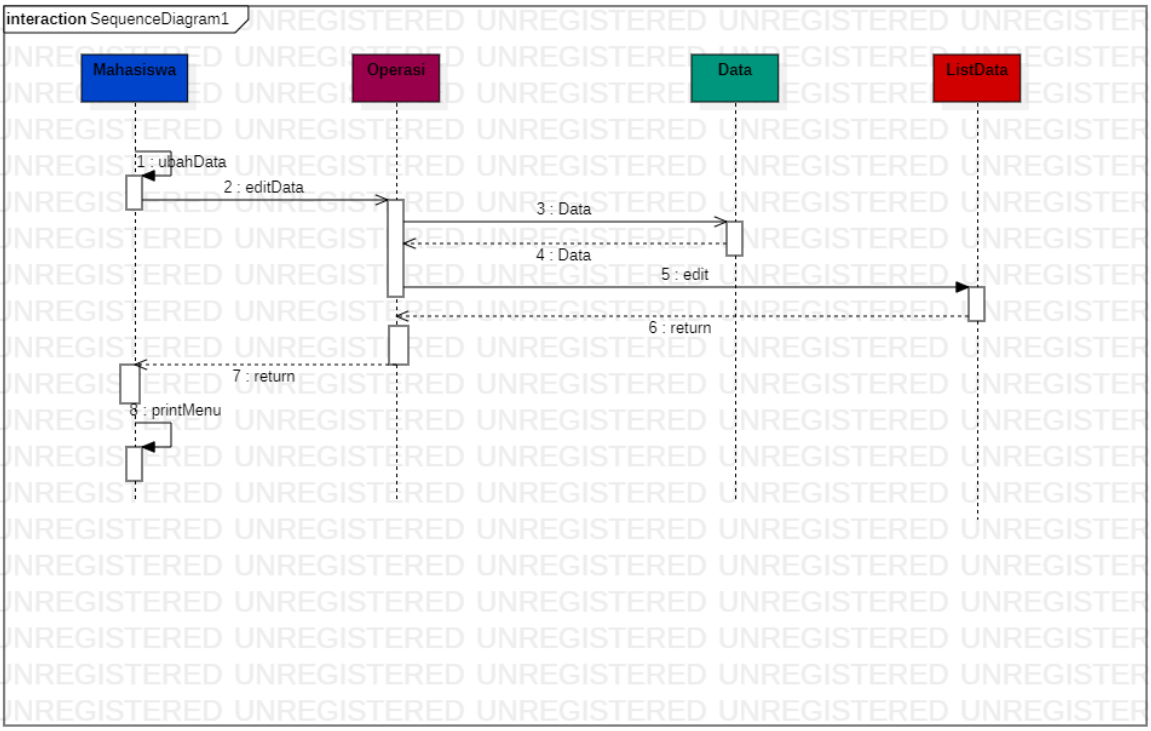
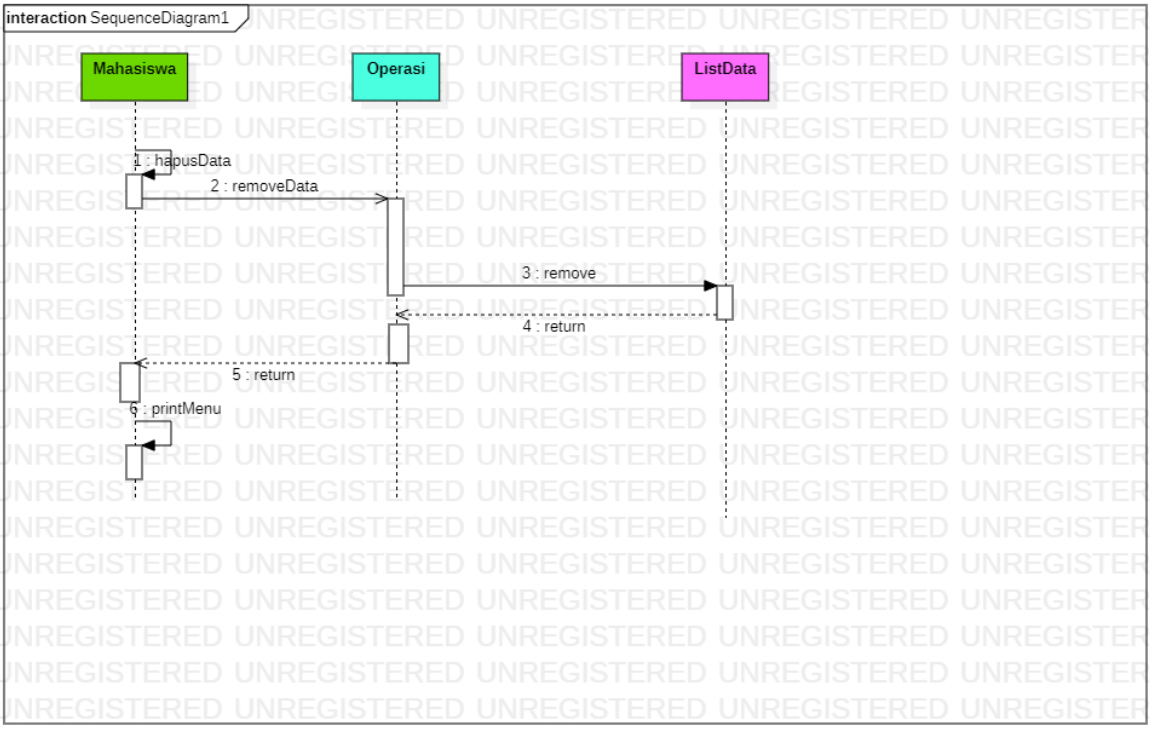


Diagram tersebut menggambarkan pada saat pengguna memilih menu pengubahan data, maka *method* ubahData() akan dipanggil untuk menampilkan formulir ubah data, setelah pengisian formulir selesai, lalu dilakukan simpan data, maka *method* editData() milik kelas Operasi, di dalam *method* ini dibentuk instan baru dari kelas Data, lalu proses berikutnya adalah mengubah data pada List<Data> dengan memanggil *method* edit(). Pada akhirnya setelah seluruh proses dijalani, program akan memanggil *method* printMenu() milik kelas Mahasiswa.

Diagram *Sequence* Penghapusan Data

Diagram ini akan terlihat seperti berikut :



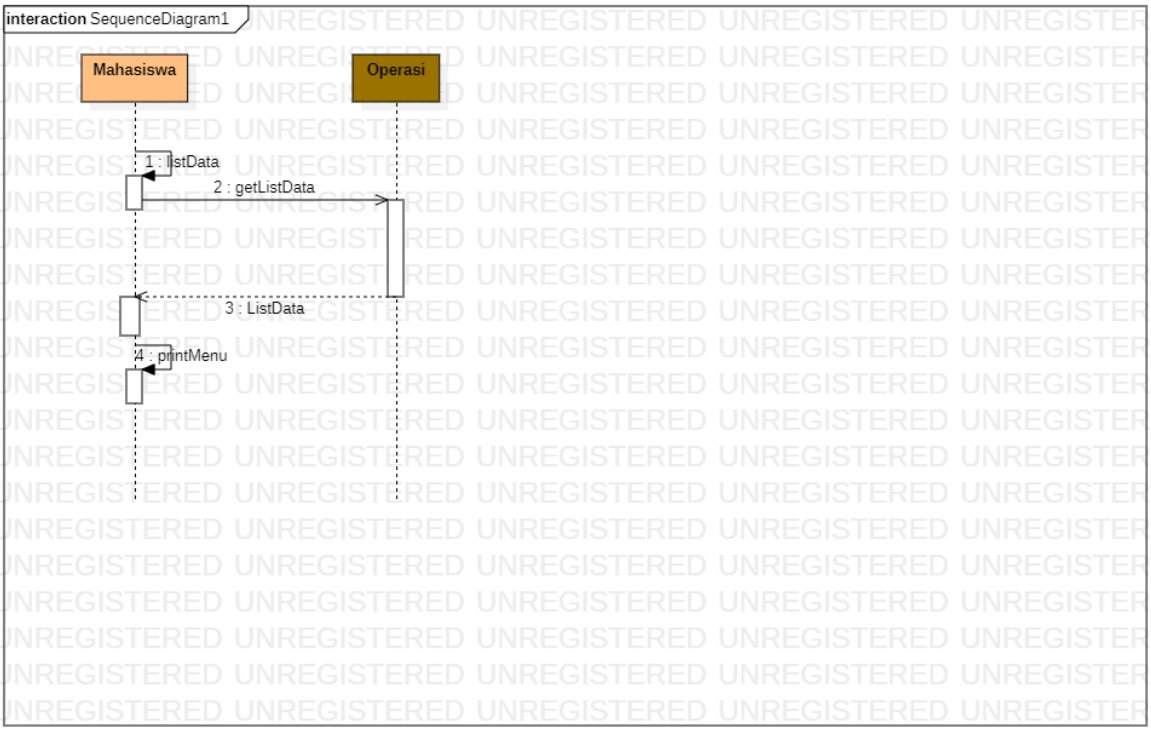
Diagramnya terlihat lebih sederhana, pada saat pengguna memilih menu hapus data, *method* hapusData() milik kelas Mahasiswa sehingga formulir untuk menghapus tampil, pada saat proses penghapusan data, nantinya *method* removeData() milik kelas Operasi akan dipanggil, yang kemudian didalam *method* ini akan memanggil *method* remove() milik kelas List, yang pada akhir proses penghapusan data pada List, kemudian dipanggil *method* printMenu() untuk memunculkan Menu Utama kembali.

Diagram *Sequence* Tampilkan Daftar Data

Diagram ini akan terlihat seperti berikut :

Diagramnya jauh lebih sederhana, menggambarkan pada saat pengguna memilih menu lihat data, maka *method* listData() akan dipanggil untuk menampilkan daftar data, di dalam *method* ini kemudian memanggil *method* getListData() milik kelas Operasi, dari nilai yang dikembalikan oleh *method* ini kemudian ditampilkan ke layar, dan langsung dipanggil kembali *method* printMenu() milik kelas

Mahasiswa untuk menampilkan Menu Utama.



Diagramnya jauh lebih sederhana, menggambarkan pada saat pengguna memilih menu lihat data, maka *method* listData() akan dipanggil untuk menampilkan daftar data, di dalam *method* ini kemudian memanggil *method* getListData() milik kelas Operasi, dari nilai yang dikembalikan oleh *method* ini kemudian ditampilkan ke layar, dan langsung dipanggil kembali *method* printMenu() milik kelas Mahasiswa untuk menampilkan Menu Utama.

Sebelumnya saya menggunakan Aplikasi Netbeans IDE 8.0.2 lalu saya membuat project baru bernama Mahasiswa dengan nama package mahasiswa. Didalam package mahasiswa terdapat 3 kelas yaitu Data.java , Mahasiswa.java dan Operasi.java.

Berikut ini source code dari Data.java

```
package mahasiswa;


public class Data {
    private String nim;
    private String nama;
    private String jurusan;
    private String alamat;
```

```
public Data() {
    nim = "";
    nama = "";
    jurusan = "";
    alamat = "";
}

public Data(String nim, String nama, String jurusan, String alamat) {
    this.nim = nim;
    this.nama = nama;
    this.jurusan = jurusan;
    this.alamat = alamat;
}

public void setnim(String nim) {
    this.nim = nim;
}

public void setnama(String nama) {
    this.nama = nama;
}

public void setjurusan(String jurusan) {
    this.jurusan = jurusan;
}
```

```
    public void setalamat(String alamat) {
        this.alamat = alamat;
    }

    public String getnim() {
        return nim;
    }

    public String getnama() {
        return nama;
    }

    public String getjurusan() {
        return jurusan;
    }

    public String getalamat() {
        return alamat;
    }
}
```

Kelas ini kita bangun dengan tujuan untuk menyiapkan *template* dari data yang akan disimpan atau dikelola. Data yang disimpan dapat dilihat pada atribut kelas yaitu: NIM, Nama, Jurusan dan Alamat.

Berikut Source Code dari Operasi.java

```
package mahasiswa;

import java.util.LinkedList;
```

```
import java.util.List;
public class Operasi {

    private static List<Data> listData = new LinkedList<Data>();

    public static void addData(Data data) {
        listData.add(data);
    }

    public static void editData(Data data, int idx) {
        listData.set(idx, data);
    }

    public static void removeData(int idx) {
        listData.remove(idx);
    }

    public static List<Data> getListData() {
        return listData;
    }
}
```

Kelas ini difungsikan memang untuk melakukan operasi data saja, dan data yang disimpan bisa kita lihat berada di atribut listData dengan tipe data berupa List yang berisi Data, perhatikan bahwa kita jadikan atribut ini dengan tanda static, agar datanya ditempatkan di alamat memori tunggal, begitu pula seluruh *method* operasi yang ada dijadikan static agar kita tidak perlu membentuk instan dari kelas Operasi ini, cukup langsung dilakukan saja operasi datanya.

Berikut Source Code dari Mahasiswa.java

```
package mahasiswa;
```

```
import java.util.List;
import java.util.Scanner;

public class Mahasiswa {

    /**
     * @param args the command line arguments
     */
    public void printMenu() {
        Scanner sc = new Scanner(System.in);
        System.out.println("-----------------------------------");
        System.out.println("APLIKASI  DATA  MAHASISWA");
        System.out.println("-----------------------------------");
        System.out.println("1. Tambah Data" + "\n2. Ubah Data" + "\n3. Hapus Data" +
"\n4. Lihat Data" + "\n5. Keluar");
        System.out.print("Masukan Pilihan Anda ---> ");
    }

    public void hapusData() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Urutan data yang dihapus ---> ");
        int idx = Integer.parseInt(sc.nextLine());
        // proses hapus data
        Operasi.removeData(idx - 1);
    }
```

```
public void ubahData() {
    Scanner sc = new Scanner(System.in);
    System.out.println();
    System.out.print("Urutan data yang diubah ---> ");
    int idx = Integer.parseInt(sc.nextLine());
    System.out.println("-------------------------");
    System.out.print("NIM       : ");
    String nim = sc.nextLine();
    System.out.print("Nama      : ");
    String nama = sc.nextLine();
    System.out.print("Jurusan     : ");
    String jurusan = sc.nextLine();
    System.out.print("Alamat    : ");
    String alamat = sc.nextLine();
    // proses ubah data
    Operasi.editData(new Data(nim, nama, jurusan, alamat), idx - 1);
}

public void listData() {
    List<Data> result = Operasi.getListData();
    for (int i = 0; i < result.size(); i++) {
        System.out.println();
        System.out.println("Data ke-" + (i + 1));
        System.out.println("  NIM       : " + result.get(i).getnim());
        System.out.println("  Nama      : " + result.get(i).getnama());
        System.out.println("  Jurusan     : " + result.get(i).getjurusan());
        System.out.println("  Alamat    : " + result.get(i).getalamat());
```

```
    }
}

public void addData() {
    Scanner sc = new Scanner(System.in);
    System.out.println();
    System.out.print("NIM : ");
    String nim = sc.nextLine();
    System.out.print("Nama : ");
    String nama = sc.nextLine();
    System.out.print("Jurusan : ");
    String jurusan = sc.nextLine();
    System.out.print("Alamat : ");
    String alamat = sc.nextLine();
    // proses tambah data
    Operasi.addData(new Data(nim, nama, jurusan, alamat));
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    Mahasiswa app = new Mahasiswa();
    while (true) {
        app.printMenu();
        int pilihan = sc.nextInt();
        switch (pilihan) {
            case 1:
                System.out.println("Formulir Tambah Data");
```

```
                app.addData();
                break;
            case 2:
                System.out.println("Daftar Data");
                app.listData();
                System.out.println("Formulir Ubah Data");
                app.ubahData();
                break;
            case 3:
                System.out.println("Daftar Data");
                app.listData();
                System.out.println("Formulir Hapus Data");
                app.hapusData();
                break;
            case 4:
                System.out.println("Daftar Data");
                app.listData();
                break;
            case 5:
                System.out.println("\nApakah Anda Ingin keluar?");
                System.out.println("1.Ya \t2.Tidak");
                char persetuju = sc.next().charAt(0);
                if (persetuju == '2') {
                    app.printMenu();
//                    pilihan = sc.nextInt();

                } else if (persetuju == '1') {
```

```
                System.out.println("Anda telah keluar dari Aplikasi Mahasiswa");
                System.out.println("TERIMA KASIH");
                System.exit(0);
            }
        }
    }
}
```

Kelas ini memiliki deklarasi terpanjang, karena berisi keseluruhan tatap muka pengguna, mulai dari Menu Utama, Formulir Penambahan Data, Pengubahan Data, Penghapusan Data, sampai ke tampilan pencetak Daftar Data pun ada disini, maka dari itu sebagian besar isinya adalah System.out.println.