**LAPORAN TUGAS AKHIR**

**PRAKTIKUM DASAR PEMROGRAMAN**

**SISTEM AKADEMIK BERBASIS COMMAND LINE**

****

Disusun oleh :

1. Annisa Eka Puspita (2341720131)

2. Noklent Fardian Erix (2341720082)

3. Muhamad Syaifullah (2341720013)

**PROGRAM STUDI D4 TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMARI**

**POLITEKNIK NEGERI MALANG**

**SEMESTER GANJIL**

**2023**

# **ABSTRAKSI**

Pada tugas akhir ini, penulis membuat sistem akademik berbasis command line. Sistem ini dibangun menggunakan bahasa pemrograman Java. Sistem ini memiliki fitur-fitur yang dapat digunakan untuk mengelola data akademik, seperti data mahasiswa, data dosen, data mata kuliah, dan data nilai.

Sistem ini dibangun dengan menggunakan struktur data dan algoritma yang sederhana. Hal ini dimaksudkan agar sistem ini mudah dipahami dan dikembangkan. Sistem ini juga dirancang agar dapat digunakan secara interaktif. Hasil pengujian menunjukkan bahwa sistem ini dapat berjalan dengan baik. Sistem ini dapat digunakan untuk mengelola data akademik dengan mudah dan efisien.

Selain itu, sistem ini juga memiliki beberapa fitur tambahan, yaitu: Sistem ini dapat digunakan untuk mencetak laporan akademik, seperti daftar mahasiswa, daftar dosen, daftar mata kuliah, dan daftar nilai. Sistem ini dapat digunakan untuk melakukan import dan export data akademik dari dan ke file. Sistem ini dapat digunakan untuk melakukan backup data akademik. Fitur-fitur tambahan ini dapat mempermudah penggunaan sistem ini dan meningkatkan efisiensi dalam pengelolaan data akademik.

# **KATA PENGANTAR**

Puji syukur kami panjatkan kehadorat Allah SWT atas segala rahmat dan hidah-Nya sehingga kami dapat menyelesaikan Laporan Tugas Akhir Praktikum Dasar Pemrograman Berbasis Command Line. Tujuan dari penyusunan laporan ini adalah sebagai bahan evaluasi hasil pembelajaran Pratikum Dasar Pemrograman.

Laporan ini merupakan salah satu syarat kelulusan mata kuliah Pratikum Dasar Pemrograman bagi Mahasiswa Program Studi Teknik Informatika , Jurusan Teknologi Infor masi Politeknik Negeri Malang.

Kami meyadari bahwa tanpa adanya dukungan dan kerja sama dari berbagai pihak, aktivitas ini tidak dapar berjalan dengan baik, untuk itu kami menyampaikan rasa terima kasih kepada.

1. Mungki Astiningrum, ST., M.Kom selaku Dosen mata kuliah Dasar Pemrograman dan Praktikum Dasar Pemrograman kelas kami.
2. Odhitya Desta Triswidrananta, S.Pd., M.Pd. selaku Dosen mata kuliah critical thinking and problem solving

# **DAFTAR ISI**

[**ABSTRAKSI** i](#_Toc154009517)

[**KATA PENGANTAR** ii](#_Toc154009518)

[**DAFTAR ISI** iii](#_Toc154009519)

[**DAFTAR GAMBAR** iv](#_Toc154009520)

[**BAB I PENDAHULUAN** 1](#_Toc154009521)

[**1.1** **Bahasa Pemrograman Java** 1](#_Toc154009522)

[**1.2 Command Line Interface** 2](#_Toc154009523)

[**BAB II SISTEM AKADEMIK** 3](#_Toc154009524)

[**2.1 Fitur Login** 3](#_Toc154009525)

[**2.2 Admin User Level** 6](#_Toc154009526)

[2.2.1 Modul Mahasiswa 7](#_Toc154009527)

[2.2.1.1 List Mahasiswa 9](#_Toc154009528)

[A. Daftarkan Mahasiswa 11](#_Toc154009529)

[B. Edit Data Mahasiswa 13](#_Toc154009530)

[C. Hapus Data Mahasiswa 15](#_Toc154009531)

[2.2.1.2 Transkip Nilai 17](#_Toc154009532)

[2.2.1.3 Presensi Mahasiswa 19](#_Toc154009533)

[2.2.1.4 Check Username dan Password Mahasiswa 21](#_Toc154009534)

[2.2.2 Modul Dosen 23](#_Toc154009535)

[2.2.2.1 Add Data Dosen 25](#_Toc154009536)

[2.2.2.2 Edit data Dosen 27](#_Toc154009537)

[2.2.2.3 Hapus BioDosen 29](#_Toc154009538)

[2.2.3 Modul Kursus 30](#_Toc154009539)

[2.2.3.1 Fitur penempatan kelas Mahasiswa 31](#_Toc154009540)

# **DAFTAR GAMBAR**

[Gambar 1.1 Bahasa Pemrograman Java 1 1](file:///D:\Tugas%20Akhir%20S1\Prakdaspro_Sistem_Adakemik\Other\Laporan\LaporanTA_kelompok2.docx#_Toc153911453)

# **BAB I PENDAHULUAN**

## **Bahasa Pemrograman Java**



Bahasa pemrograman Java telah menjadi salah satu bahasa yang paling berpengaruh dan banyak digunakan dalam dunia pengembangan perangkat lunak. Dikembangkan oleh James Gosling dan timnya di Sun Microsystems pada pertengahan tahun 1990-an, Java dirancang dengan fokus pada portabilitas lintas platform, memungkinkan program Java untuk dijalankan di berbagai sistem operasi tanpa perubahan kode. Keunggulan utama Java dalam hal ini adalah penggunaan bytecode, sebuah bentuk instruksi mesin virtual yang dapat dijalankan oleh Java Virtual Machine (JVM). JVM hadir pada berbagai sistem operasi, yang membuat program Java menjadi platform independen.

Salah satu fitur kunci dari bahasa pemrograman Java adalah paradigma pemrograman berorientasi objek. Dengan konsep objek dan kelas, Java memungkinkan pengembang untuk merancang aplikasi yang modular, mudah dipahami, dan mudah dikembangkan. Selain itu, Java juga dikenal dengan keamanan tingkat tinggi, berkat mekanisme kontrol akses yang diterapkan dalam JVM. Hal ini membuat Java menjadi pilihan populer untuk pengembangan aplikasi enterprise yang memerlukan standar keamanan yang tinggi.

Pada pembelajaran mata kuliah Praktikum dasar Pemrograman ini, kami telah diajari tentang hal hal yang ada pada bahasa pemrograman java. Seperti, Variable, Percabgan, Perulangan, Array, dan fungsi.

## **1.2 Command Line Interface**

Command Line Interface (CLI) adalah suatu antarmuka pengguna yang memungkinkan interaksi dengan sistem atau program melalui baris perintah teks. Berbeda dengan antarmuka pengguna grafis (GUI) yang menggunakan elemen visual seperti ikon dan jendela, CLI memungkinkan pengguna memberikan instruksi dengan mengetikkan perintah teks pada terminal atau baris perintah. Penggunaan CLI sering diutamakan dalam konteks efisiensi dan otomatisasi tugas, di mana pengguna dapat dengan cepat menjalankan perintah atau skrip untuk mengeksekusi tugas tertentu.

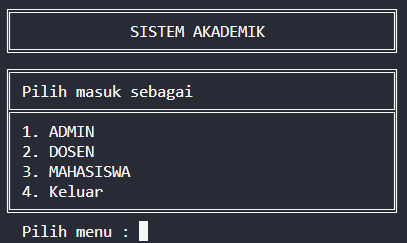
Salah satu keunggulan utama dari CLI adalah kemampuannya dalam scripting dan otomatisasi. Pengguna dapat membuat skrip atau file teks yang berisi serangkaian perintah untuk dieksekusi secara berurutan, memungkinkan otomatisasi tugas-tugas rutin. Hal ini menjadikan CLI pilihan yang populer di kalangan pengembang dan administrator sistem yang sering menghadapi tugas-tugas yang membutuhkan repetisi.

CLI umumnya digunakan pada sistem operasi Unix dan Linux, di mana terminal menyediakan akses langsung ke baris perintah. Walaupun CLI memerlukan pemahaman lebih dalam terkait sintaks dan perintah, kelebihannya dalam hal kustomisasi, kecepatan, dan otomatisasi membuatnya tetap menjadi alat yang sangat berguna dalam dunia teknologi informasi dan pengembangan perangkat lunak.

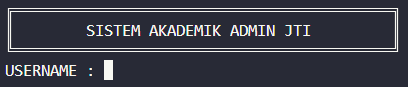
# **BAB II SISTEM AKADEMIK**

## **2.1 Fitur Login**

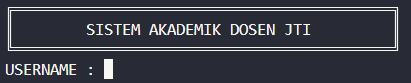
Untuk tampilan pertama dari program kami, user diminta untuk masuk sebagai role yang diinginkan. Jika role sudah meilih maka akan mengarah pada pengisian username dan password yang sudah tersimpan pada array.



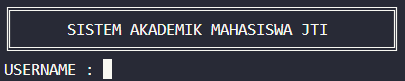
admin

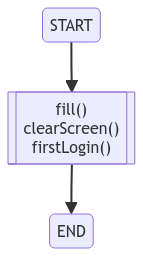


Dosen



Mahasiswa





Code fungsi main:

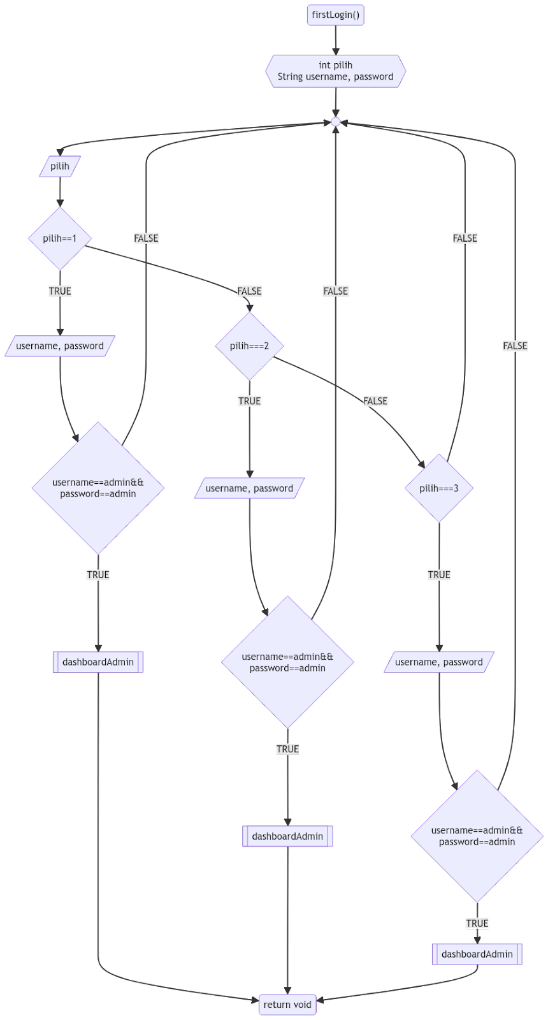
public static *void* main(String[] *args*) {

        fill();         // mengisi data array

        clearScreen();  // membersihkan layar

        firstLogin();   // login awal

    }

****

// login awal untuk login

    static *void* firstLogin() {

        while (true) {

            renderTitle("SISTEM AKADEMIK");

*int* userInput = pickMenu("Pilih masuk sebagai", role);

            clearScreen();

            switch (userInput) {

                case 1 *->* loginRole(role[0]);

                case 2 *->* loginRole(role[1]);

                case 3 *->* loginRole(role[2]);

            }

        }

    }

    // melakukan login sesuai role

    static *void* loginRole(String *level*) {

*int* counter = 0;

        while (true) {

            renderTitle("SISTEM AKADEMIK " + *level* + " JTI");

            String username = getInputStringWithLimit("USERNAME", 1, 10, true);

            String password = getInputStringWithLimit("PASSWORD", 1, 10, false);

            String userLogin;

            clearScreen();

            switch (*level*) {

                case "ADMIN" *->* {

                    userLogin = validasi(userAdmin, username, password);

                    if (userLogin != null) {

                        dashboardAdmin(userLogin);

                        return;

                    }

                }

                case "DOSEN" *->* {

                    userLogin = validasi(userDosen, username, password);

                    if (userLogin != null) {

                        dashboardDosen(userLogin);

                        return;

                    }

                }

                case "MAHASISWA" *->* {

                    userLogin = validasi(userMahasiswa, username, password);

                    if (userLogin != null) {

                        dashboardMahasiswa(userLogin);

                        return;

                    }

                }

            }

            counter++;

            if (counter == 3) {

                exitForce();

                return;

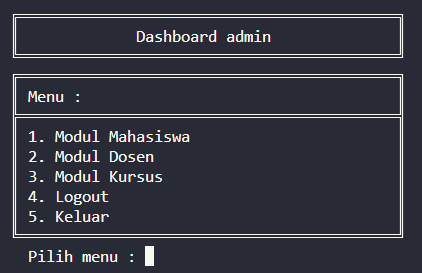
            }

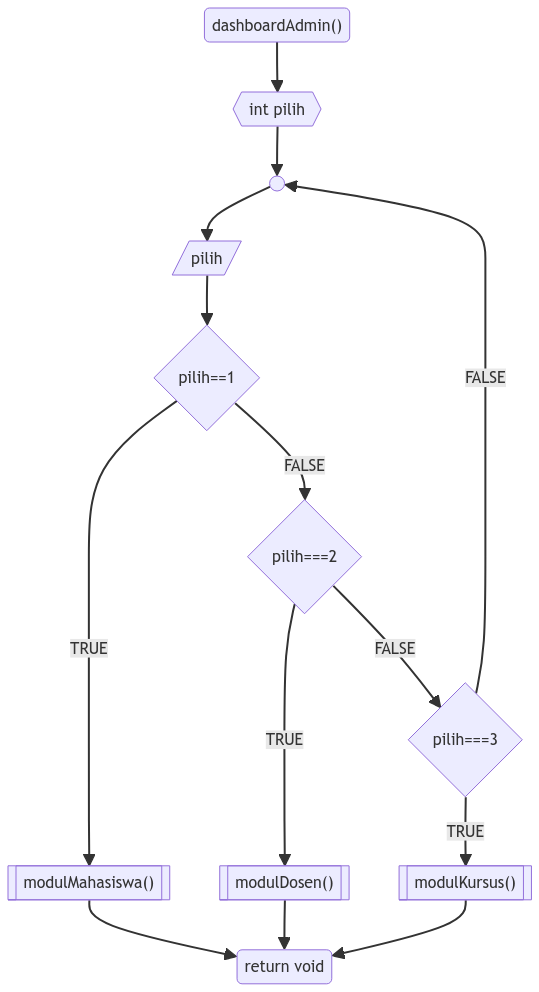
        }

    }

## **2.2 Admin User Level**

Ketika user berhasil masuk pada admin user level maka user akan ditampilkan dashboard admin. Dashboard Admin Untuk mengelola akses admin ke modul-modul dalam sistem (seperti modul mahasiswa, modul dosen, dan modul kursus) dengan opsi untuk keluar dari tampilan dashboard.





Code Dashboard admin

static *void* dashboardAdmin(String *user*) {

        while (true) {

            renderTitle("Dashboard " + *user*);

*int* userInput = pickMenu("Menu : ", **new** String[] {

                    "Modul Mahasiswa",

                    "Modul Dosen",

                    "Modul Kursus",

                    "Logout",

            });

            clearScreen();

            switch (userInput) {

                case 1 *->* modulMahasiswa();

                case 2 *->* modulDosen();

                case 3 *->* modulKursus();

                case 4 *->* {

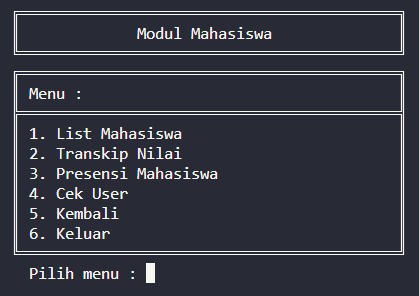
                    return;

            }

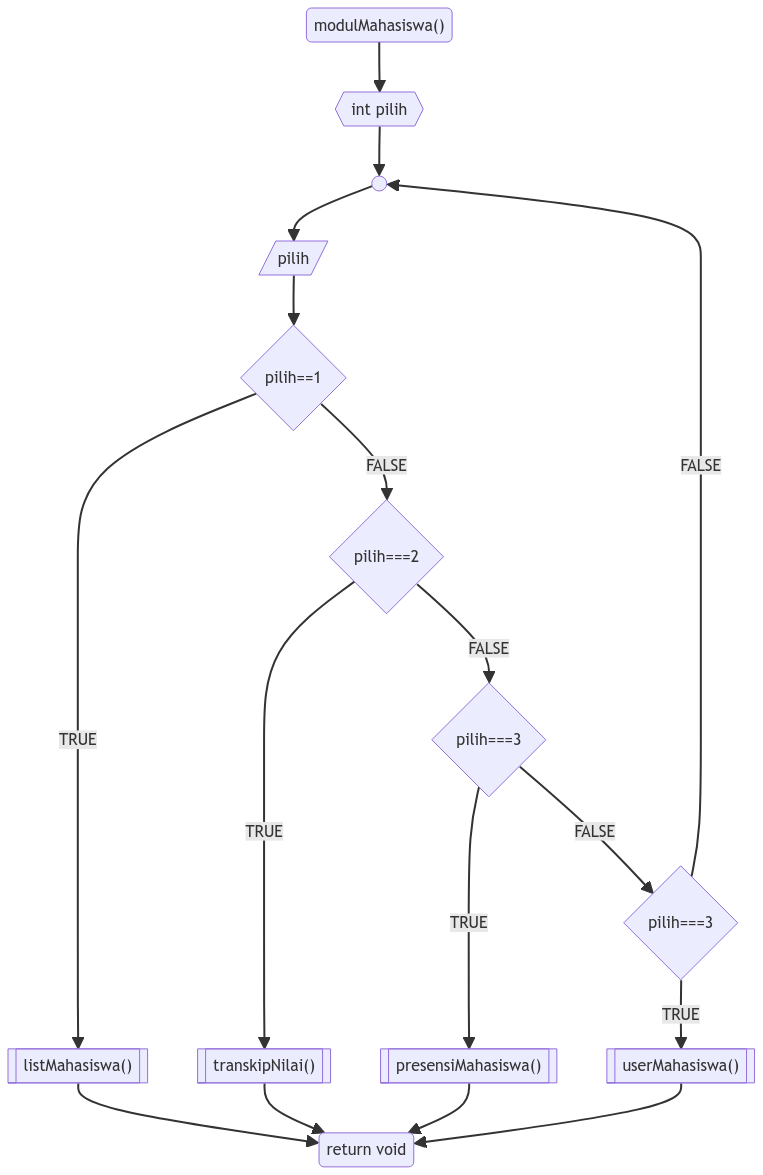
        }

    }

### 2.2.1 Modul Mahasiswa



Fitur pertama dalam user level admin adalah fitur untuk pengelolaan data mahasiswa. Admin dapat melakukan berbagai tindakan seperti melihat daftar mahasiswa, mengakses transkip nilai, mengelola presensi mahasiswa, dan melakukan pengecekan terhadap pengguna terdaftar sebelumnya.



Code modul mahasiswa :

static *void* modulMahasiswa() {

        while (true) {

            renderTitle("Modul Mahasiswa");

*int* userInput = pickMenu("Menu : ", **new** String[] {

                    "List Mahasiswa",

                    "Transkip Nilai",

                    "Presensi Mahasiswa",

                    "Cek User",

                    "Kembali",

            });

            clearScreen();

            switch (userInput) {

                case 1 *->* handleListMahasiswa();

                case 2 *->* transkipNilai();

                case 3 *->* presensiMahasiswaAdmin();

                case 4 *->* cekUserMahasiswa();

                case 5 *->* {

                    return;

            }

        }

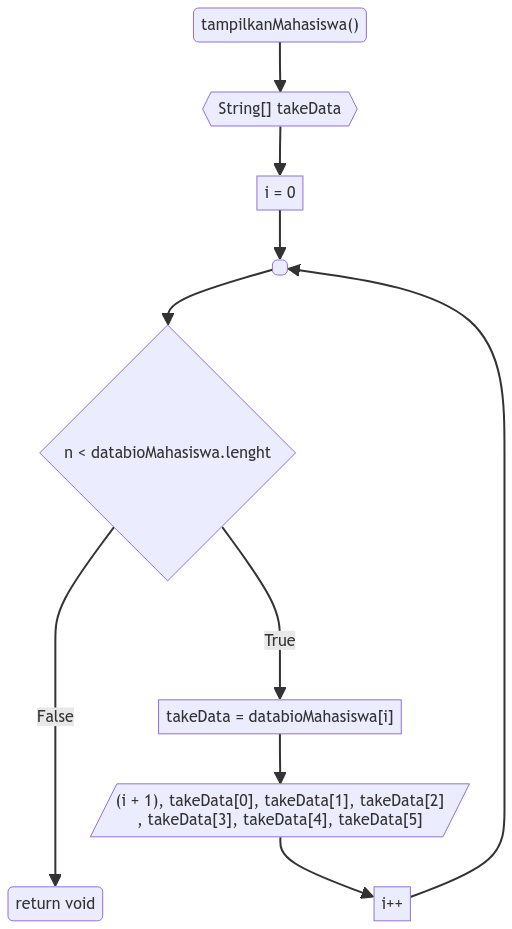
    }

}

#### 2.2.1.1 List Mahasiswa



Fitur ini memberikan fungsionalitas bagi admin untuk melihat daftar mahasiswa beserta opsi untuk menambah, mengedit, atau menghapus data mahasiswa.



Code porgram unutk list mahasiswa

static *void* handleListMahasiswa() {

        renderTitle("List Mahasiswa");

        while (true) {

            showDataBioMahasiswa(false);

*int* userInput = pickMenu("Menu : ", **new** String[] {

                    "Daftarkan Mahasiswa Baru",

                    "Edit Data Mahasiswa",

                    "Hapus Data Mahasiswa",

                    "Kembali"

            });

            clearScreen();

            switch (userInput) {

                case 1 *->* addDataBioMahasiswa();

                case 2 *->* editDataBioMahasiswa();

                case 3 *->* removeDataBioMahasiswa();

                case 4 *->* {

                    return;

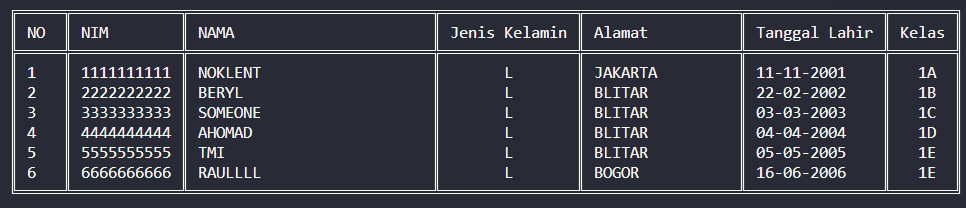
                }

            }

        }

    }

* **Funsgi untuk menampilkan tabel Biodata mahasiswa**



**Code program :**

static *void* showDataBioMahasiswa(*boolean* *isNIM*, String... *nim*) {

        String firstLine = "╔═════╦════════════╦═══════════════════════════╦═══════════════╦═════════════════╦═══════════════╦═══════╗";

        String middleLine = "╠═════╬════════════╬═══════════════════════════╬═══════════════╬═════════════════╬═══════════════╬═══════╣";

        String lastLine = "╚═════╩════════════╩═══════════════════════════╩═══════════════╩═════════════════╩═══════════════╩═══════╝";

        String formatTable = "║ %-3s ║ %-10s ║ %-25s ║       %-7s ║ %-15s ║ %-13s ║   %-3s ║%n";

        System.out.println(firstLine);

        System.out.format(

                "║ NO  ║ NIM        ║ NAMA                      ║ Jenis Kelamin ║ Alamat          ║ Tanggal Lahir ║ Kelas ║%n");

        System.out.println(middleLine);

        for (*int* i = 0; i < bioMahasiswa.length; i++) {

            String[] takeBio = bioMahasiswa[i];

            if (*isNIM* && *nim*[0].equals(takeBio[0])) {

                System.out.printf(formatTable, (i + 1), takeBio[0], takeBio[1], takeBio[2], takeBio[3], takeBio[4],

                        takeBio[5]);

                break;

            }

            if (!*isNIM*)

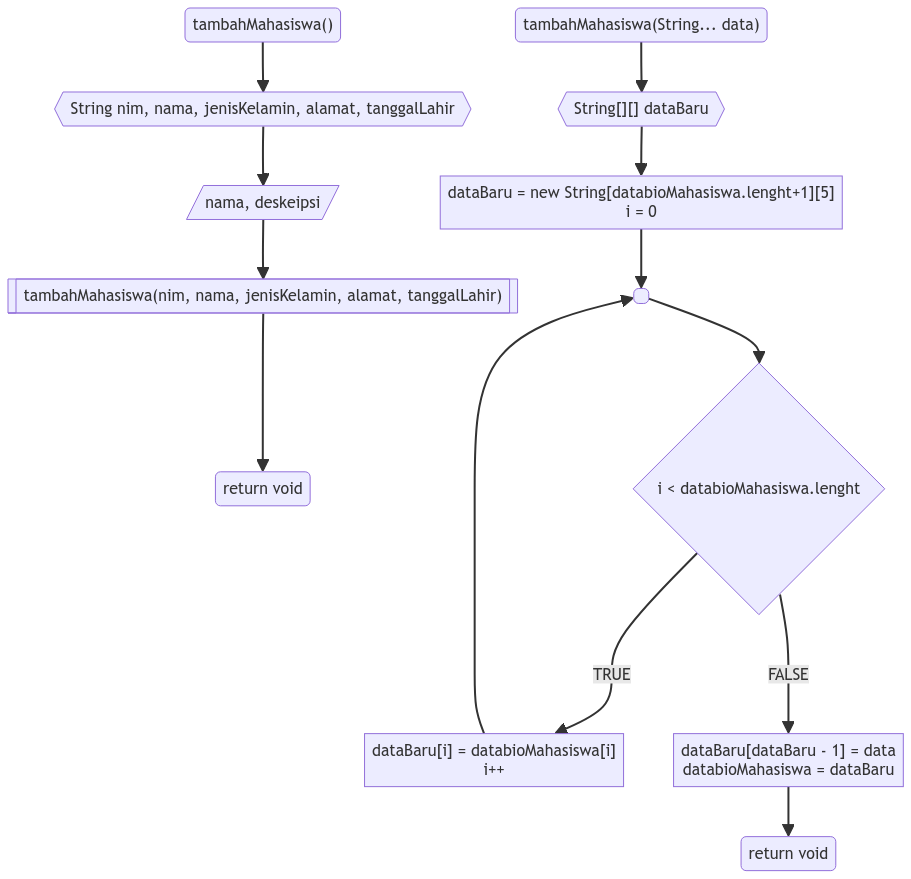
                System.out.printf(formatTable, (i + 1), takeBio[0], takeBio[1], takeBio[2], takeBio[3], takeBio[4],

                        takeBio[5]);

        }

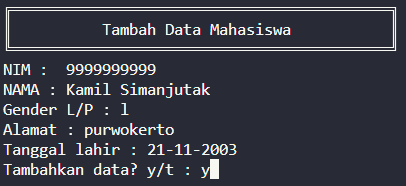
        System.out.println(lastLine);

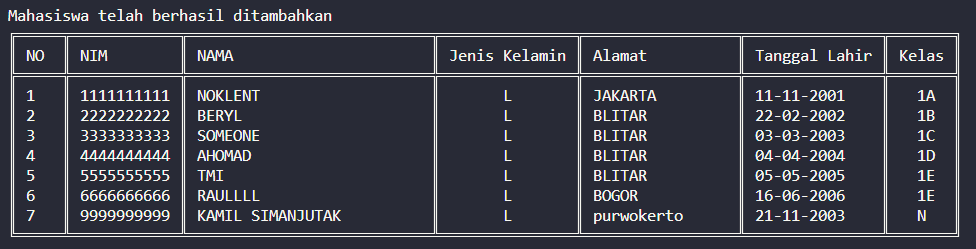
    }



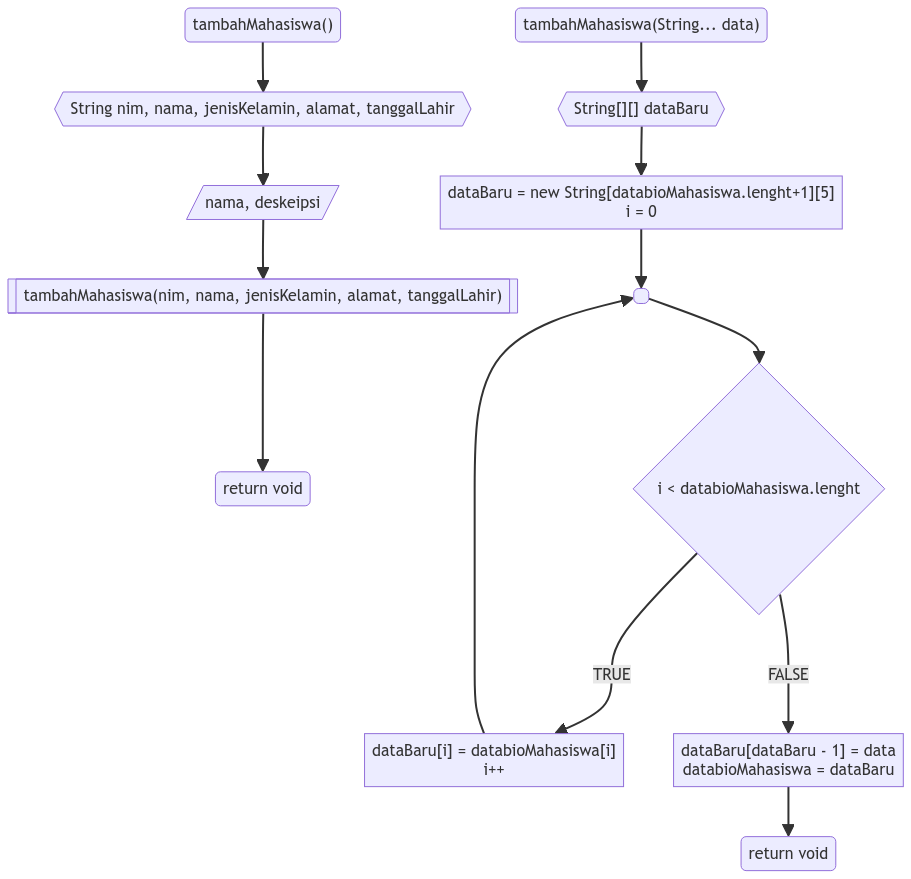
##### Daftarkan Mahasiswa

Setelah admin memilih daftarkan mahasiswa maka selanjutnya admin diminta untuk menginputkan biodata mahasiswa baru.





Jika semua pengecualian seperti persyaratan NIM harus berbeda dan semua pengecualian lainnya dapat dihindari maka data baru akan langsung muncul pada tabel.



Code untuk add mahasiswa:

static *void* addDataBioMahasiswa() {

        renderTitle("Tambah Data Mahasiswa");

        String nim = getInputStringNumberwithLimitChar("NIM", 10, 10, false);

        if (has(bioMahasiswa, nim, 0)) {

            System.out.println("NIM " + nim + " sudah terdaftar");

            return;

        }

        String nama = getInputStringWithLimit("NAMA", 1, 25, false);

        String jenisKelamin = getInputUniqueWord("Gender L/P", 1, 1, true, "l", "p");

        String alamat = getInputStringWithLimit("Alamat", 1, 15, false);

        String tanggalLahir = getInputStringWithLimit("Tanggal lahir", 10, 10, false);

        String userChoose = getInputUniqueWord("Tambahkan data? y/t", 1, 1, true, "y", "t");

        clearScreen();

        if (userChoose.equalsIgnoreCase("y"))

            addDataBioMahasiswa(nim, nama.toUpperCase(), jenisKelamin.toUpperCase(), alamat, tanggalLahir, "N");

        else

            System.out.println("Dibatalkan");

    }

    // add data bio mahasiswa

    static *void* addDataBioMahasiswa(String... *dataBio*) {

        String[][] mahasiswaBaru = **new** String[bioMahasiswa.length + 1][6];

        for (*int* i = 0; i < bioMahasiswa.length; i++)

            mahasiswaBaru[i] = bioMahasiswa[i];

        mahasiswaBaru[mahasiswaBaru.length - 1] = *dataBio*;

        bioMahasiswa = mahasiswaBaru;

        System.out.println("Mahasiswa telah berhasil ditambahkan");

        // add user mahasiswa

        String[][] userBaru = **new** String[userMahasiswa.length + 1][2];

        for (*int* i = 0; i < userMahasiswa.length; i++)

            userBaru[i] = userMahasiswa[i];

        userBaru[userBaru.length - 1] = **new** String[] { *dataBio*[0], *dataBio*[0] };

        userMahasiswa = userBaru;

        // instansisasi nilai,transkip,presensi mahasiswa

        addDataNilaiMahasiswa(*dataBio*[0]);

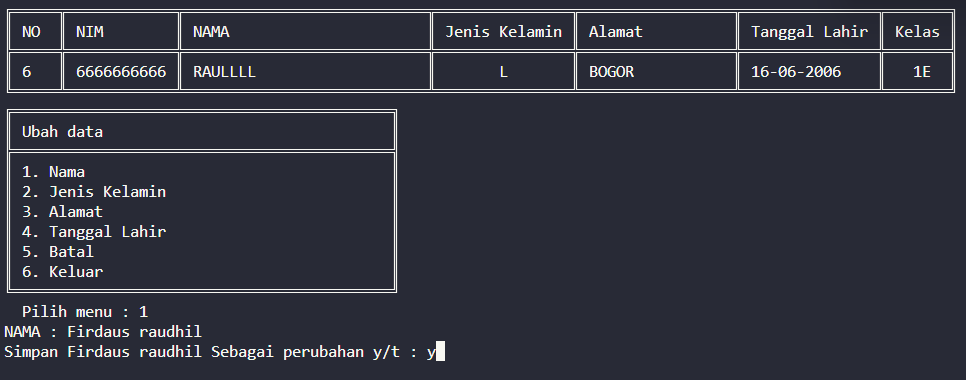
    }

##### Edit Data Mahasiswa

Jika ada laporan tenntang kesalahan dalam salah satu atau beberapa biodata mahasiswa yang tidak sesuai, maka admin harus ke fitur editBiodataMahasiswa. Admin hanya perlu memasukan NIM mahasiswa yang sudah ada dan benar.

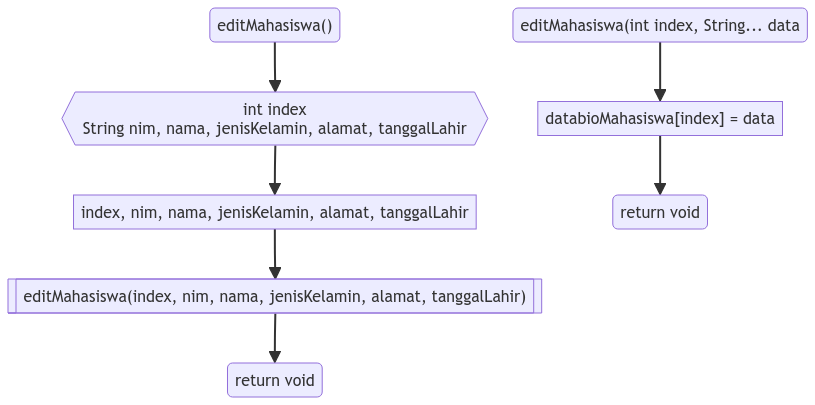


Kemudian pilih biodata apa yang ingin dirubah.



Seperti yang terlihat maka nama dari NIM 6666666666 sudah berhasil untuk diubah





Kode program

static *void* editDataBioMahasiswa() {

        renderTitle("Edit Data Mahasiswa");

        String oldNim, input = "";

*int* studentIndex = -1;

        while (true) {

            showDataBioMahasiswa(false);

            oldNim = getInputStringWithLimit("Masukan NIM yang ingin diubah", 10, 10, false);

            if (has(bioMahasiswa, oldNim, 0))

                break;

            clearScreen();

            System.out.println("NIM tidak ditemukan");

        }

        for (*int* i = 0; i < bioMahasiswa.length; i++) {

            if (bioMahasiswa[i][0].equals(oldNim)) {

                studentIndex = i;

                break;

            }

        }

        clearScreen();

        showDataBioMahasiswa(true, oldNim);

*int* userInput = pickMenu("Ubah data", **new** String[] {

                "Nama",

                "Jenis Kelamin",

                "Alamat",

                "Tanggal Lahir",

                "Batal"

        });

        switch (userInput) {

            case 1 *->* input = getInputStringWithLimit("NAMA", 1, 25, false);

            case 2 *->* input = getInputUniqueWord("Gender L/P", 1, 1, true, "l", "p");

            case 3 *->* input = getInputStringWithLimit("Alamat", 1, 15, false);

            case 4 *->* input = getInputStringWithLimit("Tanggal lahir(DD/MM/YYYY)", 10, 10, false);

            case 5 *->* {

                clearScreen();

                return;

            }

        }

        String userChoose = getInputUniqueWord("Simpan " + input + " Sebagai perubahan y/t", 1, 1, true, "y", "t");

        clearScreen();

        if (userChoose.equalsIgnoreCase("y")) {

            bioMahasiswa[studentIndex][userInput] = input;

            System.out.println("Berhasil mengedit");

        } else {

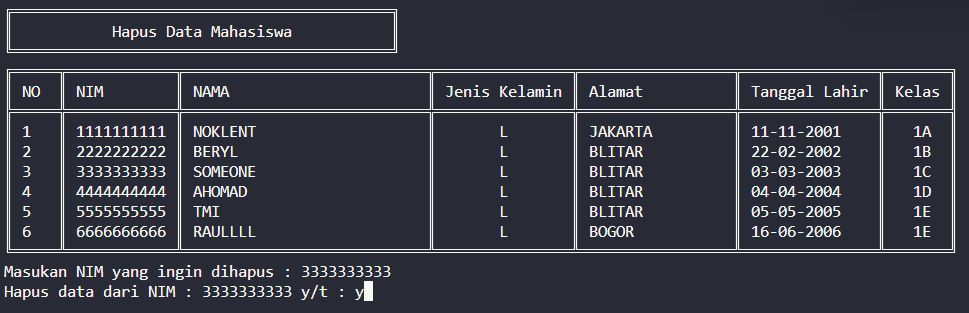
            System.out.println("Dibatalkan");

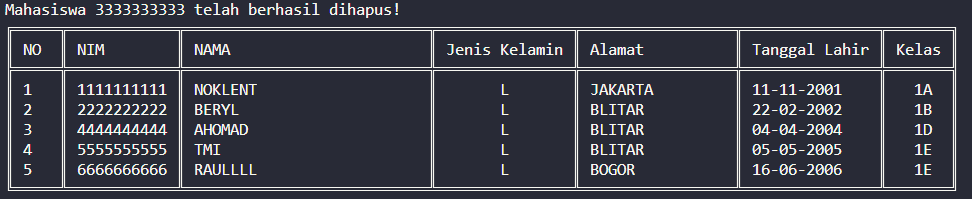
        }

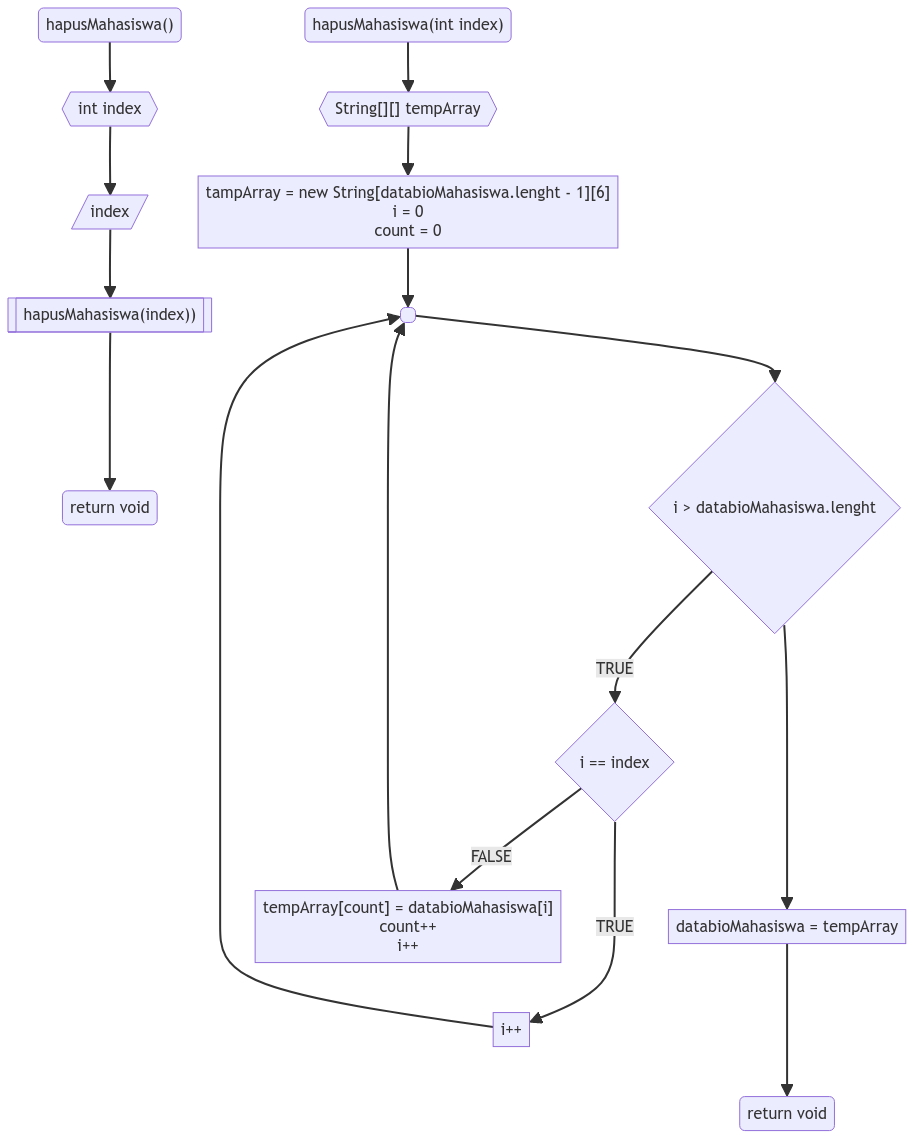
    }

##### Hapus Data Mahasiswa

Seperti namanya fitur ini dapat menghapus biodata mahasiswa







Code program

static *void* removeDataBioMahasiswa() {

        String nim;

        renderTitle("Hapus Data Mahasiswa");

        while (true) {

            showDataBioMahasiswa(false);

            nim = getInputStringWithLimit("Masukan NIM yang ingin dihapus", 10, 10, false);

            if (has(bioMahasiswa, nim, 0))

                break;

            clearScreen();

            System.out.println("Mahasiswa dengan NIM " + nim + " tidak ada!");

        }

        String userChoose = getInputUniqueWord("Hapus data dari NIM : " + nim + " y/t", 1, 1, true, "y", "t");

        clearScreen();

        if (userChoose.equalsIgnoreCase("n")) {

            System.out.println("Dibatalkan");

            return;

        }

        bioMahasiswa = removeDataBioMahasiswa(bioMahasiswa, nim);

        userMahasiswa = removeDataBioMahasiswa(userMahasiswa, nim);

        NilaiPancasila = removeDataBioMahasiswa(NilaiPancasila, nim);

        NilaiKTI = removeDataBioMahasiswa(NilaiKTI, nim);

        NilaiCTPS = removeDataBioMahasiswa(NilaiCTPS, nim);

        NilaiMATDAS = removeDataBioMahasiswa(NilaiMATDAS, nim);

        NilaiBING\_1 = removeDataBioMahasiswa(NilaiBING\_1, nim);

        NilaiDASPRO = removeDataBioMahasiswa(NilaiDASPRO, nim);

        NilaiPRAK\_DASPRO = removeDataBioMahasiswa(NilaiPRAK\_DASPRO, nim);

        NilaiK3 = removeDataBioMahasiswa(NilaiK3, nim);

        transkipNilai = removeDataBioMahasiswa(transkipNilai, nim);

        presensiMahasiswa = removeDataBioMahasiswa(presensiMahasiswa, nim);

        clearScreen();

        System.out.println("Mahasiswa " + nim + " telah berhasil dihapus!");

    }

    // remove semua data bio mahasiswa

    static String[][] removeDataBioMahasiswa(String[][] *array*, String *nim*) {

        String[][] tempArray = **new** String[*array*.length - 1][*array*[0].length];

*int* count = 0;

        for (String[] siswa : *array*) {

            if (siswa[0].equals(*nim*))

                continue;

            tempArray[count] = siswa;

            count++;

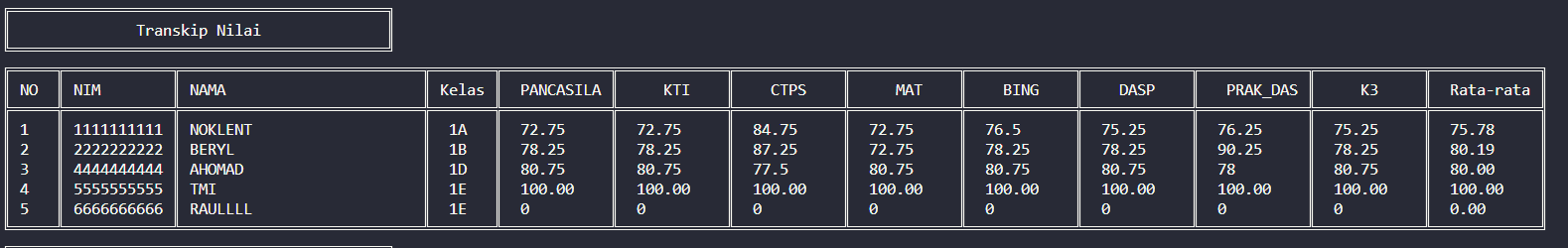
        }

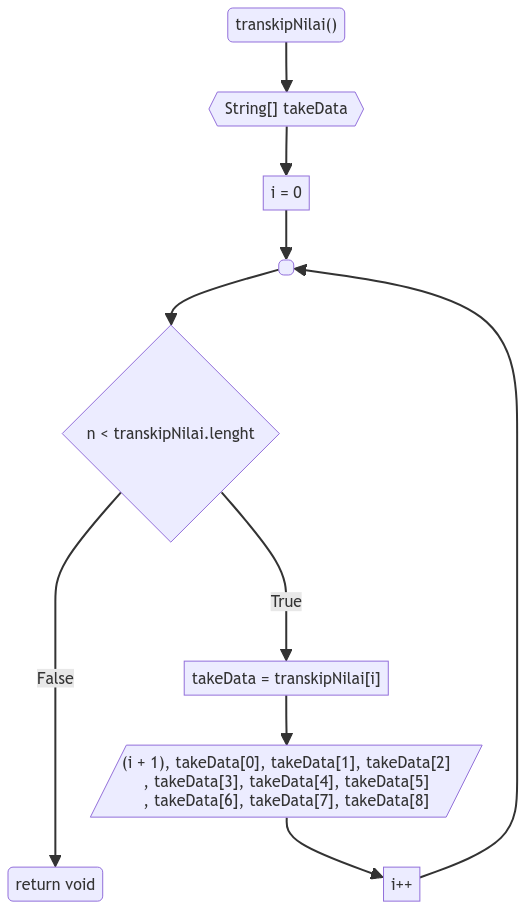
        return tempArray;

    }

#### 2.2.1.2 Transkip Nilai

Fitur kedua darai Modul mahasiswa adalah fitur transkipNilai. Menampilkan transkrip nilai 1 semester mahasiswa dalam format tabel pada layar konsol.





Kode program

 static *void* tampilkanTranskipNilai() {

        String firstLine = "╔═════╦════════════╦═══════════════════════════╦═══════" + "╦════════════".repeat(9) + "╗";

        String middleLine = "╠═════╬════════════╬═══════════════════════════╬═══════" + "╬════════════".repeat(9) + "╣";

        String lastLine = "╚═════╩════════════╩═══════════════════════════╩═══════" + "╩════════════".repeat(9) + "╝";

        String formatTable = "║ %-3s ║ %-10s ║ %-25s ║  %-3s  ║  %-9s ║  %-9s ║  %-9s ║  %-9s ║  %-9s ║  %-9s ║  %-9s ║  %-9s ║  %-9s ║%n";

        System.out.println(firstLine);

        System.out.format(

                "║ NO  ║ NIM        ║ NAMA                      ║ Kelas ║  PANCASILA ║     KTI    ║    CTPS    ║     MAT    ║    BING    ║    DASP    ║   PRAK\_DAS ║     K3     ║  Rata-rata ║%n");

        System.out.println(middleLine);

        for (*int* i = 0; i < transkipNilai.length; i++) {

            String[] takeTranskip = transkipNilai[i];

            System.out.printf(formatTable, (i + 1), takeTranskip[0], bioMahasiswa[i][1], bioMahasiswa[i][5],

                    takeTranskip[1], takeTranskip[2], takeTranskip[3], takeTranskip[4], takeTranskip[5],

                    takeTranskip[6], takeTranskip[7], takeTranskip[8], rataRataNilai(i));

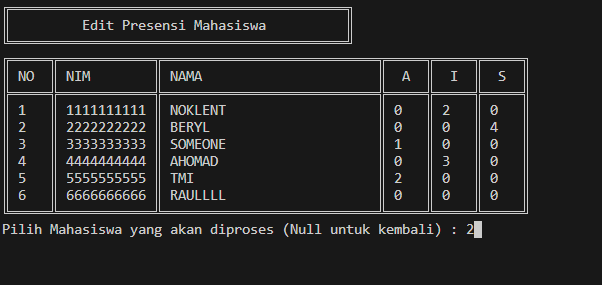
        }

        System.out.println(lastLine);

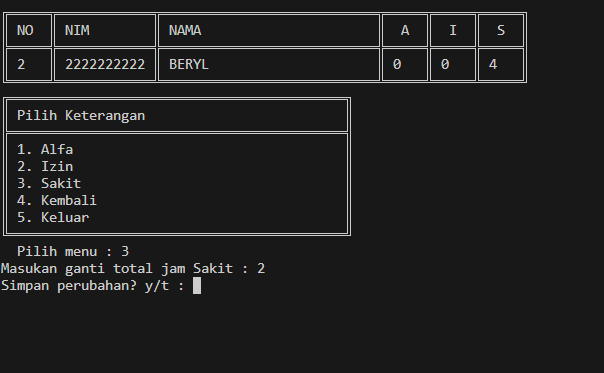
    }

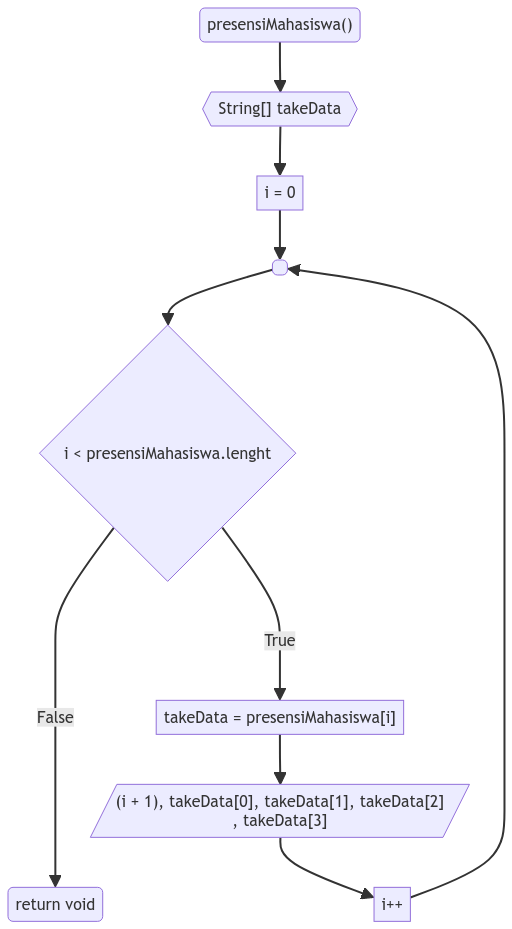
#### 2.2.1.3 Presensi Mahasiswa

Ketika terdapat kesalahan dosen dalam menginputkan absensi mahasiswa, maka admin dapat mengganti absensi tersebut dengan fitur presensi mahasiwa.



Kemudian admin memilih mahasiswa yang presensinya salah. Lalu memilih keterangan yang ingin dirubah.





Kode program

static *void* tampilkanPresensiMahasiswa(*boolean* *isSort*, String... *sort*) {

        String firstLine = "╔═════╦════════════╦═══════════════════════════╦═════╦═════╦═════╗";

        String middleLine = "╠═════╬════════════╬═══════════════════════════╬═════╬═════╬═════╣";

        String lastLine = "╚═════╩════════════╩═══════════════════════════╩═════╩═════╩═════╝";

        String formatTable = "║ %-3s ║ %-10s ║ %-25s ║ %-2s  ║ %-2s  ║ %-2s  ║%n";

        System.out.println(firstLine);

        System.out.format("║ NO  ║ NIM        ║ NAMA                      ║  A  ║  I  ║  S  ║%n");

        System.out.println(middleLine);

        for (*int* i = 0; i < presensiMahasiswa.length; i++) {

            String[] takePresensi = presensiMahasiswa[i];

            if (*isSort* && *sort*[0].equals(i + 1 + "") || *isSort* && *sort*[0].equals(takePresensi[0])) {

                System.out.printf(formatTable, (i + 1), takePresensi[0], bioMahasiswa[i][1], takePresensi[1],

                        takePresensi[2], takePresensi[3]);

                break;

            }

            if (!*isSort*)

                System.out.printf(formatTable, (i + 1), takePresensi[0], bioMahasiswa[i][1], takePresensi[1],

                        takePresensi[2], takePresensi[3]);

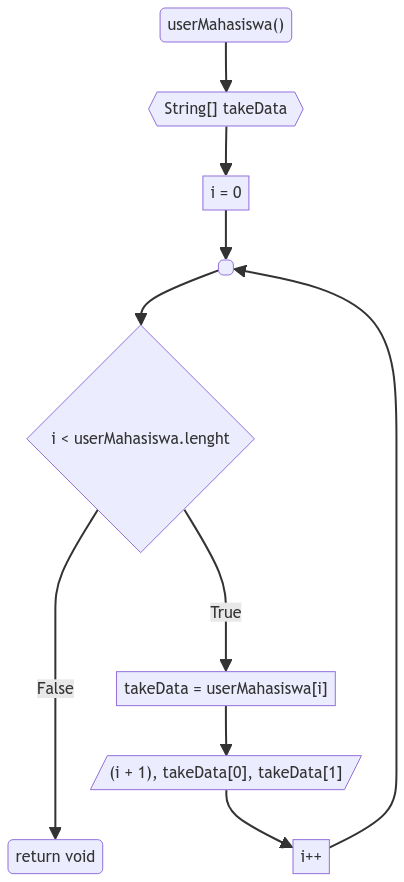
        }

        System.out.println(lastLine);

    }

#### 2.2.1.4 Check Username dan Password Mahasiswa

Fitur ini memungkinkan admin untuk mengecek username dan password yang digunakan oleh mahasiswa.

****

Untuk menampilkan user mahasiswa dalam format tabel pada layar konsol.

Kode program

**static *void* tampilkanDataUser() {**

**String firstLine = "╔═════╦════════════╦═══════════════════════════╦════════════╦═════════════════╗";**

**String middleLine = "╠═════╬════════════╬═══════════════════════════╬════════════╬═════════════════╣";**

**String lastLine = "╚═════╩════════════╩═══════════════════════════╩════════════╩═════════════════╝";**

**String formatTable = "║ %-3s ║ %-10s ║ %-25s ║ %-10s ║ %-15s ║%n";**

**System.out.println(firstLine);**

**System.out.format(**

**"║ NO  ║ NIM        ║ NAMA                      ║ USER       ║ PASSWORD        ║%n");**

**System.out.println(middleLine);**

**for (*int* i = 0; i < bioMahasiswa.length; i++) {**

**String[] takeBio = bioMahasiswa[i];**

**String[] takeUser = userMahasiswa[i];**

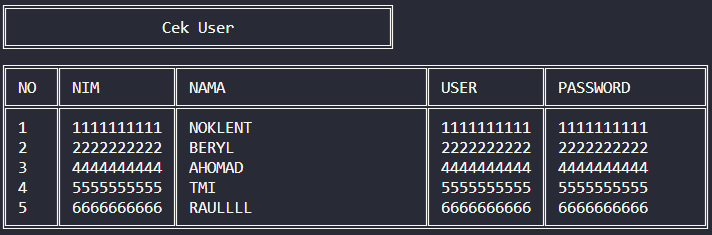
**System.out.printf(formatTable, (i + 1), takeBio[0], takeBio[1], takeUser[0], takeUser[1]);**

**}**

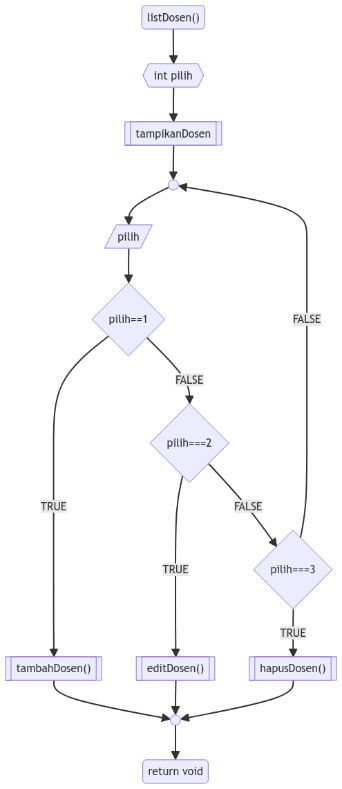
**System.out.println(lastLine);**

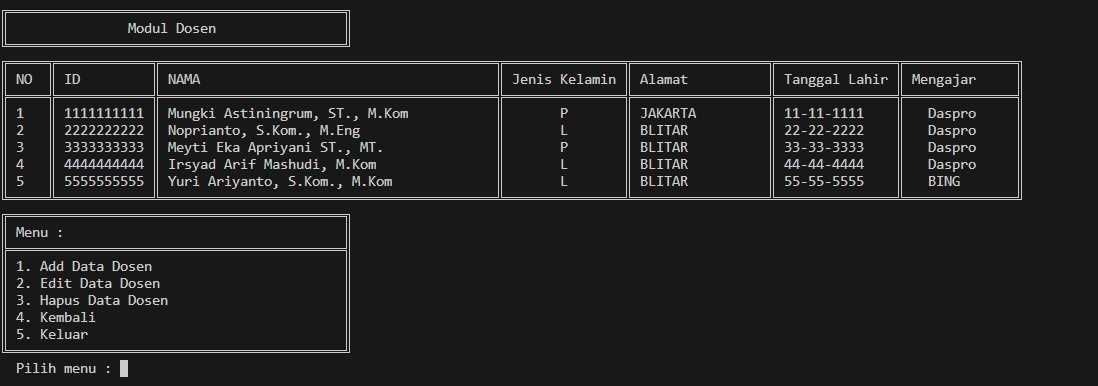
**}**

Output

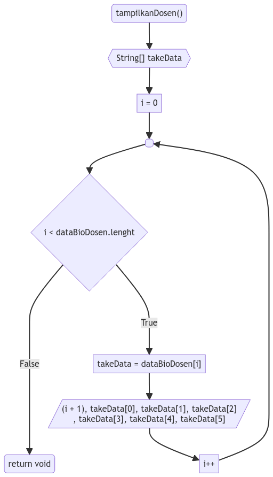


### 2.2.2 Modul Dosen





DI dalam modul Dosen kami mempunyai 3 fitur yaitu Add data Dosen, Edit Data dosen, Hapus data Dosen. Di dalam Modul dosen ini kami menggunakan fungsi showDataBioDosen untuk menampilkan biodata semua dosen yang ada pada array.



Code :  /\* <--- modulDosen ---> \*/

    static void modulDosen() {

        while (true) {

            renderTitle("Modul Dosen");

            showDataBioDosen(false);

            int userInput = pickMenu("Menu : ", new String[] {

                    "Add Data Dosen",

                    "Edit Data Dosen",

                    "Hapus Data Dosen",

                    "Kembali"

            });

            clearScreen();

            switch (userInput) {

                case 1 -> addDataDosen();

                case 2 -> editDataBioDosen();

                case 3 -> removeDataBioDosen();

                case 4 -> {

                    return;

                }

            }

        }

    }

 // Fungsi showDataBioDosen. Jika id ditemukan, maka akan menampilkan data

    static void showDataBioDosen(boolean isId, String... id) {

        String firstLine = "╔═════╦════════════╦══════════════════════════════════════════╦═══════════════╦═════════════════╦═══════════════╦══════════════╗";

        String middleLine = "╠═════╬════════════╬══════════════════════════════════════════╬═══════════════╬═════════════════╬═══════════════╬══════════════╣";

        String lastLine = "╚═════╩════════════╩══════════════════════════════════════════╩═══════════════╩═════════════════╩═══════════════╩══════════════╝";

        String formatTable = "║ %-3s ║ %-10s ║ %-40s ║       %-7s ║ %-15s ║ %-13s ║   %-10s ║%n";

        System.out.println(firstLine);

        System.out.format(

                "║ NO  ║ ID         ║ NAMA                                     ║ Jenis Kelamin ║ Alamat          ║ Tanggal Lahir ║ Mengajar     ║%n");

        System.out.println(middleLine);

        for (int i = 0; i < bioDosen.length; i++) {

            String[] takeBio = bioDosen[i];

            if (isId && id[0].equals(takeBio[0])) {

                System.out.printf(formatTable, (i + 1), takeBio[0], takeBio[1], takeBio[2], takeBio[3], takeBio[4],

                        takeBio[5]);

                break;

            }

            if (!isId)

                System.out.printf(formatTable, (i + 1), takeBio[0], takeBio[1], takeBio[2], takeBio[3], takeBio[4],

                        takeBio[5]);

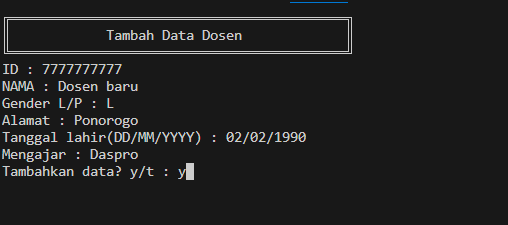
        }

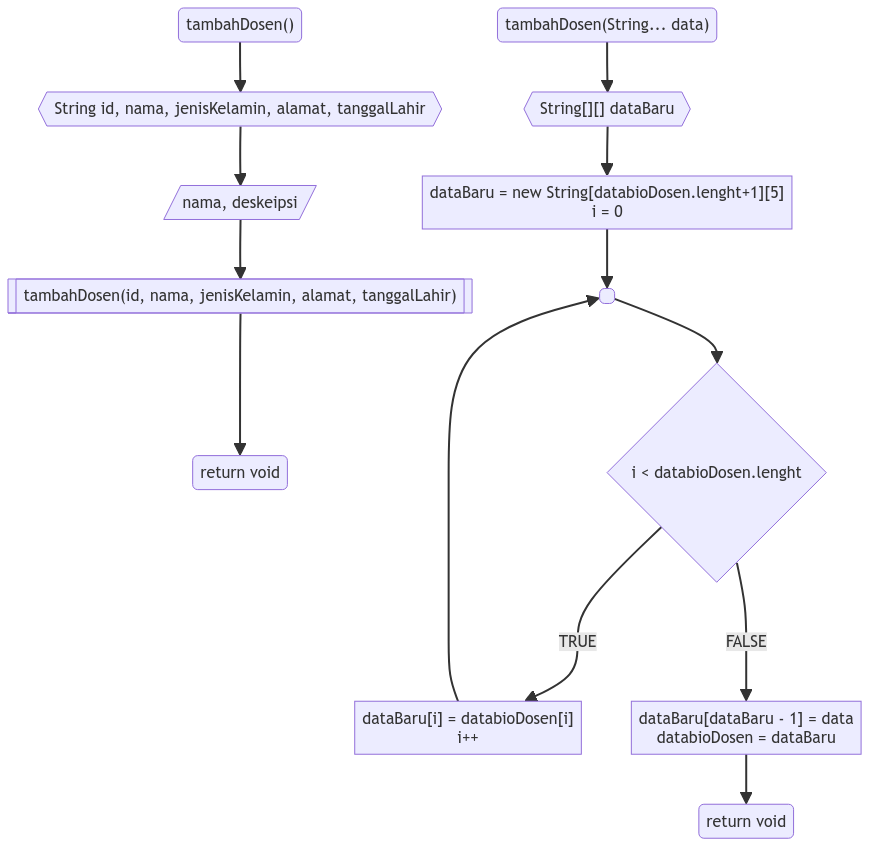
        System.out.println(lastLine);

    }

#### 2.2.2.1 Add Data Dosen

Fitur add data dosen akan berfungsi menambahkan data dosen baru ketika kita memasukan Id (Jika id susdah ada maka akan muncul pengecualian), Nama, Gender , dan lainnya





// Fungsi addDataDosen. Menambahkan data dosen

    static void addDataDosen() {

        renderTitle("Tambah Data Dosen");

        String id = getInputStringNumberwithLimitChar("ID", 10, 10, false);

        if (has(bioDosen, id, 0)) {

            System.out.println("ID " + id + " sudah terdaftar");

            return;

        }

        String nama = getInputStringWithLimit("NAMA", 1, 25, false);

        String jenisKelamin = getInputUniqueWord("Gender L/P", 1, 1, true, "l", "p");

        String alamat = getInputStringWithLimit("Alamat", 1, 15, false);

        String tanggalLahir = getInputStringWithLimit("Tanggal lahir(DD/MM/YYYY)", 10, 10, false);

        String mengajar = getInputStringWithLimit("Mengajar", 1, 10, false);

        String userChoose = getInputUniqueWord("Tambahkan data? y/t", 1, 1, true, "y", "t");

        if (userChoose.equalsIgnoreCase("y")) {

            clearScreen();

            addDataDosen(id, nama.toUpperCase(), jenisKelamin.toUpperCase(), alamat, tanggalLahir, mengajar);

            return;

        } else

            System.out.println("Dibatalkan");

    }

    // Fungsi addDataDosen. Menambahkan data dosen

    static void addDataDosen(String... dataBio) {

        String[][] dosenBaru = new String[bioDosen.length + 1][6];

        for (int i = 0; i < bioDosen.length; i++)

            dosenBaru[i] = bioDosen[i];

        dosenBaru[dosenBaru.length - 1] = dataBio;

        bioDosen = dosenBaru;

        String[][] userBaru = new String[userDosen.length + 1][2];

        for (int i = 0; i < userDosen.length; i++)

            userBaru[i] = userDosen[i];

        userBaru[userBaru.length - 1] = new String[] { dataBio[0], dataBio[0] };

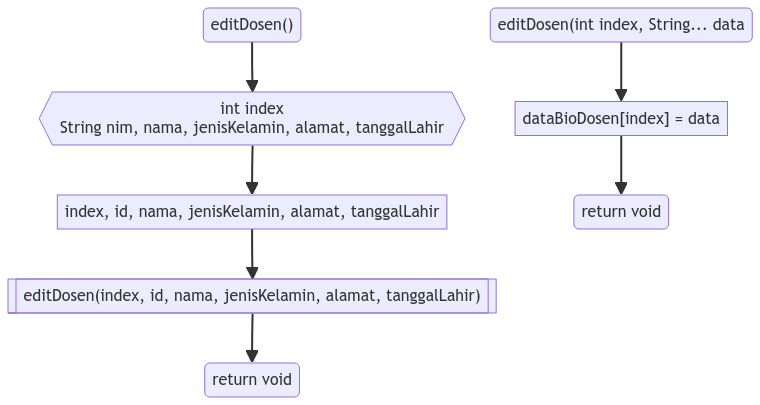
        userDosen = userBaru;

        System.out.println("Dosen telah berhasil ditambahkan");

    }

#### 2.2.2.2 Edit data Dosen

Pada kode ini akan mengizinkan admin untuk mengedit biodata yag dosen, fitur ini mirip dengan editbiomahasiswa dalam pengimplementasikan alurnya.



Kode Program

**static *void* editDataBioDosen() {**

**String ubah = "", id;**

***int* dosenIndex = -1;**

**renderTitle("Edit Data Dosen");**

**while (true) {**

**showDataBioDosen(false);**

**id = getInputStringWithLimit("Masukan ID yang ingin diubah", 10, 10, false);**

**if (has(bioDosen, id, 0))**

**break;**

**clearScreen();**

**System.out.println("ID tidak ditemukan");**

**}**

**for (*int* i = 0; i < bioDosen.length; i++) {**

**if (bioDosen[i][0].equals(id)) {**

**dosenIndex = i;**

**break;**

**}**

**}**

**clearScreen();**

**showDataBioDosen(true, id);**

***int* userInput = pickMenu("Ubah data", new String[] {**

**"Nama",**

**"Jenis Kelamin",**

**"Alamat",**

**"Tanggal Lahir",**

**"Mengajar",**

**"Batal"**

**});**

**switch (userInput) {**

**case 1 *->* ubah = getInputStringWithLimit("NAMA", 1, 25, false);**

**case 2 *->* ubah = getInputUniqueWord("Gender L/P", 1, 1, true, "l", "p");**

**case 3 *->* ubah = getInputStringWithLimit("Alamat", 1, 15, false);**

**case 4 *->* ubah = getInputStringWithLimit("Tanggal lahir(DD/MM/YYYY)", 10, 10, false);**

**case 5 *->* {**

**clearScreen();**

**return;**

**}**

**}**

**String userChoose = getInputUniqueWord("Simpan " + ubah + " Sebagai perubahan y/t", 1, 1, true, "y", "t");**

**clearScreen();**

**if (userChoose.equalsIgnoreCase("y")) {**

**bioDosen[dosenIndex][userInput] = ubah;**

**System.out.println("Berhasil mengedit");**

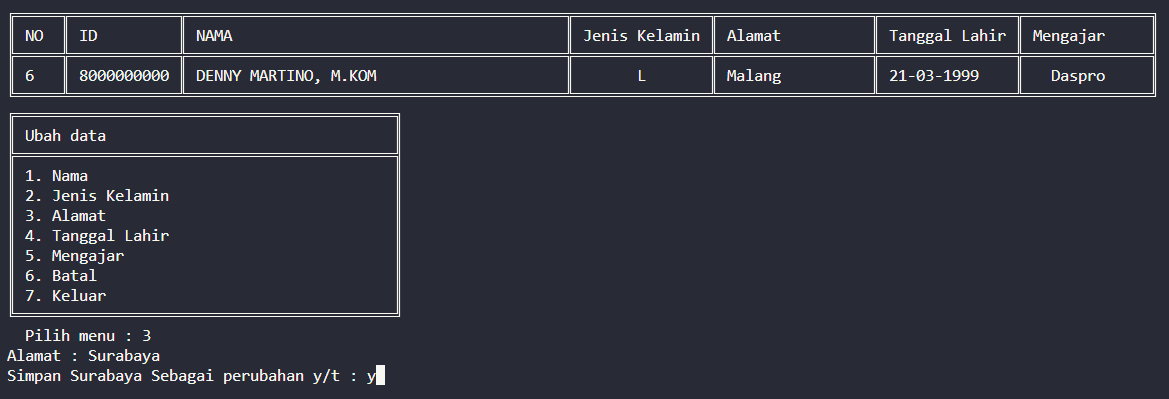
**} else {**

**System.out.println("Dibatalkan");**

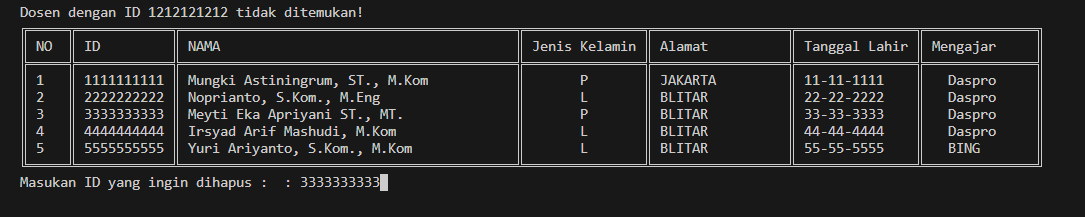
**}**

**}**

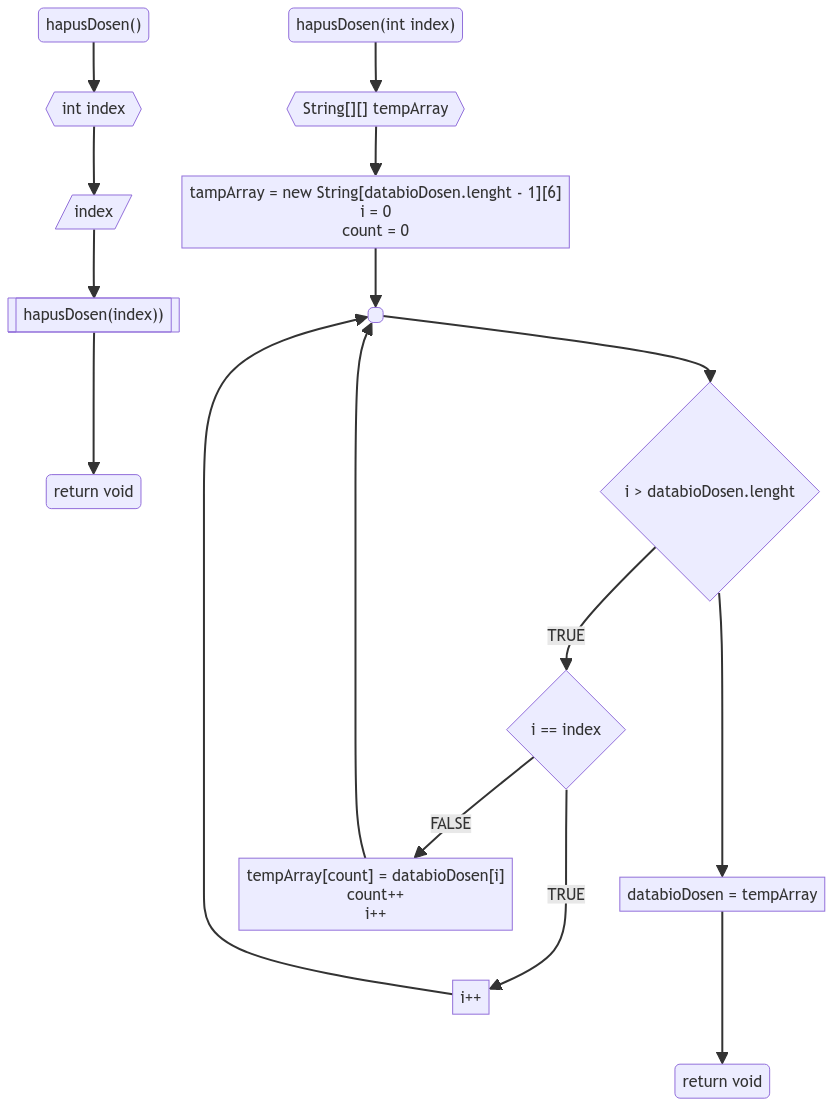
**Output**

****

#### 2.2.2.3 Hapus BioDosen



Fungsi ini bertujuan untuk menghapus data biodata dosen yang sesuai dengan ID yang dimasukan oleh admin.



**Kode Program**

static *void* removeDataBioDosen() {

        String id;

        renderTitle("Hapus Data Dosen");

        while (true) {

            showDataBioDosen(false);

            id = getInputStringWithLimit("Masukan ID yang ingin dihapus : ", 10, 10, false);

            if (has(bioDosen, id, 0))

                break;

            clearScreen();

            System.out.println("Dosen dengan ID " + id + " tidak ditemukan!");

        }

        bioDosen = removeDataBioDosen(bioDosen, id);

        userDosen = removeDataBioDosen(userDosen, id);

        clearScreen();

        System.out.println("Dosen " + id + " telah berhasil dihapus!");

    }

    // fungsi hapus data dosen

    static String[][] removeDataBioDosen(String[][] *array*, String *id*) {

        String[][] tempArray = **new** String[*array*.length - 1][*array*[0].length];

*int* count = 0;

        for (String[] dosen : *array*) {

            if (dosen[0].equals(*id*))

                continue;

            tempArray[count] = dosen;

            count++;

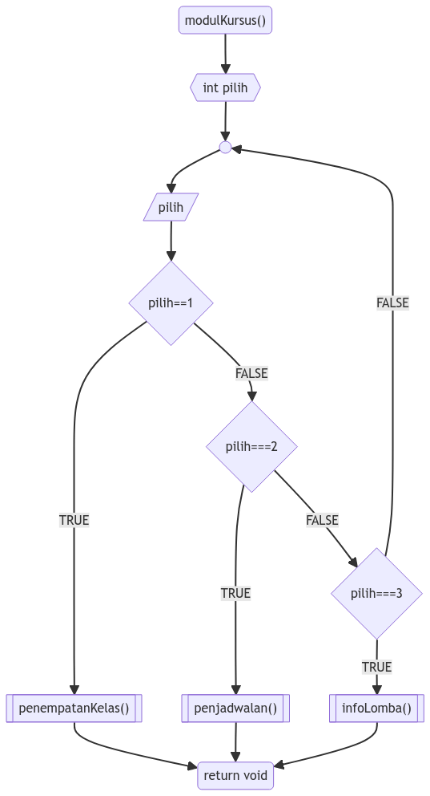
        }

        return tempArray;

    }

### 2.2.3 Modul Kursus

Pada modul kali ini berisikan hal hal yang dapat menunjang pembelajaran dan menyajikan info info tentang perlombaan yang diselengarakan di luar maupun di dalam kampus.



kode program

static *void* modulKursus() {

        while (true) {

            renderTitle("Modul Kursus");

*int* userInput = pickMenu("Menu : ", **new** String[] {

                    "Penempatan Kelas Mahasiswa",

                    "Penjadwalan",

                    "Info Lomba",

                    "Kembali"

            });

            clearScreen();

            switch (userInput) {

                case 1 *->* penempatanKelasMahasiswa();

                case 2 *->* penjadwalan();

                case 3 *->* infoLomba();

                case 4 *->* {

                    return;

                }

            }

        }

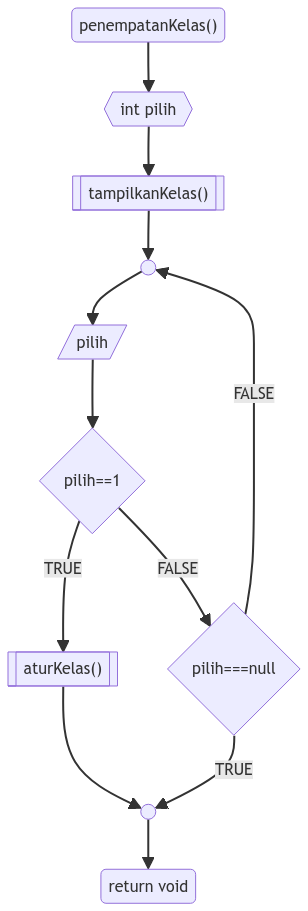
    }

output



#### 2.2.3.1 Fitur penempatan kelas Mahasiswa

Kami membuat fitur ini untuk mengelola tentang penempatan kelas bagi mahasiswa. Fungsi ini bertujuan untuk mengelola penempatan kelas mahasiswa dengan memberikan beberapa opsi kepada pengguna melalui sebuah menu interaktif pada layar.



kode program

static *void* penempatanKelasMahasiswa() {

        final String kelas = "1A|1B|1C|1D|1E|N";

        while (true) {

            renderTitle("Mahasiswa Kelas 1A|1B|1C|1D|1E|");

            tampilkanMahasiswaBerdasarkanKelas(kelas);

*int* userInput = pickMenu("Menu : ", **new** String[] {

                    "Atur Kelas",

                    "Sortir",

                    "Belum Terdaftar",

                    "Kembali"

            });

            clearScreen();

            switch (userInput) {

                case 1 *->* aturKelasMahasiswa(kelas);

                case 2 *->* sortirBerdasarkanKelas();

                case 3 *->* belumTerdaftar();

                case 4 *->* {

                    return;

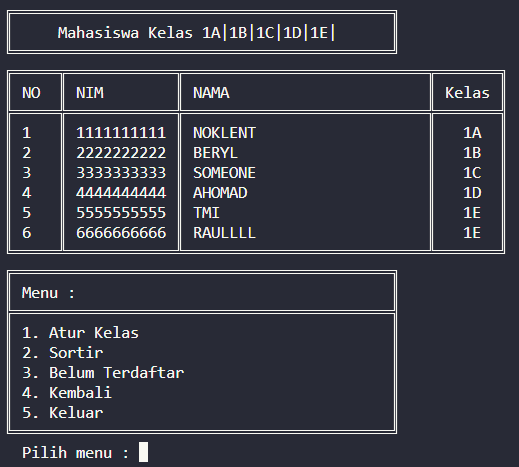
                }

            }

        }

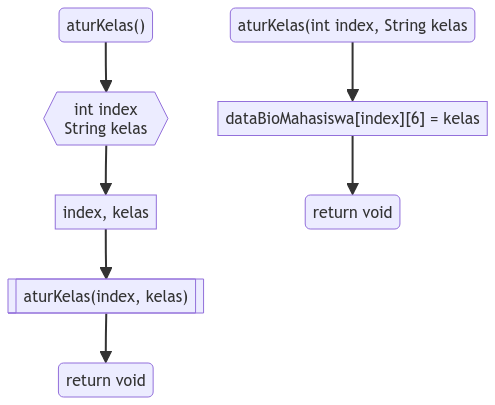
    }

output



##### Atur Kelas

Fitur ini berfungsi untuk admin dapat mengatur penempatan kelas mahasiswa.



kode program

static *void* aturKelasMahasiswa(String *kelas*) {

        String nim;

*int* studentIndex = -1;

        renderTitle("Atur Kelas Mahasiswa");

        while (true) {

            tampilkanMahasiswaBerdasarkanKelas(*kelas*);

            nim = getInputStringWithLimit("Masukan NIM", 10, 10, false);

            if (has(bioMahasiswa, nim, 0))

                break;

            clearScreen();

            System.out.println("NIM tidak ditemukan");

        }

        for (*int* i = 0; i < bioMahasiswa.length; i++) {

            if (bioMahasiswa[i][0].equals(nim)) {

                studentIndex = i;

                break;

            }

        }

        System.out.println("Kelas yang tersedia 1A|1B|1C|1D|1E|");

        String input = getInputUniqueWord("Atur kelas", 2, 2, true, "1A", "1B", "1C", "1D", "1E");

        String userChoose = getInputUniqueWord("Simpan perubahan y/t", 1, 1, true, "y", "t");

        clearScreen();

        if (userChoose.equalsIgnoreCase("y")) {

            bioMahasiswa[studentIndex][5] = input;

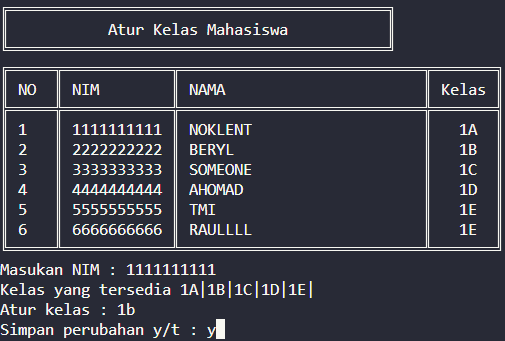
            System.out.println("Berhasil mengedit");

        } else

            System.out.println("Dibatalkan");

    }

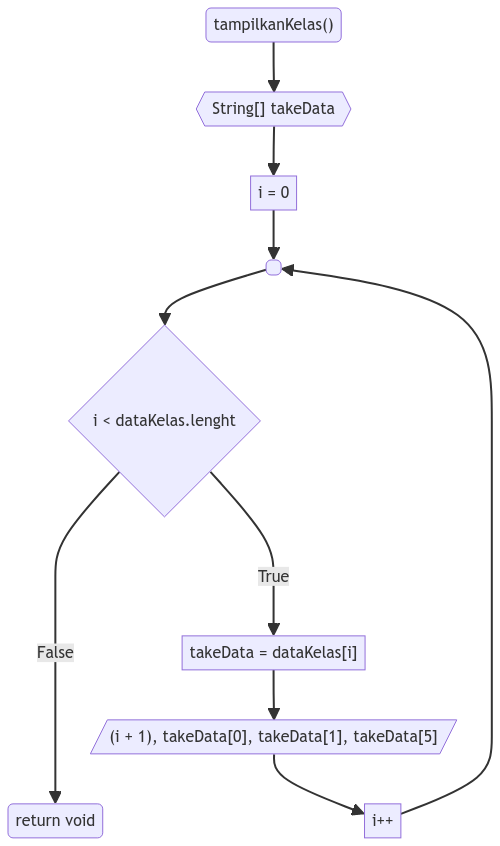
output



****

##### Sortir kelas

Seperti namanya, fitur ini dapat menyortir mahasiswa dengan kelas yang admin inginkan, dan menampilkannya dalam bentk tabel



Code program:

 static void sortirBerdasarkanKelas() {

        while (true) {

            System.out.println("Kelas yang tersedia 1A|1B|1C|1D|1E|");

            String kelas = getInputUniqueWord("Masukan Kelas", 2, 2, true, "1A", "1B", "1C", "1D", "1E");

            clearScreen();

            renderTitle("Mahasiswa Kelas " + kelas);

            tampilkanMahasiswaBerdasarkanKelas(kelas);

            int userInput = pickMenu("Menu : ", new String[] {

                    "Atur Kelas",

                    "Kembali"

            });

            clearScreen();

            switch (userInput) {

                case 1 -> aturKelasMahasiswa(kelas);

                case 2 -> {

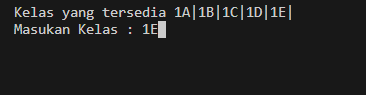
                    return;

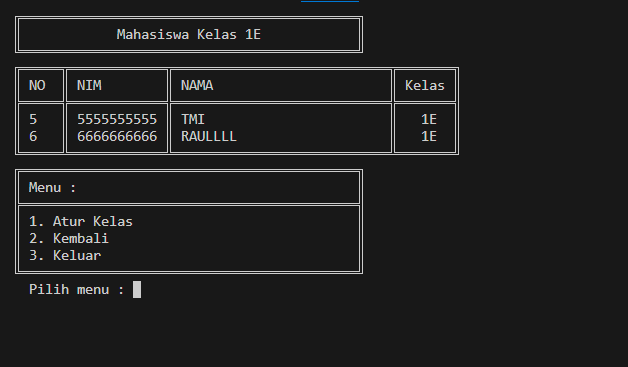
                }

            }

        }

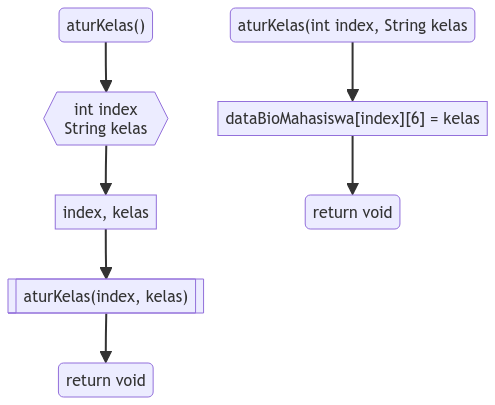
    }





##### Belum terdaftar

Fitur ini akan menempatkan mahasiswa yang baru saja diinputkan oleh admin, karena jika ada mahasiswa baru didaftarkan, kelasnya belum ditentukan.



Code

  static void belumTerdaftar() {

        while (true) {

            renderTitle("Mahasiswa Belum Terdaftar");

            tampilkanMahasiswaBerdasarkanKelas("N");

            int userInput = pickMenu("Menu : ", new String[] {

                    "Atur Kelas",

                    "Kembali"

            });

            clearScreen();

            switch (userInput) {

                case 1 -> aturKelasMahasiswa("N");

                case 2 -> {

                    return;

                }

            }

        }

    }

