**MAKALAH TUGAS BESAR**

**PEMROGRAMAN OBJEK 1**

**“Program Pendataan Mahasiswa dan Penginputan Nilai”**



**DISUSUN OLEH:**

SITI RAHMAH (3411181045)

KESYA FAUZIYAH ISDI (3411181057)

PUTRI SALSABILA RAHMADHANI (3411181182)

**JURUSAN INFORMATIKA**

**FAKULTAS SAINS DAN INFORMATIKA**

**UNIVERSITAS JENDERAL ACHMAD YANI**

**2019**

# **KATA PENGANTAR**

Puji syukur kami panjatkan kehadirat Tuhan Yang Maha Esa, karena kami dapat menyelesaikan tugas besar ini. Harapan kami semoga makalah tugas besar ini dapat membantu menambah pengetahuan dan pengalaman bagi yang membaca. Penyusunan makalah ini disusun untuk memenuhi tugas salah satu mata kuliah kami yaitu Pemrograman Objek Dasar tentang pengimplementasian basis data dengan pembuatan program Pendataan Mahasiswa dan Penginputan Nilai menggunakan bahasa Java. Selain itu tujuan dari penyusunan makalah ini juga untuk menambah wawasan tentang cara membuat program yang dapat dipakai disebuah instansi perkuliahan khususnya pada bagian Tata Usaha.

Kami menyadari bahwa makalah ini belum begitu sempurna, oleh karena itu dengan segala kerendahan hati, kami menerima kritik dan saran agar penyusunan makalah selanjutnya dapat menjadi lebih baik lagi. Kami mengucapkan banyak terima kasih dan semoga makalah ini bermanfaat bagi yang membaca.

Cimahi, 13 Desember 2019

Penulis

**DAFTAR ISI**

[**KATA PENGANTAR** i](#_Toc27385635)

[**BAB I** 1](#_Toc27385636)

[**PENDAHULUAN** 1](#_Toc27385637)

[**1.1** **Latar Belakang** 1](#_Toc27385638)

[**1.2** **Rumusan Masalah** 1](#_Toc27385639)

[**1.3** **Maksud dan Tujuan** 2](#_Toc27385640)

[**1.4** **Batasan Masalah** 2](#_Toc27385641)

[**BAB II** 3](#_Toc27385642)

[**LANDASAN TEORI** 3](#_Toc27385643)

[**2.1** **Sistem Informasi** 3](#_Toc27385644)

[**2.2**  **CRUD (*Create, Read, Update, Delete*)** 3](#_Toc27385645)

[**2.3** **OOP(*Object Oriented Programming*)** 3](#_Toc27385646)

[**2.4** **Bahasa Pemrograman JAVA** 5](#_Toc27385647)

[**BAB III** 6](#_Toc27385648)

[**PEMBAHASAN DAN RANCANGAN** 6](#_Toc27385649)

[**3.1** **Pembahasan** 6](#_Toc27385650)

[**3.2** **Perancangan** 6](#_Toc27385651)

[**BAB IV** 8](#_Toc27385652)

[**IMPLEMENTASI** 8](#_Toc27385653)

[**BAB V** 12](#_Toc27385654)

[**PENUTUP** 12](#_Toc27385655)

# **BAB I**

# **PENDAHULUAN**

* 1. **Latar Belakang**

Perkembangan teknologi informasi pada zaman ini sangat berpengaruh pada kemajuan suatu organisasi. Teknologi informasi memberikan sebuah kecepatan dan keakuratan dalam melakukan pengolahan data bila dibandingkan dengan cara manual. Dimana hal-hal seperti dokumen identitas mahasiswa pada saat ini masih diinput secara manual atau tulis tangan. Dengan cara manual tersebut terdapat beberapa kekurangan yaitu dapat terjadi kesalahan dalam penulisan dan pembacaan, tidak ringkas, lamanya proses pelayanan, dokumen yang rentan hilang atau rusak, dan tempat penyimpanan yang terbatas. Pembuatan program data mahasiswa pada saat ini diperlukan suatu teknologi informasi untuk mengolah data-data yang manual menjadi otomatisasi sehingga dapat mempermudah dalam mengelola data, data tidak akan hilang ataupun rusak karena data telah disimpan pada *database*. Tuntutan pelayanan informasi dan pengelolaan informasi secara terintegrasi menjadi sangat penting di setiap lembaga, termasuk di suatu universitas khususnya data per jurusan.

Dalam mengelola otomatisasi, pemanfaatan teknologi informasi pada data mahasiswa ini dapat memberikan suatu kecepatan dan ketepatan dalam mengolah data identitas mahasiswa. Selain itu juga dapat memberikan ruang yang tidak terbatas untuk penyimpanan data-data mahasiswa. Dengan menggunakan sistem informasi berbasis aplikasi akan dapat menyediakan layanan untuk meningkatkan performa dan kinerja dalam membuat data mahasiswa. Dengan sistem ini proses pendataan mahasiswa dapat dilakukan dengan cepat, dimana saja dan kapan saja oleh operator. Program yang dibuat ini merupakan program yang berbasis OOP yaitu suatu metode pemrograman yang berorientasi kepada objek. Tujuan dari OOP diciptakan adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari. Jadi setiap bagian dari suatu permasalahan adalah objek, nah objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil lagi.

* 1. **Rumusan Masalah**

Berdasarkan latar belakang masalah diatas, maka dapat dibuat perumusan masalah yaitu bagaimana cara membangun aplikasi pengolahan data mahasiswa.

* 1. **Maksud dan Tujuan**
     1. **Maksud**

Berdasarkan permasalahan yang diteliti, maka maksud dari penulisan tugas ini adalah membangun suatu aplikasi pengolahan Pendataan Mahasiswa.

* + 1. **Tujuan**

Adapun tujuan dari pembuatan aplikasi “Pendataan Mahasiswa” yaitu:

1. Memberikan kemudahan dalam mengolah data mahasiswa.
2. Memberikan efisiensi waktu dalam pendataan mahasiswa.
3. Kemudahan dalam pencarian data.
4. Penyimpanan lebih aman karena disimpan pada *database.*
   1. **Batasan Masalah**

Dalam penulisan proposal ini, ada beberapa batasan masalah yang digunakan yaitu:

1. Aplikasi ini merupakan aplikasi CRUD (*Create, Read, Update, Delete*) sederhana.
2. Pembuatan *database* dilakukan di localhost/phpmyadmin.
3. Aplikasi ini digunakan oleh bagian administrasi untuk mendata mahasiswa yang mengikuti suatu mata kuliah.
4. Pengolahan data mahasiswa bisa menambah, mengubah dan menghapus.

# **BAB II**

# **LANDASAN TEORI**

## **2.1 Sistem Informasi**

Sistem informasi adalah kombinasi dari teknologi dan aktivitas orang yang menggunakan teknologi dan aktivitas orang yang menggunakan teknologi itu untuk mendukung operasi dan manajemen . dalam arti yang sangat luas, istilah system informasi yang sering digunakan merujuk kepada interaksi orang, proses algoritmik,data dan teknolog. Sistem informasi adalah gabungan terorganisasi dari manusia, perangkat lunak, perangkat keras, jaringan komunikasi dan sumber data dalam mengumpulkan, mengubah dan menyebarkan informasi dan organisasi.

## **2.2 CRUD (*Create, Read, Update, Delete*)**

CRUD adalah singkatan dari Create Read Update Delete , yang sering digunakan pada aplikasi-aplikasi pengolahan data yang kebanyakan mengguanakan fungsi CRUD didalamnya . Fungsi ini digunakan untuk menambahkan data, menghapus data, serta mengupdate data.CRUD merupakan dasar-dasar operasi di dalam database dalam sebuah sistem, baik berbasis web maupun desktop.

## **2.3 OOP(*Object Oriented Programming*)**

OOP (Object Oriented Programming) adalah suatu metode pemrograman yang berorientasi kepada objek. Tujuan dari OOP diciptakan adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari.

Jadi setiap bagian dari suatu permasalahan adalah objek, nah objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil lagi. Saya ambil contoh Pesawat, Pesawat adalah sebuah objek. Pesawat itu sendiri terbentuk dari beberapa objek yang lebih kecil lagi seperti mesin, roda, baling-baling, kursi, dll. Pesawat sebagai objek yang terbentuk dari objek-objek yang lebih kecil saling berhubungan, berinteraksi, berkomunikasi dan saling mengirim pesan kepada objek-objek yang lainnya. Begitu juga dengan program, sebuah objek yang besar dibentuk dari beberapa objek yang lebih kecil, objek-objek itu saling berkomunikasi, dan saling berkirim pesan kepada objek yang lain.

**2.3.1 Konsep OOP**

* Abstrak Class
  1. Kelas merupakan deskripsi abstrak informasi dan tingkah laku dari sekumpulan data.
  2. Kelas dapat diilustrasikan sebagai suatu cetak biru(blueprint) atau prototipe yang digunakan untuk menciptakan objek.
  3. Kelas merupakan tipe data bagi objek yang mengenkapsulasi data dan operasi pada data dalam suatu unit tunggal.
  4. Kelas mendefinisikan suatu struktur yang terdiri atas data kelas (data field), prosedur atau fungsi (method), dan sifat kelas (property).
* Encapsulation
  1. Istilah enkapsulasi sebenarnya adalah kombinasi data dan fungsionalitas dalam sebuah unit tunggal sebagai bentuk untuk menyembunyikan detail informasi.
  2. Proses enkapsulasi memudahkan kita untuk menggunakan sebuah objek dari suatu kelas karena kita tidak perlu mengetahui segala hal secara rinci.
  3. Enkapsulasi menekankan pada antarmuka suatu kelas, atau dengan kata lain bagaimana menggunakan objek kelas tertentu.
  4. Contoh: kelas mobil menyediakan antarmuka fungsi untuk menjalankan mobil tersebut, tanpa kita perlu tahu komposisi bahan bakar, udara dan kalor yang diperlukan untuk proses tersebut.
* Inheritance
  1. Kita dapat mendefinisikan suatu kelas baru dengan mewarisi sifat dari kelas lain yang sudah ada.
  2. Penurunan sifat ini bisa dilakukan secara bertingkattingkat, sehingga semakin ke bawah kelas tersebut menjadi semakin spesifik.
  3. Sub kelas memungkinkan kita untuk melakukan spesifikasi detail dan perilaku khusus dari kelas supernya.
  4. Dengan konsep pewarisan, seorang programmer dapat menggunakan kode yang telah ditulisnya pada kelas super berulang kali pada kelas-kelas turunannya tanpa harus menulis ulang semua kodekode itu.
* Polymorphism
  1. Polimorfisme merupakan kemampuan objekobjek yang berbeda kelas namun terkait dalam pewarisan untuk merespon secara berbeda terhadap suatu pesan yang sama.
  2. Polimorfisme juga dapat dikatakan kemampuan sebuah objek untuk memutuskan method mana yang akan diterapkan padanya, tergantung letak objek tersebut pada jenjang pewarisan.
  3. Method overriding.
  4. Method name overloading.

## **2.4 Bahasa Pemrograman JAVA**

Java adalah bahasa pemrograman tingkat tinggi yang berorientasi objek dan program java tersusun dari bagian yang disebut kelas. Kelas terdiri atas metode-metode yang melakukan pekerjaan dan mengembalikan informasi setelah melakukan tugasnya. Para pemrogram  Java banyak mengambil keuntungan dari kumpulan kelas di pustaka kelas Java, yang disebut dengan Java Application Programming Interface (API). Kelas-kelas ini diorganisasikan menjadi sekelompok yang disebut paket (package). Java API telah menyediakan fungsionalitas yang memadai untuk menciptakan applet dan aplikasi canggih. Jadi ada dua hal yang harus dipelajari dalam Java, yaitu mempelajari bahasa Java dan bagaimana mempergunakan kelas pada Java API. Kelas merupakan satu-satunya cara menyatakan bagian eksekusi program, tidak ada cara lain. Pada Java program javac untuk mengkompilasi file kode sumber Java menjadi kelas-kelas bytecode. File kode sumber mempunyai ekstensi \*.java. Kompilator javac menghasilkan file bytecode kelas dengan ekstensi \*.class. Interpreter merupakan modul utama sistem Java yang digunakan aplikasi Java dan menjalankan program bytecode Java.

# **BAB III**

# **PEMBAHASAN DAN RANCANGAN**

## **3.1 Pembahasan**

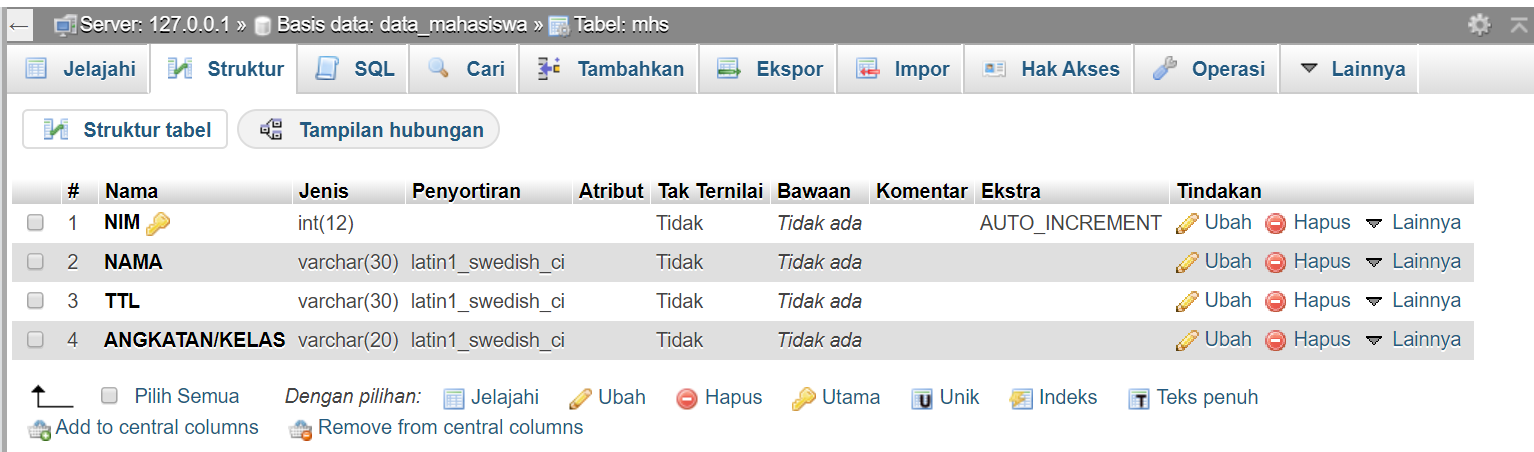
Aplikasi pendataan mahasiswa ini merupakan aplikasi GUI sederhana menggunakan bahasa pemrograman JAVA Netbeans. Dimana dalam program/aplikasi ini dapat menginput data mahasiswa berupa NIM, Nama, Tempat Tanggal Lahir dan Angkatan/ kelas apa. Data tersebut disimpan pada database yang dibuat di localhost/phpmyadmin. Aplikasi ini merupakan aplikasi CRUD (*Create, Read, Update, Delete*).

## **3.2 Perancangan**

Perancangan aplikasi “Pendataan Mahasiswa” menggunakan bahasa Java dan menyimpannya ke *database.* Program dibuat menggunakan NetBeans IDE dengan menggunakan konsep *Object Oriented Program.*

1. Langkah awal yang dilakukan yaitu menentukan entitas yang akan digunakan dan membuat *database* seperti:

* NIM (*primary key*)
* Nama
* TTL
* Angkatan/kelas



Gambar 1. Pembuatan Database

1. Perancangan antarmuka perangkat lunak dalam pembuatan program pengolahan data mahasiswa sangat penting karena merupakan tampilan visual dari perangkat lunak tersebut sehingga dapat berinteraksi dengan pengguna.
2. Membuat halaman login untuk pengguna, pada halaman tersebut terdapat text field untuk menginput username dan password. Juga terdapat tombol masuk untuk login ke halaman berikutnya, terdapat juga tombol close yang berguna untuk keluar dari halaman tersebut
3. Membuat halaman home yang isinya terdapat pilihan mata kuliah, agar pengguna dapat mendata siswa yang mengikuti matakuliah berdasarkan matakuliahnya sendiri.
4. Membuat halaman pendataan mahasiswa yang didalamnya terdapat:
5. Membuat tampilan form untuk menginput data mahasiswa yang akan mengikuti suatu mata kuliah. Admin dapat menginput data dengan mengisi *textfield* sesuai keterangan yang ada.
6. Membuat tabel yang berguna untuk menampilkan data yang telah di input oleh admin
7. Membuat tombol:

* *Add*

Tombol *add,* untuk menambah data.

* *Change*

Tombol *change,* untuk mengganti data yang telah di input.

* *Delete*

Tombol *delete,* untuk menghapus data yang telah ada.

* *Reset*

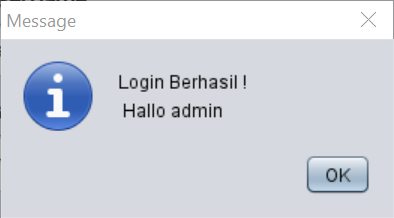
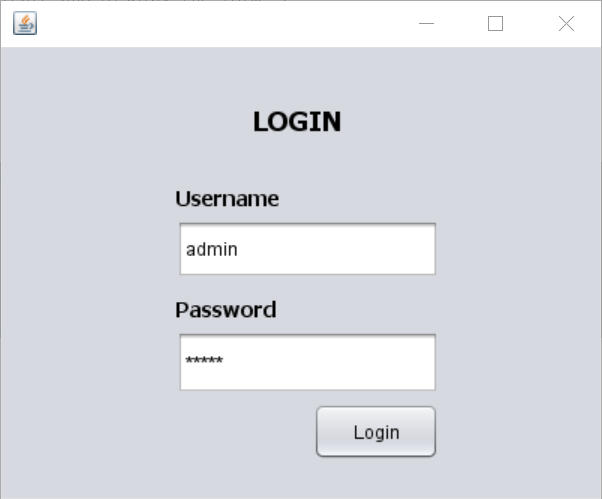
Tombol *reset,* untuk mereset data yang baru diinput ke dalam *textfield* (belum di *add*)

* *Search*

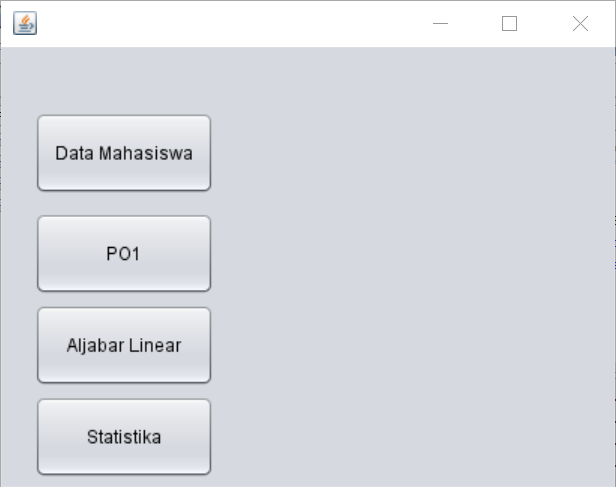
Tombol *search,* digunakan untuk mencari data yang telah ada. Admin dapat mengisi nama mahasiswa pada *textfield* yang disediakan, setelah itu tekan tombol *search* untuk mengetahui data-data mahasiswa yang dicari tersebut.

# **BAB IV**

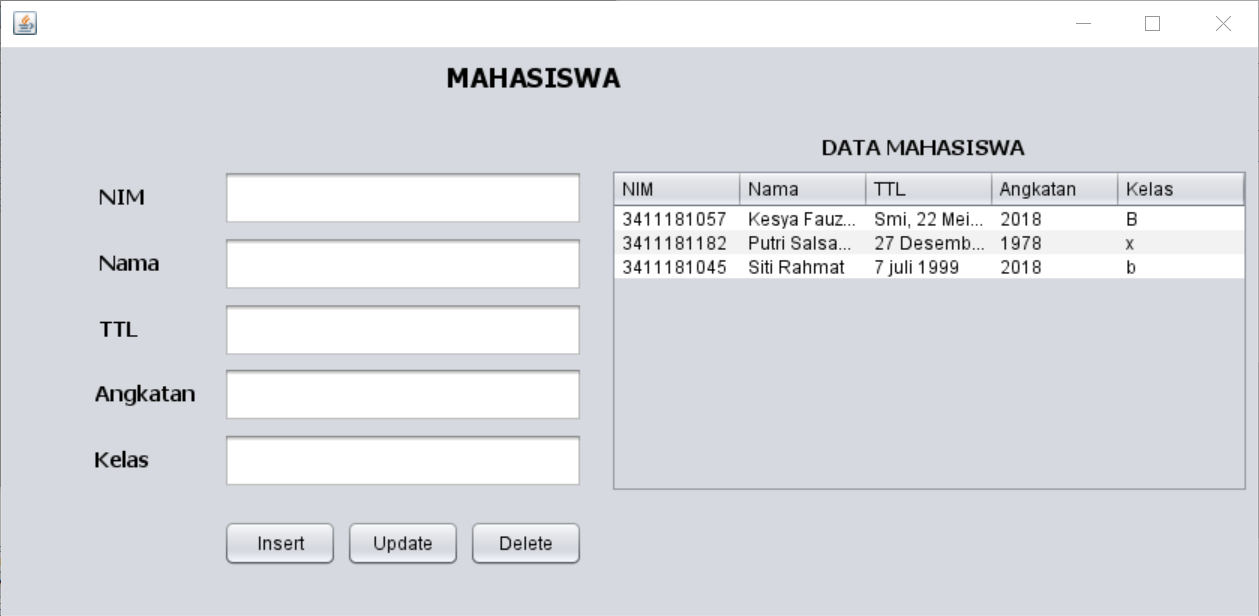
# **IMPLEMENTASI**

****

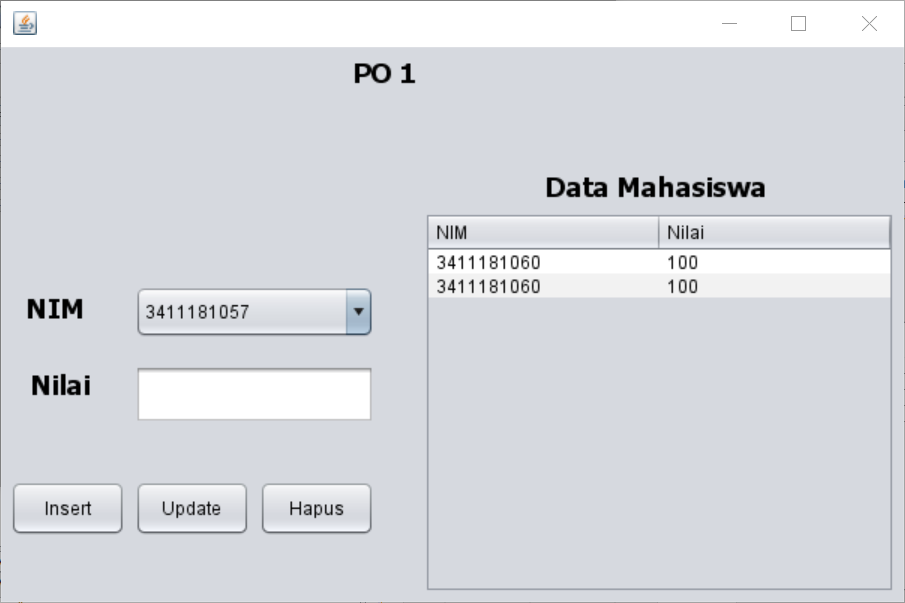
Halaman ini merupakan form utama program untuk masuk ke halaman setelahnya, username dan password telah ditentukan sehingga hanya admin yang bisa mengakses program ini.

****

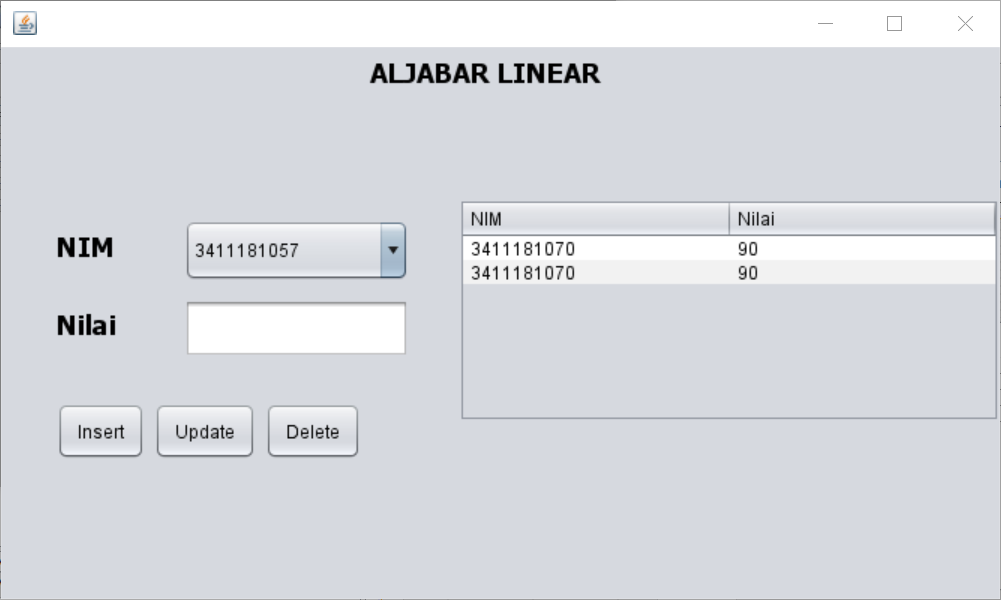
Pada halaman ini, pengguna bisa mengklik tombol “Data Mahasiswa” untuk masuk ke halaman penginputan data. “PO 1” “Aljabar Linear” dan tombol “Statistika” untuk masuk ke penginputan nilai matakuliah yang dipilih

****

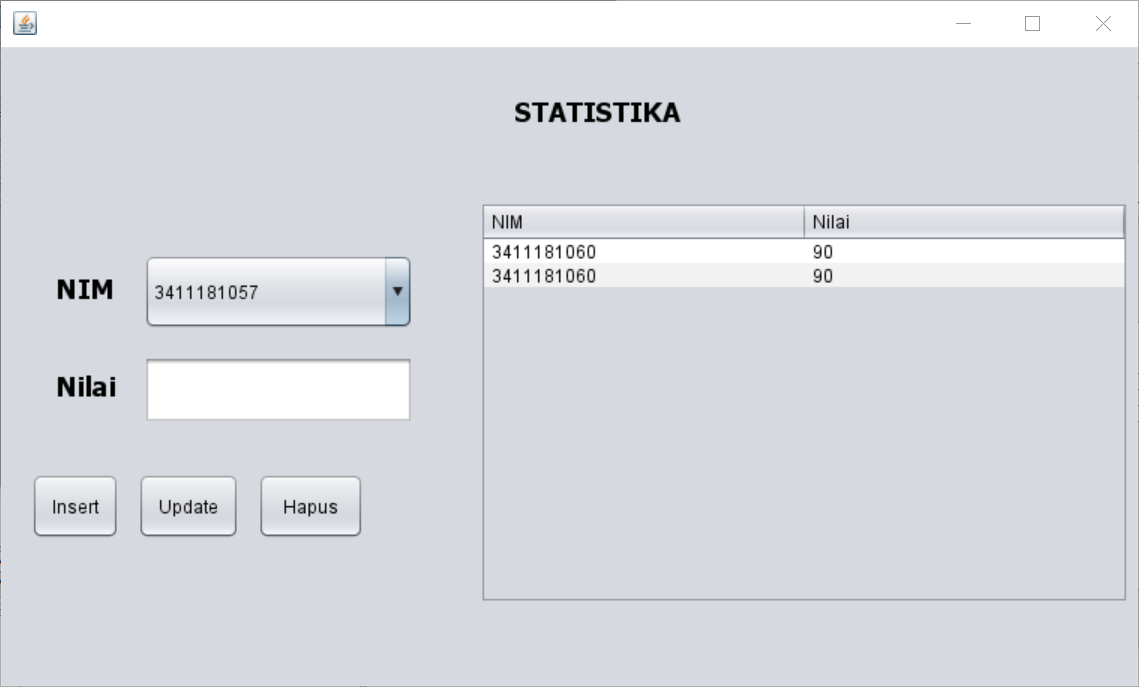
Halaman mahasiswa

****

Halaman PO 1

****

Halaman aljabar linear

****

Halaman statistika

# **BAB V**

# **PENUTUP**

**Kesimpulan :**

Aplikasi pendataan mahasiswa ini merupakan aplikasi GUI sederhana menggunakan bahasa pemrograman JAVA Netbeans. Dimana dalam program/aplikasi ini dapat menginput data mahasiswa berupa NIM, Nama, Tempat Tanggal Lahir dan Angkatan/ kelas apa. Data tersebut disimpan pada database yang dibuat di localhost/phpmyadmin. Aplikasi ini merupakan aplikasi CRUD (*Create, Read, Update, Delete*). Aplikasi ini dibuat untuk digunakan admin suatu Tata Usaha agar dapat mempermudah pengelolaan data.