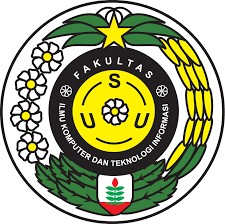
# LAPORAN TUGAS BESAR SISTEM BASIS DATA

**“E-Learning Universitas Sumatera Utara”**



Dosen Pengampu :

Ibu Sarah Purnamawati, ST., M.Sc.

Dikerjakan Oleh :

|  |  |
| --- | --- |
| Rizki Sahat | (201402020) |
| Christine Amanda | (201402035) |
| Muhammad Fauzan Fadhil Kurnia | (201402056) |
| Dzakiy Dzakwan | (201402068) |
| Amelia Angelita Silalahi | (201402128) |

# FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI UNIVERSITAS SUMATERA UTARA

**MEDAN 2021**

# KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa atas segala rahmatnya sehingga laporan yang berjudul ‘*E-Learning* Universitas Sumatera Utara’ ini dapat tersusun hingga selesai. Tidak lupa juga kami mengucapkan terima kasih kepada Ibu Sarah Purnamawati, ST., M.Sc. selaku dosen pengampu mata kuliah Sistem Basis Data yang telah membimbing dan memberi sumbangan pengetahuannya kepada kami sehingga dapat menyelesaikan laporan ini secara tepat waktu.

Kami menyadari sepenuhnya bahwa masih ada kekurangan dalam laporan dan tugas besar kami, baik dari segi penyusunan kalimat maupun tata bahasanya. Oleh karena itu, dengan tangan terbuka kami menerima segala bentuk saran maupun kritik dari pembaca agar kami dapat memperbaiki laporan ‘*E-Learning* Universitas Sumatera Utara’ ini.

Akhir kata, kami berharap semoga laporan yang telah kami buat dapat bermanfaat dalam meningkatkan pengetahuan dan pemahaman pembaca terkait penerapan Sistem Basis Data.

Medan, 06 Desember 2021

Kelompok 4

# BAB I PENDAHULUAN

Aplikasi web *E-Learning* merupakan sebuah aplikasi yang menghubungkan antara dosen dan mahasiswa. Dimana pada aplikasi ini dosen dapat melakukan kegiatan belajar mengajar melalui aplikasi web seperti membuat pertemuan, memberikan tugas, memberikan materi dan juga mebuat absensi pengguna/pasien dapat membuat janji temu di rumah sakit dengan dokter yang diinginkan . Mahasiswa juga dapat melakukan mengisi absensi, membaca materi yang diberikan dosen bahkan mengakses pertemuan dan tugas yang diberikan dosen.

* 1. Latar Belakang

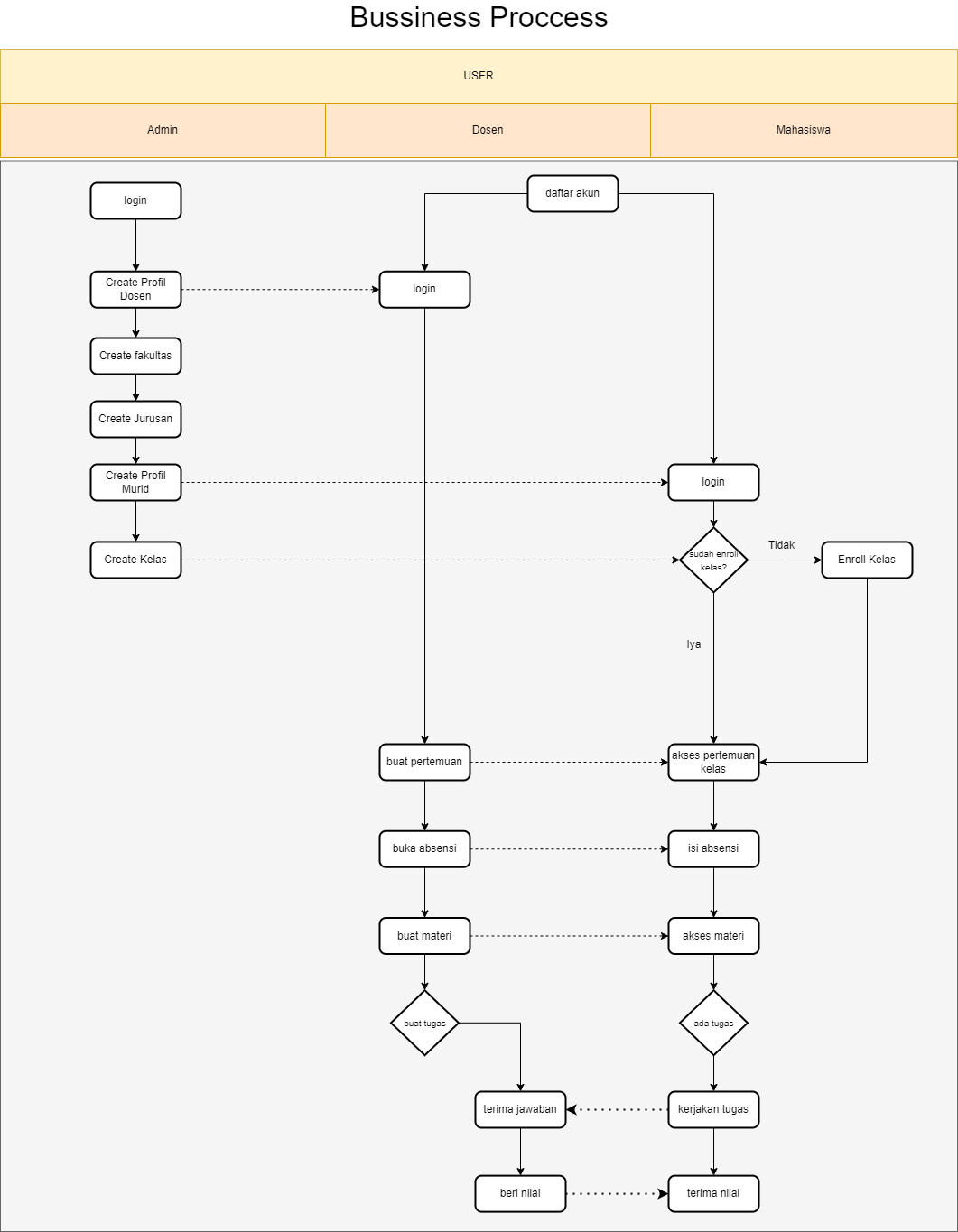
Pada era saat ini, teknologi semakin berkembang bahkan berkembang pesat akibat dari pandemi Covid-19 yang telah melanda seluruh dunia selama kurang lebih dua tahun ini. Pada masa pandemi ini, teknologi dijadikan media telekomunikasi utama akibat adanya pembatasan tatap muka. Oleh karena itu, tidak dapat dipungkiri bahwa semakin banyak masyarakat yang menggunakan *smartphone* sebagai media telekomunikasi dan hiburan.

Salah satu manfaat dari perkembangan teknologi saat ini adalah memudahkan masyarakat untuk memperoleh informasi yang cepat dan akurat, baik dari segi sosial, keuangan, bahkan kesehatan. Oleh karena itu, hadirlah teknologi-teknologi yang berperan penting dalam penyediaan informasi, misalnya *website. Website* atau *web* adalah halaman informasi yang dapat diakses dengan menggunakan internet, baik dari segi sosial, pengembangan diri, bahkan kesehatan.

Berdasarkan hal di atas, kami membuat Aplikasi *website E-Learning* yang diperuntukkan bagi mahasiswa dan dosen agar kegiatan belajar mengajar tetap berlangsung walaupun pandemi sedang berlangsung. Selain itu, Adapun alasan lain penulis dalam pembuatan *Website E-Learning* ini adalah untuk memenuhi tugas besar mata kuliah Sistem Basis Data.

* 1. Tujuan
     1. Memudahkan masyarakat untuk melakukan konsultasi dengan dokter, dimana melakukan pemesanan secara online.
     2. Memberikan informasi kesehatan secara cepat dan akurat kepada masyarakat.
     3. Memberikan gambaran manajemen Halodoc.
  2. Keluaran yang Diharapkan
     1. Memahami sistem basis data pada *Website E-Learning* .
     2. Memahami pengaplikasian sistem basis data dengan menggunakan *framework* Laravel dan MySQL
  3. Proses Bisnis

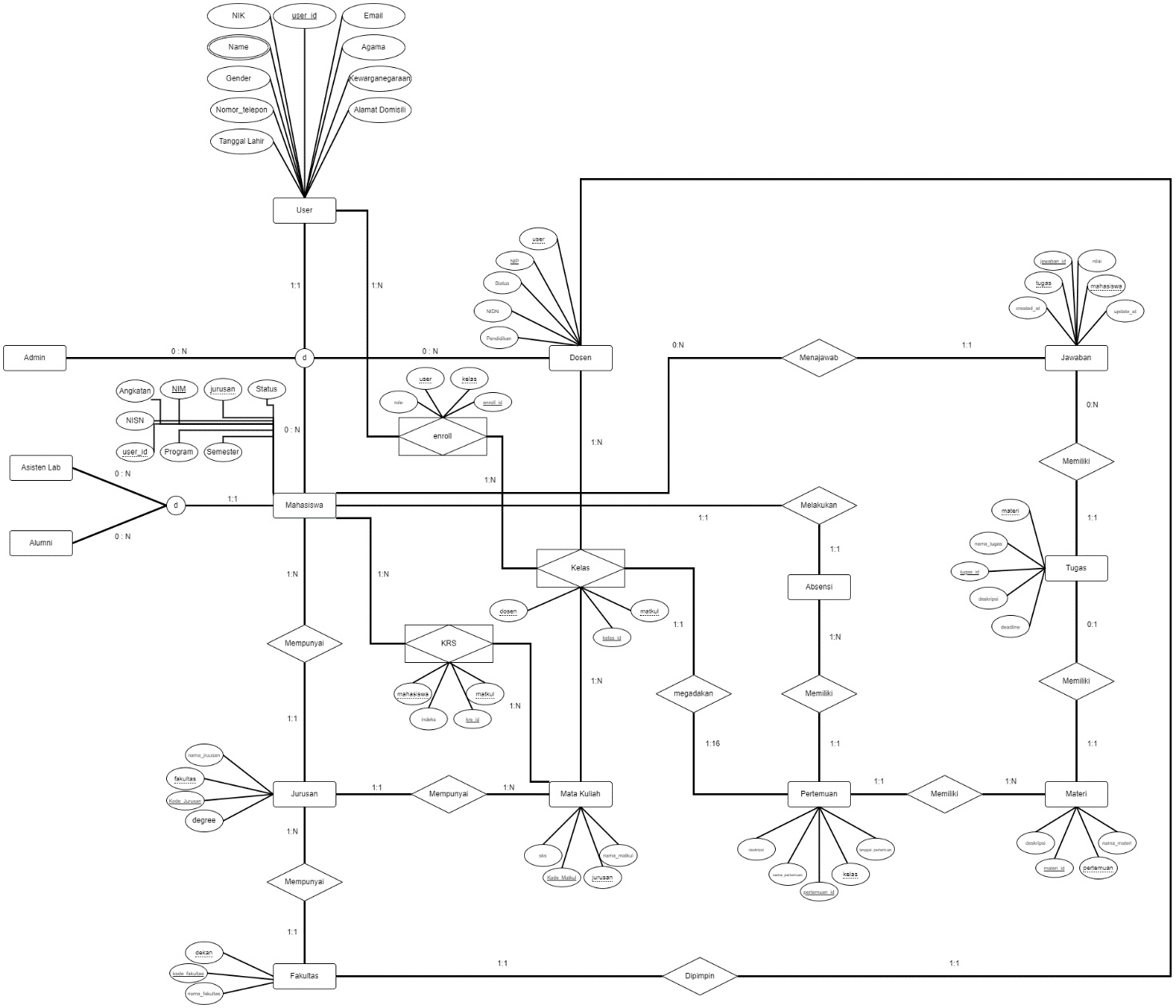
# BAB II PEMBAHASAN



Gambar 1. Proses Bisnis

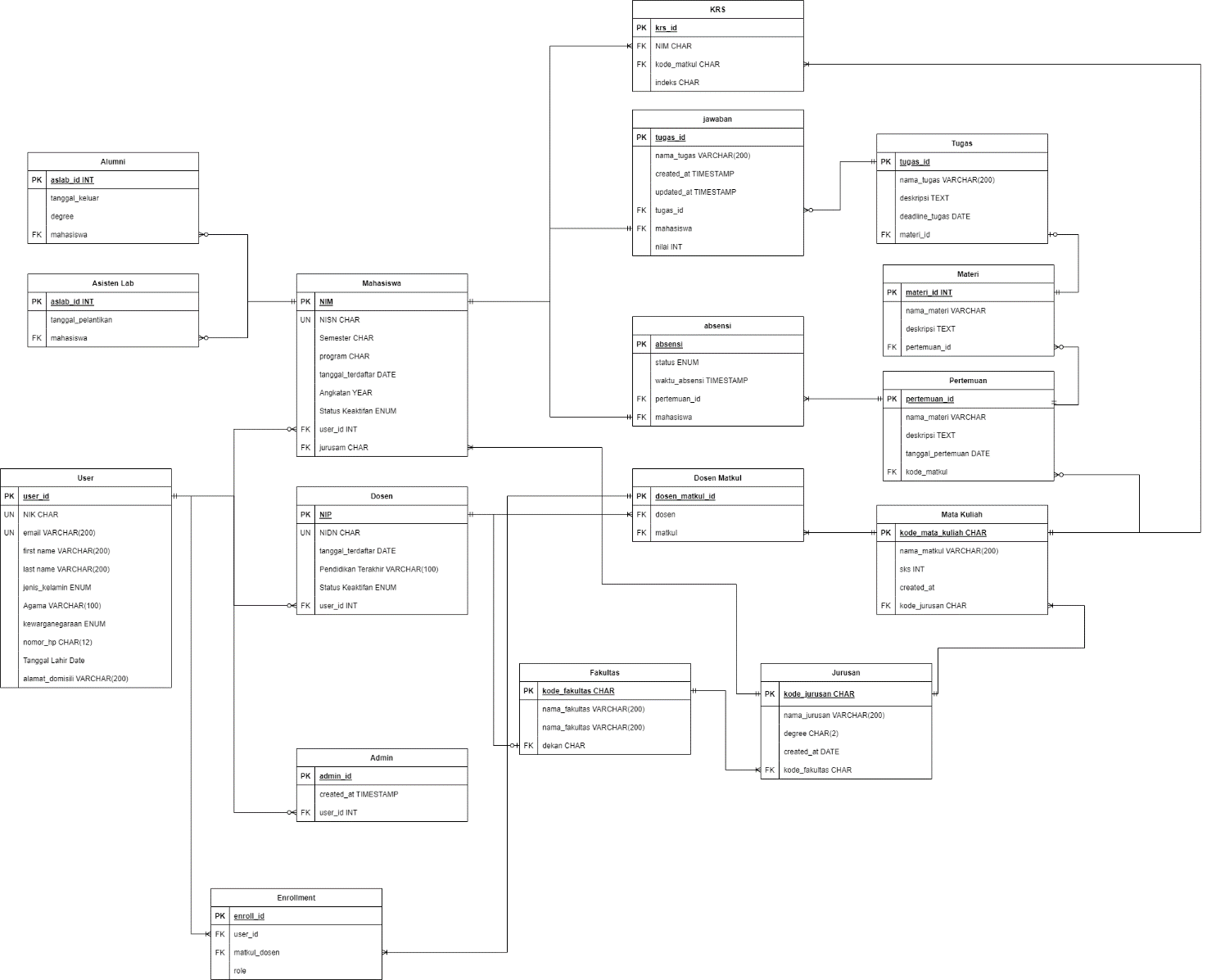
Seperti yang dapat kita lihat pada gambar diatas, gambar diatas merupakan proses bisnis dari Halodoc. Pertama-tama user diwajibkan untuk mengisi biodata diri terlebih dahulu, kemudian user dapat menunggu hingga admin membuat profil dari user. Lalu user yang berstatus sebagai dosen dapat membuat pertemuan, absensi, materi dan tugas yang akan disampaikan ke user yang berstatus sebagai mahasisa. User yang berstatus sebagai mahasiswa diharuskan mengambil kelas yang tersedia sebelum dapat mengakses kelas. Lalu user mahasiswa dapat mengakses setiap pertemuan yang terdapat dalam kelas. Begitu juga dengan materi, absensi dan tugas. Mahasiswa dapat mengirimkan jawaban tugas yang kemudian dapat dinilai oleh dosen.

* 1. Analisis Entitas, Atribut, dan *Relationship* (ERD)



Gambar 2. ERD

* 1. *Relational Relationship Diagram*



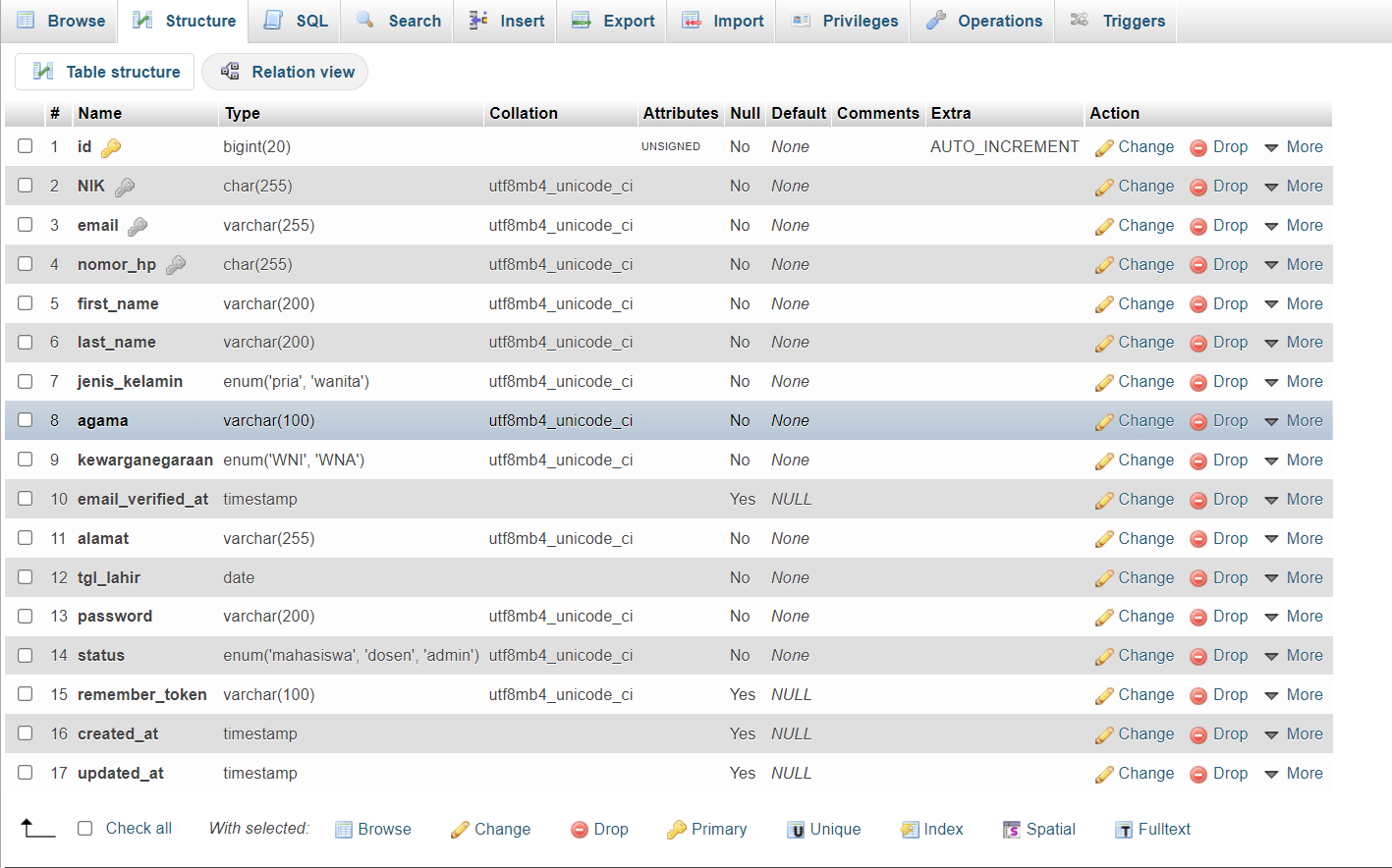
Gambar 3. *Relational Relationship Diagram*

* 1. *Relational Database Schema*

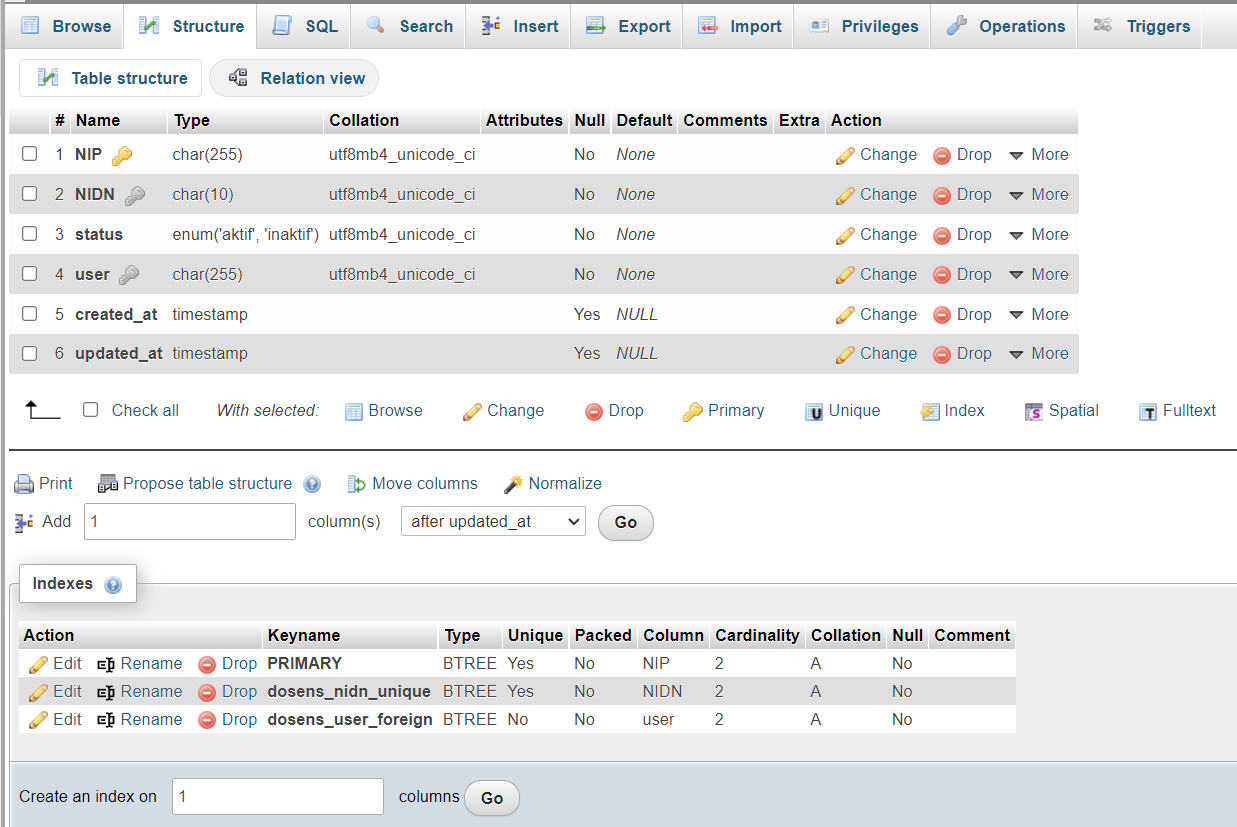
Melalui gambar, *relational relationship diagram* di atas dapat digambarkan bahwa

*relational database schema* yang ada ialah sebagai berikut.

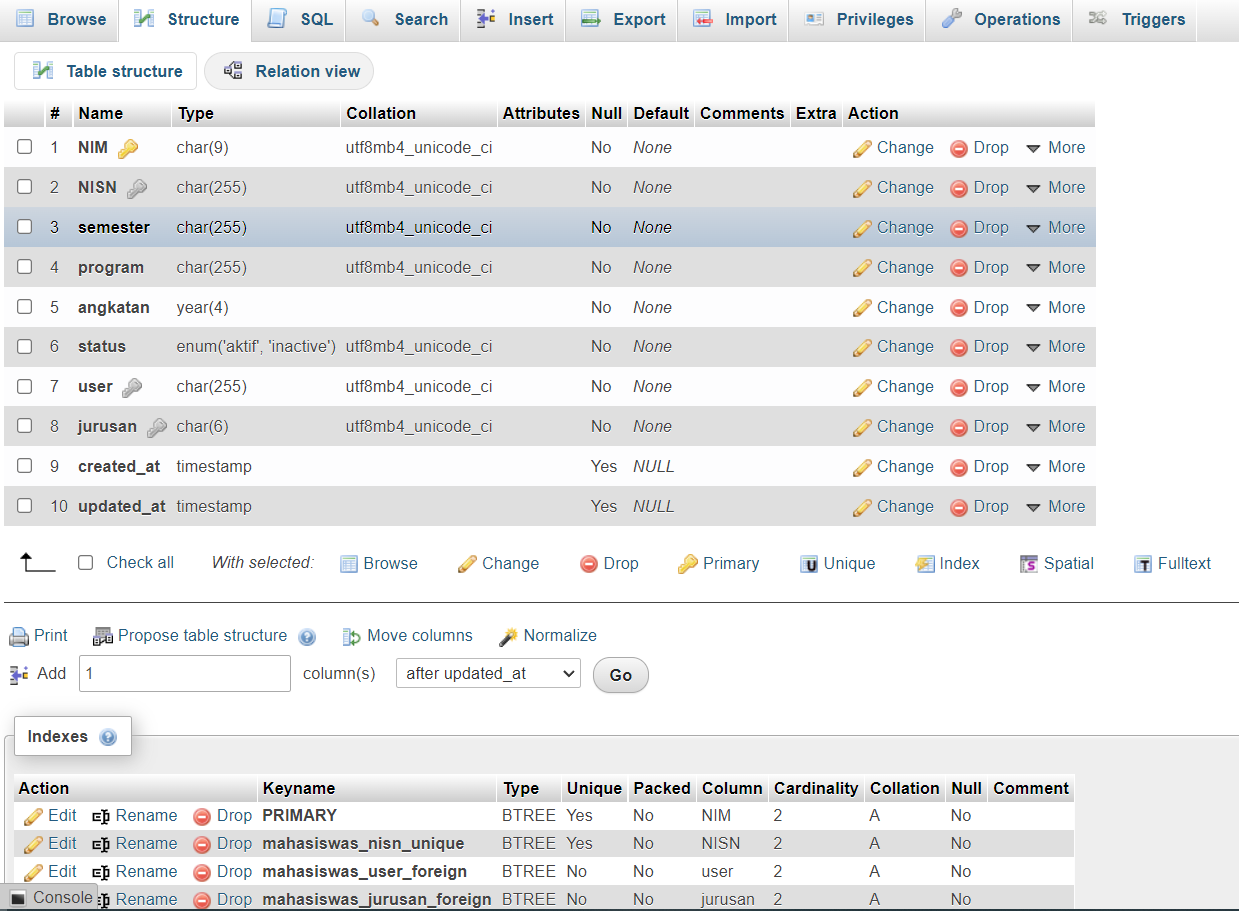
* + - users (id, NIK, email, first\_name, last\_name, jenis\_kelamin, agama, kewarganegaraan, alamat, tgl\_lahir, password, status)
    - dosens (NIP, NIDN, status, user)
    - mahasiswas (NIM, \_NISN, semester, program, angkatan, status, berat, user, jurusan)
    - fakultas (kode\_fakultas, nama\_fakultas, dekan,)
  1. Database

1. Tabel users

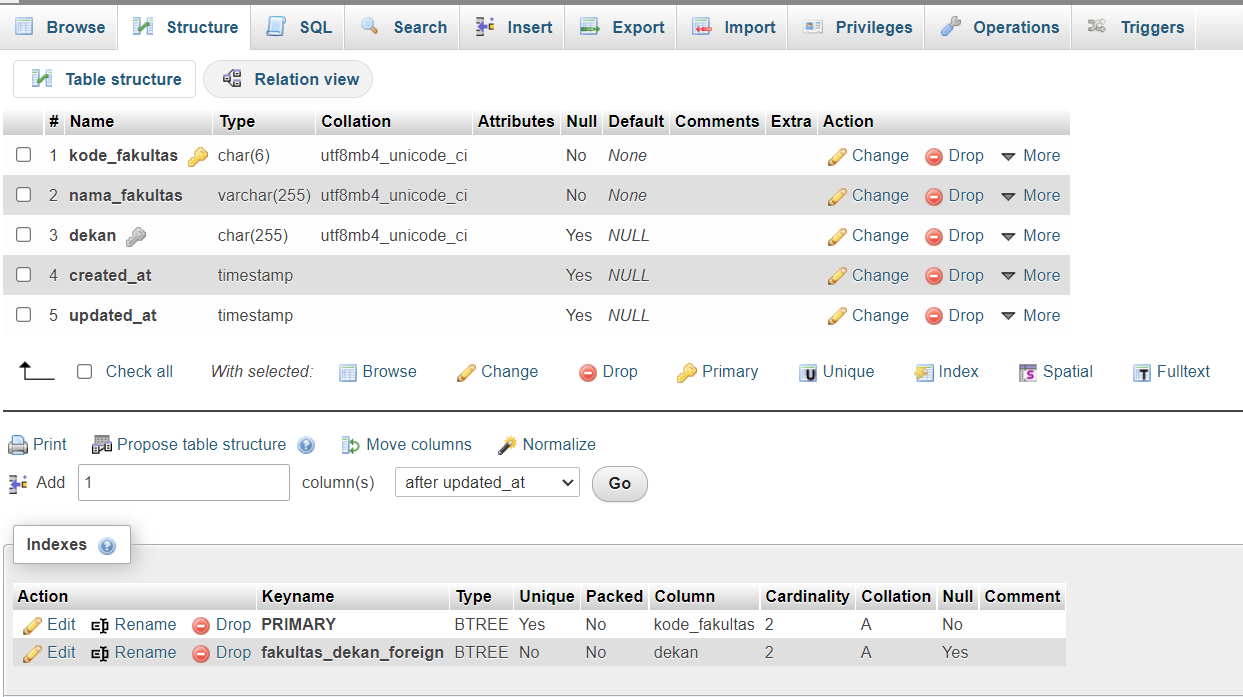
Gambar 4. Struktur Tabel users

1. Tabel dosens

Gambar 5. Struktur Tabel dosens

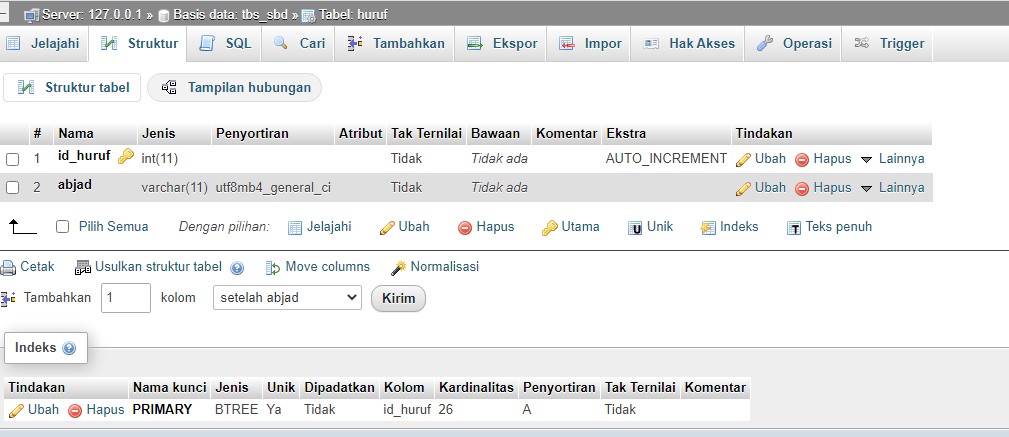
1. Tabel mahasiswas

Gambar 6. Struktur Tabel mahasiswas

1. Tabel fakultas

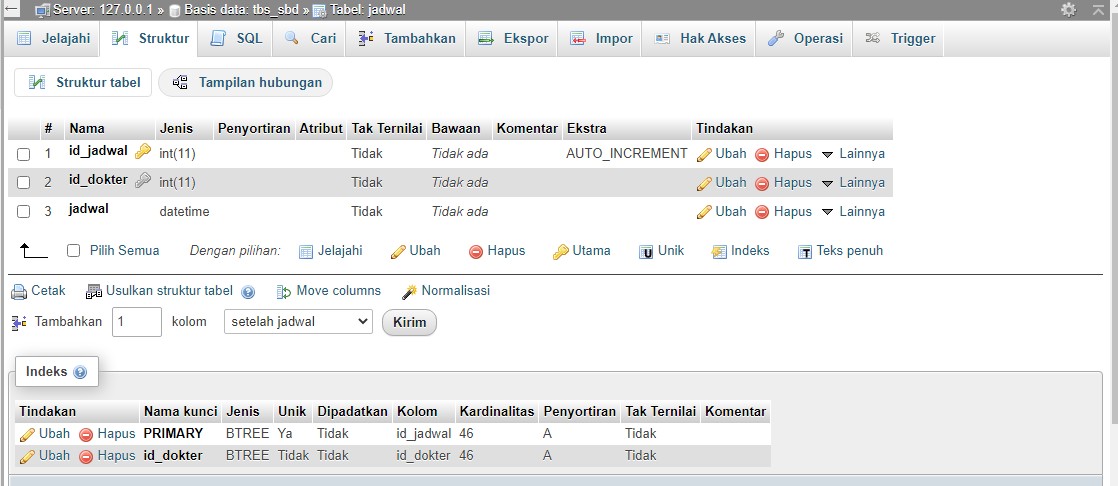
Gambar 7. Struktur Tabel fakultas

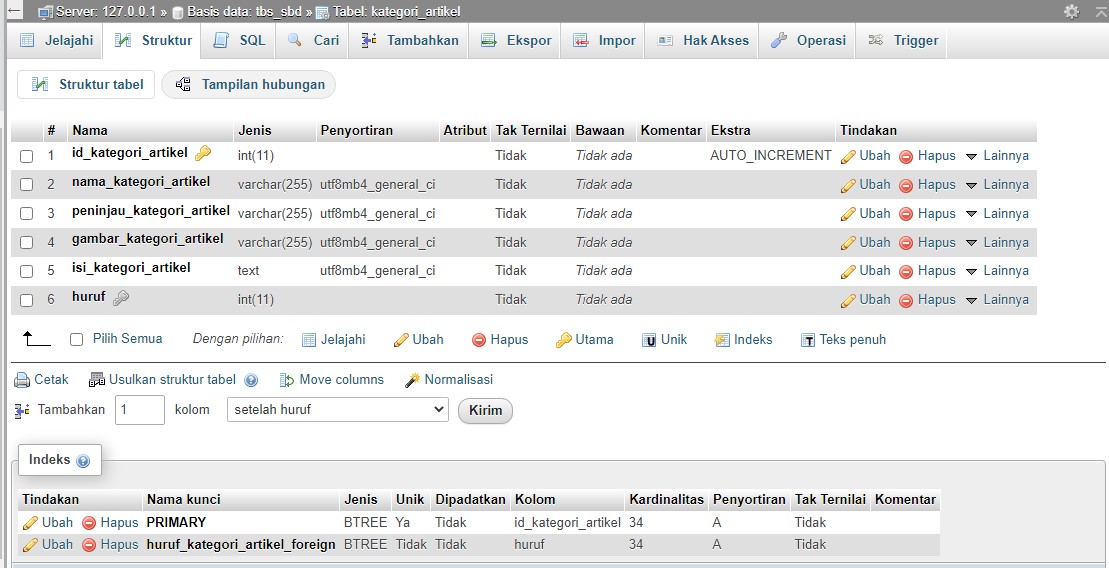
1. Tabel huruf



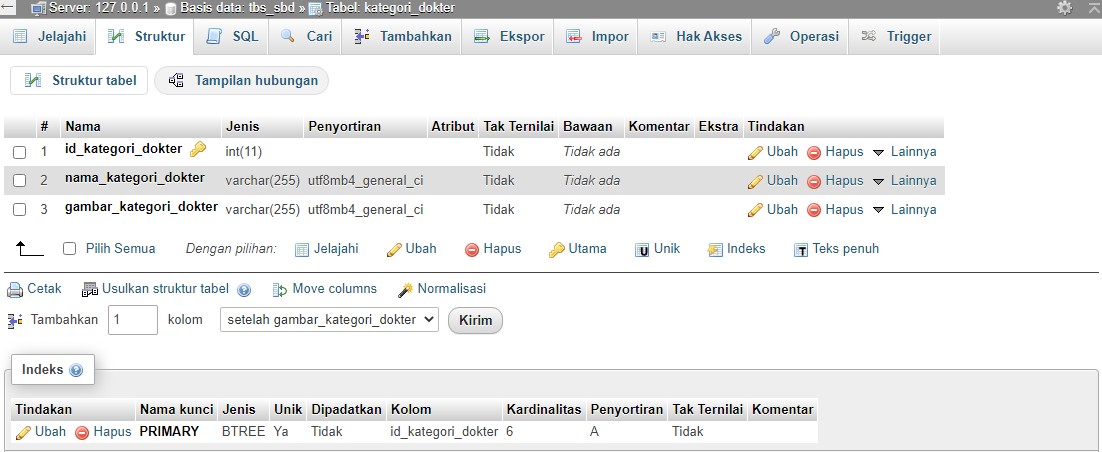
1. Tabel jadwal

Gambar 8. Struktur Tabel huruf

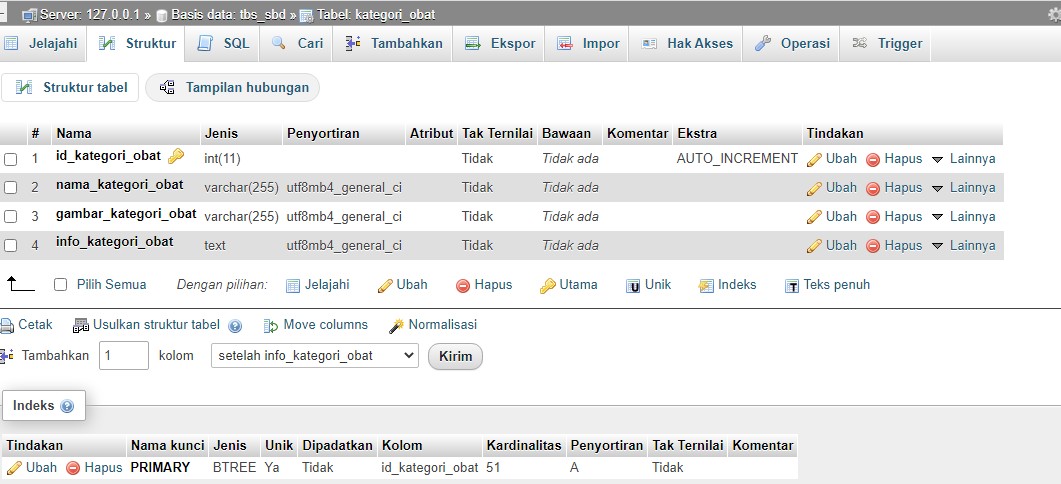
Gambar 9. Struktur Tabel jadwal

1. Tabel kategori\_artikel

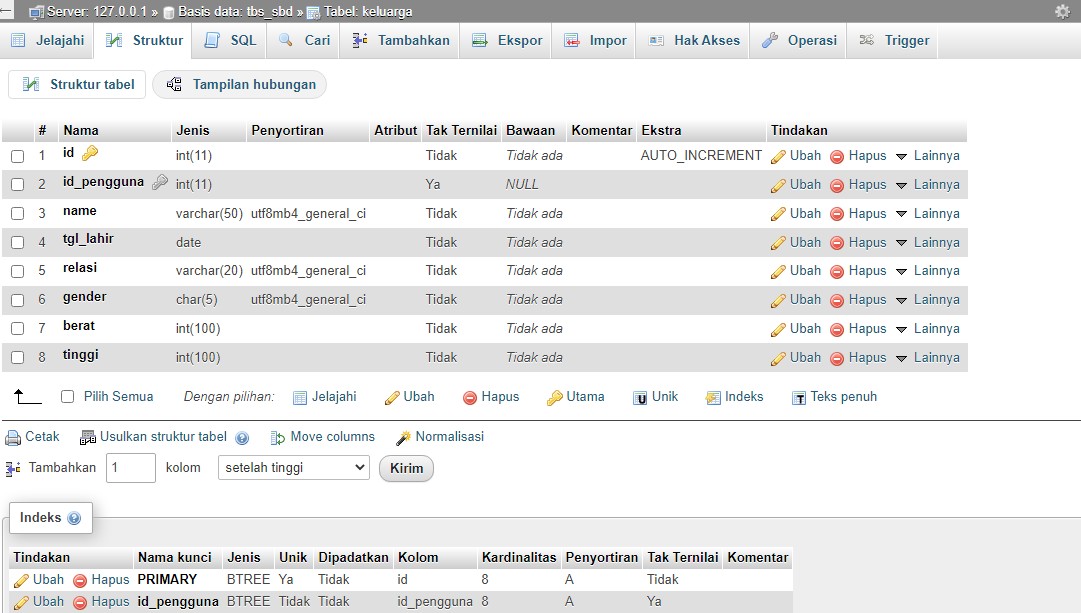
Gambar 10. Tabel Struktur kategori\_artikel

1. Tabel kategori\_dokter

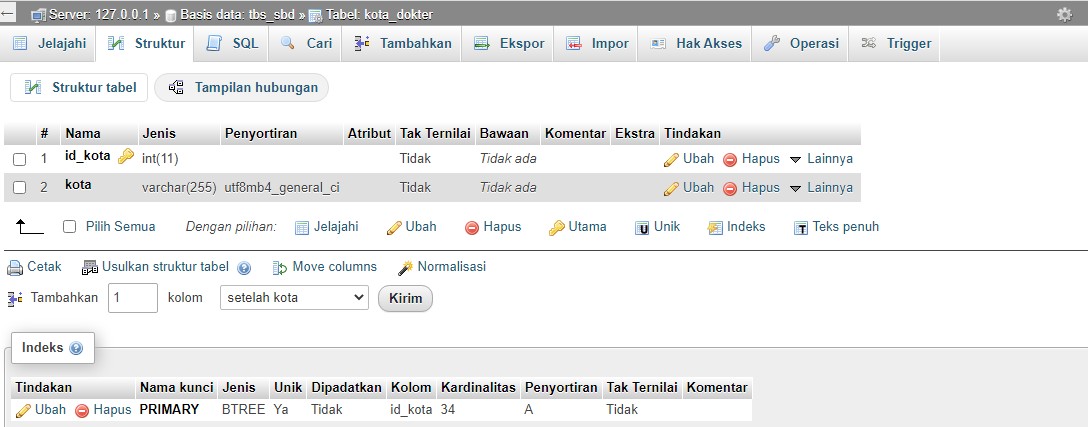
Gambar 11. Tabel Struktur kategori\_dokter

1. Tabel kategori\_obat

Gambar 12. Tabel Struktur kategori\_obat

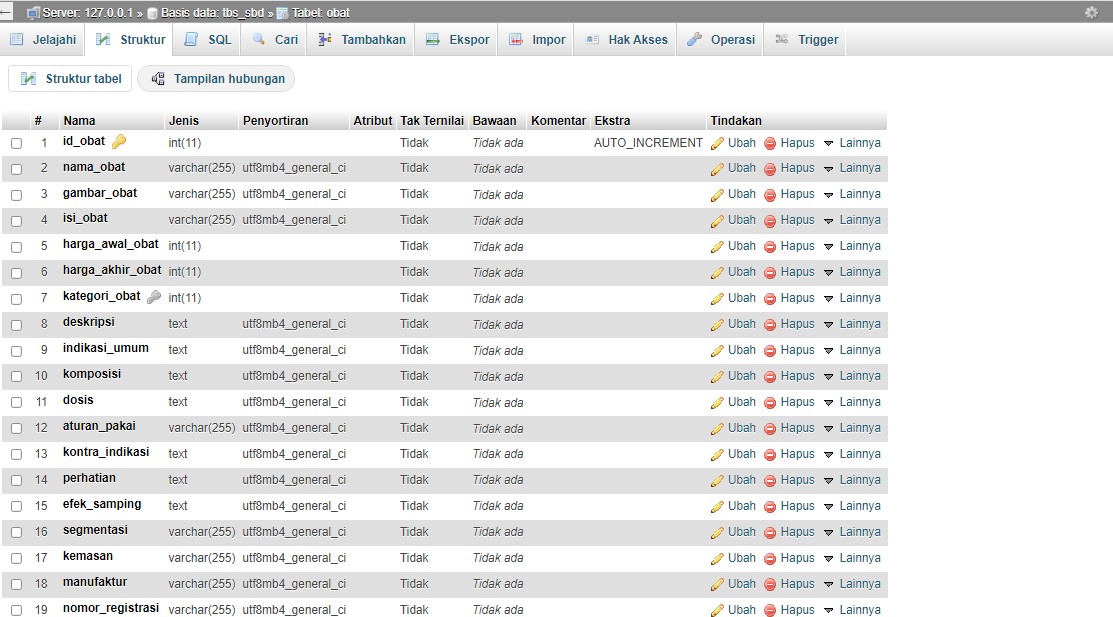
1. Tabel keluarga

Gambar 13. Tabel Struktur keluarga

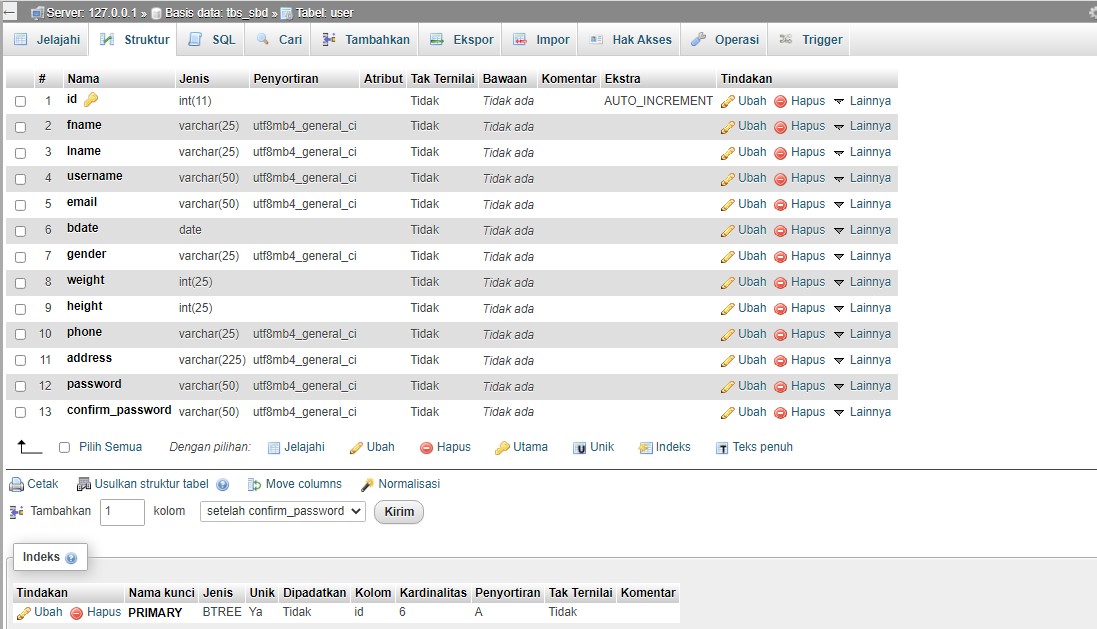
1. Tabel kota\_dokter

Gambar 14. Tabel kota\_dokter

1. Tabel obat



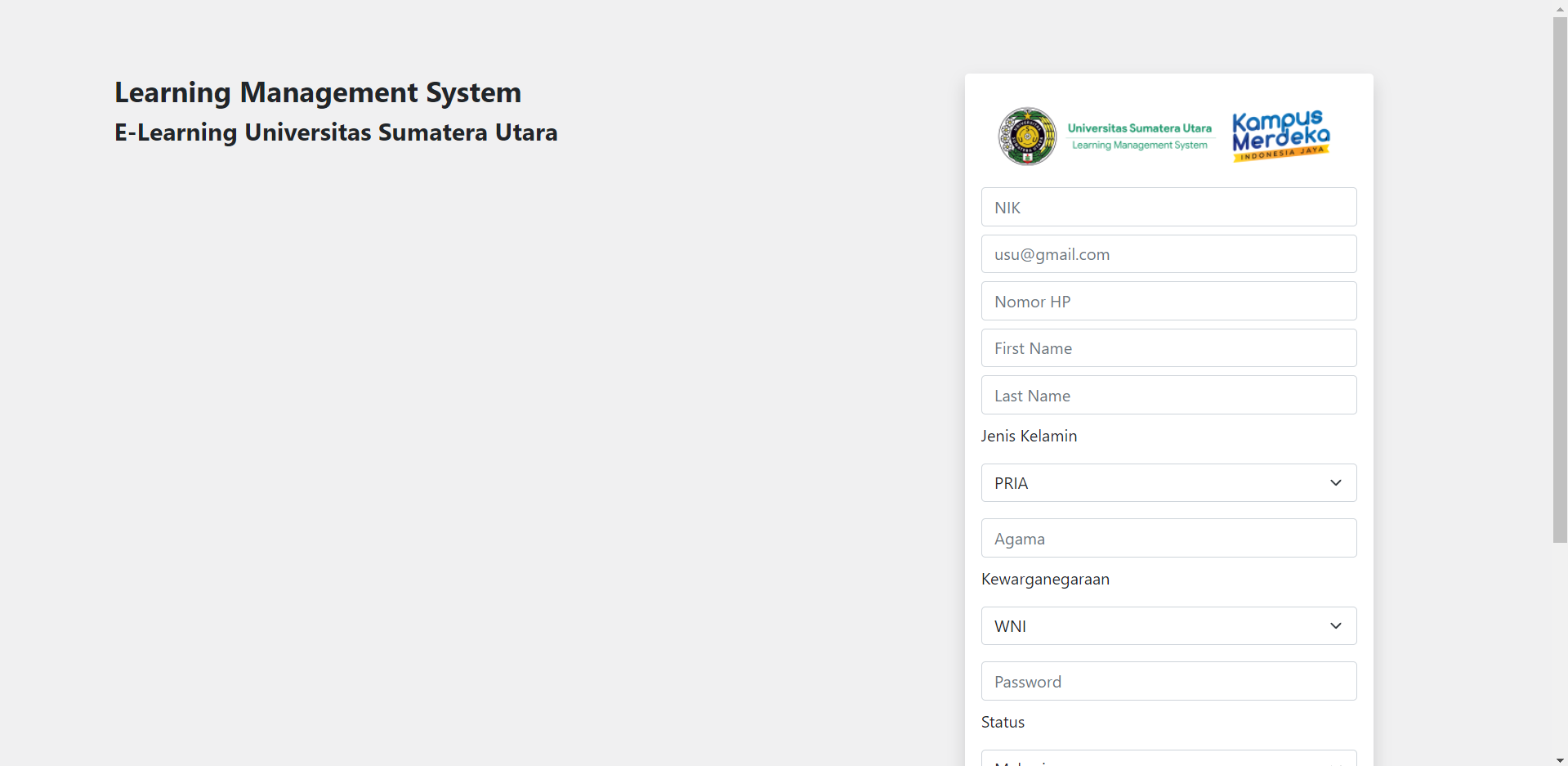
Gambar 15. Struktur Tabel obat

1. Tabel user

Gambar 16. Struktur Tabel user

* 1. Tampilan Program dan Penjelasan *Query*

Website ini menggunakan *framework* Laravel dimana untuk menampilkan data yang ada di database menggunakan Model yaitu program perantara antara database dengan *project* Laravel ini.

1. **Halaman Login dan Registrasi**
   1. Registrasi

Gambar 17. Halaman Registrasi

Gambar diatas menunjukkan halaman registrasi bagi pengguna/pengunjung yang ingin mendaftar *elearning* .

Query :

$validatedData = $request->validate([

            'NIK' => 'required|min:1|max:16|unique:users',

            'email' => 'required|email:dns|unique:users',

            'nomor\_hp' => 'required|min:8|max:255|unique:users',

            'jenis\_kelamin' => 'required',

            'first\_name' => 'required|min:3|max:200',

            'last\_name' => 'required|min:3|max:200',

            'agama' => 'required|min:5|max:255',

            'kewarganegaraan' => 'required',

            'alamat' => 'required|min:1|max:255',

            'tgl\_lahir' => 'required',

            'password' => 'required|min:5|max:255',

            'status' => 'required',

        ]);

        $validatedData['password'] = Hash::make($validatedData['password']);

        User::create($validatedData);

Query di atas akan melakukan pengecekan terhadap beberapa data untuk validasi. Ada beberapa data seperti NIK, email, nomor\_hp yang diharuskan unik.

SELECT \* FROM users WHERE NIK = ‘NIK’;

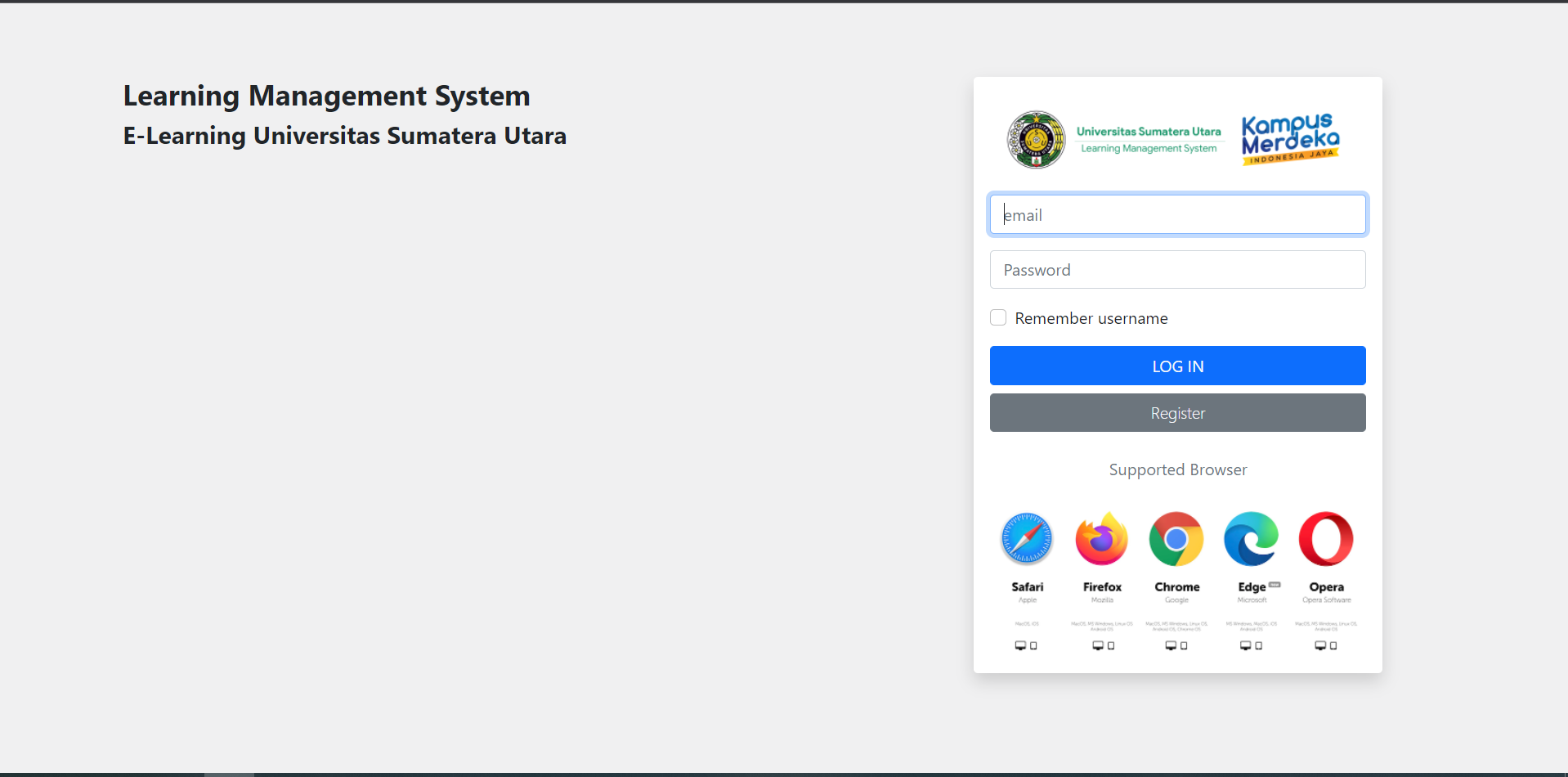
SELECT \* FROM users WHERE email = ‘email’

SELECT \* FROM users WHERE nomor\_hp = ‘nomor\_hp’

Lalu Jika Kondisi false maka Program akan menginsert data kedalam database

INSERT \* INTO users (NIK, email, nomor\_hp, first\_name, last\_name, jenis\_kelamin, agama, kewarganegaraan, alamat, tgl\_lahir, password, status) VALUES ( ‘$NIK’, ‘$email’, ‘$nomor\_hp’, ‘$first\_name’, ‘$jenis\_kelamin’, ‘$agama’, ‘$kewarganegaraan’, ‘$alamat’, ‘$tgl\_lahir’, ‘$password’, ‘$status’

* 1. Login



Gambar 17. Halaman Login (User)

Gambar diatas merupakan tampilan login.

Query :

 $credentials = $request->validate([

            'email' => 'required',

            'password' => 'required'

        ]);

        if(Auth::attempt($credentials)){

            $request->session()->regenerate();

            //return redirect()->intended('/testadmin');

            if(Auth::user()->status === "mahasiswa" || Auth::user()->status === "dosen") {

                return redirect()->intended('/dashboard');

            } else {

                return redirect()->intended('/admin');

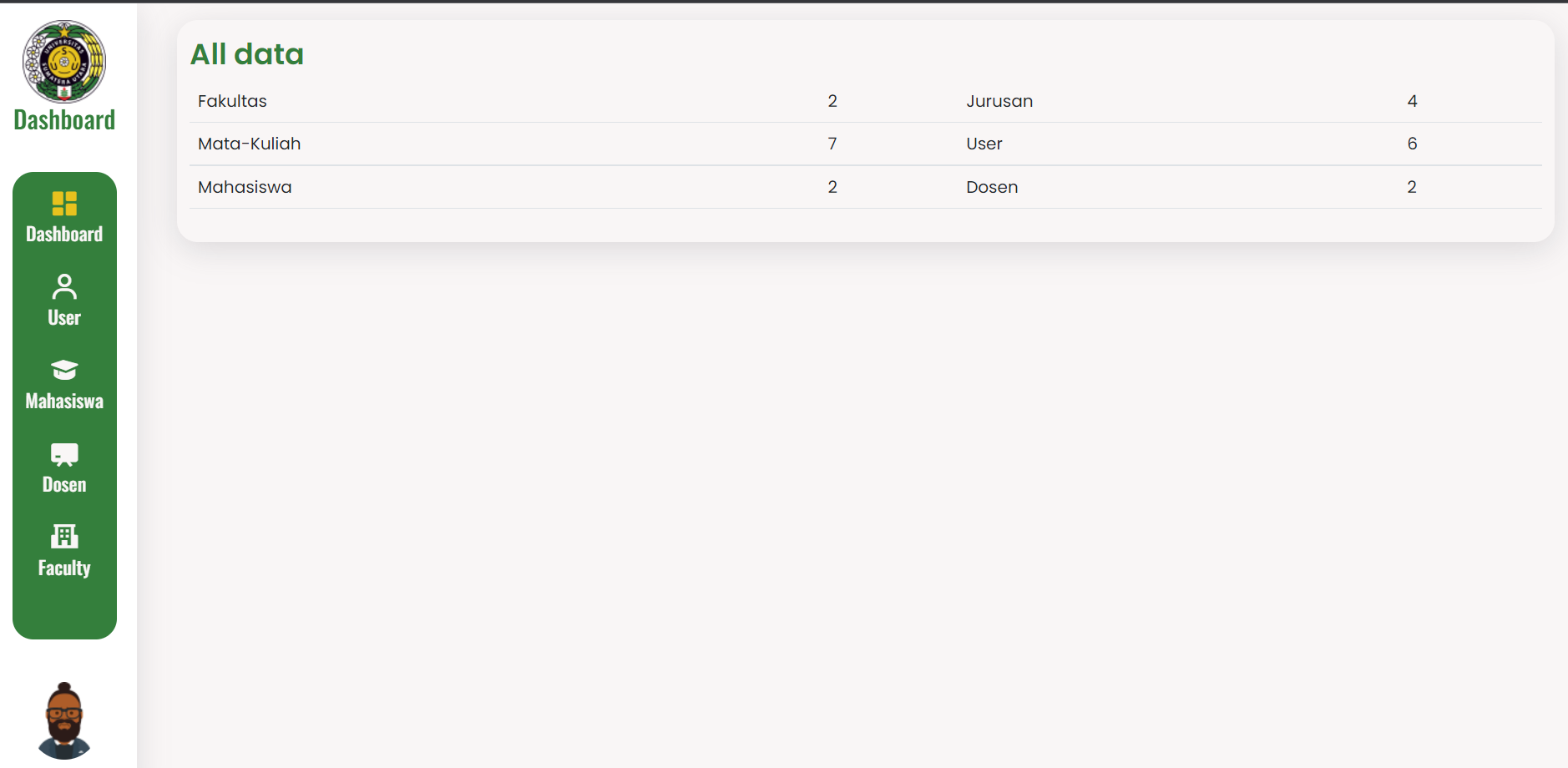
            }

        }

SELECT \* FROM users WHERE email='$email' AND password='$password'

Dengan menggunakan *query* diatas, program akan mengecek apakah email dan password yang dimasukkan user sudah sesuai atau tersimpan di dalam database. Apabila telah tersimpan maka user akan diarahkan ke rute dashboard apabila status nya dosen atau mahasiswa. Jika status admin maka user akan dikembalikan ke rute admin, namun jika tidak tersimpan atau tidak sesuai user akan dikembalikan ke halaman login.

1. **Halaman Admin**
   1. Dashboard Admin



Gambar 18. Halaman Dashboard (Admin)

Query :

$user = User::count();

        $fakultas = Fakultas::count();

        $jurusan = Jurusan::count();

        $matkul = Mata\_kuliah::count();

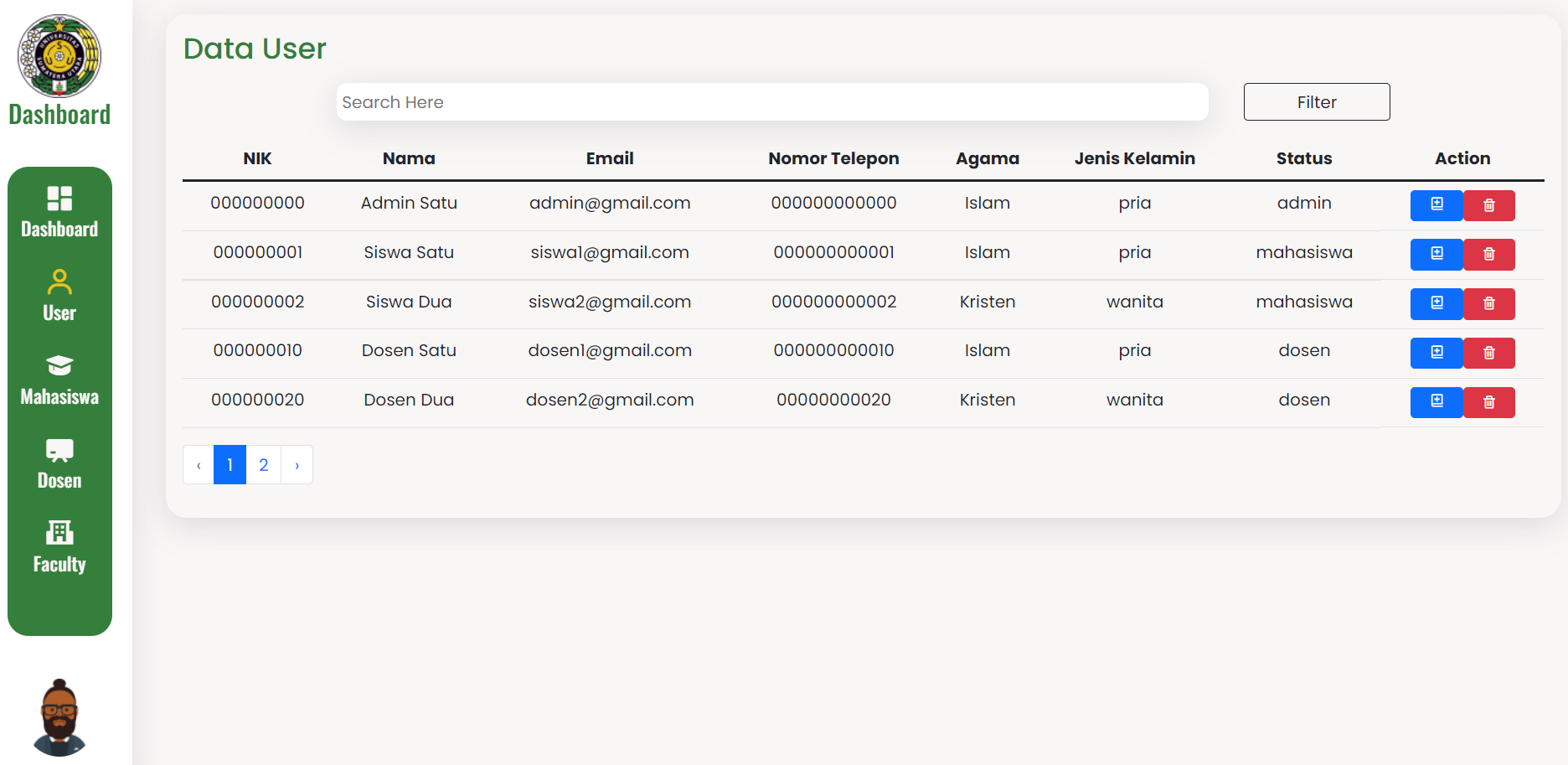
        $mahasiswa = Mahasiswa::count();

        $dosen = Dosen::count();

SELECT COUNT(id\_table) FROM nama\_table

Dengan query diatas didapat jumlah record yang ada di table yang kita pilih. Sehingga tampil jumlah record dari ke enam table

* 1. User Admin



Gambar 18. Halaman Table User (Admin)

User::select('NIK', 'first\_name', 'last\_name', 'email', 'nomor\_hp', 'agama', 'jenis\_kelamin', 'status')->paginate(5)

SELECT NIK, first\_name, last\_name, email, no\_hp, agama, jenis\_kelamin, status FROM users LIMIT 5 OFFSET 0

Query tersebut akan menampilkan data dari table user ( NIK, first\_name, last\_name , email, nomor\_hp, agama, jenis\_kelamin, status)

Admin tidak dapat mengedit data user kecuali user tersebut tersendiri

Admin dapat mengahpus dengan query

User::where('NIK', $nik)->delete();

DELETE FROM users WHERE id = ‘id’

Admin dapat membuat profil mahasiswa dan dosen sehingga mereka dapat login

Mahasiswa::create([

            'NIM'=> $request->NIM,

            'NISN'=> $request->NISN,

            'program'=>$request->program,

            'angkatan'=>$request->angkatan,

            'jurusan'=>$request->jurusan,

            'semester'=> 'I',

            'status'=>'aktif',

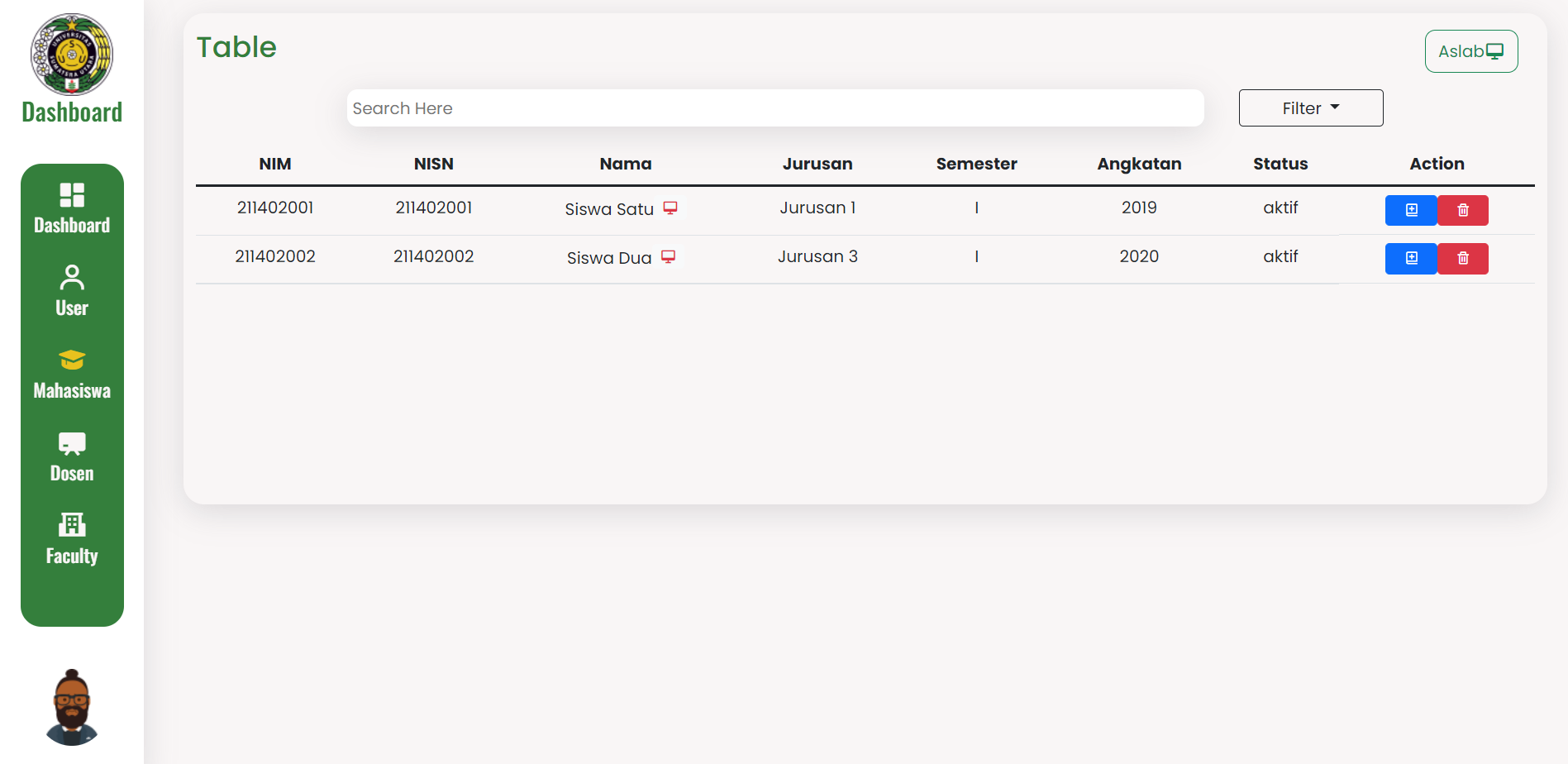
            'user'=>$request->NIK

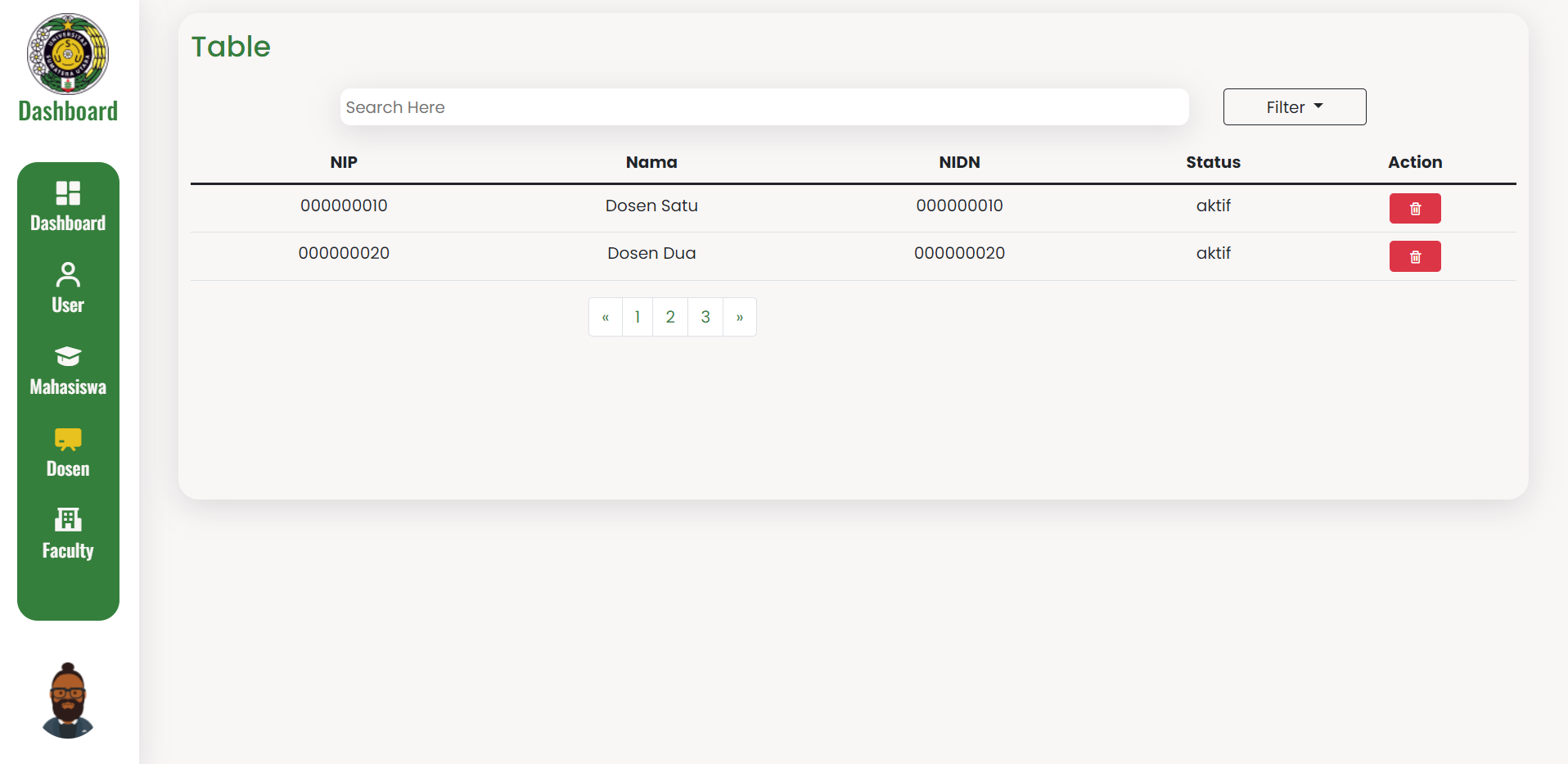
        ]);

INSERT \* INTO mahasiswas (NIM, NISN program, angkatan, jurusan, semester, status, user) VALUES ( ‘$NIM’, ‘$NISN’, ‘$program’, ‘$angkatan’, ‘$jurusan’, ‘$semester’, ‘aktif’, ‘$NIK’)

* 1. Mahasiswa dan Dosen Admin

Gambar 18. Halaman Table Siswa (Admin)





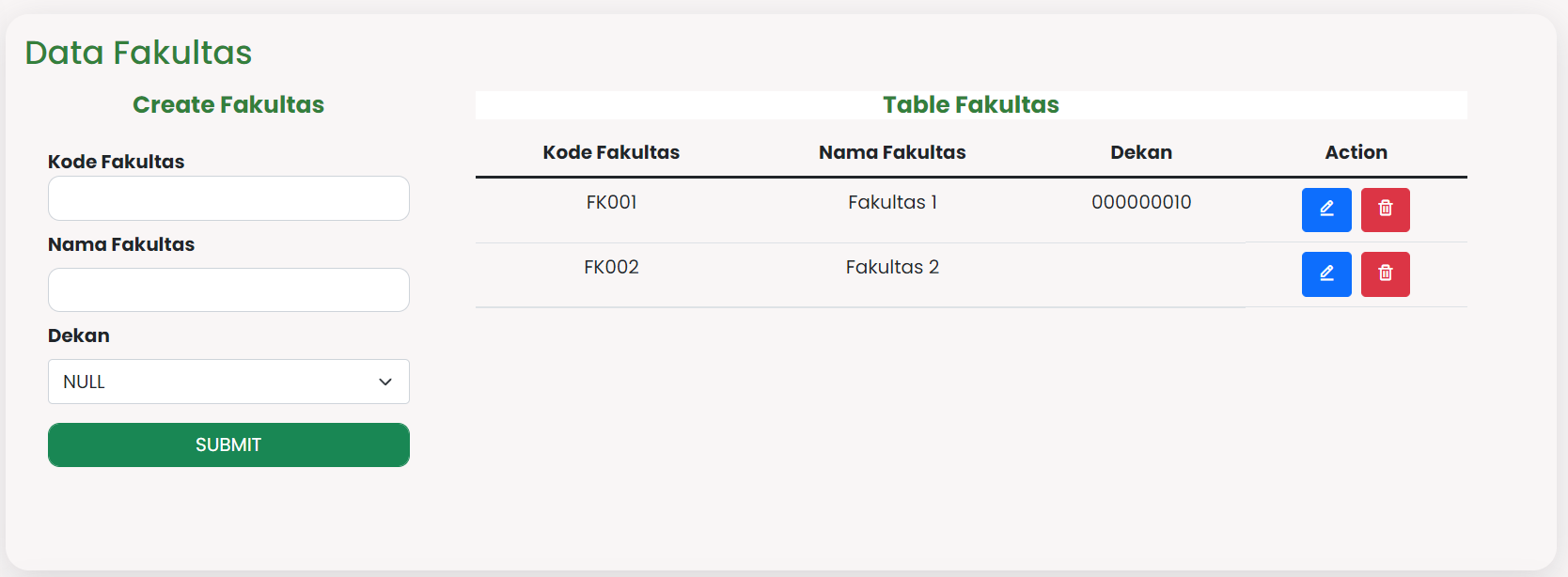
Gambar 18. Halaman Table Dosen (Admin)

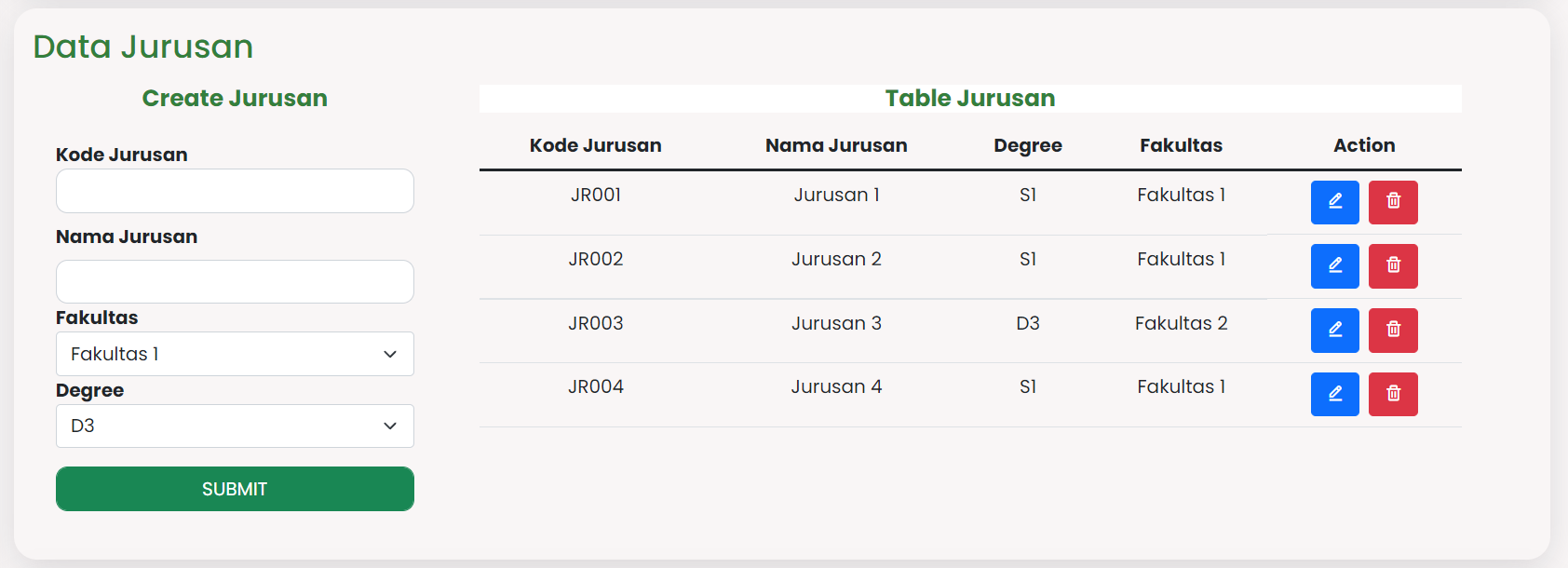
Mahasiswa::join('jurusans','mahasiswas.jurusan', '=', 'jurusans.kode\_jurusan')->join('users', 'mahasiswas.user', '=', 'users.NIK')->select('mahasiswas.NIM','mahasiswas.NISN' ,'mahasiswas.semester', 'mahasiswas.angkatan', 'mahasiswas.status' ,'users.first\_name', 'users.last\_name', 'jurusans.nama\_jurusan')->paginate(5)

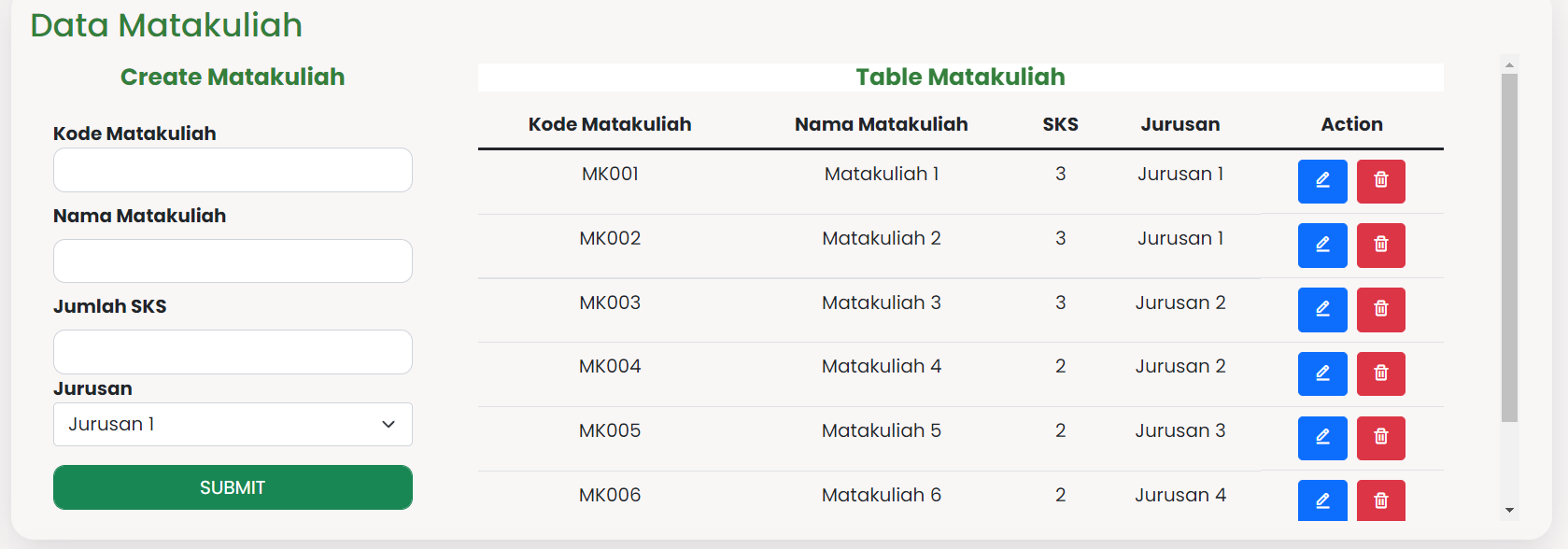
SELECT mahasiswas.NIM, mahasiswas.NISN , mahasiswas.semester, mahasiswas.angkatan, mahasiswas.status, users.first\_name, users.last\_name, jurusans.nama\_jurusan FROM mahasiswas JOIN jurusans ON mahasiswas.jurusan = jurusans.kode\_jurusan JOIN users ON mahasiswas.user = users.id LIMIT 5 OFFSET 0

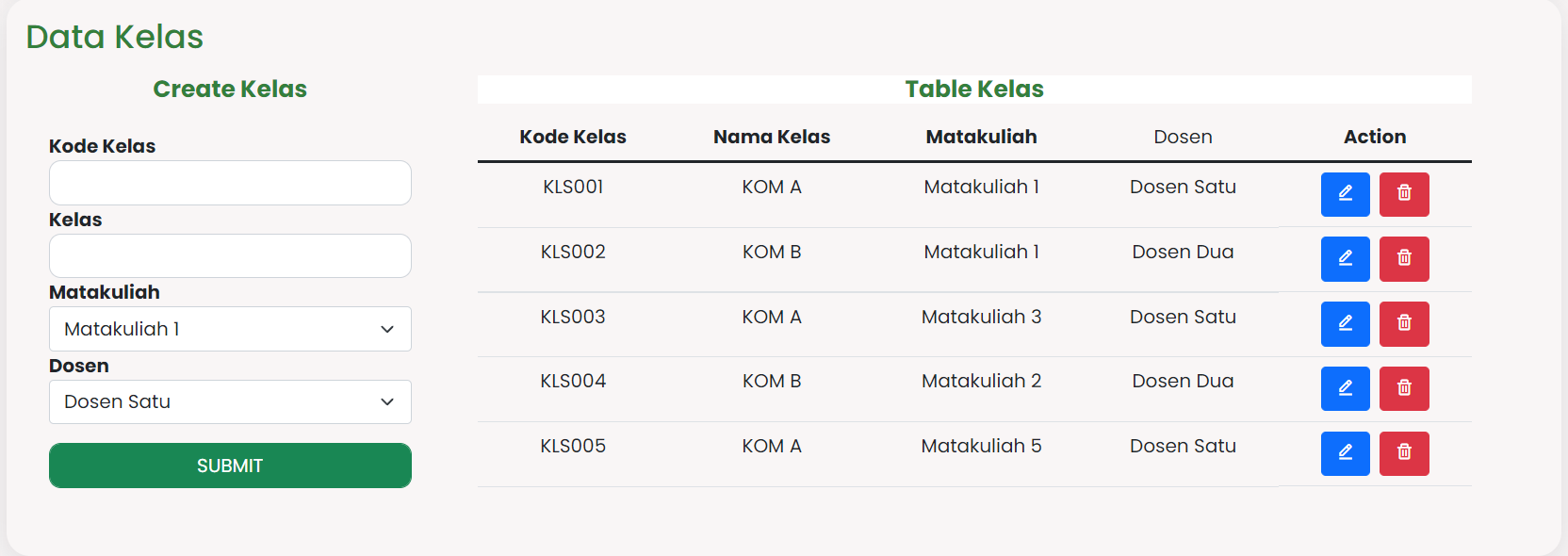
Query tersebut akan menampilkan data dari table mahasiswa ( NIM, NISN, semester, Angkatan, status ), table user ( first\_name, last\_name ) dan table jurusan ( nama\_jurusan ) dengan limit 5 dimulai dari indeks ke 0

* 1. Fakultas Admin









Query :

$dosen =  Dosen::join('users', 'dosens.user', '=', 'users.NIK')->select('dosens.NIP', 'users.first\_name', 'users.last\_name')->get();

fakultas = Fakultas::get();

$jurusan = Jurusan::join('fakultas', 'jurusans.fakultas\_id' , '=', 'fakultas.kode\_fakultas')->get();

$matakuliah = Mata\_kuliah::join('jurusans', 'mata\_kuliahs.jurusan', '=', 'jurusans.kode\_jurusan')->get();

$kelas = Kelas::join('mata\_kuliahs', 'kelas.mata\_kuliah', '=', 'mata\_kuliahs.kode\_mata\_kuliah')->join('dosens', 'kelas.dosen', '=', 'dosens.NIP')->join('users', 'dosens.user', '=', 'users.NIK')->select('kelas.kelas\_id', 'kelas.kelas', 'mata\_kuliahs.nama\_matkul', 'users.first\_name', 'users.last\_name')->get();

Dosen = SELECT dosens.NIP, users.first\_name, users.last\_name FROM dosens JOIN users ON dosens.user = users.NIK

Fakultas = SELECT \* FROM fakultas

Jurusan = SELECT \* FROM jurusans JOIN fakultas ON jurusans.fakultas\_id = fakultas.kode\_fakultas

Matakuliah = SELECT \* FROM mata\_kuliahs JOIN jurusans ON mata\_kuliahs.jurusan = jurusans.kode\_jurusan

Kelas = SELECT kelas.kelas\_id, kelas.kelas, mata\_kuliahs.nama\_matkul, users.first\_name, users.last\_name FROM kelas JOIN mata\_kuliahs ON kelas.mata\_kuliah = mata\_kuliahs.kode\_mata\_kuliah JOIN dosens ON kelas.dosen = dosens.NIP JOIN users ON dosens.user = users.NIK

Ada pula dibagian ini fitur untuk menghapus data dan juga update data

Contoh update data :

$validate = $request->validate([

                'kode\_fakultas'=>"required",

                'nama\_fakultas'=>"required",

            ]);

            Fakultas::where('kode\_fakultas', $request->kode\_fakultas)->update([

                'kode\_fakultas'=>$request->kode\_fakultas,

                'nama\_fakultas'=>$request->nama\_fakultas,

                'dekan'=>$request->dekan

            ]);

Update fakultas SET kode\_fakultas = $kode\_fakultas, nama\_fakultas = $nama\_fakultas, dekan = $NIP WHERE kode\_fakutlas = $id

Hal yang sama juga diterapkan di update data yang lain

Untuk Delete

Fakultas::where('kode\_fakultas', $id)->delete();

Delete FROM Fakultas where kode\_fakultas = $id

1. **Halaman User**
2. **Prioritas CRUD**
3. CRUD Admin

CRUD Artikel merupakan fitur dimana admin dapat mengelola user, mahasiswa, dosen, fakultas, jurusan, matakuliah dan kelas dari database.

1. CRUD Kelas Dosen

CRUD Kelas Dosen fitur dimana dosen dapat mengelola sebuah kelas dari database

3.1. KESIMPULAN

# BAB III PENUTUP

Pembuatan aplikasi website Halodoc ditujukan untuk penyediaan layanan pembuatan janji dengan dokter secara online, pencarian informasi kesehatan dan obat-obatan, pengecheckan riwayat janji, serta memudahkan admin dalam pengelolaan artikel, obat, dan permohonan janji dari pengguna.