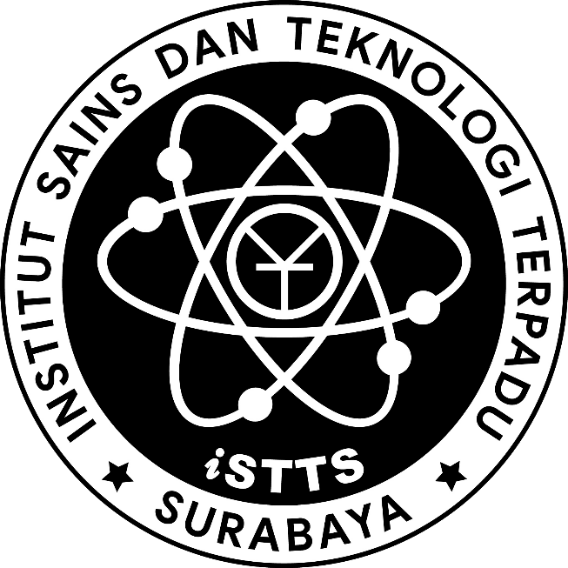
PROYEK AKHIR

SOFTWARE DEVELOPMENT PROJECT

WEBSITE SISTEM INFORMASI MAHASISWA



Oleh:

Andyco Nicky 218116676

Andy 218116710

David Brave 218116715

Marco Holiwono 218116730

PROGRAM STRATA-1

JURUSAN TEKNIK INFORMATIKA

INSTITUT SAINS DAN TEKNOLOGI TERPADU SURABAYA

SURABAYA

2020

SOFTWARE DEVELOPMENT PROJECT

SOFTWARE DEVELOPMENT PROJECT

WEBSITE SISTEM INFORMASI MAHASISWA

Diajukan Untuk Memenuhi Persyaratan Menyelesaikan Mata Kuliah Software Development Project

Institut Sains dan Teknologi Terpadu Surabaya

Disetujui oleh Dosen Pembimbing:

1. REDDY ALEXANDRO H., S.Kom., M.Kom. (Pembimbing)

SURABAYA

November 2020

ii

KATA PENGANTAR

Segala puji dan syukur dipanjatkan kepada Tuhan Yang Maha Kuasa karena atas rahmat dan karunia yang telah diberikan-Nya, penyusunan buku “Website Sistem Informasi Mahasiswa” dapat diselesaikan dengan baik. Semoga seluruh pembaca buku ini bisa mendapatkan sesuatu yang berguna dan bermanfaat. Buku ini disusun dengan tujuan menjadi salah satu syarat kelulusan mata kuliah Software Development Project.

Tentunya penulis juga menyampaikan ucapan terima kasih kepada dosen pembimbing, Bapak Reddy Alexandro H., S.Kom., M.Kom. untuk semua saran dan bimbingannya dalam proses pembelajaran juga penyusunan buku ini. Buku ini jauh dari kata sempurna dan atas semua kekurangan dan kesalahan yang terdapat di dalam buku ini, penulis menminta maaf. Saran dan kritik sangat diharapkan untuk mengembangkan dan menjadi masukan bagi penulis kedepannya. Atas perhatian seluruh pembaca, diucapkan terima kasih.

Surabaya, November 2020

Penulis

iii

DAFTAR ISI

Halaman

HALAMAN JUDUL i

KATA PENGANTAR iii

DAFTAR ISI iv

DAFTAR GAMBAR vi

DAFTAR TABEL ix

DAFTAR SEGMEN PROGRAM xii

BAB I PENDAHULUAN 1

1.1 Latar Belakang 1

1.2 Tujuan 2

1.3 Ruang Lingkup 2

1.4 Sistematika Pembahasan 3

BAB II TEORI PENUNJANG 6

2.1 HTML 6

2.2 Php 6

2.3 JavaScript 6

2.4 MySQL 6

2.5 Git 7

2.6 Github 7

2.7 Trello 7

2.8 Visual Studio Code 7

iv

2.9 Power Designer 8

2.10 Scrum 8

2.11 User Interface 9

2.12 User Experience 10

iv

DAFTAR ISI

Halaman

BAB III PENGUMPULAN DATA DAN DESAIN SISTEM 11

3.1 Pengumpulan Data 11

3.2 Analisa Sistem 12

3.3 Desain Database 14

BAB IV IMPLEMENTASI SISTEM 29

4.1 Implementasi 29

BAB V UJI COBA 62

BAB VI PENUTUP 70

DAFTAR PUSTAKA 71

LAMPIRAN A ENTITY RELATIONSHIP DIAGRAM A-1

v

DAFTAR GAMBAR

Gambar Halaman

1.1 Gambar 7

vi

DAFTAR GAMBAR

Gambar Halaman

1.1 Gambar 1

6.19 Gambar 2

vii

DAFTAR GAMBAR

Gambar Halaman

1.1 Gambar 7

6.12 Gambar 80

viii

DAFTAR TABEL

Tabel Halaman

ix

DAFTAR TABEL

Tabel Halaman

x

DAFTAR TABEL

Tabel Halaman

xi

DAFTAR SEGMEN PROGRAM

Segmen Program Halaman

xii

**BAB I**

**PENDAHULUAN**

1. **Latar Belakang**

Situs sistem informasi mahasiswa (SIM) adalah sebuah sistem online yang dibuat dengan tujuan membantu seluruh peserta kegiatan akademik mendapatkan data akademik atau informasi yang telah diolah. SIM menjadi pusat pencarian informasi penting bagi seluruh mahasiswa, mulai dari jadwal perkuliahan, data absensi maupun rekapitulasi nilai yang diperoleh sepanjang kuliah. SIM juga digunakan untuk mendukung dosen melalukan proses pengajaran dan pembibingan mahasiswa.

Alasan utama dibentuknya SIM ini adalah untuk mempermudah proses perwalian bagi seluruh mahasiswa dan para dosen. Mengingat sulit dan merepotkannya hal tersebut jika seluruh prosesnya harus dilaksanaakan secara manual atau offline. Mengingat jumlah mahasiswa yang dibimbing tidaklah sedikit, situs SIM tentunya sangat membantu dosen dalam pencatatan data para mahasiswa yang menjadi siswa wali ataupun siswa bimbingan dosen tersebut.

1. **Tujuan**

Tujuan utama pembuatan proyek ini adalah sebagai metode pembelejaran dan penerapan prinsip-prinsip development project secara nyata. Hal ini terkait dengan kerja sama tim, penggunaan metodologi, pembuatan dokumentasi maupun referensi dokumen lainnya.

Selain itu, pemahaman akan sistem project juga menjadi faktor penting, sehingga tujuan dibentuknya aplikasi juga menjadi bahan pembelajaran project ini. Seperti SIM yang diperuntukan menjadi pusat integrasi data dan informasi akademik. Dengan menggunakan SIM, data yang diperlukan dapat terintegrasi dan terhubung sehingga membantu akses data oleh seluruh peserta kegiatan belajar mengajar. Terkait dengan integrasi data, SIM juga tentunya akan memudahkan proses pengelolaan dan pengolahan data, dimana waktu yang diperlukan untuk memproses dan mengelola data akan jauh lebih efektif dibandingkan dikerjakan secara manual.

Data di dalam SIM juga dapat digunakan sebagai penyimpanan data akademik yang tidak hanya dimiliki oleh para mahasiswa saat ini, namun juga para mahasiswa sebekumnya. Hal ini dapat digunakan untuk mengecek peningkatan nilai maupun kualitas pendidikan universitas.

Rekap data di dalam SIM dapat digunakan untuk banyak hal. Seperti memberi laporan kenaikan atau penurunan nilai pembelajaran mahasiswa berdasarkan IP setiap semester. Yang mana hal ini akan sangat memudahkan pemantauan perkermbangan nilai mahasiswa dalam proses belajar mengajar berdasarkan index prestasi yang diperoleh pada semester tersebut. Tidak hanya mahasiswa, laporan pengajaran dosen juga dapat diperoleh untuk mengukur beban pengajaran dosen berdasar data mata kuliah yang diajar dan jumlah sks ataupun total jam kerja pada setiap mata kuliah tersebut.

1. **Ruang Lingkup**

Pembuatan situs SIM ini terfokus pada cakupan kegiatan akademik. Terutama kegiatan akademik yang dengan berhubungan lansung dengan pembelajaran mahasiswa. Seperti proses pengambilan mata kuliah dan perwalian. Untuk seluruh mahasiswa baik baru maupun lama, setiap semester baru akan dilakukan proses pengambilan mata kuliah. Jumlah mata kuliah yang diambil dibatasi oleh Indeks Prestasi Semester (IPS) pada semester sebelumnya. Setelah pengambilan mata kuliah dilakukan melalui formular rencana studi (FRS), dilakukan perwalian untuk persetujuan FRS tersbut dari dosen wali mahasiwa.

Setelah perwalian dan FRS selesai, mahasiswa dapat mengikuti kegiatan belajar mengajar sesuai jadwal mata kuliah yang sudah diambil sebelumnya. Absensi mahasiswa pada semua pertemuan pada jadwal mata kuliah juga akan dicatat pada SIM. Pencatatan absensi tersebut dapat digunakan sebagai penentu apakah seorang mahasiswa dapat mengambil ujian, baik tengah semester UTS maupun akhir semester UAS. Secara penjadwalan, hari dan waktu ujian akan disamakan dengan hari dan waktu pertemuan mata kuliah.

Nilai mahasiswa baik sebelum maupun saat ujian juga akan disimpan sebagai data penentu kelulusan mahasiswa untuk mata kuliah tertentu. Selain itu, hasil perhitungan seluruh data nilai akan digunakan untuk mendapatkan IPS. IPS ini digunakan sebagai batas pengambilan sks dan nilai perhitunagn Indeks Prestasi Kumulatif (IPK).

Mahasiswa dapat mebatalkan mata kuliah yang dirasa kurang sesuai dengan melakukan batal tambah. Jika batal tambah dilakukan, maka nilai mata kuliah tersebut tidak akan dimasukkan pada proses penghitungan IPS. Namun mahasiswa perlu mengulang mata kuliah tersebut. Sama halnya dengan mahasiswa yang mendapat nilai dibawah standar kriteria nilai. Pengulangan dilakukan dengan mengisi FRS pada semester baru dan memilih mata kuliah tersebut Kembali.

Penjadwalan yang terdapat pada SIM tidak hanya jadwal mata kuliah ataupun ujian. Namun juga jadwal tugas akhir, bagi mahasiswa semester 7 atau 8. Jadwal tugas akhir mencakup jadwal sidang tugas akhir dan data dosen pembimbing serta penguji dari setiap mahasiswa.

1. **Sistematika Pembahasan**

* BAB I : PENDAHULUAN

Pada bab ini akan dibahas tujuan utama pengerjaan project serta pembuatan situs SIM, cakupan atau ruang lingkup atau cakupan kerja SIM serta sistematika dan keterang singkat masing-masing bab.

* BAB II : TEORI PENUNJANG

Pada bab ini akan dibahas semua alat dan metode yang digunakan dalam proses pembuatan website SIM ini. Alat dan metode yang digunakan mencakup HTML5, Php, Javascript, MySQL, Git, Github, Trello, Visual Studio Code, Power Designer, metodologi Scrum dan teori dasar dalam pembuatan User Interface dan User Experience.

* BAB III : PENGUMPULAN DATA DAN DESAIN SISTEM

Pada bab ini akan dibahas semua metode pengumpulan data, mulai dari wawancara dan pengamatan. Juga proses desain sistem, seperti entiti yang termasuk dalam sistem.

* BAB IV : IMPLEMENTASI SISTEM

Pada bab ini akan dibahas cara implementasi sistem website SIM beserta cara kerja seluruh fitur yang terdapat pada website SIM. Mulai dari database di mana sebuah data disimpan, perubahan data saat dilakukan suatu kegiatan dan cara penampilan data pada halaman pengguna.

* BAB V : UJI COBA

Pada bab ini akan dilampirkan uji coba yang telah dilakukan untuk semua fitur website SIM, beserta hasil yang diharapkan dalam uji coba tersebut.

* BAB VI : PENUTUP

Bab ini berisi kata penutup dan kesimpulan dari buku website SIM ini beserta saran untuk mengembangkan buku ini kedepannya.

* Daftar Pustaka

Daftar Pustaka berisi sumber-sumber yang digunakan dalam proses pembuatan website SIM dan penyusunan buku ini.

* Lampiran A : ENTITY RELATIONSHIP DIAGRAM

Lampiran A berisi gambar *entity relationship diagram* atau ERD, yang terdiri atas *conceptual data model* (CDM) dan *physical data model* (PDM).

**BAB II**

**TEORI PENUNJANG**

Pada bab 2 ini, akan dijelaskan seluruh *tools*, metode dan teori yang digunakan untuk mendesain situs SIM.

1. **HTML 5**

HTML atau Hyper-Text Markup Language adalah markup language yang digunakan pada hampir seluruh website di dunia. HTML 5 sendiri merukan versi pengembangan dari HTML. Perbedaan utama HTML 5 adalah dukungan grafik dan audio yang jauh lebih luas dibandingkan versi HTML sebelumnya.

1. **Php**

PHP yang dikenal sebagai HyperText Preprocessor adalah bahasa pemrograman web yang dapat ditanamakan kedalam HTML. PHP diciptakan oleh Rasmus Lerdorf. Awalnya PHP merupakan kependekan dari Personal Home Page dan pada awalnya PHP digunakan untuk mengelola data formulir dari web. PHP merupakan bahasa pemrograman yang diciptakan secara gratis dan opensource yang bersifat bebas.

1. **JavaScript**

JavaScript adalah scripting language yang dipelopori oleh Brandan Eich. JavaScript adalah bahasa yang tidak memerlukan compiler dan dapat digunakan oleh seluruh website dan browser. JavaScript dipilih karena sangat interaktif dan dapat dipahami dengan cukup mudah oleh banyak orang.

1. **MySQL**

SQL merupakan kependekan dari Structure Query Language adalah sebuah program yang dikhususkan untuk membuat database. SQL juga merupakam program pengakses database yang dapat diakses melalui jaringan sehingga dapat digunakan oleh banyak pengguna atau Multi User. SQL digunakan dalam membangun sebuah database dengan banyak bahasa pemrograman dan tentunya dalam berbagai platform. MySQL sendiri adalah salah satu database management system, yang dapat digunakan untukm menyimpan jumlah data kecil hingga menengah ke atas.

1. **Git**

Git adalah version control system, yaitu aplikasi yang mencatatat tahapan perubahan dan perbaikan file. Git dikenal juga dengan distributed revision control (VCS terdistribusi), artinya penyimpanan database Git tidak hanya berada dalam satu tempat saja. Git sangat membantu proses kolaborasi project dengan tim lebih dari 2 orang.

1. **Github**

Github adalah website yang menghosting atau menyimpan data version control dari Git. Github memungkinkan collaborators untuk melakukan perubahan pada file secara remote. Perubahan file pada folder atau repository dapat dilakukan melalui branch terlebih dahulu maupun secara langsung pada master.

1. **Trello**

Trello adalah aplikasi kolaborasi yang memungkinkan seluruh angota untuk mengatur berbagai bagian proyek dalam satu tempat. Dengan Trello, semua orang di proyek tersebut bisa tahu apa yang sedang dikerjakan, siapa yang mengerjakannya, dan sudah sejauh mana proses pengerjaannya.

1. **Visual Studio Code**

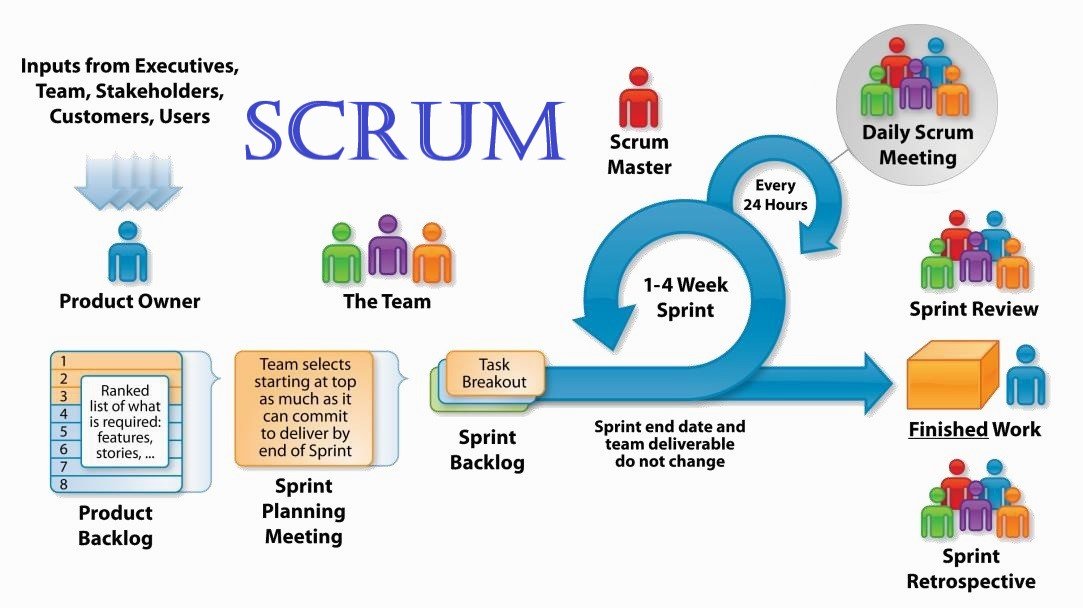
Visual studio code (VS Code) adalah sebuah aplikasi text editor yang bersifat open source, yang memiliki arti bahwa kode dari dari VS Code dapat diakses dan digunakan oleh siapapun secara bebas dan terbuka. VS Code digunakan sebagai tool utama dalam pengeditan kode aplikasi. Seluruh file dalam project dapat dibuka dan diubah secara bersamaan menggunakan VS Code. VS Code juga menyediakan berbagai extension yang sangat membantu dalam penggunaan syntax dan snippet dalam mengembangkan aplikasi. Karena alasan inilah, VS Code dipilih sebagai tool yang digunakan untuk mengedit kode aplikasi.

1. **Power Designer**

Power designer adalah modeling tool yang digunakan untuk merancang database dengan relasi antar data. Beberapa model data yang sering digunakan adalah Conceptual Data Model (CDM) dan Physical Data Model (PDM). Power designer dapat menvisualisasikan kedua model data tersebut untuk membantu mendesain database.

1. **Scrum**

Scrum adalah metodologi pengerjaan sebuah proyek, secara agile. Sehingga pengerjaan dilakukan secara berulang dengan testing pada setiap perulangan. Perulangan pada scrum disebut dengan sprint. Pada setiap sprint dilakukan testing dan review untuk menyiapkan sprint selanjutnya.



**Gambar 2.1** Penggambaran Metodologi Scrum

**2.11 User Interface**

User interface (UI) adalah tampilan visual sebuah website atau aplikasi yang memastikan bahwa pengguna dapat menggunakan website tersebut. UI juga mencakup bagaimana informasi ditampilkan kepada user. UI menggunakan konsep desain visual, dan infrastruktur informasi. Tujuan dari user interface adalah untuk memudahkan pengguna mendapatkan informasi yang dibutuhkan dan memberikan User Experience yang baik. Karakteristik yang dimiliki oleh UI yang baik adalah sebagai berikut :

1. Jelas

Kejelasan UI merupakan salah satu faktor penting, karena dengan desain UI yang jelas, para pengguna dapat berinteraksi dengan sistem secara mudah. Hal ini juga membantu navigasi antar halaman pada sebuah website.

1. Ringkas

UI yang memiliki penjelasan akan sangat membantu, namun penjelasan yang dimiliki haruslah singkat dan tidak bertele-tele. Karena penjelasan yang berlebih akan membuat kesan kacau dan tidak rapi.

1. Familiar

UI yang familiar memiliki arti UI yang akrab dan intuitif kepada pengguna. Sehingga tampilan yang website yang dibuat dapat dimengerti secara mudah dan natural. Hal ini dapat dicapai dengan menggunakan logo atau ikon yang sudah umum dipakai, menggunakan warna yang dengan action yang dilakukan. Seperti tombol logout dibuat berwarna merah.

1. Responsif

Karatkteristik responsive pada UI memiliki arti bahwa website dapat memberikan *feedback* yang cepat dan deskriptif. Sehingga jika dilakukan sebuah action pada website, terdapat pemberitahuan status action tersebut.

1. Konsisten

Konsistensi dapat membantu pengguna mengoperasikan fungsi interface yang ada pada website. Jika pengguna menemukan interface yang mirip, maka pengguna akan dapat mengerti kegunaan component pada interface tersebut. Dengan demikian, pengguna dapat menyelesaikan pekerjaan dengan lebih cepat.

1. Efisien

UI juga perlu dibuat secara efisien untuk memudahkan user melaksanakan tugas yang diinginkan. Untuk mendapatkan UI yang efisien perlu diketahui cara kerja dan alur website, beserta fungsi dan apa kegunaan komponen pada website tersebut.

1. Memaafkan

Tentunya pengguna bisa melakukan kesalahan saat menggunakan website. Seperti tidak sengaja menekan tombol hapus pada sebuah data. Sehingga konfirmasi sederhana tentunya akan sangat membantu mencegah terjadinya kesalahan penghapusan.

**2.12 User Experience**

User Experience (UX) adalah pengalaman dari seorang pengguna dalam menggunakan website. Yang juga berarti perasaan pengguna selama pengguna memakai semua fitur dari website tersebut. Mulai dari tampilan, tema, kemudahan penggunaan dan kelengkapan fitur yang dibutuhkan pengguna.

**BAB III**

**PENGUMPULAN DATA**

**DAN DESAIN SISTEM**

**3.1 Pengumpulan Data**

Metode pengumpulan data yang digunakan adalah wawancara dan pengamatan. Narasumber yang diwawancarai adalah dosen yang mengurusi kegiatan akademik dan website SIM. Pengamatan dilakukan sesuai pengalaman kami sebagai mahasiswa untuk seluruh kegiatan yang berhubungan dengan website SIM.

1. **Wawancara / Interview**

Wawancara dilakukan untuk mengetahui cara kerja system website SIM pada iSTTS. Sistem berfokus pada kegiatan akademik seperti seperti perwalian, batal tambah dan penjadwalan mata kuliah. Narasumber yang diwawancarai adalah Pak Kevin Setiono sebagai salah satu dosen pengurus Bagian Administrasi Akademik dan Kemahasiswaan (BAAK). BAAK sendiri bertugas mengurusi kegiatan akademik dan kemahasiswaan pada universitas. Mengingat website SIM ini berfokus kepada kegiatan akademik, maka hanya data masalah-masalah yang mencakup hal tersebut yang dikumpulkan. Beberapa informasi penting yang didapatkan dari wawancara ini adalah alur perwalian, penjadwalan mata kuliah, absensi kuliah, dan data nilai yang diambil.

Narasumber lain yang diwawancarai adalah Pak Michael Setiwan, yang mengurusi website SIM untuk iSTTS. Wawancara ini dilakukan untuk mendapatakan data website SIM, seperti isi database, data yang dapat diakses pada halaman mahasiswa dan data yang dapat diakses pada halaman dosen. Pada website SIM yang kami buat, kami juga memiliki halaman admin yang dapat mengubah dan mengakses database. Fungsi utama halaman admin tersebut adalah melakukan perbaikan data jika diperlukan.

1. **Pengamatan / Observasi**

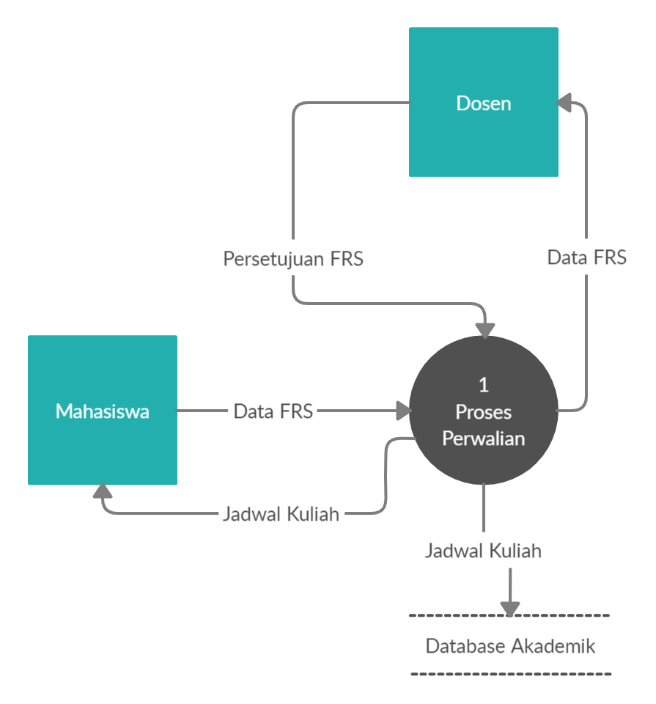
Observasi dilakukan berdasarkan pengalaman kami sebagai mahasiswa yang melakukan kegiatan akademik pada website SIM. Pengamatan dilakukan untuk mengetahui alur dari kegiatan kademik. Salah satu contoh dari kegiatan akademik yang dilakukan di website SIM adalah perwalian. Perwalian pada iSTTS saat ini dilakukan secara online melalui website SIM. Pada saat perwalian, mahasiswa dapat memilih mata kuliah yang ingin diambil memalui FRS. Setelah mahasiswa sudah menyelesaikan FRS, data tersebut akan dikirim kepada dosen wali mahasiswa. Data FRS tersbut harus disetujui sehingga jadwal dari mata kuliah yang diambil mahasiswa dapat dibuat. Jika jadwal kuliah sudah dibuat, maka mahasiwa dapat melihat jadwal. Jadwal ini juga berlaku sebagai jadwal ujian.

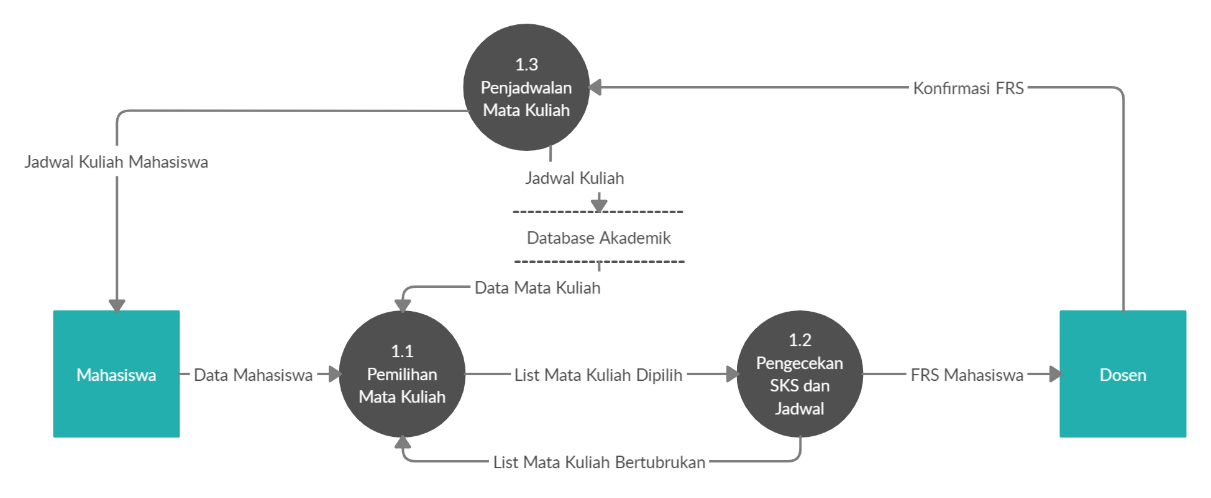
**3.2 Analisa Sistem**

Berdasarkan pengamatan dan wawancara yang dilakukan sebagai mahasiswa iSTTS, sistem perwalian, batal tambah, dan penjadwalan dilakukan secara online dengan menggunakan website SIM iSTTS. Dengan beberapa ketentuan dilakukan secara tatap muka seperti pada konfirmasi pengisian FRS dan konfirmasi batal tambah mata kuliah.

Tata urutan proses perwalian untuk mendapatkan data jadwal pelajaran selama satu semester :

1. Mahasiswa dating ke BAA untuk rmemenuhi seluruh kekebutuhan data maupun pembayaran yang belum terpenuhi.
2. Jika seluruh persyaratan terpenuhi, mahasiswa dapat mengisi FRS melalui menu rencana studi pada website SIM.
3. Mahasiswa mengisi mata kuliah yang ingin diambil sesuai batas SKS maksimal yang dimiliki mahasiswa.
4. Mahasiswa datang kepada dosen wali untuk meminta persetujuan.
5. Jadwal yang sudah disetujui akan disimpan dan menjadi jadwal ujian.

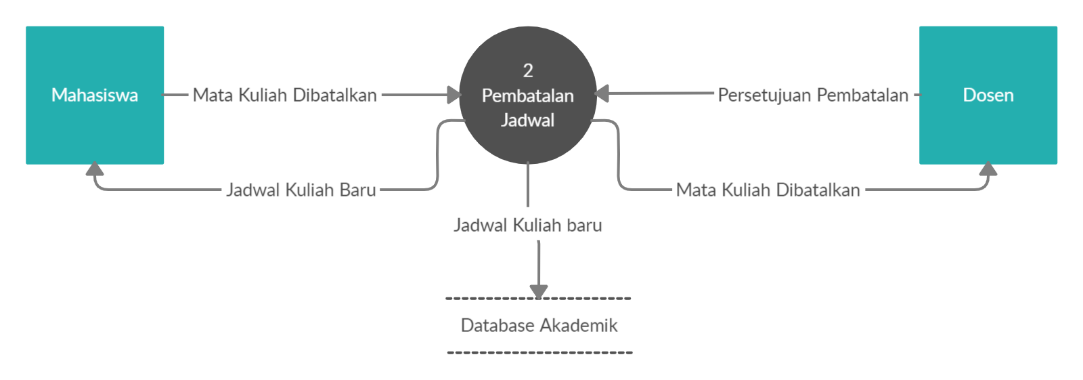
****

****

**Gambar 3.1** Context Diagram Perwalian

Jika terdapat mata kuliah yang kurang sesuai dengan kebutuhan mahasiswa, dapat dilakukan pembatalan mata kuliah. Pembatalan memiliki waktu terbatas. Tata cara pembatalan mata kuliah dilakukan sebagai berikut:

1. Mahasiswa memilih menu rencana studi pada website SIM. Kemudian mahasiswa memilih pembatalan mata kuliah.
2. Kemudian, mahasiswa memilih mata kuliah yang ingin dibatalkan.
3. Setelah mengisi memilih mata kuliah yang ingin dibatalkan, mahasiswa dapat mendatangi, dosen wali untuk mengkonfirmasi pembatalan.
4. Jika pembatalan telah diterima oleh dosen wali, maka jadwal mahasiswa akan diperbaharui dengan mnghilangkan mata kuliah yang telah dibatalkan.

****

**Gambar 3.2** Context Diagram Batal Tambah

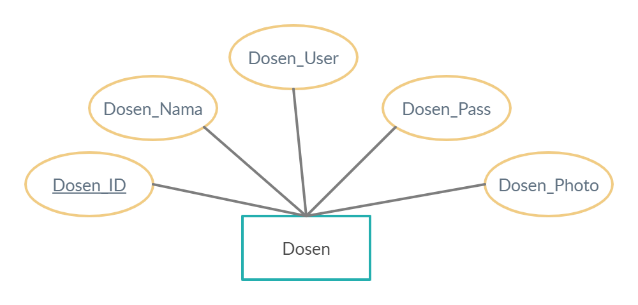
**3.3 Desain Database**

**3.3.1 Entiti**

Data pada SIM dapat dikelompokan sebagai beberapa enititas atau entiti yang dapat saling berhubungan satu dengan lain. Berikut adalah data entiti dan attribute yang dimiliki setiap entiti dari website SIM.

1. Dosen

Entiti dosen memliki beberapa atribut, diantaranya Dosen\_ID, Dosen\_Nama, Dosen\_User, Dosen\_Pass, dan Dosen\_Photo.

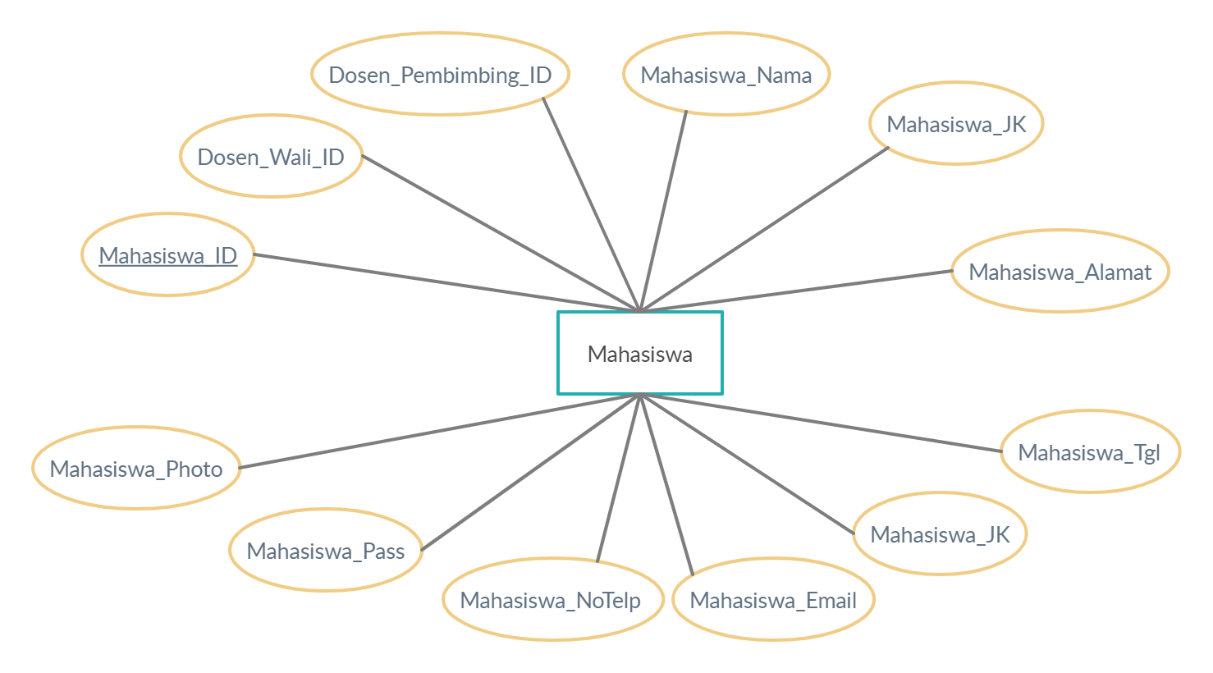


**Gambar 3.3** Entiti Dosen

Data Dosen\_ID, Dosen\_Nama, Dosen\_User dan Dosen\_Pass harus diisi ke dalam entiti sebagai informasi data dosen.

1. Mahasiswa

Entiti mahasiswa memliki beberapa atribut, diantaranya Mahasiswa\_ID, Dosen\_Wali\_ID, Dosen\_Pembimbing\_ID, Mahasiswa\_Nama, Mahasiswa\_JK, Mahasiswa\_Alamat, Mahasiswa\_NoTelp, Mahasiswa\_Pass, dan Mahasiswa\_Photo.

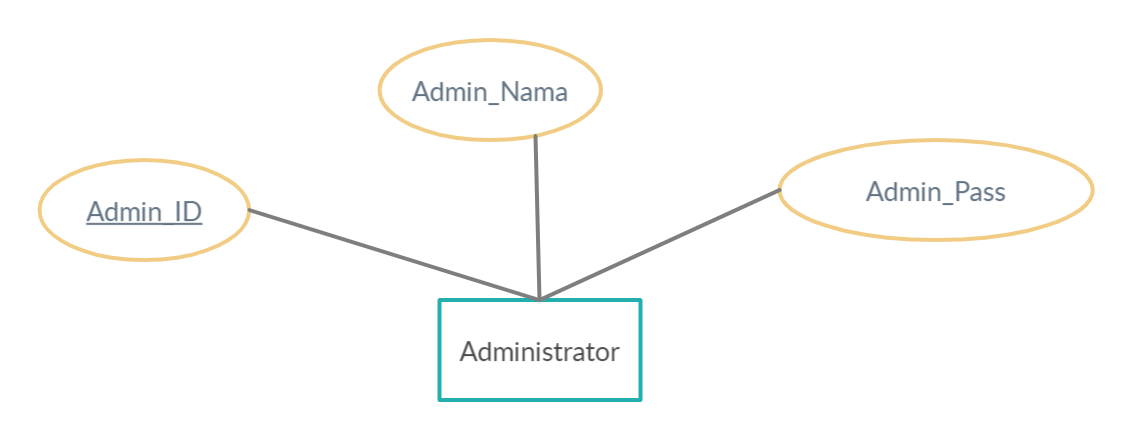


**Gambar 3.4** Entiti Mahasiswa

Data Mahasiswa\_ID, Dosen\_Wali\_ID, Mahasiswa\_Nama, Mahasiswa\_JK, Mahasiswa\_Alamat, Mahasiswa\_NoTelp, dan Mahasiswa\_Pass harus diisi ke dalam entity sebagai informasi data mahasiswa. Data Dosen\_Pembimbing\_ID akan diisi saat mahasiswa dalam proses tugas akhir.

1. Administrator

Entiti administrator memliki beberapa atribut, diantaranya Admin\_ID, Admin\_Nama, dan Admin\_Pass.

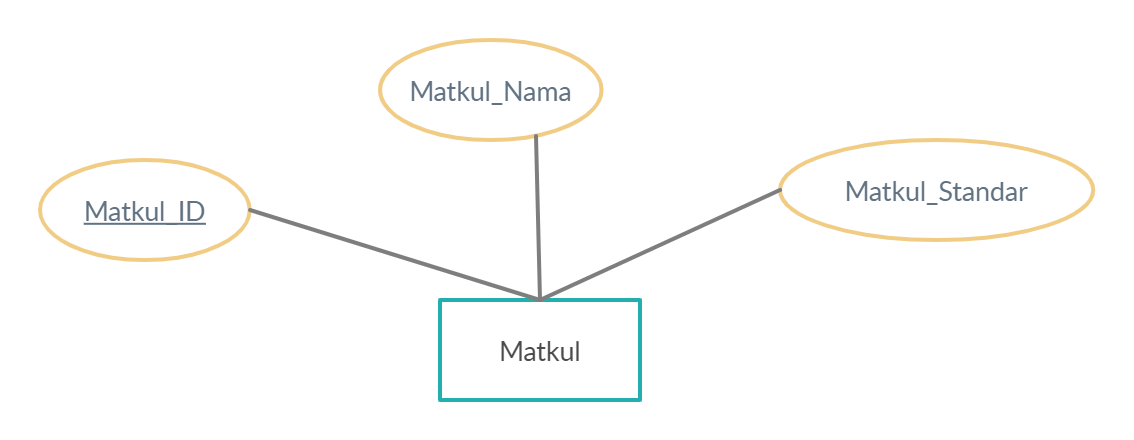


**Gambar 3.5** Entiti Administrator

Data Admin\_ID, Admin\_Nama, dan Admin\_Pass harus diisi ke dalam entity sebagai informasi data administrator.

1. Mata Kuliah

Entity Matkul memiliki beberapa atribut, diantaranya Matkul\_ID, Matkul\_Nama, dan Matkul\_Standar.

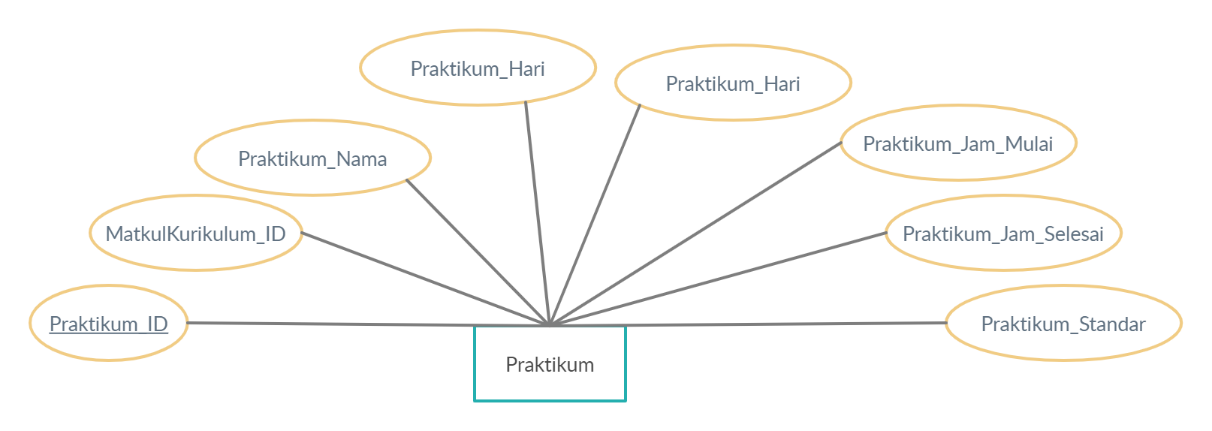


**Gambar 3.6** Entiti Matkul

Data Matkul\_ID, Matkul\_Nama, dan Matkul\_Standar harus diisi ke dalam entity sebagai informasi data mata kuliah.

1. Praktikum

Entity praktikum memiliki beberapa atribut, diantaranya Praktikum\_ID, Matkulkurikulum\_ID, Praktikum\_Nama, Praktikum\_Hari, Praktikum\_Ruangan, Praktikum\_Jam\_Mulai, Praktikum\_Jam\_Selesai, Praktikum\_Kapasitas, dan Praktikum\_Standat.

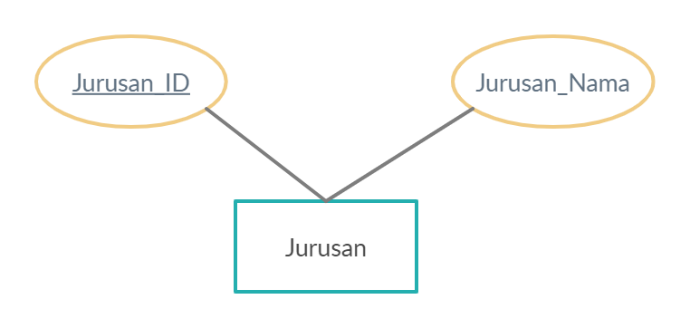


**Gambar 3.7** Entiti Praktikum

Data Praktikum\_ID, Matkulkurikulum\_ID, Praktikum\_Nama, Praktikum\_Hari, Praktikum\_Ruangan, Praktikum\_Jam\_Mulai, Praktikum\_Jam\_Selesai, Praktikum\_Kapasitas, dan Praktikum\_Standat.harus diisi ke dalam entity sebagai informasi data praktikum. Praktikum mengikuti suatu mata kuliah, namun tidak semua mata kuliah memiliki praktikum.

1. Jurusan

Entity jurusan memiliki beberapa atribut, diantaranya Jurusan\_ID, dan Jurusan\_Nama.

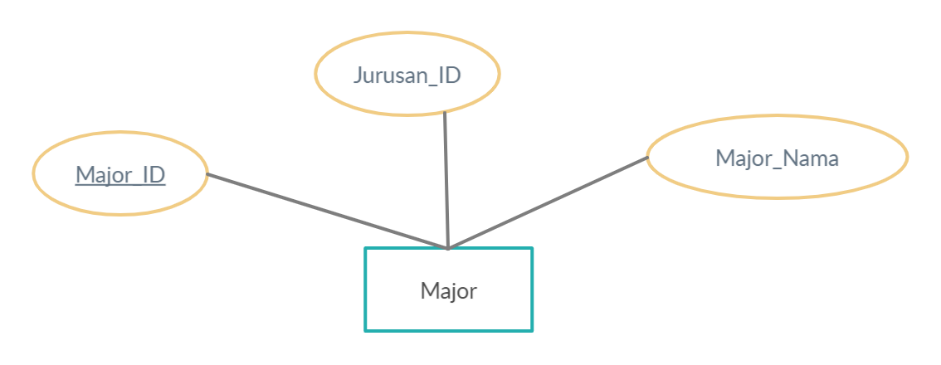


**Gambar 3.8** Entiti Jurusan

Data Jurusan\_ID, dan Jurusan\_Nama harus diisi ke dalam entity sebagai informasi data mata kuliah.

1. Major

Entity major memiliki beberapa atribut, diantaranya Major\_ID, Jurusan\_ID dan Major\_Nama.

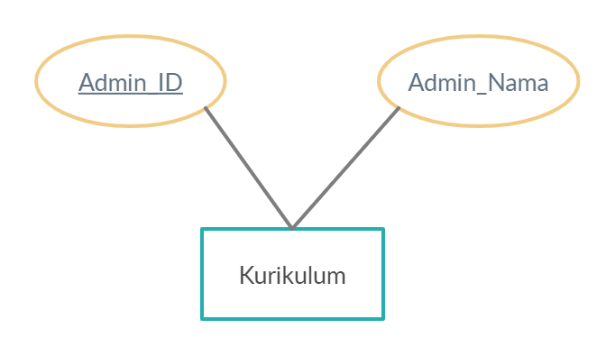


**Gambar 3.9** Entiti Major

Data Major\_ID, Jurusan\_ID dan Major\_Nama harus diisi ke dalam entity sebagai informasi data mata kuliah. Major mengikuti suatu jurusan, walalupun jurusan tersebut dapat memiliki beberapa major.

1. Kurikulum

Entity kurikulum memiliki beberapa atribut, diantaranya Kurikumum\_ID, dan Kurikulum\_Nama.

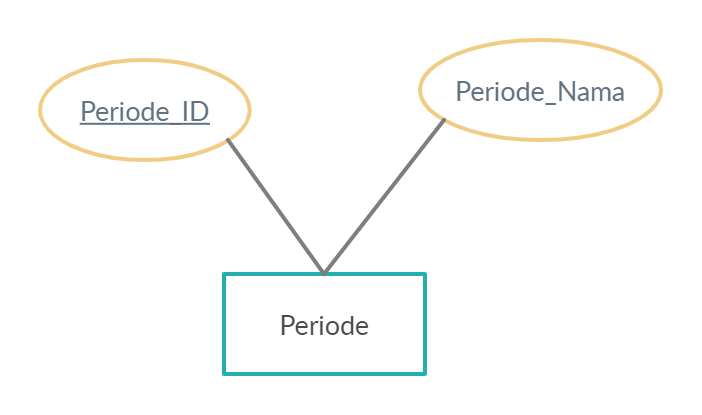


**Gambar 3.10** Entiti Kurikulum

Data Kurikumum\_ID, dan Kurikulum\_Nama harus diisi ke dalam entity sebagai informasi data kurikulum.

1. Periode

Entity periode memiliki beberapa atribut, diantaranya Periode\_ID, dan Periode\_Nama.

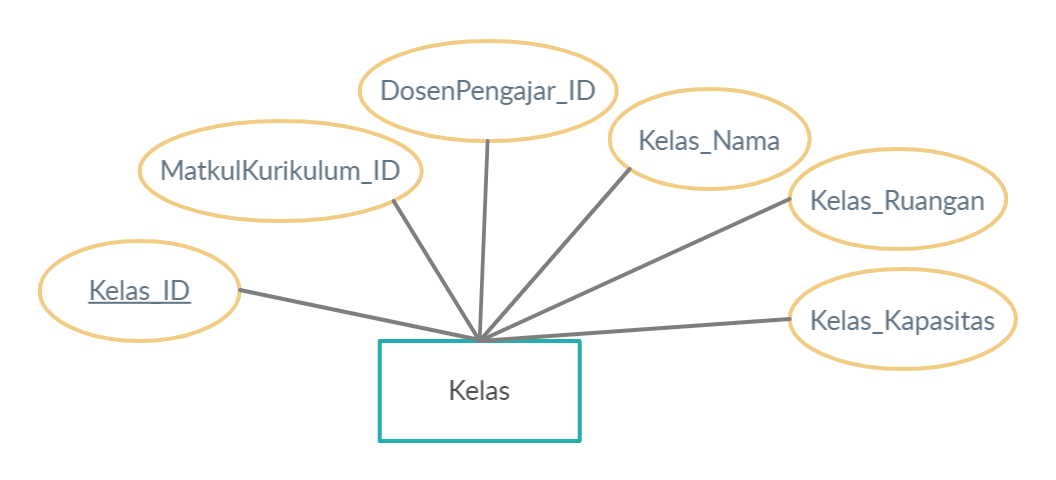


**Gambar 3.11** Entiti Periode

Data Periode\_ID, dan Periode\_Nama harus diisi ke dalam entity sebagai informasi data periode.

1. Kelas

Entity Kelas memiliki beberapa atribut, diantaranya Kelas\_ID, MatkulKurikulum\_ID, DosenPengajar\_ID, Kelas\_Nama, Kelas\_Ruangan, dan Kelas\_Kapasitas.

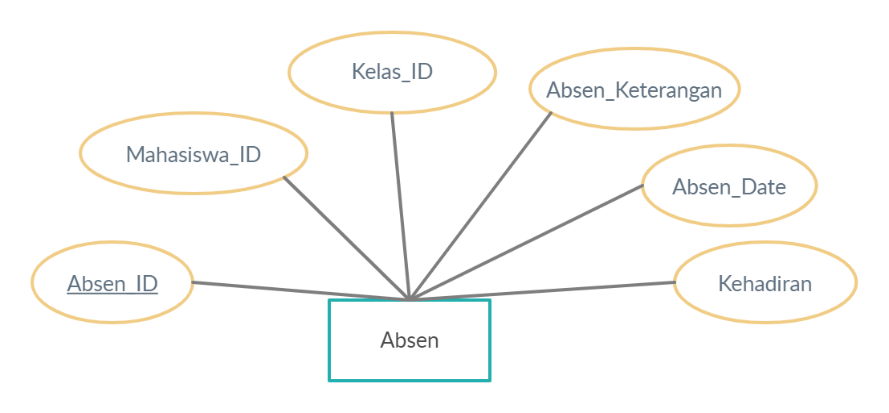


**Gambar 3.12** Entiti Kelas

Data Kelas\_ID, MatkulKurikulum\_ID, DosenPengajar\_ID, Kelas\_Nama, Kelas\_Ruangan, dan Kelas\_Kapasitas harus diisi ke dalam entity sebagai informasi data kelas.

1. Absen

Entity Absen memiliki beberapa atribut, diantaranya Absen\_ID, Mahasiswa\_ID, Kelas\_ID, Absen\_Keterangan, Absen\_Date, dan Kehadiran.

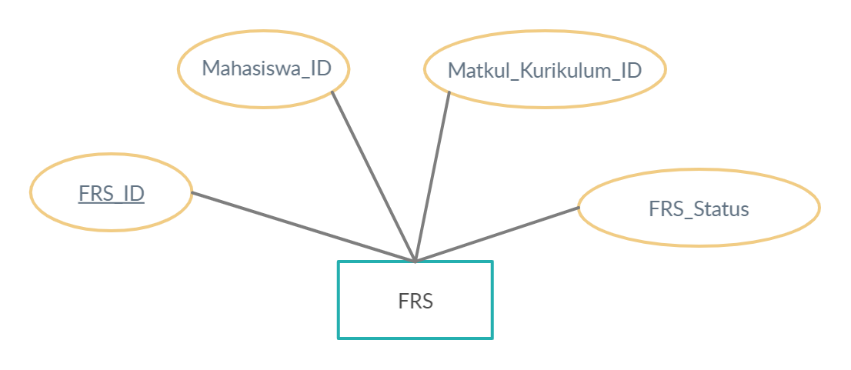


**Gambar 3.13** Entiti Absen

Data Absen\_ID, Mahasiswa\_ID, Kelas\_ID, Absen\_Keterangan, Absen\_Date, dan Kehadiran harus diisi ke dalam entity sebagai informasi data absen.

1. FRS

Entity FRS memiliki beberapa atribut, diantaranya FRS\_ID, Mahasiswa\_ID, Matkul\_Kurikulum\_ID dan FRS\_Status.

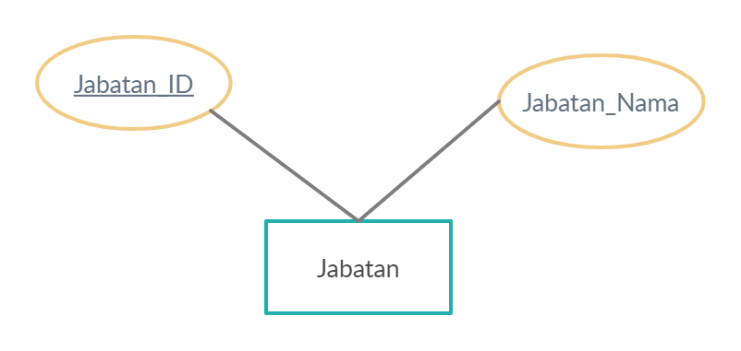


**Gambar 3.14** Entiti FRS

Data FRS\_ID, Mahasiswa\_ID, Matkul\_Kurikulum\_ID dan FRS\_Status harus diisi ke dalam entity sebagai informasi FRS.

1. Jabatan

Entity Jabatan memiliki 2 atribut, yaitu Jabatan\_ID dan Nama\_Jabatan.

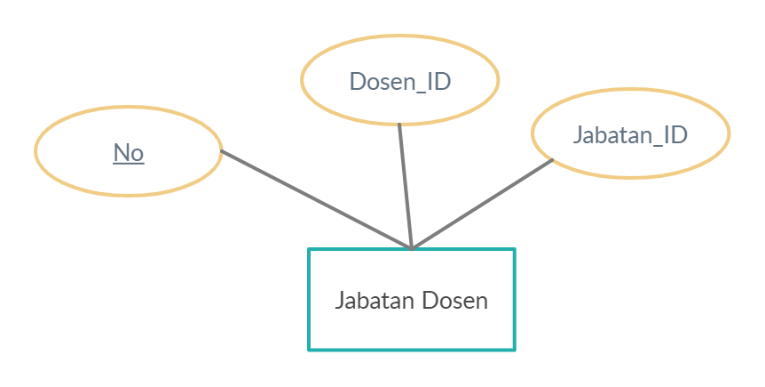


**Gambar 3.15** Entiti Jabatan

Data Jabatan\_ID dan Nama\_Jabatan.harus diisi ke dalam entity sebagai informasi Jabatan.

1. Jabatan Dosen

Entity Jabatan Dosen memiliki beberapa atribut, diantaranya No, Dosen\_ID, dan Jabatan\_ID.

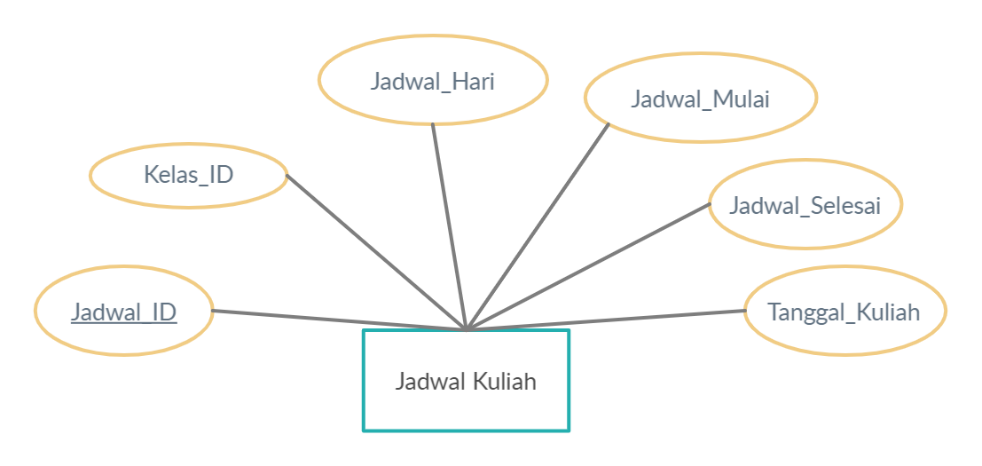


**Gambar 3.16** Entiti Jabatan Dosen

Data Dosen\_ID dan Jabatan\_ID harus diisi ke dalam entity sebagai informasi Jabatan Dosen.

1. Jadwal Kuliah

Entity Jadwal Kuliah memiliki beberapa atribut, diantaranya Jadwal\_ID, Kelas\_ID, Jadwal\_Hari, Jadwal\_Mulai, Jadwal\_Selesai dan Tanggal\_Kuliah.

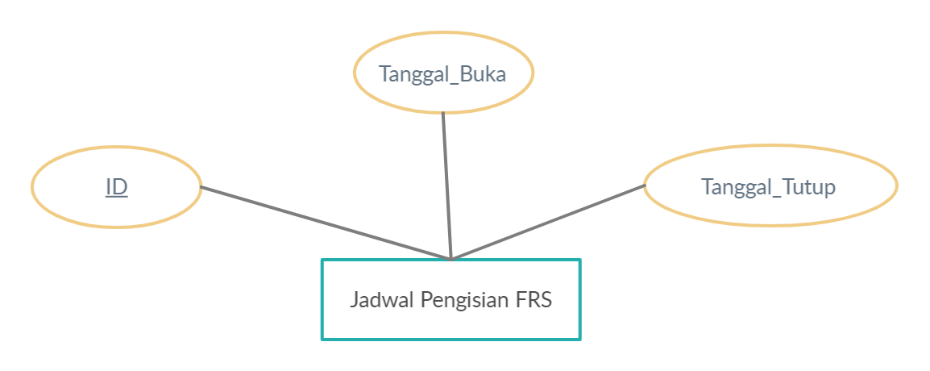


**Gambar 3.17** Entiti Jadwal Kuliah

Data Jadwal\_ID, Kelas\_ID, Jadwal\_Hari, Jadwal\_Mulai, Jadwal\_Selesai dan Tanggal\_Kuliah harus diisi ke dalam entiti sebagai informasi Jadwal Kuliah.

1. Jadwal Pengisian FRS

Entity Jadwal Pengisian FRS memiliki beberapa atribut, diantaranya ID, Tanggal\_Buka dan Tanggal\_Tutup.

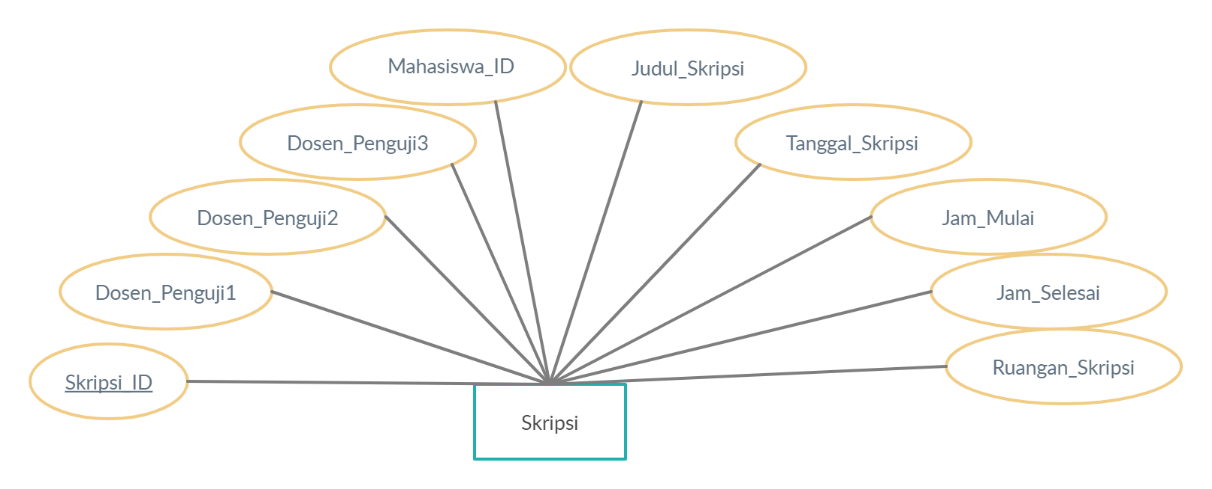


**Gambar 3.18** Entiti Jadwal Pengisian FRS

Data ID, Tanggal\_Buka dan Tanggal\_Tutup harus diisi ke dalam entiti sebagai informasi Jadwal Pengisian FRS. Jadwal ini menentukan batas akhir pengumpulan form FRS bagi mahasiswa.

1. Skripsi

Entity Skripsi memiliki beberapa atribut, diantaranya Skripsi\_ID, Dosen\_Penguji1, Dosen\_Penguji2, Dosen\_Penguji3, Mahasiswa\_ID, Judul\_Skripsi, Tanggal\_Skripsi, Jam\_Mulai, Jam\_Selesai dan Ruangan Skripsi.

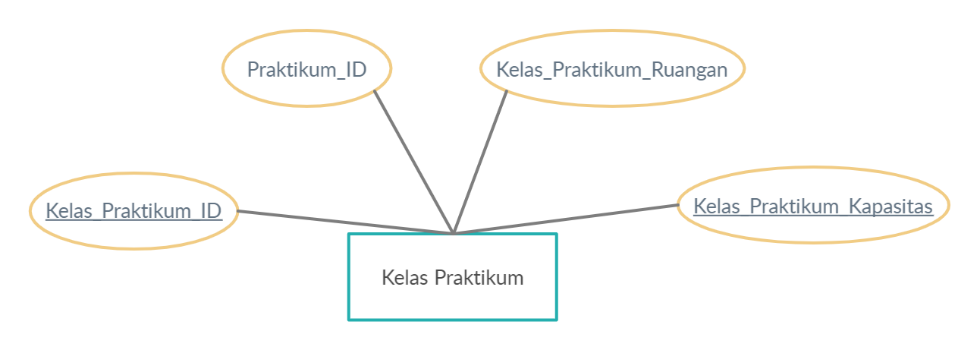


**Gambar 3.19** Entiti Skripsi

Data Skripsi\_ID, Dosen\_Penguji1, Dosen\_Penguji2, Dosen\_Penguji3, Mahasiswa\_ID, Judul\_Skripsi, Tanggal\_Skripsi, Jam\_Mulai, Jam\_Selesai dan Ruangan Skripsi harus diisi ke dalam entiti sebagai informasi Skripsi.

1. Kelas Praktikum

Entity Kelas Praktikum memiliki beberapa atribut, diantaranya Kelas\_Praktikum\_ID, Praktikum\_ID, Kelas\_Praktikum\_Ruangan dan Kelas\_Praktikum\_Kapasitas.

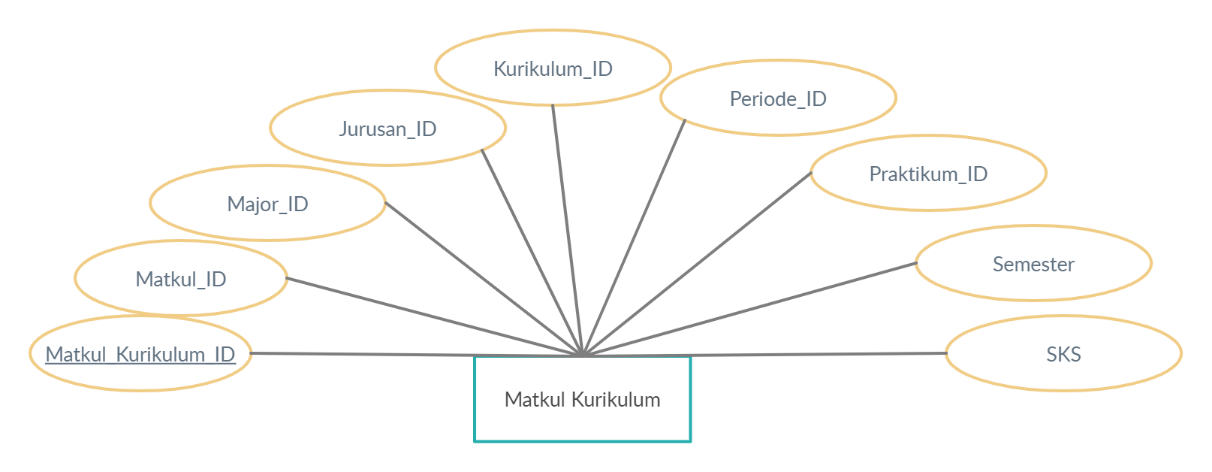


**Gambar 3.20** Entiti Kelas Praktikum

Data Kelas\_Praktikum\_ID, Praktikum\_ID, Kelas\_Praktikum\_Ruangan dan Kelas\_Praktikum\_Kapasitas harus diisi ke dalam entiti sebagai informasi Kelas Praktikum.

1. Matkul Kurikulum

Entity Matkul Kurikulum memiliki beberapa atribut, diantaranya Matkul\_Kurikulum\_ID, Matkul\_ID, Major\_ID, Jurusan\_ID, Kurikulum\_ID, Periode\_ID, Praktikum\_ID, Semester dan SKS.

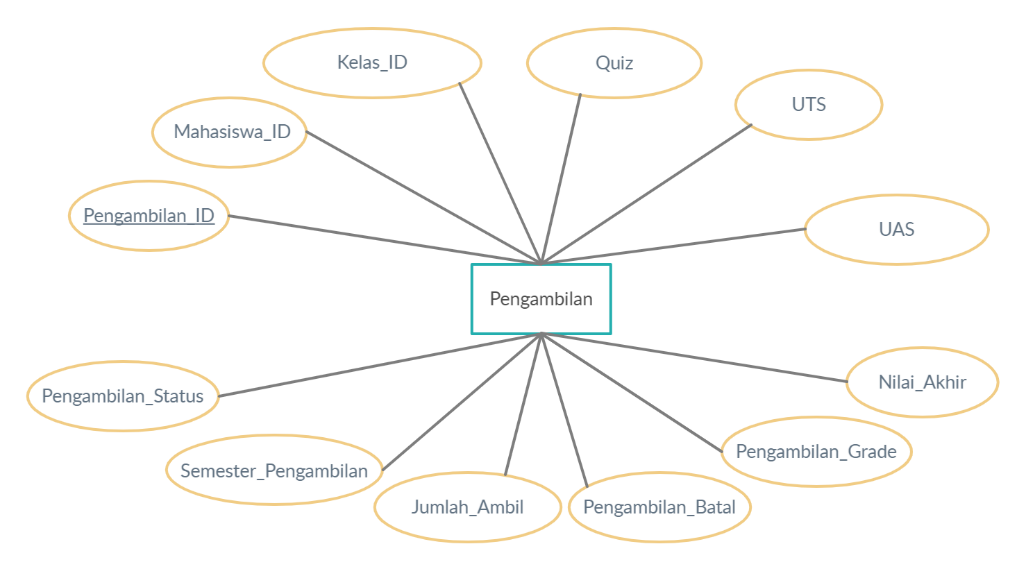


**Gambar 3.21** Entiti Matkul Kurikulum

Data Matkul\_Kurikulum\_ID, Matkul\_ID, Major\_ID, Jurusan\_ID, Kurikulum\_ID, Periode\_ID, Praktikum\_ID, Semester dan SKS harus diisi ke dalam entiti sebagai informasi Matkul Praktikum.

1. Pengambilan

Entity Pengambilan memiliki beberapa atribut, diantaranya Pengambilan\_ID, Mahasiswa\_ID, Kelas\_ID, Quiz, UTS, UAS, Nilai\_Akhir, Pengambilan\_Grade, Pengambilan\_Batal, Jumlah\_Ambil, Semester\_Pengambilan dan juga Pengambilan\_Status.

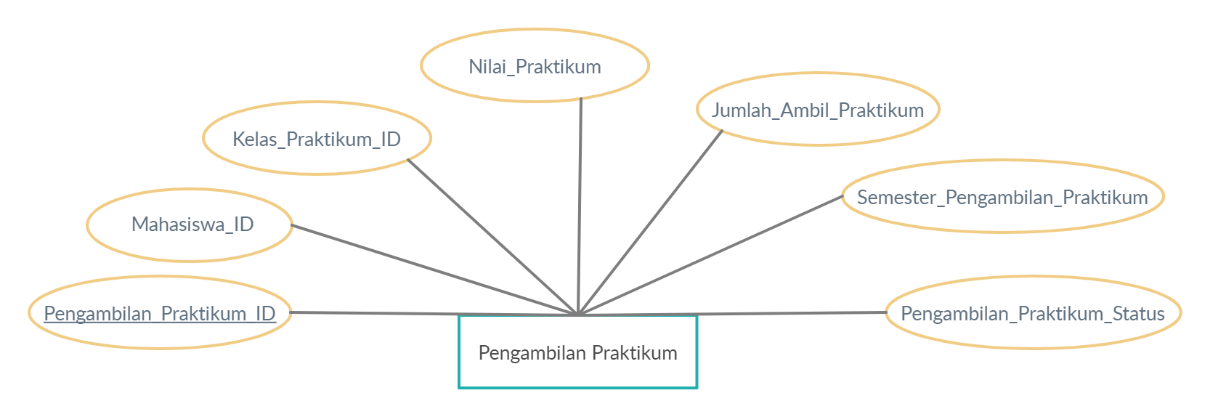


**Gambar 3.22** Entiti Pengambilan

Data Pengambilan\_ID, Mahasiswa\_ID, Kelas\_ID, Quiz, UTS, UAS, Nilai\_Akhir, Pengambilan\_Grade, Pengambilan\_Batal, Jumlah\_Ambil, Semester\_Pengambilan dan juga Pengambilan\_Status harus diisi ke dalam entiti sebagai informasi Pengambilan. Data pengambilan inilah yang dapat digunakan menjadi list mata kuliah yang diambil mahasiswa. Hal ini dapat dikembangkan menjadi jadwal mata kuliah mahasiswa dan laporan nilai mahasiswa.

1. Pengambilan Praktikum

Entity Pengambilan Praktikum memiliki beberapa atribut, diantaranya adalah Pengambilan\_Praktikum\_ID, Mahasiswa\_ID, Kelas\_Praktikum\_ID, Nilai\_Praktikum, Jumlah\_Ambil\_Praktikum, Semester\_Pengambilan\_Praktikum dan Pengambilan\_Praktikum\_Status.

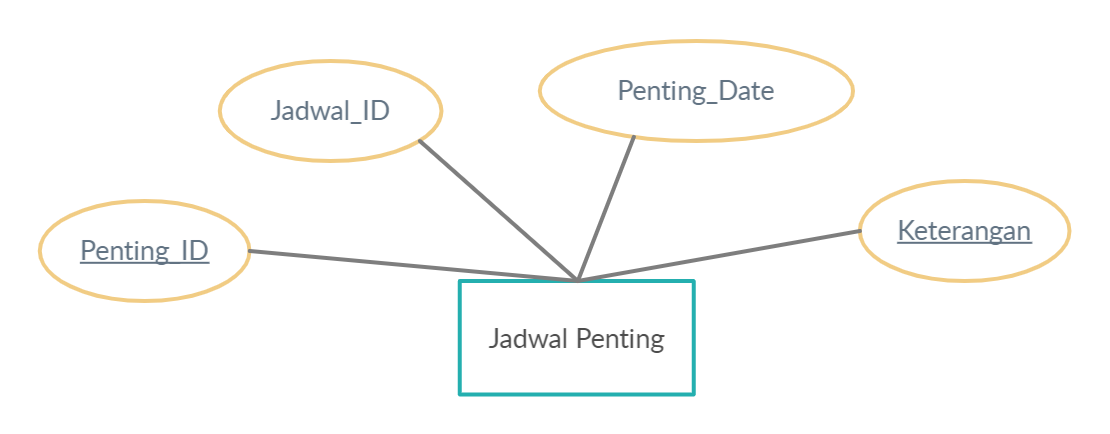


**Gambar 3.23** Entiti Pengambilan Praktikum

Data Pengambilan\_Praktikum\_ID, Mahasiswa\_ID, Kelas\_Praktikum\_ID, Nilai\_Praktikum, Jumlah\_Ambil\_Praktikum, Semester\_Pengambilan\_Praktikum dan Pengambilan\_Praktikum\_Status harus diisi ke dalam entiti sebagai informasi Pengambilan Praktikum.

1. Jadwal Penting

Entity Jadwal Penting memiliki beberapa atribut, diantaranya Penting\_ID, Jadwal\_ID, Penting\_Date dan Keterangan.



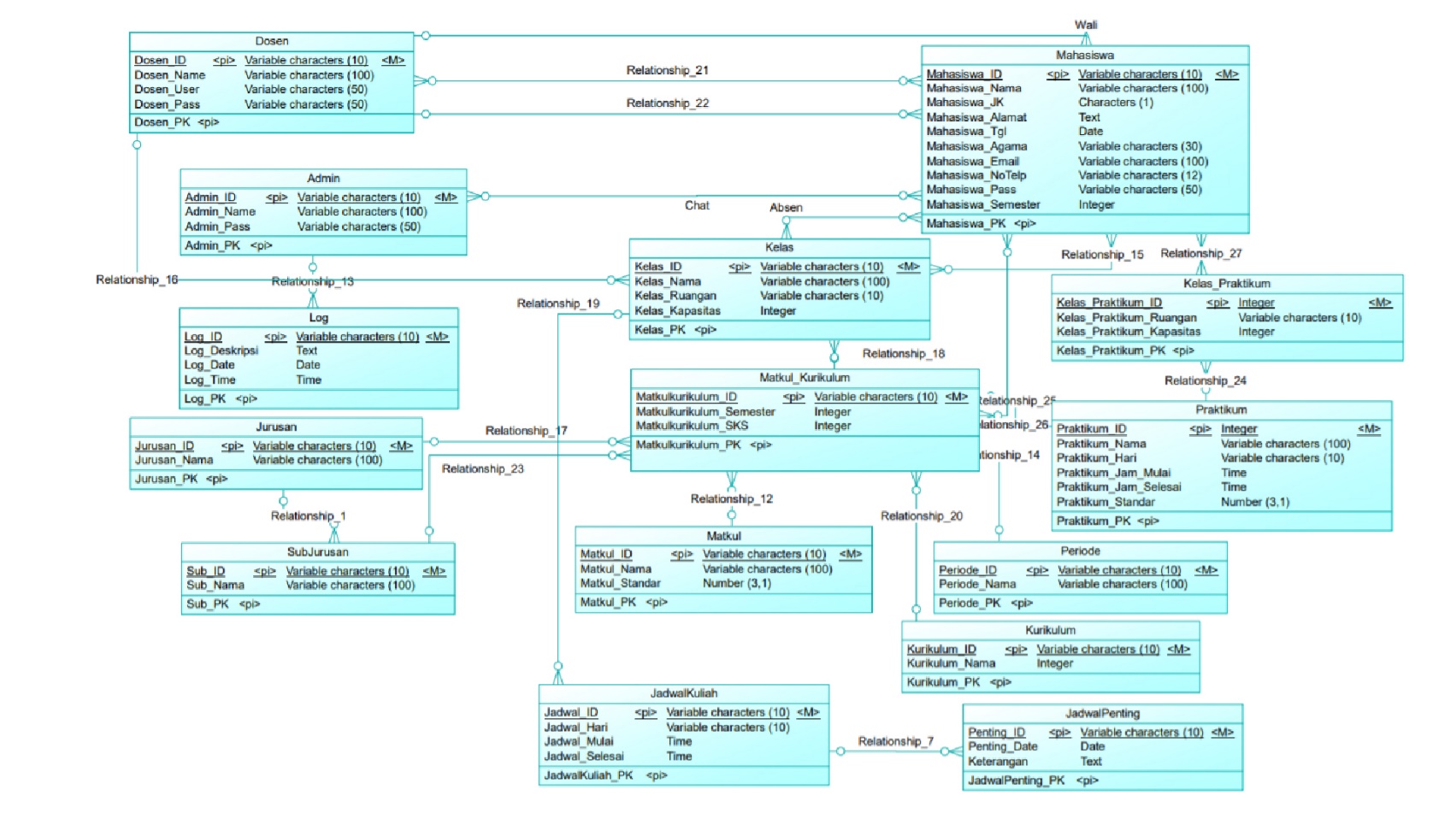
**Gambar 3.24** Entiti Jadwal Penting

Data Penting\_ID, Jadwal\_ID, Penting\_Date dan Keterangan harus diisi ke dalam entiti sebagai informasi Jadwal Penting. Jadwal penting digunakan sebagai penanda kegiatan penting pada proses pembelajaran seperti Quiz pada sebuah mata kuliah.

**3.3.2 Conceptual Data Model**

Conceptual Data Model (CDM) digunakan untuk memetakan hubungan antar entity. Seperti Namanya, CDM dibuat sesuai konsep pemikiran dari pengembang data, mulai dari entity apa saja yang ada, apakah entity tertentu memiliki hubungan dengan entity lainnya, dan bagaimana bentuk hubungan entity tersebut.

Pada database SIM, dapat dilihat bahwa bentuk hubungan dapat berbeda antar entity. Seperti 1 to many pada entity dosen dan kelas, yang mana seorang dosen dapat mengajar banyak kelas, namun sebuah kelas hanya memiliki 1 dosen pengajar. Relasi lain seperti many to many juga dapat ditemukan pada hubungan antara entity kelas dan entity mahasiswa. Dengan maksud seorang mahasiswa dapat mengikuti banyak kelas, dan satu kelas dapat menampung banyak mahasiswa.

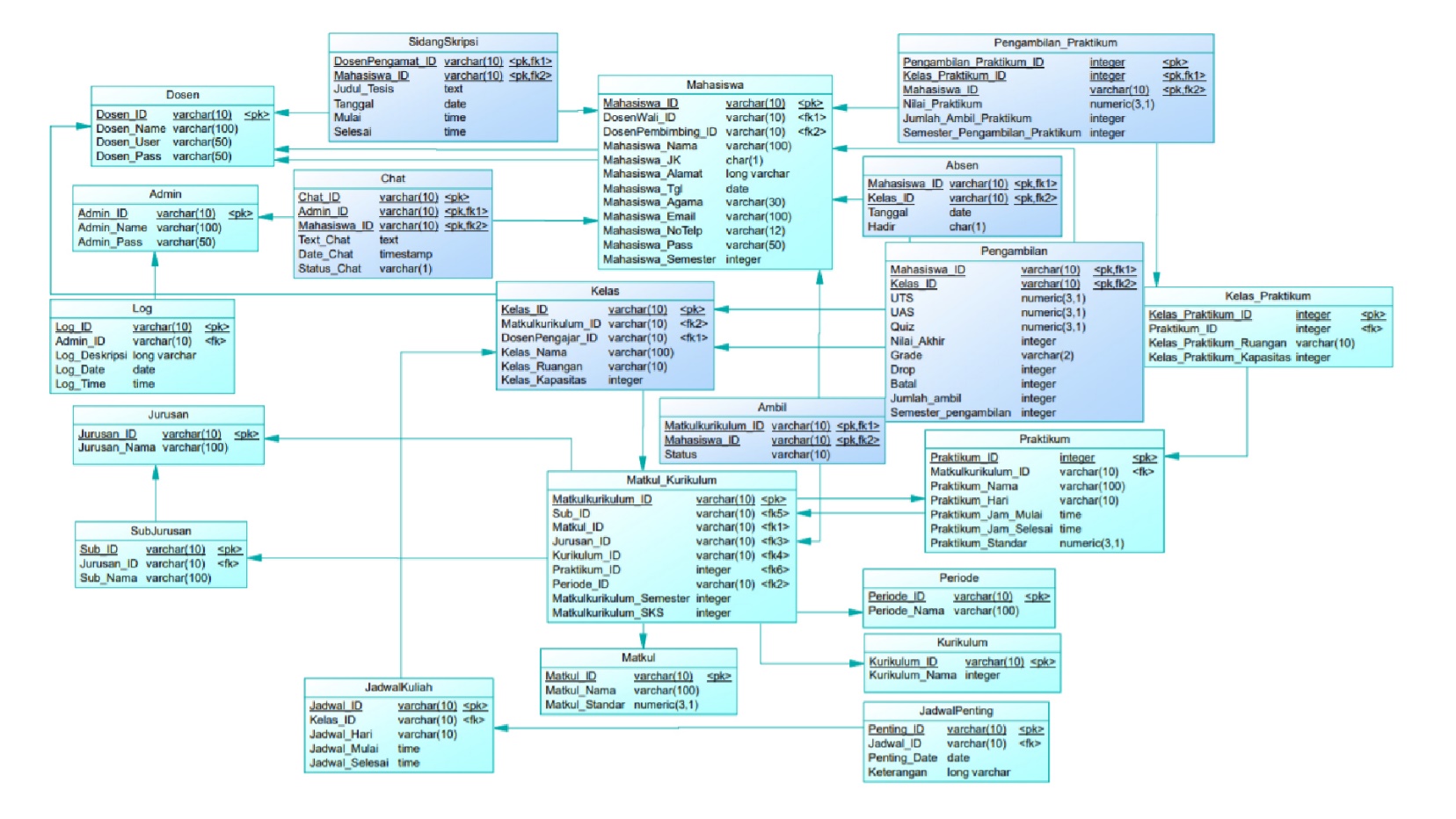


**Gambar 3.25** Conceptual Data Model Website SIM

**3.3.3 Physical Data Model**

Physical Data Model (PDM) digunakan untuk menggambarkan model data pada database base sesungguhnya di dunia nyata. Hal yang paling menonjol pada PDM adalah adanya sebuah table tambahan yang muncul karena hubungan many to many dari 2 buah entity. Tabel tersebut memetakan anggota apa saja dari sebuah entity yang memiliki hubungan dengan anggota lain pada entity yang berbeda.

Pada website SIM ini, salah satu contoh tabel baru yang muncul karena hubungan many to many adalah seperti tabel pengambilan yang menghubungkan tabel kelas dan mahasiswa. Tabel pengambilan tersebut mencatat kelas apa saja yang diambil oleh setiap mahasiswa.



**Gambar 3.26** Physical Data Model Website SIM

**BAB IV**

**IMPLEMENTASI SISTEM**

**4.1 Implementasi**

Implementasi sistem adalah sebuah tahap penerapan dari rancangan sistem yang telah dibuat. Pada bagian ini akan dibahas implementasi inti dari perancangan Sistem Informasi Mahasiswa berbasis Web. Dimulai dari rancangan sistem, relasi tabel, rancangan basis data dan alur langkah system.

Website Sistem Informasi Mahasiswa ini menggunakan Bahasa pemrograman Php dengan framework Materialize untuk membantu tampilan front-end dan XAMPP sebagai alat untuk membantu local server development.

**4.1.1 Penghubung Database**

Untuk menghubungkan database, dibuat sebuah file dengan nama Connection.php yang memberikan variable $conn kepada seluruh file lain dengan tujuan menerima dan mengambil maupun mngubah data pada database. Database yang digunakan adalah MySQL yang disimpan pada sebuah situs hosting.

<?php

    $con\_servername = "185.232.14.1";

    $con\_username = "u855625606\_ProjectSDP";

    $con\_password = "ProjectSDP2020";

    $con\_database = "u855625606\_ProjectSDP";

    $conn = new mysqli($con\_servername, $con\_username, $con\_password, $con\_database);

    if ($conn->connect\_error) {

      die("Failed: " . $conn->connect\_error);

    }

    $con\_procedural = mysqli\_connect($con\_servername, $con\_username, $con\_password, $con\_database);

    if (!$con\_procedural) {

       die("Failed : ".mysqli\_connect\_error());

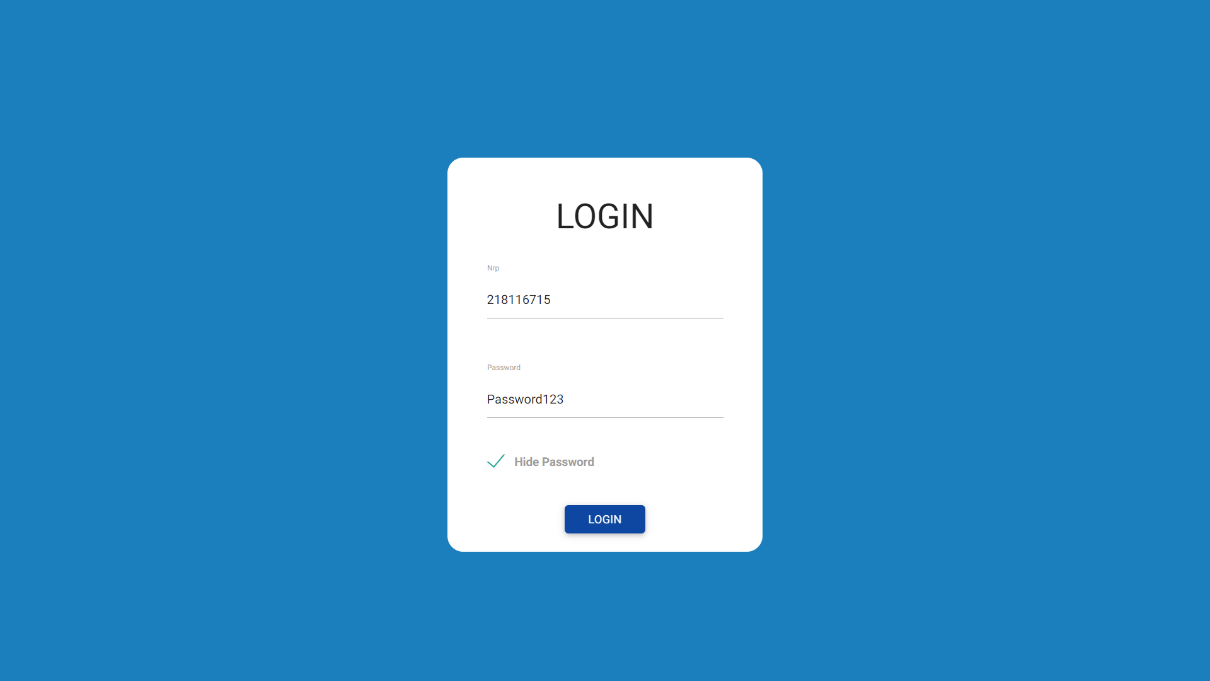
    }

?>

**Segmen Program 4.1** Koneksi Database

**4.1.2 Halaman Login**

Seluruh user akan diarahkan menuju halaman login jika berusaha mengunjungi halaman lain tanpa login atau tidak memiliki akses untuk halaman tersebut. Pada halam login terdapat 2 input yang dapat diisi user, yaitu NRP dan password. Untuk melakukan proses login, terdapat button login yang dapat ditekan setelah mengisi 2 input tersebut. Untuk mengankan password, maka input pada password akan berformat ‘\*’, yang dapat ditampilkan jika user mencentang checkbox show password.



**Gambar 4.1** Halaman Login

**4.1.3 Proses Login**

Setelah user menekan tombol login, jika terdapat input yang kosong, maka akan muncul pesan error yang mengingatkan user tentang input yang kosong. Jika seluruh input telah diisi, akan dilakukan pengecekan kombinasi NRP dan password pada database. Jika terdapat data yang sesuai, maka user akan dialihkan menuju halaman home mahasiswa. Namun jika data yang ada sesuai dengan username dan password dosen, user akan dialihkan menuju halaman home dosen. Jika data tersebut sesuai dengan data admin pada database, maka akan dilanjutakan menuju halaman home admin. Sebalikanya jika tidak terdapat data yang sesuai, maka proses login dianggap gagal dan dimunculkan pesan error.

<?php

if(isset($\_POST['btnLogin'])){

$user = $\_POST['user']; $pass = $\_POST['pass'];

$cekAdmin = false;$cekDosen = false;$cekMahasiswa = false;

$query = "SELECT \* FROM Administrator";

$listadmin = $conn->query($query);

foreach ($listadmin as $key => $value) {

if($user == $value['Admin\_ID'] && $pass == $value['Admin\_Pass']){

$\_SESSION['user']['user'] = $value['Admin\_ID'];

$\_SESSION['user']['name'] = $value['Admin\_Nama'];

$\_SESSION['user']['pass'] = $value['Admin\_Pass'];

$cekAdmin = true; }}

$query = "SELECT \* FROM Mahasiswa";

$listmahasiswa = $conn->query($query);

foreach ($listmahasiswa as $key => $value) {

if($user == $value['Mahasiswa\_ID'] && $pass == $value['Mahasiswa\_Pass']){

$\_SESSION['user']['user']=$value['Mahasiswa\_ID'];

$\_SESSION['user']['name']=$value['Mahasiswa\_Nama'];

$\_SESSION['user']['pass']=$value['Mahasiswa\_Pass'];

$\_SESSION['user']['photo']=

$value['Mahasiswa\_Photo'];

$cekMahasiswa = true; }}

$query = "SELECT \* FROM Dosen";

$listdosen = $conn->query($query);

foreach ($listdosen as $key => $value) {

if($user == $value['Dosen\_User'] && $pass == $value['Dosen\_Pass']){

$\_SESSION['user']['user'] = $value['Dosen\_ID'];

$\_SESSION['user']['name'] = $value['Dosen\_Nama'];

$\_SESSION['user']['pass'] = $value['Dosen\_Pass'];

$\_SESSION['user']['photo'] = $value['Dosen\_Photo'];

$cekDosen = true; }

if($cekAdmin){

header("location: Admin/Admin.php");

}else{

if($cekMahasiswa){

header("location: Mahasiswa/Home.php");

}else{

if($cekDosen){

header("location: Dosen/Home.php");

}else{

echo "<label style='color: red;'>

NRP / Password yang anda masukkan salah

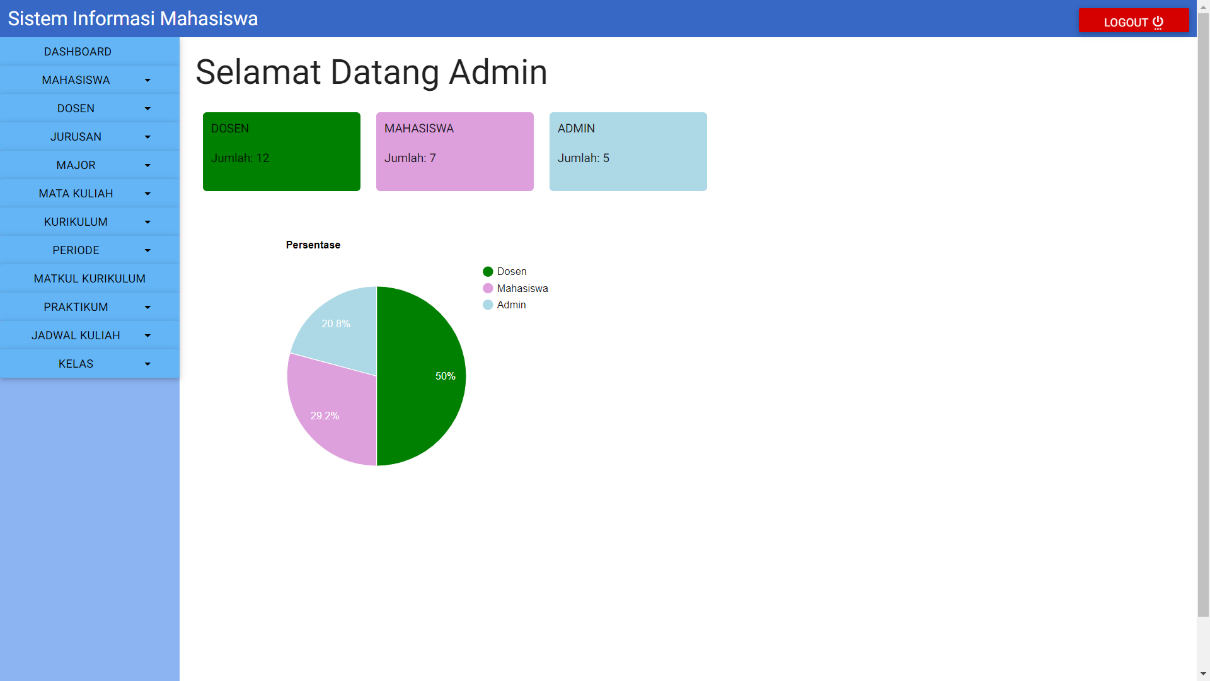
</label>";}}}}

?>

**Segmen Program 4.2** Pengecekan Login

**4.1.4 Halaman Home Admin**

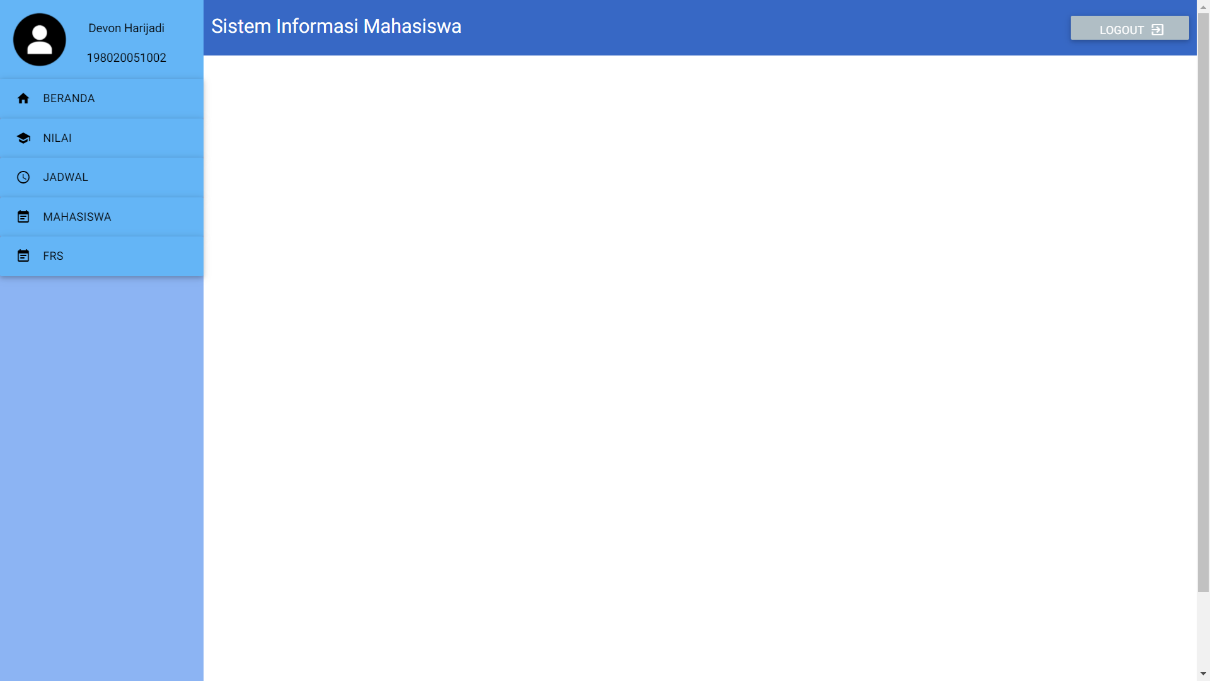
Halaman home admin digunakan sebagai halaman navigasi menuju halaman insert, edit, delete dan informasi pada database. Terdapat navigation bar pada bagian kiri dari halaman admin dapat ditekan sesuai kebutuhan informasi yang ingin ditampilkan, seperti data mahasiswa, dosen, jurusan, major dan data-data lainnya.



**Gambar 4.2** Halaman Home Admin

**4.1.5 Halaman Home Dosen**

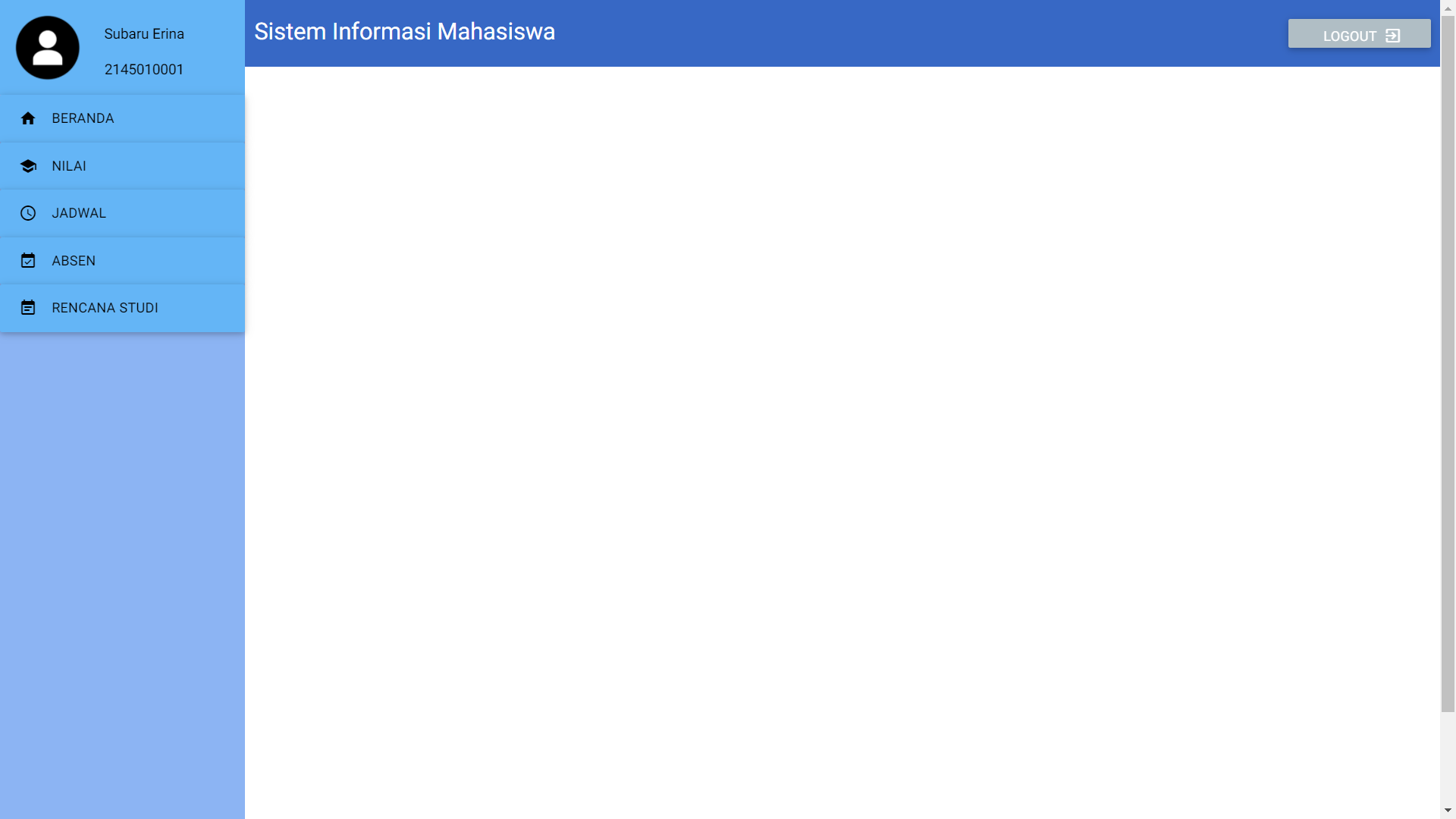
Halaman home dosen digunakan sebagai halaman navigasi menuju halaman jadwal mengajar, nilai mahasiswa yang diajar, daftar mahasiswa yang diajar, dan jika dosen memiiliki jabatan sebagai dosen wali, maka dosen juga dapat menyetujui form FRS dan batal tambah dari mahasiswa wali pada setiap semester baru.



**Gambar 4.3** Halaman Home Dosen

**4.1.6 Halaman Home Mahasiswa**

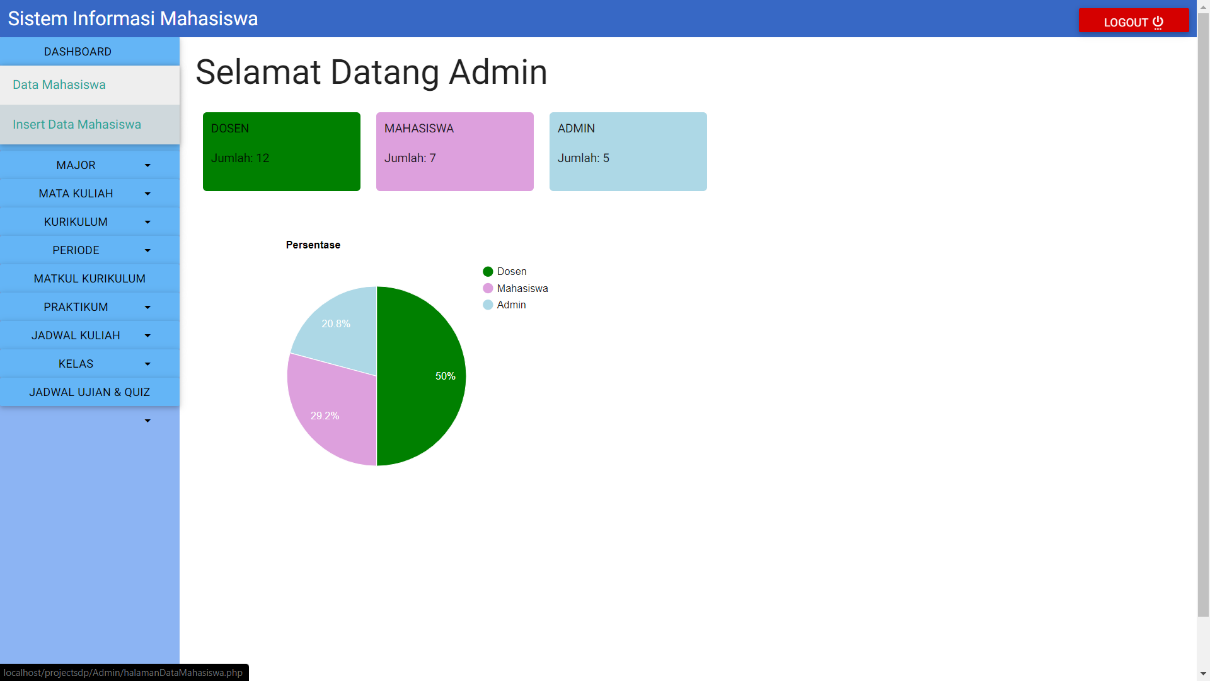
Halaman home mahasiswa digunakan sebagai halaman navigasi menuju halaman daftar nilai mahasiswa, jadwal pelajaran dan ujian mahasiswa, absensi mahasiswa dan pada sememster baru, mahasiswa dapat melakukan pengisian form FRS untuk memilih mata kuliah apa saja yang akan diambil selama 1 semester ke depan. FRS harus disetujui oleh dosen wali sebelum dapat dijadikan jadwal mata kuliah.



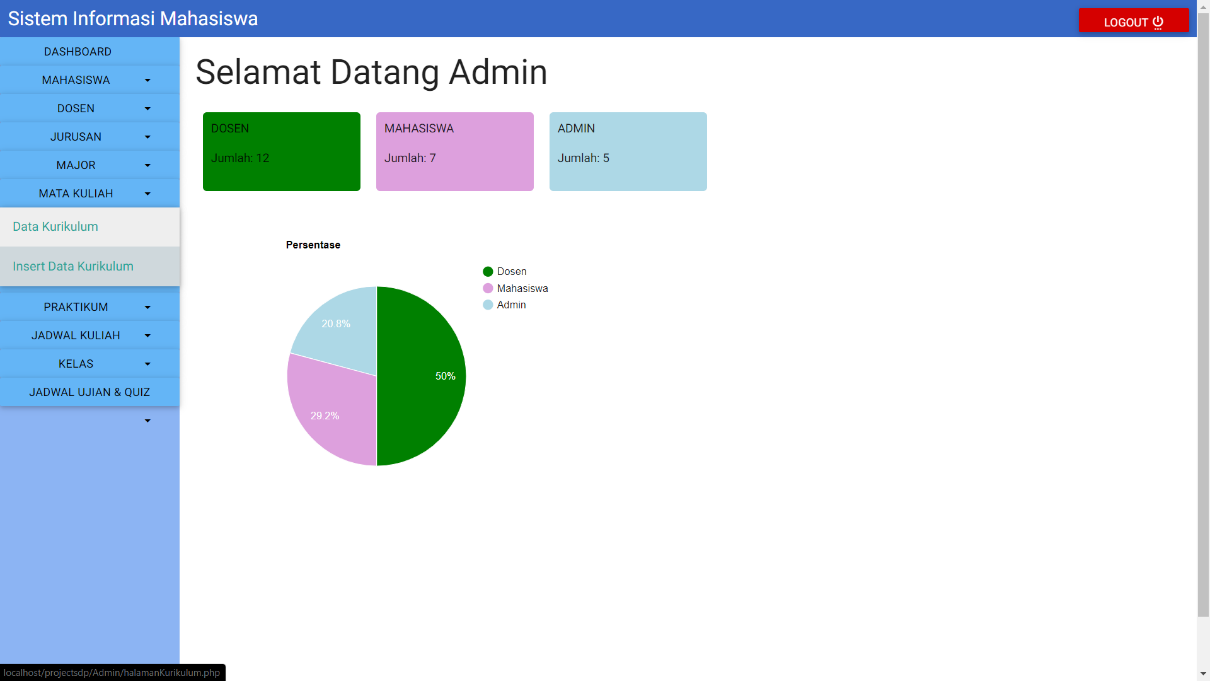
**Gambar 4.4** Halaman Home Mahasiswa

**4.1.7 Halaman Insert dan Edit Data**

Jika login sebagai salah satu admin, maka admin akan diarahkan menuju halaman home atau dashboard admin. Pada bagian kiri halaman tersebut terdapat navigation bar yang akan memberikan pilihan untuk melihat data pada suatu table tertentu, ataupun menginputkan data pada table tersebut. Setiap pilihan pada navigation bar merujuk kepada table yang berbeda.

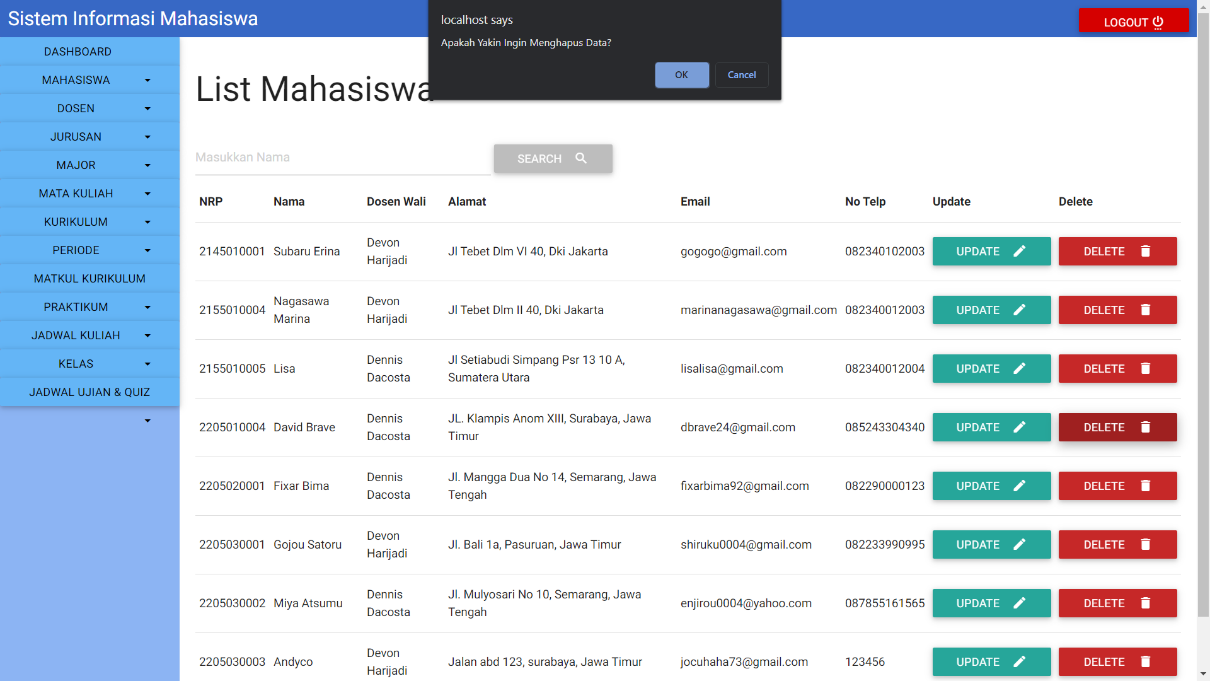


**Gambar 4.5** Pilihan Navigation Bar Mahasiswa

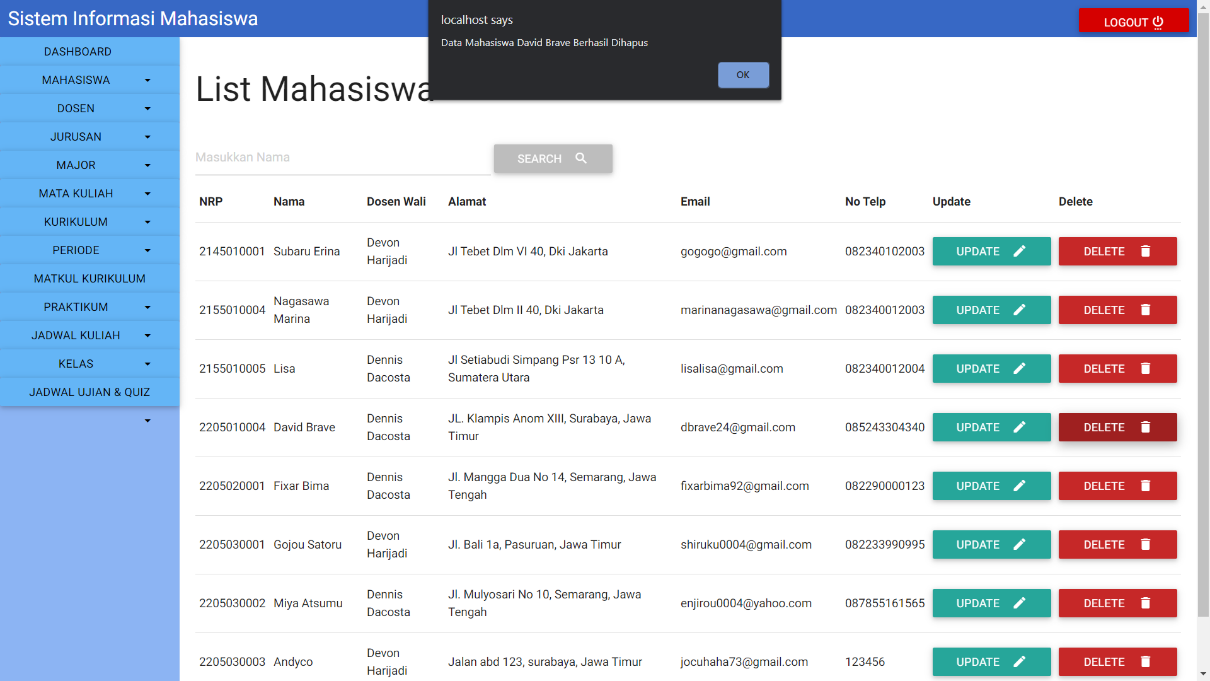


**Gambar 4.6** Pilihan Navigation Bar Kurikulum

Ketika menu data dipilih, maka akan ditampilkan sebuah table mengenai data tersebut. Tabel tersebut juga memiliki button update dan delete. Jika button delete ditekan, akan muncul pop up konfirmasi penghapusan data. Jika konfirmasi disetujui, maka data akan terhapus dari database, dan akan dimunculkan pesan status.

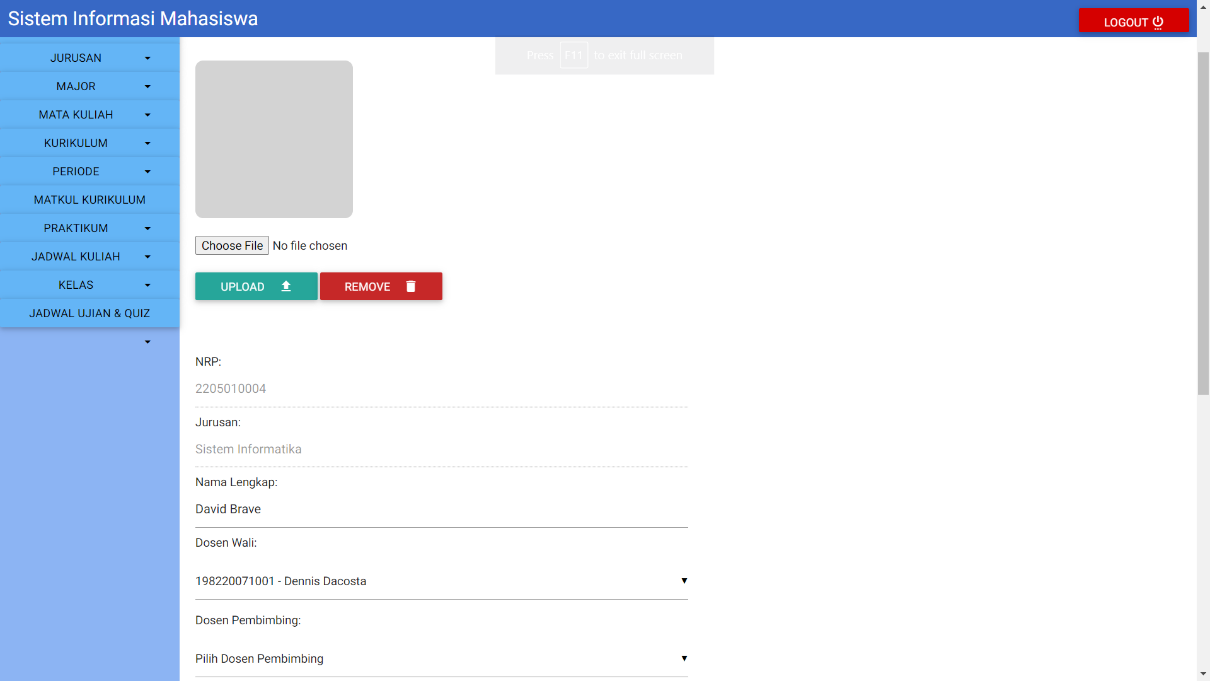


**Gambar 4.7** Konfirmasi Delete Mahasiswa



**Gambar 4.8** Pesan Status Delete

Button update akan mengalihkan admin ke halaman detail profile sesuai data yang dipilih. Di Halaman ini, admin dapat mengubah dan memperbaharui data-data tertentu. Pada contoh mahasiswa, admin dapat merubah nama mahasiswa, alamat mahasiswa, untuk memperbaiki data jika terdapat kesalahan. Beberapa data tidak dapat diubah semisal NRP mahasiswa.



**Gambar 4.9** Halaman Update Mahasiswa

Data-data yang dapat dimasukan dan diperbaharui oleh admin adalah sebagai berikut:

* Mahasiswa
* Dosen
* Jurusan
* Major (Sub-Jurusan)
* Mata Kuliah
* Kurikulum
* Periode
* Matkul Kurikulum
* Praktikum
* Jadwal Kuliah
* Kelas
* Jadwal Ujian

Data lain yang mencakup proses pembelajaran seperti nilai dan absensi dapat dimasukkan oleh dosen pengajar. Data pengambilan jadwal seperti FRS dan batal tambah akan diisi oleh mahasiswa dan disetujui oleh dosen wali dari mahasiswa tersebut.

**4.1.8 Halaman Pengisian FRS**

Pada semester baru, setiap mahasiswa dapat mengisi form FRS untuk menentukan mata kuliah yang ingin diambil pada satu semester ke depan. Mahasiswa memiliki batas mata kuliah yang dapat diambil, hal tersebut dihitung dari nilai maksimal sks yang dimiliki mahasiswa pada semester sembelumnya. Untuk mahasiswa baru, nilai maksimal sks adalah 18. Mata kuliah yang wajib diambil adalah mata kuliah aktif pada semester mahasiswa saat itu. Sebagai contoh mata kuliah Intro To Programming adalah mata kuliah aktif bagi mahasiswa jurusan informatika pada semester 1, maka seluruh mahasiswa dengan jurusan informatika, harus mengambil mata kuliah tersebut pada semester 1. Jika seluruh mata kuliah wajib telah diambil, maka mahasiswa dapat mengambil mata kuliah tambahan selama total sks seluruh mata kuliah tidak melebihi maksimal sks. Untuk mata kuliah yang memiliki praktikum, mahasiswa dapat memilih waktu dan ruang praktikum berdasarkan pilihan yang ditentukan. Mahasiswa wajib mengambil praktikum jika pada mata kuliah yang diambil memiliki praktikum.

Setelah form FRS diisi, mahasiswa harus menunggu persetujuan dosen wali untuk mendapatkan jadwal mata kuliah pada semester baru. Pada masa persetujuan ini, mahasiswa dapat menemui atau menghubungi dosen wali jika mahasiswa ingin melakukan penambahan mata kuliah atau pembatalan mata kuliah. Jika FRS telah disetujui oleh dosen wali, maka jadwal kuliah dibuat berdasarkan FRS yang telah diisi.

**4.1.9 Sistem Pengisian FRS**

Setiap mata kuliah yang dapat dipilih akan disimpan setelah form FRS dikumpulkan. Data FRS tersebut akan disimpan pada database, dan akan diubah menjadi jadwal jika telah disetujui oleh dosen pembimbing.

<?php

$success = true;

if(isset($\_SESSION['matkul']) &&

$\_SESSION['matkul'] != null){

for ($i=0; $i < sizeof($selectedMatkuls); $i++) {

$selectedMatkul = $selectedMatkuls[$i];

$query = "SELECT mk.Matkul\_Kurikulum\_ID, m.Matkul\_Nama, jk.Jadwal\_Hari, jk.Jadwal\_Mulai, jk.Jadwal\_Selesai,mk.Semester, mk.SKS FROM Matkul\_Kurikulum mk, Matkul m, Kelas k, Jadwal\_Kuliah jk

WHERE mk.Matkul\_ID = m.Matkul\_ID AND mk.Matkul\_Kurikulum\_ID = k.Matkulkurikulum\_ID AND k.Kelas\_ID = jk.Kelas\_ID AND mk.Matkul\_Kurikulum\_ID = '$selectedMatkul'";

$matkul = mysqli\_fetch\_array($conn-> query($query));

$jadwalMulai2 = $matkul2['Jadwal\_Mulai'];

$jadwalHari2 = $matkul2['Jadwal\_Hari'];

if($jadwalHari2 == $jadwalHari && $jadwalMulai2 == $jadwalMulai && $selectedMatkuls[$j] != $selectedMatkul){

$isCollision = true; } }

if($isCollision){

$success = false;

echo "<tr style='background-color: maroon; color: white;'>";

}else{ echo "<tr>"; }

echo "<td>$matkulkurikulumid</td>";

echo "<td>$matkulnama</td>";

echo "<td>$jadwalHari</td>";

echo "<td>$jadwalMulai</td>";

echo "<td>$jadwalSelesai</td>";

echo "<td>$semester</td>";

echo "<td>$sks</td>";

echo "<td><form action='#' method='post'><input type='submit' name='btnRemove' value='Remove' class='btn waves-effect red darken-3' style='height: 35px;'><input type='hidden' name='hidId' value='$matkulkurikulumid'></form></td>";

echo "</tr>"; }

}else{

echo "<tr>";

echo "<td colspan='7' style='text-align: center;'>Tidak ada matkul yang dipilih</td>";

echo "</tr>"; }

?>

**Segmen Program 4.3** Pilihan Mata Kuliah Pada FRS

<?php

if(isset($\_POST['btnSubmit'])){

if(isset($\_POST['praktikum'])){

$praktikum = $\_POST['praktikum'];

$mahasiswa = $\_SESSION['user']['user'];

for ($i=0; $i < sizeof($praktikum); $i++) {

$kelas = $praktikum[$i];

$query = "INSERT INTO Pengambilan\_Praktikum VALUES('', '$mahasiswa', '$kelas', 0, 1, $semester+1, '')";

$conn->query($query); }

for ($i=0; $i < sizeof($selectedMatkuls); $i++) {

$matkul = $selectedMatkuls[$i];

$query = "INSERT INTO FRS VALUES('', '$mahasiswa', '$matkul', '')";

$conn->query($query); }

}else{

$mahasiswa = $\_SESSION['user']['user'];

for ($i=0; $i < sizeof($selectedMatkuls); $i++) {

$matkul = $selectedMatkuls[$i];

$query = "INSERT INTO FRS VALUES('', '$mahasiswa', '$matkul', '')";

$conn->query($query); } }

$query = "UPDATE Mahasiswa SET Mahasiswa\_Semester = $semester+1 WHERE Mahasiswa\_ID = '$nrp'";

$conn->query($query);

$\_SESSION['pesan'] = "FRS berhasil";

header("location: Home.php"); }

?>

**Segmen Program 4.4** Proses Penyimpanan FRS Pada Database

<?php

if(isset($\_POST['id'])){

$id = $\_POST['id']; $semester = 0; $nama = "";

$query = "SELECT \* FROM Mahasiswa WHERE Mahasiswa\_ID = '$id'";

$temp = $conn->query($query);

foreach($temp as $key => $value) {

$nama = $value['Mahasiswa\_Nama'];

$semester = $value['Mahasiswa\_Semester']; }

$query = "UPDATE FRS SET FRS\_Status = 'Diterima' WHERE Mahasiswa\_ID = '$id'";

$conn->query($query);

$query = "SELECT \* FROM FRS WHERE Mahasiswa\_ID='$id'";

$matkuls = $conn->query($query);

foreach($matkuls as $key => $value) {

$query = "INSERT INTO Pengambilan VALUES('', '$id', '', 0,0,0,0,'',0,0,1,$semester,'')";

$conn->query($query); }

$message = "Gagal Delete";

if($conn){

$message = "Berhasil Terima FRS Mahasiswa $nama";

}else{

$message = "Gagal Terima FRS Mahasiswa $nama"; }

echo "<script type ='text/javascript'>alert('$message');</script>"; }

$conn->close();

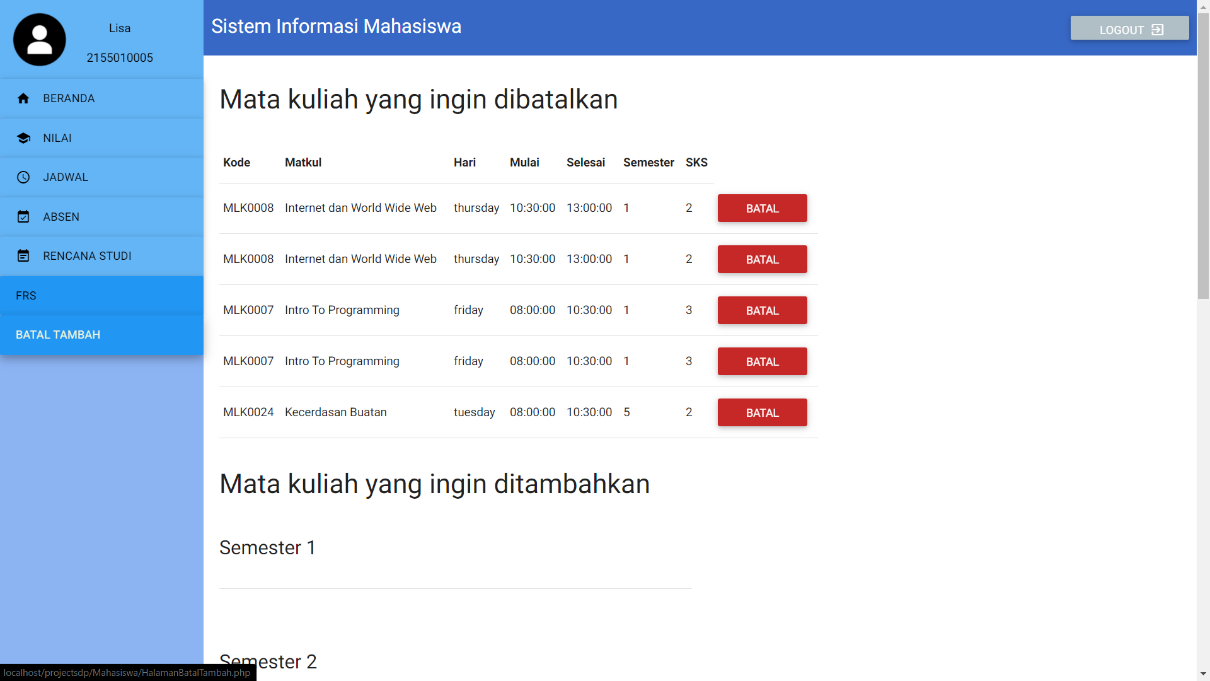
?>

**Segmen Program 4.5** Proses Persetujuan FRS

Saat FRS disetujui, status FRS pada database akan diubah menjadi diterima dan data FRS akan dimasukan ke dalam data pengambilan mata kuliah untuk semua pelajaran yang terdapat pada FRS tersebut.

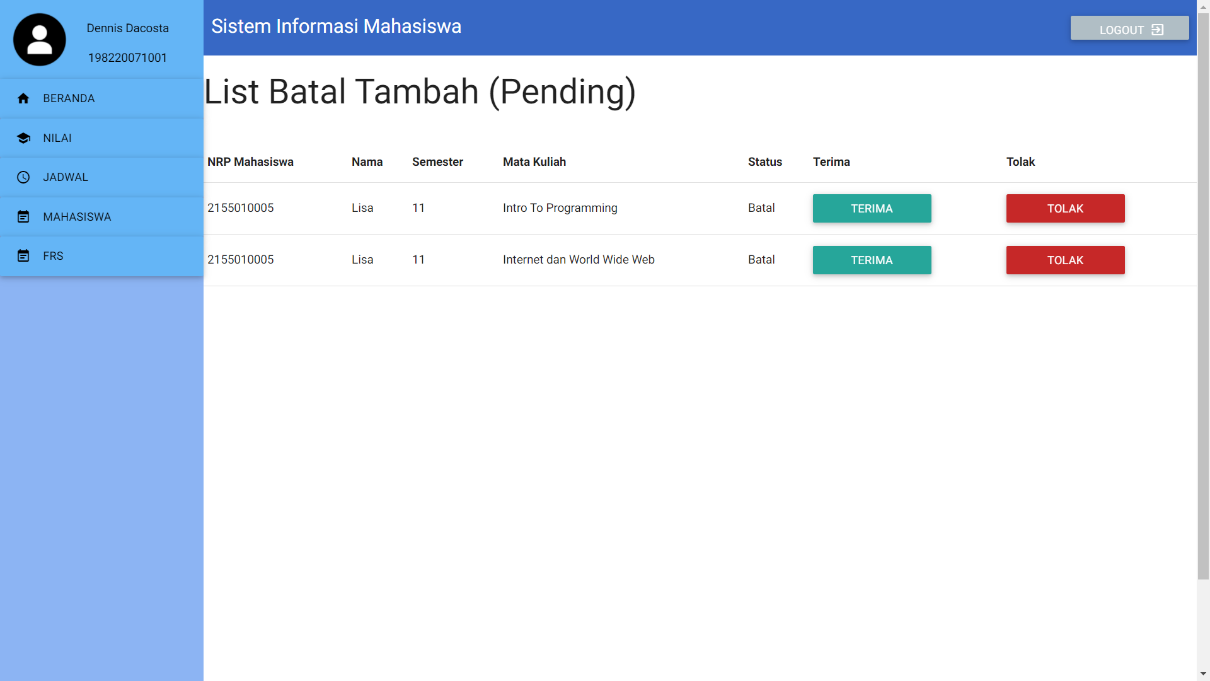
**4.1.10 Halaman Batal Tambah**

Mata kuliah yang telah dimasukan ke dalam pengambilan, dapat dibatalkan oleh mahasiswa melalui proses batal tambah. Mahasiswa dapat memilih mata kuliah yang ingin dibatalkan melalui menu rencana studi, batal tambah. Kemudian mahasiswa dapat menekan tombol batal mata pelajaran yang ingin dibatalkan.



**Gambar 4.10** Halaman Batal Tambah Mahasiswa

Seluruh data informasi mata kuliah yang ingin dibatalkan akan dikirim kepada dosen wali mahasiswa. Setelah mendapat persetujuan, maka mata kuliah tersebut tidak akan dimasukan sebagai mata kuliah yang diambil mahasiswa pada semester itu.



**Gambar 4.11** Halaman Persetujuan Batal Tambah Dosen

**4.1.11 Sistem Batal Tambah**

Batal tambah akan mengubah status FRS menjadi batal, sembari menunggu persetujuan dari dosen wali mahasiswa. Jika dosen wali menyetujui batal tambah, maka data pengambilan dan FRS dari mata kuliah tersebut akan dihapus dari database. Jika batal tambah ditolak oleh dosen, maka data mata kuliah tersebut tetap disimpan pada pengambilan mata kuliah mahasiswa yang mengajukan batal tambah.

` if(isset($\_POST['btnBatal'])){

$id = $\_POST['hidId'];

$query = "UPDATE FRS SET FRS\_Status = 'Batal' WHERE Mahasiswa\_ID = '$nrp' AND Matkul\_Kurikulum\_ID = '$id'";

$conn->query($query);

}

$query = "SELECT mk.Matkul\_Kurikulum\_ID, m.Matkul\_Nama, jk.Jadwal\_Hari, jk.Jadwal\_Mulai, jk.Jadwal\_Selesai,mk.Semester, mk.SKS

FROM Matkul\_Kurikulum mk, Matkul m, Kelas k, Jadwal\_Kuliah jk, Pengambilan p, Mahasiswa mhs, FRS f

WHERE p.Kelas\_ID = k.Kelas\_ID AND mk.Matkul\_ID = m.Matkul\_ID AND mk.Matkul\_Kurikulum\_ID = k.Matkulkurikulum\_ID AND k.Kelas\_ID = jk.Kelas\_ID AND p.Mahasiswa\_ID = '$nrp'

AND p.Mahasiswa\_ID = mhs.Mahasiswa\_ID AND p.Semester\_Pengambilan = mhs.Mahasiswa\_Semester AND k.Matkulkurikulum\_ID = f.Matkul\_Kurikulum\_ID AND f.Mahasiswa\_ID = mhs.Mahasiswa\_ID

AND p.Pengambilan\_Batal <> 1 AND f.FRS\_Status <> 'Batal'";

$selectedMatkuls = $conn->query($query);

$tahun1 = date("Y");

$tahun2 = $tahun1 + 1;

$bulan = date("m");

$periodeID = $tahun1 . $tahun2;

if ($bulan >= 8 && $bulan <= 12 || $bulan == 1) {

$periodeID = $periodeID . "11";

} else if ($bulan >= 2 && $bulan <= 7) {

$periodeID = $periodeID . "21";

}

**Segmen Program 4.6** Proses Pembatalan Mata Kuliah

if(isset($\_POST['id'])){

$id = $\_POST['id']; $matkul = $\_POST['matkul'];

$nama = "";

$query = "SELECT \* FROM Mahasiswa WHERE Mahasiswa\_ID = '$id'";

$temp = $conn->query($query);

foreach($temp as $key => $value) {

$nama = $value['Mahasiswa\_Nama']; }

$query = "DELETE FROM FRS WHERE Mahasiswa\_ID = '$id' AND Matkul\_Kurikulum\_ID = '$matkul'";

$conn->query($query);

$query = "SELECT k.Kelas\_ID FROM Kelas k, Pengambilan p WHERE k.Kelas\_ID = p.Kelas\_ID AND p.Mahasiswa\_ID = '$id' AND k.Matkulkurikulum\_ID = '$matkul'";

$temp = $conn->query($query); $kelas = "";

foreach($temp as $key => $value) {

$kelas = $value['Kelas\_ID']; }

$query = "DELETE FROM Pengambilan WHERE Mahasiswa\_ID = '$id' AND Kelas\_ID = '$kelas'";

$message = "Gagal Delete";

if($conn){

$message = "Berhasil Terima Batal Mahasiswa $nama";

}else{

$message = "Gagal Terima Batal Mahasiswa $nama"; }

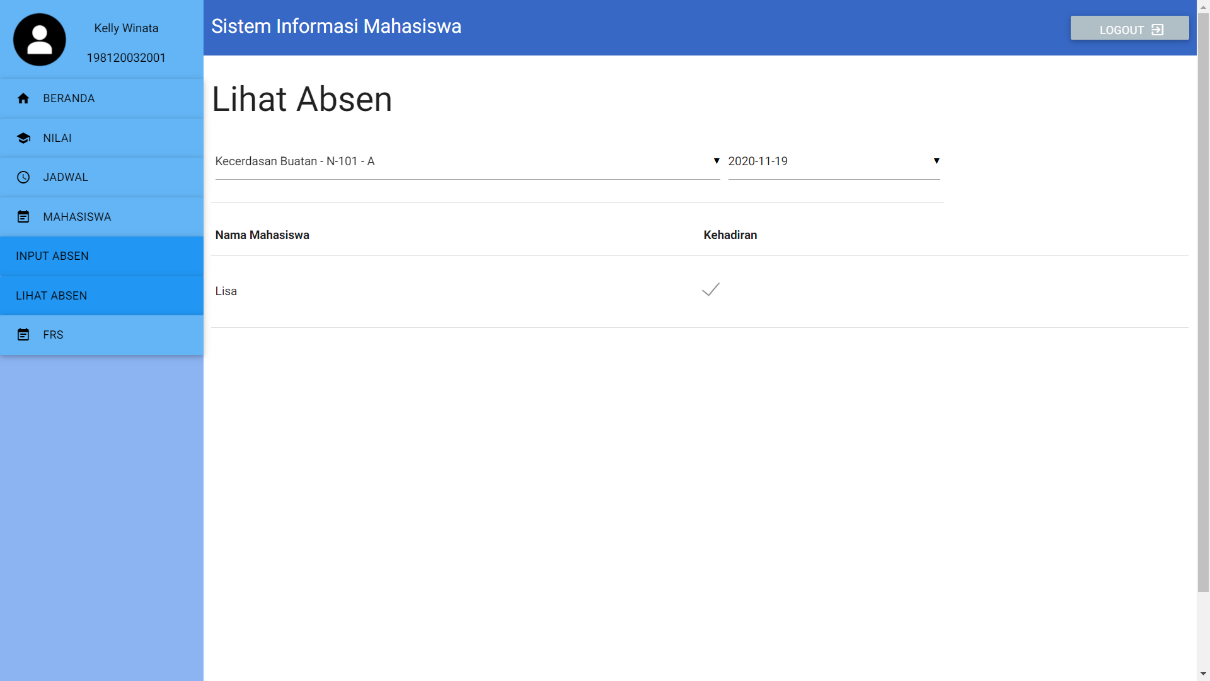
echo "<script type='text/javascript'> alert('$message');</script>"; }

$conn->close();

**Segmen Program 4.7** Proses Persetujuan Pembatalan

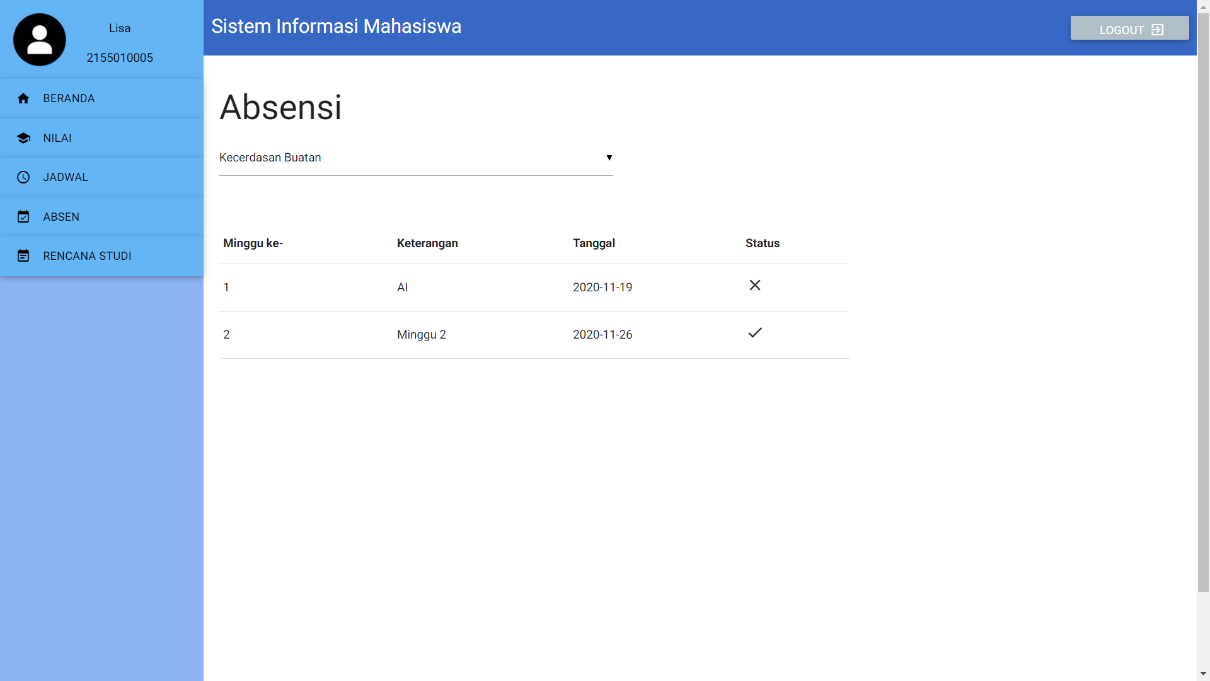
**4.1.12 Halaman Absensi**

Absensi dapat dilakukan dosen setelah kelas berakhir. Dosen dapat melihat data absensi untuk seluruh mahasiswa yang diajar pada sebuah mata kuliah, dengan memilih menu lihat absen, dan memasukan mata kuliah beserta tanggal pelajaran mata kuliah yang ingin dilihat.



**Gambar 4.12** Halaman Lihat Absensi Dosen

Pada halaman absensi milik mahasiswa, mahasiswa dapat memilih mata kuliah yang diikuti. Setelah mata kuliah dipilih, maka seluruh data absensi pada mata kuliah tersebut akan dimunculkan. Jika mahasiswa mengikuti mata kuliah, akan terdapat tanda centang dan akan terdapat tanda silang jika mahasiswa tidak mengikuti mata kuliah pada tanggal tersebut.

****

**Gambar 4.13** Halaman Lihat Absensi Mahasiswa

**4.1.13 Sistem Absensi**

Absensi dapat dilakukan dosen setelah kelas berakhir. Dosen dapat melihat data absensi untuk seluruh mahasiswa yang diajar pada sebuah mata kuliah, dengan memilih menu lihat absen, dan memasukan mata kuliah beserta tanggal pelajaran mata kuliah yang ingin dilihat.

if(isset($\_POST['btnSubmit'])) {

$kelas = $\_POST['kelasID'];

$mhs = $\_POST['kehadiran'];

$tanggal = $\_POST['tanggalAbsen'];

$keterangan = $\_POST['keterangan'];

$query = "SELECT a.Absen\_ID, m.Mahasiswa\_Nama, m.Mahasiswa\_ID FROM Absen a, Mahasiswa m WHERE a.Kelas\_ID = '$kelas' AND a.Absen\_Date = '$tanggal' AND a.Mahasiswa\_ID = m.Mahasiswa\_ID";

$listMhs = $conn->query($query);

foreach ($listMhs as $key => $value) {

$query = "UPDATE Absen SET Hadir = 0, Absen\_Keterangan = '$keterangan' WHERE Absen\_ID = $value[Absen\_ID]";

$conn->query($query); }

if (isset($\_POST['kehadiran'])) {

for($i = 0 ; $i < sizeof($mhs) ; $i++) {

$query = "UPDATE Absen SET Hadir = 1 WHERE Kelas\_ID = '$kelas' AND Mahasiswa\_ID = '$mhs[$i]'";

$conn->query($query); }

}

echo "<script>alert('Berhasil Edit Absen')</script>";

}

$dosenID = $\_SESSION['user']['user'];

$query = "SELECT DISTINCT k.Kelas\_Nama, k.Kelas\_Ruangan, m.Matkul\_Nama, k.Kelas\_ID

FROM Matkul\_Kurikulum mk, Matkul m, Kelas k, Dosen d, Jadwal\_Kuliah jk

WHERE d.Dosen\_ID = '$dosenID' AND k.DosenPengajar\_ID = d.Dosen\_ID AND k.Matkulkurikulum\_ID = mk.Matkul\_Kurikulum\_ID AND mk.Matkul\_ID = m.Matkul\_ID AND jk.Kelas\_ID = k.Kelas\_ID AND jk.Tanggal\_Kuliah = DATE(NOW()) AND jk.Jadwal\_Selesai < DATE\_ADD(CURRENT\_TIME, INTERVAL 7 hour)";

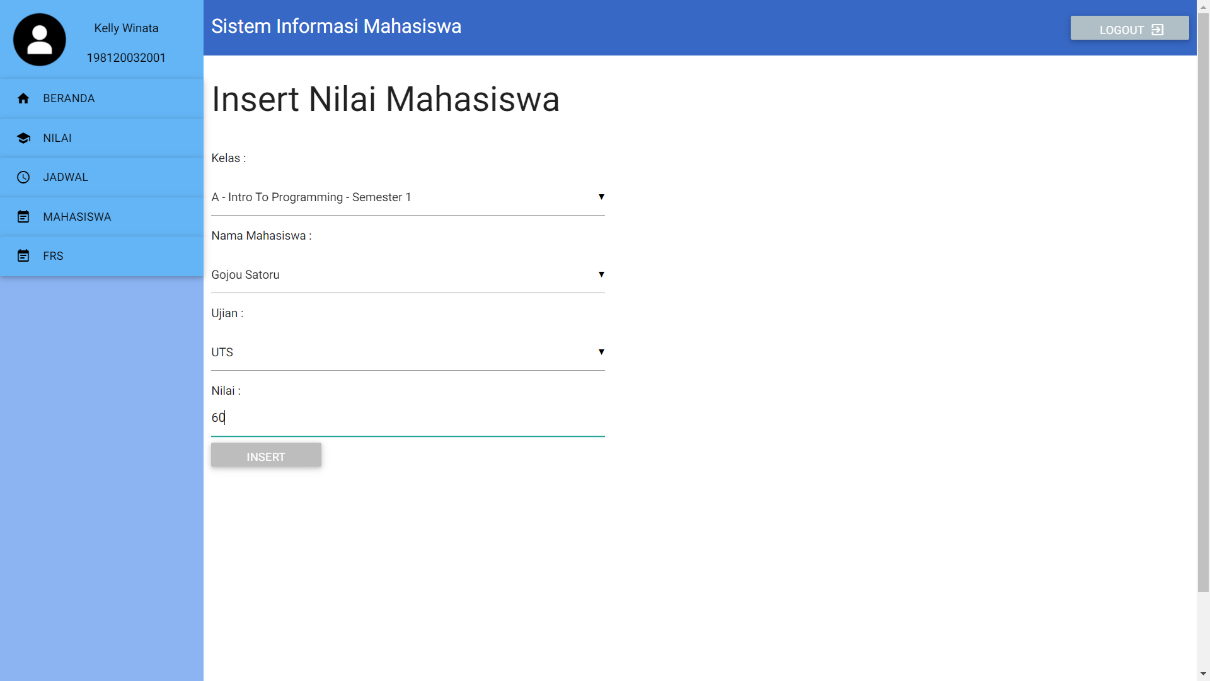
$listKelas = $conn->query($query);

}

**Segmen Program 4.8** Proses Mengubah Absen

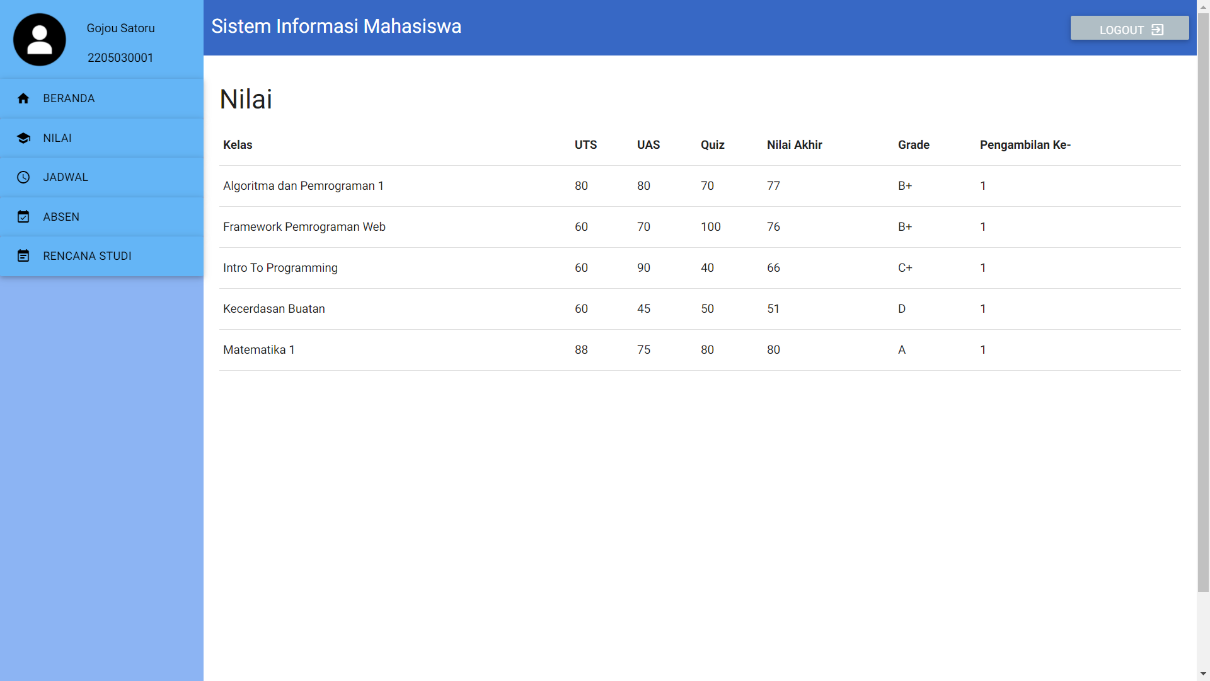
**4.1.14 Halaman Nilai**

Dosen dapat memasukan nilai mahasiswa yang diajar pada menu insert nilai. Pada halaman tersebut, memilih kelas yang diajar, beserta nama mahasiswa yang ingin diberi nilai. Terdapat 4 tipe nilai yang bisa diinputkan, yaitu quiz, uts, uas dan praktikum. Setelah memilih tipe nilai, dosen dapat menuliskan nilai dari mahasiswa yang telah dipilih.



**Gambar 4.14** Halaman Dosen Insert Nilai

Mahasiswa dapat melihat laporan nilai dari menu nilai. Halaman tersebut kemudian akan menampilakn seluruh data nilai mahasiswa berdasarkan mata kuliah yang telah dinilai oleh dosen pengajar. Halaman nilai juga akan menunjukan penilaian grade dari keseluruhan nilai mahasiswa, beserta status pengambilan mata kuliah. Jika mahasiswa baru mengambil mata kuliah tersebut 1 kali, maka akan ditampilkan sebagai pengambilan ke-1, dan seterusnya sesuai jumlah pengambilan yang telah dilakukan pada mata kuliah tersebut.



**Gambar 4.15** Halaman Laporan Nilai Mahasiswa

**4.1.15 Sistem Penilaian**

Saat dosen memasukan nilai mahasiswa, data nilai pada tabel pengambilan mata kuliah tersebut diubah sesuai tipe nilai yang dipilih dosen. Jika seluruh nilai telah dimasukan, akan dilakukan perhitungan nilai akhir. Nilai akhir adalah 30% quiz + 30% uts + 40% uas. Setelah nilai akhir didapatkan, maka akan dibuat grade pada pengambilan tersebut. Nilai akhir di atas 80 akan mendapat grade ‘A’, nilai akhir yang berkisar di antara 75 hingga 80 akan mendapat grade ‘B+’, nilai akhir yang berkisar di antara 70 hingga 75 akan mendapat grade ‘B’, nilai akhir yang berkisar di antara 65 hingga 70 akan mendapat grade ‘C+’, nilai akhir yang berkisar di antara 56 hingga 65 akan mendapat grade ‘C’, nilai akhir yang berkisar di antara 45 hingga 56 akan mendapat grade ‘D’ dan nilai dibawah 45 akan mendapat grade ‘E’. Setelah diberi grade, apabila nilai berada di atas standar kelulusan, maka status pengambilan akan menjadi lulus, sebaliknya jika nilai dibawah standar kelulusan, maka status pengambilan diubah menjadi tidak lulus.

if (isset($\_POST['btnInsert'])) {

if (isset($\_POST['ujian']) && isset($\_POST['mahasiswa']) && isset($\_POST['nilai'])) {

$nrp = $\_POST['mahasiswa'];

$kelasId = $\_POST['kelas'];

$nilai = $\_POST['nilai'];

if ($\_POST['ujian'] == "quiz"){

$query = "UPDATE Pengambilan SET Quiz = $nilai WHERE Mahasiswa\_ID = '$nrp' AND Kelas\_ID = '$kelasId'";

$conn->query($query);

echo "<script>alert('Berhasil insert nilai')</script>";

}else if ($\_POST['ujian'] == "uts"){

$query = "UPDATE Pengambilan SET UTS = $nilai WHERE Mahasiswa\_ID = '$nrp' AND Kelas\_ID = '$kelasId'";

$conn->query($query);

echo "<script>alert('Berhasil insert nilai')</script>";

}else if ($\_POST['ujian'] == "uas"){

$query = "UPDATE Pengambilan SET UAS = $nilai WHERE Mahasiswa\_ID = '$nrp' AND Kelas\_ID = '$kelasId'";

$conn->query($query);

$query = "SELECT UTS FROM Pengambilan WHERE Mahasiswa\_ID='$nrp' AND Kelas\_ID='$kelasId'";

$nilaiUTS = mysqli\_fetch\_array($conn->query($query));

$query = "SELECT UAS FROM Pengambilan WHERE Mahasiswa\_ID='$nrp' AND Kelas\_ID='$kelasId'";

$nilaiUAS = mysqli\_fetch\_array($conn->query($query));

$query = "SELECT Quiz FROM Pengambilan WHERE Mahasiswa\_ID='$nrp' AND Kelas\_ID='$kelasId'";

$nilaiQuiz = mysqli\_fetch\_array($conn->query($query));

$nilaiAkhir = ((int)$nilaiQuiz['Quiz']\*(30/100)+ ((int)$nilaiUTS['UTS']\*(30/100)+((int)$nilaiUAS['UAS']\*(40/100))));

$query = "SELECT m.Matkul\_Standar FROM Kelas k, Matkul\_Kurikulum mk, Matkul m WHERE k.Matkulkurikulum\_ID = mk.Matkul\_Kurikulum\_ID AND mk.Matkul\_ID = m.Matkul\_ID AND k.Kelas\_ID = '$kelasId'";

$standar = mysqli\_fetch\_array($conn->query($query));

if($nilaiAkhir >= $standar['Matkul\_Standar']){

$query="UPDATE Pengambilan SET Nilai\_Akhir = $nilaiAkhir, Pengambilan\_Status = 'Lulus' WHERE Mahasiswa\_ID = '$nrp' AND Kelas\_ID = '$kelasId'";

$conn->query($query);

}else{

$query="UPDATE Pengambilan SET Nilai\_Akhir = $nilaiAkhir, Pengambilan\_Status = 'Tidak Lulus' WHERE Mahasiswa\_ID = '$nrp' AND Kelas\_ID = '$kelasId'";

$conn->query($query); }

$grade="";

if ($nilaiAkhir>=80) $grade="A";

elseif (($nilaiAkhir<80)&&($nilaiAkhir>=75))

$grade="B+";

elseif (($nilaiAkhir<75)&&($nilaiAkhir>=70))

$grade="B";

elseif (($nilaiAkhir<70)&&($nilaiAkhir>=65))

$grade="C+";

elseif (($nilaiAkhir<65)&&($nilaiAkhir>=56))

$grade="C";

elseif (($nilaiAkhir<56)&&($nilaiAkhir>=45))

$grade="D";

Else {$grade="E";}

$query="UPDATE Pengambilan SET Pengambilan\_Grade='$grade' WHERE Mahasiswa\_ID='$nrp' AND Kelas\_ID='$kelasId'";

$conn->query($query);

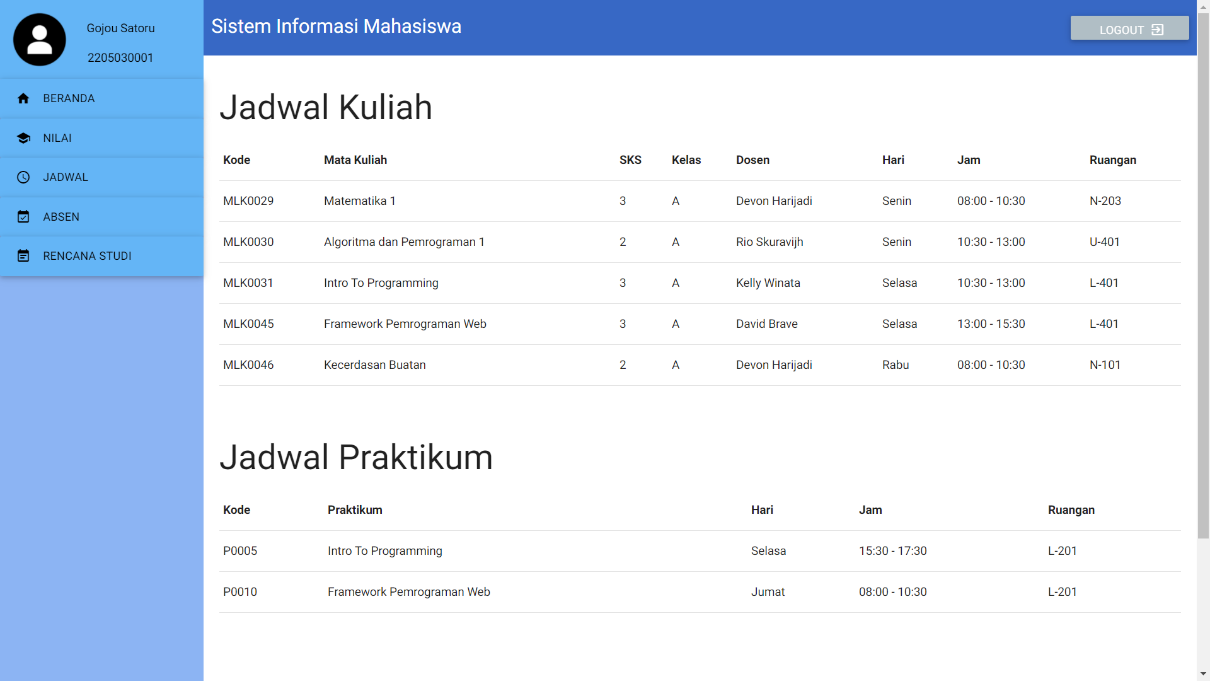
echo "<script>alert('Berhasil insert nilai')</script>";

}

**Segmen Program 4.9** Proses Insert Nilai

**4.1.16 Halaman Jadwal Mata Kuliah**

Jadwal kuliah yang akan ditampilkan adalah jadwal kuliah yang diambil mahasiswa pada saat pengisian FRS dan telah disetujui dosen pembimbing, beserta jadwal praktikum yang harus diambil jika mata kuliah pada FRS, memiliki praktikum. Jadwal kuliah akan menampilkan hari, waktu dan ruang yang akan digunakan beserta dosen pengajar dari mata kuliah yang dijadwalkan. Informasi lain yang disediakan adalah jumlah sks mata kuliah dan kelas dari mahasiswa yang memiliki jadwal tersebut. Untuk praktikum, akan ditampilkan hari, waktu dan ruangan berlangsunya praktikum.

****

**Gambar 4.16** Halaman Jadwal Kuliah Mahasiswa

**4.1.17 Cara Menampilan Jadwal Kuliah**

Nama mata kuliah dan jumlah sks yang ditampilkan diambil dari tabel matkul yang memiliki id yang sama dengan data matkul id pada tabel matkul\_kurikulum. Nama dosen pengajar diambil dari tabel dosen yang memiliki id sama dengan DosenPengajar\_id pada tabel kelas, sesuai dengan kelas yang diambil oleh mahasiswa yang sedang login. Untuk data hari, jam dan ruangan didapatkan dari data jadwal kuliah yang yang memiliki kelas\_id sesuai dengan id kelas yang diambil mahasiswa. Kemudian data jadwal kuliah tersebut diurutkan berdasarkan hari berlangsungya perkuliahan.

$query = "SELECT DISTINCT mk.Matkul\_Kurikulum\_ID, m.Matkul\_Nama, mk.SKS, k.Kelas\_Nama, d.Dosen\_Nama, jk.Jadwal\_Hari, SUBSTR(jk.Jadwal\_Mulai,1,5) as Jadwal\_Mulai, SUBSTR(jk.Jadwal\_Selesai,1,5) as Jadwal\_Selesai, k.Kelas\_Ruangan, (CASE jk.Jadwal\_Hari WHEN 'monday' THEN 1 WHEN 'tuesday' THEN 2 WHEN 'wednesday' THEN 3 WHEN 'thursday' THEN 4 WHEN 'friday' THEN 5 WHEN 'saturday' THEN 6 WHEN 'sunday' THEN 7 END) AS day

FROM Matkul\_Kurikulum mk, Matkul m, Kelas k, Dosen d, Jadwal\_Kuliah jk, Pengambilan p, Mahasiswa mhs, FRS f

WHERE mk.Matkul\_Kurikulum\_ID = k.Matkulkurikulum\_ID AND mk.Matkul\_ID = m.Matkul\_ID AND p.Mahasiswa\_ID = mhs.Mahasiswa\_ID AND p.Kelas\_ID = k.Kelas\_ID AND jk.Kelas\_ID = k.Kelas\_ID AND k.DosenPengajar\_ID = d.Dosen\_ID AND p.Mahasiswa\_ID = '$nrp' AND p.Semester\_Pengambilan = $semester

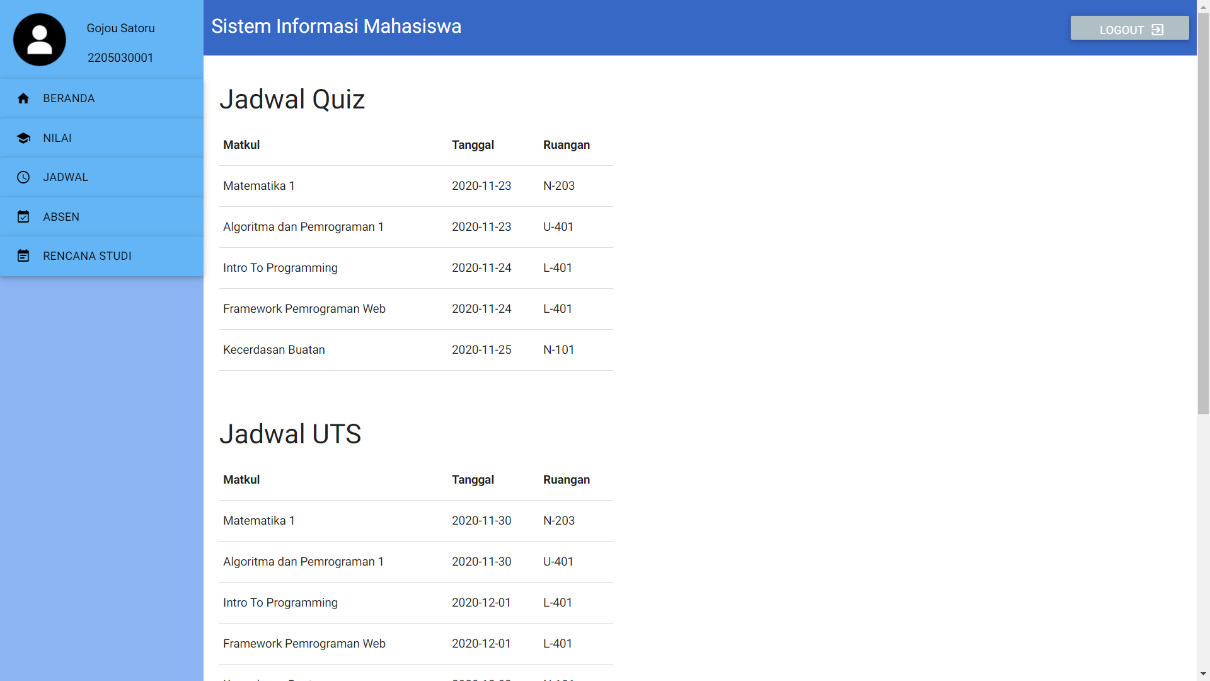
AND k.Matkulkurikulum\_ID = f.Matkul\_Kurikulum\_ID AND p.Pengambilan\_Batal <> 1 AND f.FRS\_Status <> 'Batal'

ORDER BY day ASC";

**Segmen Program 4.10** Query Jadwal Mata Kuliah

**4.1.18 Halaman Jadwal Ujian**

Halaman jadwal ujian dapat diakses memalui menu jadwal ujian. Halaman tersebut menampilkan jadwal kegiatan penting seperti quiz, uts dan uas. Informasi yang ditampilkan adalah mata kuliah yang diujikan, tanggal ujian dan ruang ujian.

****

**Gambar 4.17** Halaman Jadwal Ujian Mahasiswa

**4.1.19 Cara Menampilan Jadwal Ujian**

Nama mata kuliah yang ditampilkan diambil dari tabel matkul yang memiliki id yang sama dengan data matkul id pada tabel matkul\_kurikulum. Untuk data hari, diambil dari data tanggal pada tabel jadwal penting yang memiliki jadwal id sama dengan jadwal id pada tabel jadwal kuliah.

Jadwal penting dibagi menjadi 3 bagian, yaitu jadwal quiz, jadwal uts dan jadwal uas. Selain pengelompokan tersebut, pengurutan jadwal didasarkan pada tanggal berlangsungnya ujian tersebut.

$query = "SELECT DISTINCT m.Matkul\_Nama, jp.Penting\_Date, k.Kelas\_Ruangan FROM Jadwal\_Penting jp, Jadwal\_Kuliah jk, Kelas k, Matkul\_Kurikulum mk, Matkul m, Pengambilan p, FRS f

WHERE p.Kelas\_ID = k.Kelas\_ID AND jp.Jadwal\_ID = jk.Jadwal\_ID AND jk.Kelas\_ID = k.Kelas\_ID AND k.Matkulkurikulum\_ID = mk.Matkul\_Kurikulum\_ID AND mk.Matkul\_ID = m.Matkul\_ID AND jp.Keterangan = 'quiz' AND p.Kelas\_ID = k.Kelas\_ID AND p.Mahasiswa\_ID = '$nrp' AND p.Semester\_Pengambilan = '$semester'

AND k.Matkulkurikulum\_ID = f.Matkul\_Kurikulum\_ID AND p.Pengambilan\_Batal <> 1 AND f.FRS\_Status <> 'Batal'

ORDER BY Penting\_Date";

$listQuiz = $conn->query($query);

**Segmen Program 4.11** Query Jadwal Penting Quiz

$query = "SELECT DISTINCT m.Matkul\_Nama, jp.Penting\_Date, k.Kelas\_Ruangan FROM Jadwal\_Penting jp, Jadwal\_Kuliah jk, Kelas k, Matkul\_Kurikulum mk, Matkul m, Pengambilan p, FRS f

WHERE p.Kelas\_ID = k.Kelas\_ID AND jp.Jadwal\_ID = jk.Jadwal\_ID AND jk.Kelas\_ID = k.Kelas\_ID AND k.Matkulkurikulum\_ID = mk.Matkul\_Kurikulum\_ID AND mk.Matkul\_ID = m.Matkul\_ID AND jp.Keterangan = 'uts' AND p.Kelas\_ID = k.Kelas\_ID AND p.Mahasiswa\_ID = '$nrp' AND p.Semester\_Pengambilan = '$semester'

AND k.Matkulkurikulum\_ID = f.Matkul\_Kurikulum\_ID AND p.Pengambilan\_Batal <> 1 AND f.FRS\_Status <> 'Batal'

ORDER BY Penting\_Date";

$listUts = $conn->query($query);

**Segmen Program 4.12** Query Jadwal Penting UTS

$query = "SELECT DISTINCT m.Matkul\_Nama, jp.Penting\_Date, k.Kelas\_Ruangan FROM Jadwal\_Penting jp, Jadwal\_Kuliah jk, Kelas k, Matkul\_Kurikulum mk, Matkul m, Pengambilan p, FRS f

WHERE p.Kelas\_ID = k.Kelas\_ID AND jp.Jadwal\_ID = jk.Jadwal\_ID AND jk.Kelas\_ID = k.Kelas\_ID AND k.Matkulkurikulum\_ID = mk.Matkul\_Kurikulum\_ID AND mk.Matkul\_ID = m.Matkul\_ID AND jp.Keterangan = 'uas' AND p.Kelas\_ID = k.Kelas\_ID AND p.Mahasiswa\_ID = '$nrp' AND p.Semester\_Pengambilan = '$semester'

AND k.Matkulkurikulum\_ID = f.Matkul\_Kurikulum\_ID AND p.Pengambilan\_Batal <> 1 AND f.FRS\_Status <> 'Batal'

ORDER BY Penting\_Date";

$listUas = $conn->query($query);

**Segmen Program 4.13** Query Jadwal Penting UAS

**BAB V**

**UJI COBA**

Bab uji coba akan menjelaskan semua hasil uji coba yang telah dilakukan pada website SIM yang telah dibuat. Uji coba dilakukan untuk mengecek apakah terdapat masalah saat website digunakan.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Masalah Yang Diujikan | Hasil yang Diharapkan | Hasil Uji Coba | Status Uji Coba |
| 1 | Login dengan NRP dan password yang terdapat pada database mahasiswa | User diarahkan pada halaman mahasiswa | User diarahkan pada halaman mahasiswa | Valid |
| 2 | Login dengan username dan password yang terdapat pada database dosen | User diarahkan pada halaman dosen | User diarahkan pada halaman dosen | Valid |
| 3 | Login dengan username dan password yang terdapat pada database admin | User diarahkan pada halaman admin | User diarahkan pada halaman admin | Valid |
| 4 | Login dengan username atau password kosong | Munculkan pesan error gagal login | Munculkan pesan error gagal login | Valid |
| 5 | Login dengan username atau password yang salah | Munculkan pesan error gagal login | Munculkan pesan error gagal login | Valid |

**Tabel 5.1** Tabel Uji Coba Login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Masalah Yang Diujikan | Hasil yang Diharapkan | Hasil Uji Coba | Status Uji Coba |
| 1 | Memasukkan semua input dan generate nrp | Data mahasiswa dimasukan ke dalam database | Data mahasiswa dimasukan ke dalam database | Valid |
| 2 | Memasukkan semua input dan tanpa generate nrp | Data mahasiswa tidak dimasukan ke dalam database dan muncul pesan error | Data mahasiswa tidak dimasukan ke dalam database dan muncul pesan error | Valid |
| 3 | Terdapat input kosong dan generate nrp | Data mahasiswa tidak dimasukan ke dalam database dan muncul pesan error | Data mahasiswa tidak dimasukan ke dalam database dan muncul pesan error | Valid |
| 4 | Terdapat input kosong dan tanpa generate nrp | Data mahasiswa tidak dimasukan ke dalam database dan muncul pesan error | Data mahasiswa tidak dimasukan ke dalam database dan muncul pesan error | Valid |
| 5 | Input jurusan mahasiswa diubah sebelum generate nrp | Data mahasiswa tidak dimasukan ke dalam database dan muncul pesan error | Data mahasiswa tidak dimasukan ke dalam database dan muncul pesan error | Valid |
| 6 | Input jurusan mahasiswa diubah setelah generate nrp | Nrp mahasiswa diperbaharui berdasarkan jurusan yang baru | Nrp mahasiswa diperbaharui berdasarkan jurusan yang baru | Valid |

**Tabel 5.2** Tabel Uji Coba Insert Mahasiswa

Tabel di atas merupakan tabel uji coba untuk proses insert mahasiswa. Jika terdapat input yang kosong atau NRP yang belum di buat, maka insert tidak akan dilakukan. Jika jurusan mahasiswa dirubah, maka NRP yang telah dibuat juga akan berubah berdasarkan jurusan mahasiswa tersebut. Jika seluruh input valid, maka data mahasiswa akan dimasukan pada database.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Masalah Yang Diujikan | Hasil yang Diharapkan | Hasil Uji Coba | Status Uji Coba |
| 1 | Memasukkan semua input data dosen | Data dosen dimasukan pada database dosen | Data dosen dimasukan pada database dosen | Valid |
| 2 | Terdapat data dosen yang kosong | Data dosen tidak dimasukan ke dalam database dosen dan muncul pesan error | Data dosen tidak dimasukan ke dalam database dan muncul pesan error | Valid |

**Tabel 5.3** Tabel Uji Coba Insert Dosen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Masalah Yang Diujikan | Hasil yang Diharapkan | Hasil Uji Coba | Status Uji Coba |
| 1 | Memasukkan semua input data jurusan | Data jurusan dimasukan pada database jurusan | Data jurusan dimasukan pada database jurusan | Valid |
| 2 | Terdapat data jurusan yang kosong | Data jurusan tidak dimasukan pada database jurusan dan muncul pesan error | Data jurusan tidak dimasukan pada database jurusan dan muncul pesan error | Valid |

**Tabel 5.4** Tabel Uji Coba Insert Jurusan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Masalah Yang Diujikan | Hasil yang Diharapkan | Hasil Uji Coba | Status Uji Coba |
| 1 | Memasukkan semua input data major | Data major dimasukan pada database major | Data major dimasukan pada database major | Valid |
| 2 | Terdapat data major yang kosong | Data major tidak dimasukan pada database major dan muncul pesan error | Data major tidak dimasukan pada database major dan muncul pesan error | Valid |

**Tabel 5.5** Tabel Uji Coba Insert Major

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Masalah Yang Diujikan | Hasil yang Diharapkan | Hasil Uji Coba | Status Uji Coba |
| 1 | Memasukkan semua input data mata kuliah | Data mata kuliah dimasukan pada database mata kuliah | Data mata kuliah dimasukan pada database mata kuliah | Valid |
| 2 | Terdapat data mata kuliah yang kosong | Data mata kuliah tidak dimasukan pada database mata kuliah dan muncul pesan error | Data mata kuliah tidak dimasukan pada database mata kuliah dan muncul pesan error | Valid |

**Tabel 5.6** Tabel Uji Coba Insert Mata Kuliah

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Masalah Yang Diujikan | Hasil yang Diharapkan | Hasil Uji Coba | Status Uji Coba |
| 1 | Memasukkan input nama kurikulum | Data kurikulum dimasukan pada database kurikulum | Data kurikulum dimasukan pada database kurikulum | Valid |
| 2 | Input nama kurikulum kosong | Data kurikulum tidak dimasukan pada database kurikulum dan muncul pesan error | Data kurikulum tidak dimasukan pada database kurikulum dan muncul pesan error | Valid |

**Tabel 5.7** Tabel Uji Coba Insert Kurikulum

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Masalah Yang Diujikan | Hasil yang Diharapkan | Hasil Uji Coba | Status Uji Coba |
| 1 | Memasukkan semua input data periode | Data periode dimasukan pada database periode | Data periode dimasukan pada database periode | Valid |
| 2 | Terdapat data periode yang kosong | Data periode tidak dimasukan pada database periode dan muncul pesan error | Data periode tidak dimasukan pada database periode dan muncul pesan error | Valid |

**Tabel 5.8** Tabel Uji Coba Insert Periode

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Masalah Yang Diujikan | Hasil yang Diharapkan | Hasil Uji Coba | Status Uji Coba |
| 1 | Memasukkan semua input data matkul kurikulum | Data matkul kurikulum dimasukan pada database matkul\_kurikulum | Data matkul kurikulum dimasukan pada database matkul\_kurikulum | Valid |
| 2 | Terdapat data matkul kurikulum yang kosong | Data matkul kurikulum tidak dimasukan pada database matkul\_kurikulum dan muncul pesan error | Data matkul kurikulum tidak dimasukan pada database matkul\_kurikulum dan muncul pesan error | Valid |

**Tabel 5.9** Tabel Uji Coba Insert Matkul Kurikulum

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Masalah Yang Diujikan | Hasil yang Diharapkan | Hasil Uji Coba | Status Uji Coba |
| 1 | Memasukkan semua input data praktikum | Data praktikum dimasukan pada database praktikum | Data praktikum dimasukan pada database praktikum | Valid |
| 2 | Terdapat data praktikum yang kosong | Data praktikum tidak dimasukan pada database praktikum dan muncul pesan error | Data praktikum tidak dimasukan pada database praktikum dan muncul pesan error | Valid |
| 3 | Jadwal praktikum yang diinputkan bertubrukran dengan jadwal lain | Data praktikum tidak dimasukan pada database praktikum dan muncul pesan error | Data praktikum tidak dimasukan pada database praktikum dan muncul pesan error | Valid |

**Tabel 5.10** Tabel Uji Insert Jadwal Praktikum

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Masalah Yang Diujikan | Hasil yang Diharapkan | Hasil Uji Coba | Status Uji Coba |
| 1 | Memasukkan semua input data ruang praktikum | Data ruang praktikum dimasukan pada database kelas\_praktikum | Data ruang praktikum dimasukan pada database kelas\_praktikum | Valid |
| 2 | Terdapat data ruang praktikum yang kosong | Data ruang praktikum tidak dimasukan pada database kelas\_praktikum dan muncul pesan error | Data ruang praktikum tidak dimasukan pada database kelas\_praktikum dan muncul pesan error | Valid |

**Tabel 5.11** Tabel Uji Coba Insert Kelas Praktikum

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Masalah Yang Diujikan | Hasil yang Diharapkan | Hasil Uji Coba | Status Uji Coba |
| 1 | Memasukkan semua input data jadwal kuliah | Data jadwal kuliah dimasukan pada database jadwal\_kuliah | Data jadwal kuliah dimasukan pada database jadwal\_kuliah | Valid |
| 2 | Terdapat data jadwal kuliah yang kosong | Data jadwal kuliah tidak dimasukan pada database jadwal\_kuliah dan muncul pesan error | Data jadwal kuliah tidak dimasukan pada database jadwal\_kuliah dan muncul pesan error | Valid |

**Tabel 5.12** Tabel Uji Coba Insert Jadwal Kuliah

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Masalah Yang Diujikan | Hasil yang Diharapkan | Hasil Uji Coba | Status Uji Coba |
| 1 | Memasukkan semua input data kelas | Data kelas dimasukan pada database kelas | Data kelas dimasukan pada database kelas | Valid |
| 2 | Terdapat data kelas yang kosong | Data kelas tidak dimasukan pada database kelas dan muncul pesan error | Data kelas tidak dimasukan pada database kelas dan muncul pesan error | Valid |

**Tabel 5.13** Tabel Uji Coba Insert Kelas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Masalah Yang Diujikan | Hasil yang Diharapkan | Hasil Uji Coba | Status Uji Coba |
| 1 | Memasukkan semua input jadwal penting | Data jadwal penting dimasukan pada database jadwal\_penting | Data jadwal penting dimasukan pada database jadwal\_penting | Valid |
| 2 | Terdapat data jadwal penting yang kosong | Data jadwal penting tidak dimasukan pada database jadwal\_penting dan muncul pesan error | Data jadwal penting tidak dimasukan pada database jadwal\_penting dan muncul pesan error | Valid |

**Tabel 5.14** Tabel Uji Insert Jadwal Penting

Pada proses insert oleh admin, akan dilakukan pengecekan untuk semua field inputan. Jika terdapat field yang kosong atau tidak valid, maka data tidak akan dimasukkan ke dalam database dan dimunculkan pesan error yang menginfokan admin bahwa terdapat inputan yang belum diisi atau tidak benar.

Pada beberapa data, seperti jadwal praktikum, akan dilakukan pengecekan apakah jadwal yang ingin diinputkan bertabrakan dengan jadwal praktikum yang sudah ada. Jika terjadi tabrakan jadwal, maka data tidak akan dimasukan dan akan dimunculkan pesan error yang memberitahu admin bahwa data bertubrukan.

**BAB VI**

**PENUTUP**

Penutup adalah bab terakhir yang berisi kata-kata penutup, kesimpulan dan saran yang berhubungan dengan pembuatan website SIM. Bagian kesimpulan merupakan ringkasan yang diambil dari buku ini. Saran yang terdapat dalam buku ini diharapkan dapat digunakan untuk mengembangkan website SIM yang telah dibuat agar menjadi lebih baik.

**6.1 Kesimpulan**

Website SIM yang dibuat telah memiliki fitur-fitur untuk dapat mengakomodasi seluruh kegiatan belajar mengajar. Berdasarkan Analisa yang telah dilakukan, dapat ditarik beberapa kesimpulan mengenai fitur-fitur pada website SIM. Seperti FRS online yang disediakan dapat memudahkan proses penjadwalan mata kuliah. Rekap nilaipun akan lebih mudah diproses dengan laporan nilai yang jelas.

**6.2 Saran**

Untuk membuat website SIM lebih baik, tentunya terdapat saran-saran konstruktif yang dapat mendukung pengembangan program. Saran dapat diambil berdasarkan kesimpulan yang sudah dibuat, seperti data absensi yang ada, dapat ditambah sisipan folder untuk membantu mahasiswa mendapatkan sumber pelajaran.

**Daftar Pustaka**

Usability Post. 2009. Characteristics Of Successful User Interfaces.

https://www.usability.gov/what-and-why/user-interface-design.html/

Worcester Institute. Effective Visual Communication for Graphical User Interfaces

http://web.cs.wpi.edu/~matt/courses/cs563/talks/smartin/int\_design.html

Usability Post. Improving the User Experience

https://www.usability.gov/what-and-why/user-experience.html

Php Documentation

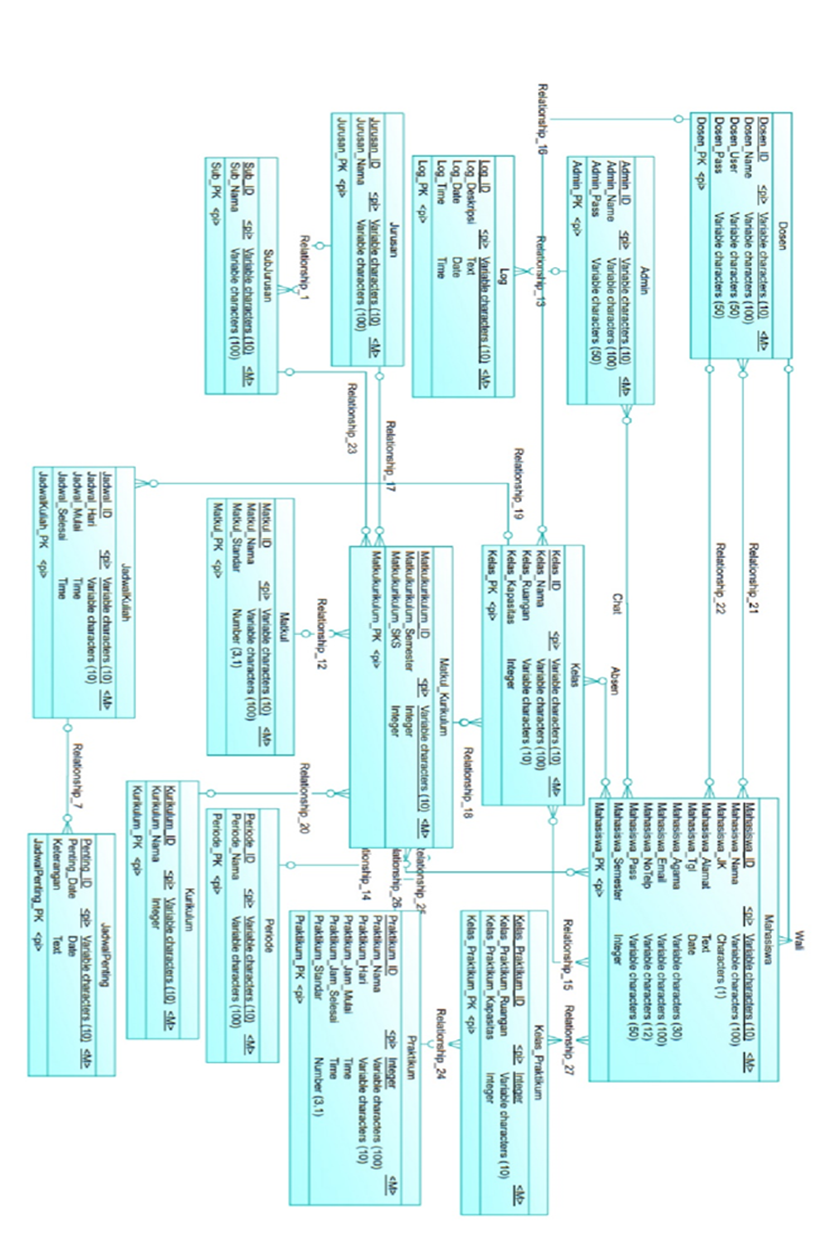
https://www.php.net/docs.php

Website SIM iSTTS

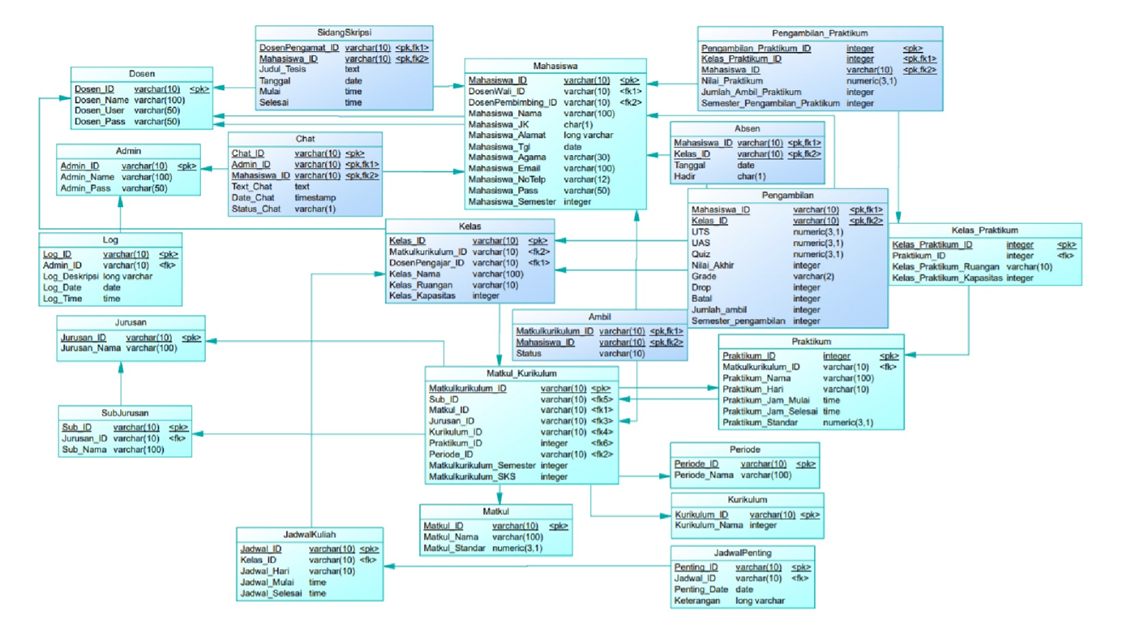
https://sim.stts.edu

LAMPIRAN A

ENTITY RELATIONSHIP DIAGRAM

****

**Gambar A-1** CDM Database Website SIM

****

**Gambar A-2** PDM Database Website SIM