**

TUGAS AKHIR – “”

**Rancang Bangun Aplikasi Sistem Basis Data Online Judge (SBDOJ) untuk Proses Pembelajaran Mata Kuliah Sistem Basis Data di Jurusan Teknik Informatika ITS**

DEMSY IMAN MUSTASYAR

NRP 5113 100 015

Dosen Pembimbing 1

Rizky Januar Akbar, S.Kom., M.Eng.

Dosen Pembimbing 2

Nurul Fajrin A., S.Kom, M.Sc.

JURUSAN TEKNIK INFORMATIKA

Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember

Surabaya 2016

**



TUGAS AKHIR – “”

**Rancang Bangun Aplikasi Sistem Basis Data Online Judge (SBDOJ) untuk Proses Pembelajaran Mata Kuliah Sistem Basis Data di Jurusan Teknik Informatika ITS**

DEMSY IMAN MUSTASYAR

NRP 5113 100 015

Dosen Pembimbing 1

Rizky Januar Akbar, S.Kom., M.Eng.

Dosen Pembimbing 2

Nurul Fajrin A., S.Kom, M.Sc.

JURUSAN TEKNIK INFORMATIKA

Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember

Surabaya 2016

*[Halaman ini sengaja dikosongkan]*

**

UNDERGRADUATE THESIS – “”

**APPLICATION DESIGN OF DATABASE SYSTEM ONLINE JUDGE (SBDOJ) FOR LEARNING PROCESS OF DATABASE SYSTEM COURSE AT THE DEPARTMENT OF INFORMATION ENGINEERING ITS**

DEMSY IMAN MUSTASYAR

NRP 5113 100 015

Supervisor 1

RIZKY JANUAR AKBAR, S.Kom., M.Eng.

Supervisor 2

Nurul Fajrin A., S.Kom, M.Sc.

INFORMATICS ENGINEERING DEPARTMENT

Information Technology Faculty

Sepuluh Nopember Institute of Technology

Surabaya 2017

*[Halaman ini sengaja dikosongkan]*

# LEMBAR PENGESAHAN

**RANCANG BANGUN APLIKASI SISTEM BASIS DATA ONLINE JUDGE (SBDOJ) UNTUK PROSES PEMBELAJARAN MATA KULIAH SISTEM BASIS DATA DI JURUSAN TEKNIK INFORMATIKA ITS**

**TUGAS AKHIR**

Diajukan Guna Memenuhi Salah Satu Syarat   
Memperoleh Gelar Sarjana Komputer  
pada   
Rumpun Mata Kuliah Manajemen Informasi  
Program Studi S-1 Jurusan Teknik Informatika  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh :  
**DEMSY IMAN MUSTASYAR  
NRP. 5113 100 015**

Disetujui oleh Dosen Pembimbing Tugas Akhir:

1. Rizky Januar Akbar, S.Kom., M.Eng. ........................

NIP. 198701032014041001 (Pembimbing 1)

1. R Nurul Fajrin A., S.Kom, M.Sc. ........................

NIP. 198607222015042003 (Pembimbing 2)

**SURABAYA**

**JUNI 2016**

*[Halaman ini sengaja dikosongkan]*

**RANCANG BANGUN APLIKASI SISTEM BASIS DATA ONLINE JUDGE (SBDOJ) UNTUK PROSES PEBELAJARAN MATA KULIAH SISTEM BASIS DATA DI JURUSAN TEKNIK INFORMATIKA ITS**

**Nama : Demsy Iman Mustasyar**

**NRP : 5113100015**

**Jurusan : Teknik Informatika**

**Fakultas Teknologi Informasi ITS**

**Dosen Pembimbing 1 : Rizky Januar Akbar, S.Kom.,M.Eng.**

**Dosen Pembimbing 2 : Nurul Fajrin A., S.Kom., M.Sc.**

# ABSTRAK

***Kata Kunci:***

**APPLICATION DESIGN OF DATABASE SYSTEM ONLINE JUDGE (SBDOJ) FOR LEARNING PROCESS OF DATABASE SYSTEM COURSE AT THE DEPARTMENT OF INFORMATION ENGINEERING ITS**

**Name : Demsy Iman Mustasyar**

**NRP : 5113100015**

**Department : Department of Informatics**

**Faculty of Information Technology ITS**

**Supervisor I : Rizky Januar Akbar, S.Kom., M.Eng.**

**Supervisor II : Nurul Fajrin A., S.Kom., M.Sc.**

# ABSTRACT

***Keywords:***

*[Halaman ini sengaja dikosongkan]*

# KATA PENGANTAR

Surabaya, …. 2017

Penulis

# BAB I PENDAHULUAN

Pada bab ini dijelaskan hal-hal yang menjadi latar belakang, permasalahan yang dihadapi, batasan masalah, tujuan, metodologi dan sistematika penulisan yang digunakan dalam pembuatan buku Tugas Akhir ini.

## Latar Belakang

Basis data (*database*) adalah kumpulan data, biasanya menjelaskan mengenai aktifitas satu atau lebih organisasi yang terkait. Basis dataterdiri dari entitas dan hubungan antar entitas. Basis databiasanya diatur menggunakan sebuah *software* yaitu *Database Management System* (DBMS). Basis datamerupakan dasar dari sistem penyimpanan dalam sebuah aplikasi, baik itu aplikasi *desktop*, *website*, maupun *mobile*. Oleh karena itu teori-teori mengenai basis dataharus dikuasai oleh semua *developer* perangkat lunak.

Dengan landasan seperti itu, maka hampir di semua jurusan yang berbau Teknologi Informasi mengajarkan pelajaran basis data. Termasuk di jurusan Teknik Informatika ITS. Sistem Basis Data (SBD) adalah mata kuliah di Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya yang mengajarkan mengenai basis data. Mata kuliah ini mempelajari mengenai dasar-dasar basis data.

Di dalam mata kuliah SBD diajarkan mengenai *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML). DDL merupakan perintah-perintah yang biasa digunakan *administrator* basis datauntuk mendefinisikan skema dan subskema basis data. DDL mempunyai fungsi utama untuk mendefinisikan data dalam database secara logika. Sedangkan DML merupakan merupakan perintah-perintah yang memungkinkan pengguna melakukan akses dan manipulasi data sebagaimana yang telah diorganisasikan sebelumnya dalam model data yang tepat. DML digunakan untuk memanipulasi basis datayang telah didefinisikan dengan DDL.

Di dalam matakuliah SBD juga terdapat praktikum mengenai materi DDL maupun DML. Penilaian praktikum SBD saat ini masih manual, yaitu praktikan mengeksekusi *query* di DBMS lalu ditunjukkan kepada asistennya untuk dinilai. Sistem praktikum seperti ini tentu sangat tidak efektif, karena membutuhkan waktu lama dan membutuhkan sumber daya asisten yang banyak. Oleh karena itu, diperlukan sebuah aplikasi yang bisa membantu praktikum SBD tersebut. Aplikasi yang akan dibangun adalah Sistem Basis Data Online Judge (SBDOJ).

SBDOJ sama seperti konsep-konsep *online judge* pada umumnya. SBDOJ merupakan aplikasi berbasis *website*. Aplikasi ini akan digunakan dalam praktikum SBD khusunya pada materi DML yaitu *query* SELECT. Praktikan diharuskan mengumpulkan *query* jawaban dari sebuah soal, kemudian *query* tersebut akan dieksekusi oleh sistem dan akan dinilai kebenarannya. Dengan adanya aplikasi SBDOJ ini, diharapkan bisa mempermudah asisten maupun praktikan dalam praktikum SBD materi DML.

## Rumusan Permasalahan

Rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana cara agar server bisa melakukan proses *judgement* secara cepat?
2. Bagaimana membangun aplikasi berbasis web agar mudah digunakan oleh praktikan dan asisten Sistem Basis Data?

## Batasan Masalah

Permasalahan yang akan diselesaikan dalam Tugas Akhir ini memiliki beberapa batasan, yaitu sebagai berikut:

1. Aplikasi berbasis web dengan bahasa pemrograman PHP.
2. Aplikasi berbasis kerangka kerja *Laravel*.
3. Basis data yang digunakan adalah PostgreSQL dan OracleSQL.
4. *Grader* menggunakan Bahasa pemrograman Python.
5. Sistem digunakan untuk DML query SELECT.

## Tujuan

Tujuan dari pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Membangun aplikasi yang dapat mempermudah praktikum Sistem Basis Data di jurusan Teknik Informatika ITS pada materi DML *query* SELECT.

## Manfaat

Tugas akhir ini diharapkan dapat memberikan solusi pada praktikum Sistem Basis Data materi DML query SELECT di jurusan Teknik Informatika ITS yang selama ini dianggap kurang efektif dengan cara memberikan penilaian secara otomatis kepada query yang dikumpulkan praktikan..

## Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. **Studi Literatur**

Pada tahap ini dilakukan pengumpulan informasi mengenai kerangka kerja Laravel beserta pustaka pendukung lainnya yaitu, bahasa pemgrograman PHP, JavaScript, Python dan *database-python connector*. Selain itu juga dilakukan perumusan awal proses *online judge* pada umumnya, terutama bagaimana jalannya sebuah *grader* yang menjadi kunci utama dari *online judge*. Kemudian akan dilakukan pengumpulan informasi mengenai fitur apa saja yang akan diimplementasikan kepada sistem. Informasi dan pengetahuan yang didapat tersebut digunakan dalam perancangan dan implementasi sistem yang akan dibangun.

1. **Analisis dan Perancangan Sistem**

Pada tahap ini dilakukan analisis dan perancangan untuk aplikasi Sistem Basis Data Online Judge. Perancangan sistem dilakukan dalam beberapa tahap, yaitu analisis aktor yang terlibat di dalam sistem, analisis kebutuhan fungsional yang kemudian dituangkan pada diagram kasus penggunaan, perancangan arsitektur aplikasi, dan perancangan antarmuka pengguna.

1. **Implementasi**

Pada tahap ini dilakukan pembuatan elemen aplikasi yang merupakan implementasi yang berpedoman pada rancangan yang telah dibuat pada proses analisis dan perancangan sistem.

1. **Pengujian dan evaluasi**

Pada tahap ini dilakukan pengujian terhadap elemen aplikasi yang dibangun. Pengujian dan evaluasi perangkat dilakukan untuk mencari masalah yang mungkin timbul, mengevaluasi jalannya program, dan mengadakan perbaikan jika ada kekurangan.

1. **Penyusunan buku Tugas Akhir**

Pada tahap ini dilakukan proses dokumentasi dan pembuatan laporan dari seluruh konsep, tinjauan pustaka, metode, implementasi, proses yang telah dilakukan, pengujian, evaluasi dan hasil-hasil yang didapatkan selama pengerjaan Tugas Akhir.

## Sistematika Penulisan

Penulisan buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran pengerjaan Tugas Akhir dan diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini:

**Bab I Pendahuluan**

Bab ini berisi latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat pembuatan Tugas Akhir, serta metodologi yang digunakan, dan sistematika penulisan buku Tugas Akhir.

**Bab II Tinjauan Pustaka**

Bab ini berisi penjelasan mengenai dasar-dasar penunjang yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

**Bab III Analisis dan Perancangan Sistem**

Bab ini berisi analisis dan perancangan aplikasi yang akan dibangun. Analisi apliksi meliputi deskripsi umum aplikasi, analisis proses bisnis, analisis spesifikasi kebutuhan sistem, dan analisis aktor. Perancangan aplikasi meliputi kasus penggunaan, perancangan arsitektur sistem, dan perancangan antarmuka pada aplikasi.

**Bab IV Implementasi**

Bab ini berisi hasil penerapan perancangan aplikasi yang telah dibuat pada bab sebelumnya. Implementasi dalam Tugas Akhir ini adalah dalam bentuk kode sumber beserta penjelasannya dan implementasi antarmuka pengguna pada aplikasi.

**Bab V Pengujian dan Evaluasi**

Bab ini berisi pengujian dengan metode *black box* untuk mengetahui aspek nilai fungsionalitas dari aplikasi yang dibangun dan nilai kegunaan yang dibuat dengan memperhatikan ketertarikan pada calon penggiuna untuk menggunakan aplikasi ini.

**Bab VI Kesimpulan dan Saran**

Bab ini berisi kesimpulan dari pengerjaan Tugas Akhir ini dan saran untuk pengembangan aplikasi kedepannya.

**Daftar Pustaka**

Merupakan daftar referensi yang digunakan dalam Tugas Akhir ini.

**Lampiran**

Merupakan bab tambahan yang berisi daftar istilah yang penting pada aplikasi ini dan hasil-hasil dari pengujian aplikasi ini.

# BAB II TINJAUAN PUSTAKA

Pada bab ini dijelaskan pustaka penunjang yang berkaitan dengan pembuatan Tugas Akhir ini. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap aplikasi dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

## *Online Judge*

*Online Judge* secara umum adalah server yang berisi deskripsi dari masalah dalam kontes yang berbeda. Serta data yang diatur untuk menilai apakah sebuah solusi tertentu menyelesaikan soal-soal yang berada di server tersebut

## *Database*

Basis data (*database*) adalah kumpulan data, biasanya menjelaskan mengenai aktifitas satu atau lebih organisasi yang terkait. Basis data terdiri dari entitas dan hubungan antar entitas. Basis data biasanya diatur menggunakan sebuah software yaitu *Database Management System* (DBMS)

## SQL

SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk menggambarkan dan memanipulasi basis data relasional. Standar SQL yang digunakan saat ini adalah SQL-99. Ada 2 aspek pada SQL, yaitu DDL (*Data Definition Language*) dan DML (*Data Manipulation Language*)

### Simple Query

Pertanyaan :

Tampilkan NRP,nama, dan Tempat lahir mahasiwa yang mendapatkan nilai E pada FRS urutkan berdasarkan NRP secara descending.

Jawaban :

select distinct f.nrp, m.nama, m.tempat\_lhr

from mahasiswa m, frs f where m.NRP = f.NRP and nilai\_huruf = 'E' order by f.NRP DESC;

Hasil :



### Medium Query

Pertanyaan :

Tampilkan semua data mata kuliah beserta nilai hurufnya pada data FRS yang memiliki nilai angka di atas rata-rata nilai angka pada data nilai serta urutkan berdasarkan ID mata kuliah secara ascending dan berdasarkan nilai huruf secara descending, tanpa ada row yang ganda / terduplikat.

Jawaban :

select distinct mk.\*, f.nilai\_huruf

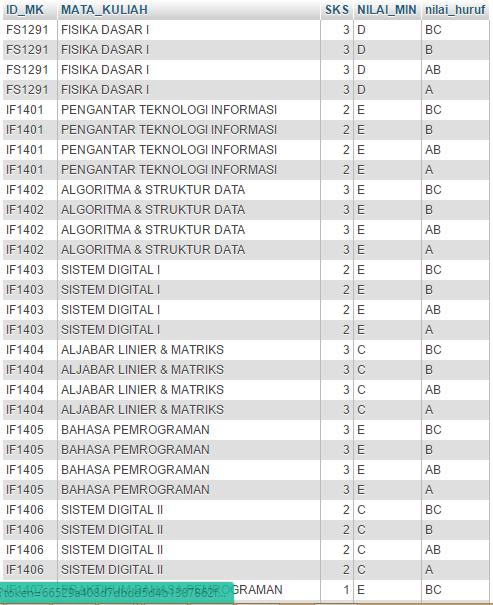
from matakuliah mk, mk\_tawar mkt, frs f

where mk.ID\_MK = mkt.ID\_MK

and mkt.ID\_MK\_TAWAR = f.ID\_MK\_TAWAR

and f.NILAI\_HURUF in (select NILAI\_HURUF from NILAI where nilai\_angka > (select avg(nilai\_angka) from nilai))

order by mk.ID\_MK ASC, f.nilai\_huruf DESC;

Hasil :

### Complex Query

Pertanyaan :

Tampilkan semua data mahasiswa yang mengambil mata kuliah ARSITEKTUR KOMPUTER yang diajar oleh Ibu Sarwosri, S.Kom urutkan berdasarkan NRP secara ascending.

Jawaban :

select distinct m.\*

from mahasiswa m, dosen d, matakuliah mk, mk\_tawar mkt,

frs f

where m.NRP = f.NRP

and f.ID\_MK\_TAWAR = mkt.ID\_MK\_TAWAR

and mk.ID\_MK = mkt.ID\_MK

and d.NIP = mkt.NIP

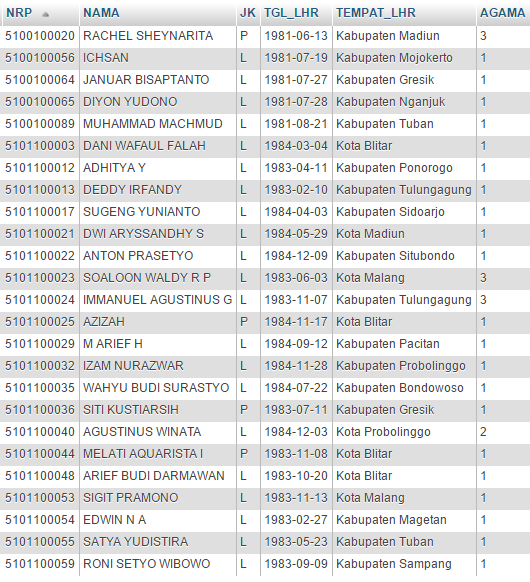
and mk.ID\_MK in (select ID\_MK from matakuliah where

mata\_kuliah = 'ARSITEKTUR KOMPUTER')

and d.NIP = (select NIP from dosen where nama LIKE

'%Sarwosri%' )

order by m.NRP ASC

Hasil :

## Kerangka Kerja Laravel

*Laravel* merupakan *Framework* PHP yang menekankan pada kesederhanaan dan fleksibilitas pada desainnya. *Laravel* dirilis dibawah lisensi MIT dengan sumber kode yang disediakan di Github. Sama seperti *Framework* PHP lainnya, *Laravel* dibangun dengan basis MVC (Model-View-Controller). *Laravel* dilengkapi command line tool ynag bernama “Artisan” yang bisa digunakan untuk packaging bundle dan instalasi bundle. Menurut survey yang lakukan oleh Sitepoint.com pada Desember 2013 dalam popularitas *Framework* PHP, *Laravel* menduduki urutan teratas. Sehingga menjadikan *Laravel* sebagai *Framework* PHP terbaik untuk tahun 2014. Saat ini Lavarel merupakan *Framework* dengan versi PHP yang up-to-date, karena *Laravel* menisyaratkan PHP versi 5.3 keatas.

## PHP

PHP (Hypertext Preprocessor) adalah bahasa skrip yang dapat diletakkan ke dalam HTML. PHP banyak digunakan untuk memrogram situs web dinamis. PHP memiliki beberapa kelebihan jika dibandingkan dengan bahasa pemrograman yang lain, diantaranya adalah :

• Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.

• Web Server yang mendukung PHP dapat ditemukan dimana-mana dari mulai Apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.

• Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan developer yang siap membantu dalam pengembangan.

• Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak.

• PHP adalah bahasa open source yang dapat digunakan berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah sistem.

• PHP memiliki delapan tipe data, yaitu : Boolean, Integer, Float/Double, String, Array, Object, Resource, NULL.

## OracleSQL

Oracle adalah relational database management system (RDBMS) untuk mengelola informasi secara terbuka, komprehensif dan terintegrasi. Oracle menyediakan solusi yang efisien dan efektif karena kemampuannya.

Kekurangan :

- Merupakan software DMBS yang paling mahal, paling rumit, dan paling sulit untuk dipelajari.

- Membutuhkan spesifikasi hardware yang tinggi untuk dapat menjalankan software DMBS Oracle supaya berjalan dengan stabil.

- Hanya diperuntukan bagi perusahaan berukuran besar, dan tidak cocok untuk perusahaan kecil maupun menengah.

- Data yang bertambah ukurannya akan mengalami kelambatan proses, jadi harus ada database management.

Kelebihan :

- Merupakan software DBMS yang handal dan memiliki kemampuan yang tinggi.

- Dapat menangani jumlah data dalam ukuran yang besar.

- Dapat mengolah data dalam ukuran besar dan mengolahnya dengan cepat sehingga didapatkan informasi yang akurat sesuai permintaan pengguna.

- Memiliki kemampuan akan fleksibilitas dan skalabilitas yang dapat memenuhi tuntutan akan data dan informasi yang bervolume besar dan terus-menerus bertambah besar.

- Memiliki kemampuan Technology Cluster Server, dimana jika terdapat lebih dari satu unit server misalnya 100 unit server maka Oracle dapat menjadikan 100 unit server tersebut aktif bekerja bersama sebagai 100 aktif server.

- Memiliki kemampuan untuk manajemen user dan tiap user bisa diatur hak akses terhadap suatu database oleh database administrator.

- Bisa berjalan pada lebih dari satu platform sistem operasi.

- Ketika kita mengakses database dan kemudian ada kejadian seperti listrik mati misalnya maka data yang sudah kita simpan tidak rusak/hilang. Oracle memiliki kemampuan flashback, sehingga semua jenis transaksi yang salah akan dapat dikembalikan. Dan dapat menampung data dalam sekala besar.

- Dapat bekerja di lingkungan client/server (pemrosesan tersebar).

*[Halaman ini sengaja dikosongkan]*

# BAB III ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini menjelaskan tahap analisis dan perancangan dari aplikasi yang akan dibangun pada Tugas Akhir ini..

## Analisis Sistem

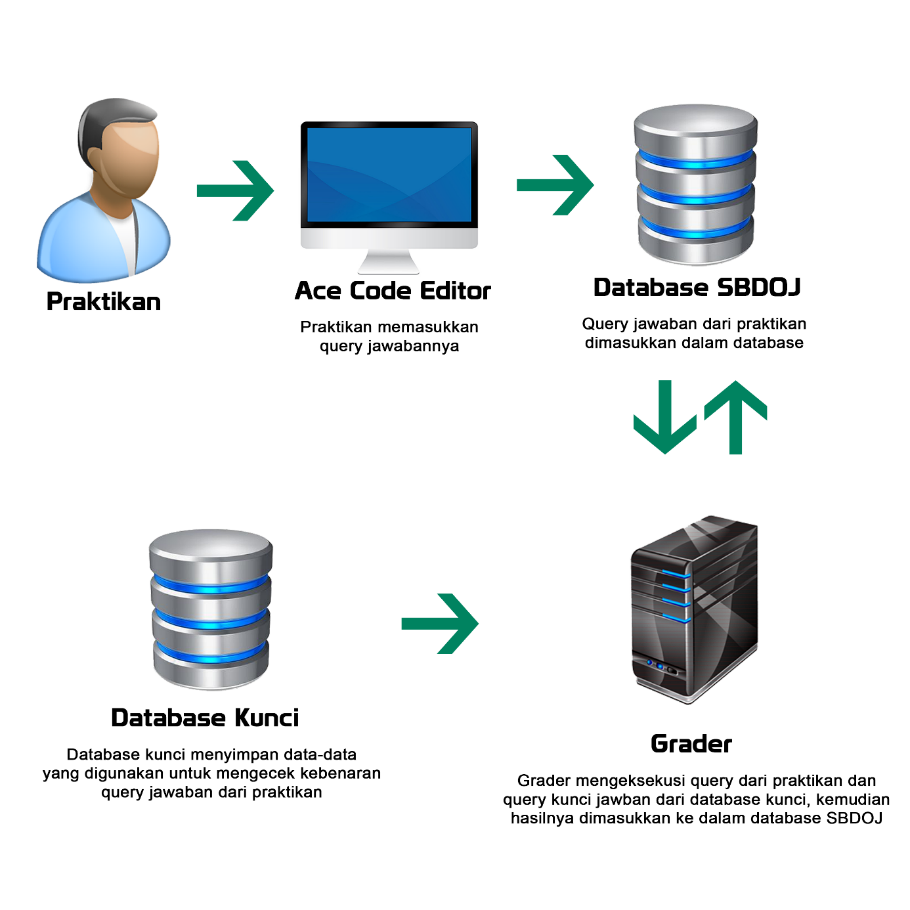
Tahap analisis sistem ini terbagi menjadi beberapa bagian antara lain deskripsi umum perangkat lunak, analisis proses bisnis, spesifikasi kebutuhan sistem, dan identifikasi aktor.

### Deskripsi Umum Aplikasi

Sistem Basis Data Online Judge adalah aplikasi yang ditujukan untuk mempermudah penilaian praktikum Sistem Basis Data di Jurusan Teknik Informatika ITS khususnya pada materi DML.

Pada aplikasi ini, asisten bias memasukkan soal-soal yang digunakan untuk praktikum beserta jawaban dari soal tersebut. Kemudian praktikan akan memasukkan *query* jawaban dari masing-masing soal yang kemudian akan dicek oleh *grader* untuk mengetahui benar atau tidaknya jawaban dari praktikan.

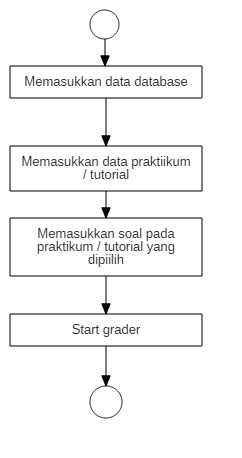
Selain digunakan untuk praktikum, SBDOJ juga bisa digunakan asisten untuk memberi *tutorial* kepada mahasiswa kelas. Di setiap *event* *tutorial* dan praktikum, terdapat scoreboard untuk melihat rekapitulasi nilai dari mahasiswa. Nilai tersebut antara lain 100 untuk jawaban benar dan tidak melewati batas waktu eksekusi, 0 untuk jawaban salah dan TLE (*time limit exceeded*) untuk jawaban yang melebihi batas waktu eksekusi. Dengan adanya aplikasi SBDOJ ini, asisten akan dimudahkan unruk mendapatkan nilai dari mahasiswa kelasnya dalam praktikum Sistem Basis Data. Deskripsi umum aplikasi dapat dilihat pada Gambar 3.1.



Gambar 3.1 Deskripsi Umum Perangkat Lunak

### Analisis Proses Bisnis

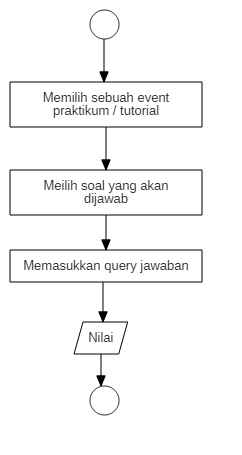
Proses bisnis yang digunakan dalam aplikasi ini adalah proses bisnis yang umum terdapat pada *online judge* lainnya. Proses bisnis pada sisi asisten dapat dilihat pada diagram alir pada Gambar 3.2.



Gambar 3.2 Diagram Alir untuk Asisten

Pada gambar di atas, asisten harus memasukkan database kunci atau database yang akan digunakan untuk praktikum. Kemudian asisten harus membuat event praktikum atau tutorial. Langkah selanjutnya adalah membuat soal untuk praktikum tersebut. Soal harus disertai quer jawaban yang benar dari asisten untuk proses penilaian. Kemudian asisten harus memulai grader dari database yang digunakan agar penilaian jawaban dari praktikan bisa berlangsung di *background*.

Pada aplikasi ini, data praktian dimasukkan oleh asisten masing-masing kelas. Jadi praktikan tidak bisa mendaftarkan dirinya. Proses bisnis pada sisi praktikan dapat dilihat pada diagram alir Gambar 3.3.



Gambar 3.3 Diagram Alir Proses untuk Praktikan

Pada gambar tersebut, praktikan bisa memilih event praktikum atau tutorial. Event praktikum bisa dipilih apabila waktu sudah memasuki waktu mulai praktikum dan belum melewati waktu akhir praktikum. Praktikan kemudian memaukkan query jawaban pada soal yang dipilih. Setelah itu *grader* akan mengecek jawaban praktikan dan akan memberikan nilai dari jawaban tersebut.

### Analisis Kebutuhan Sistem

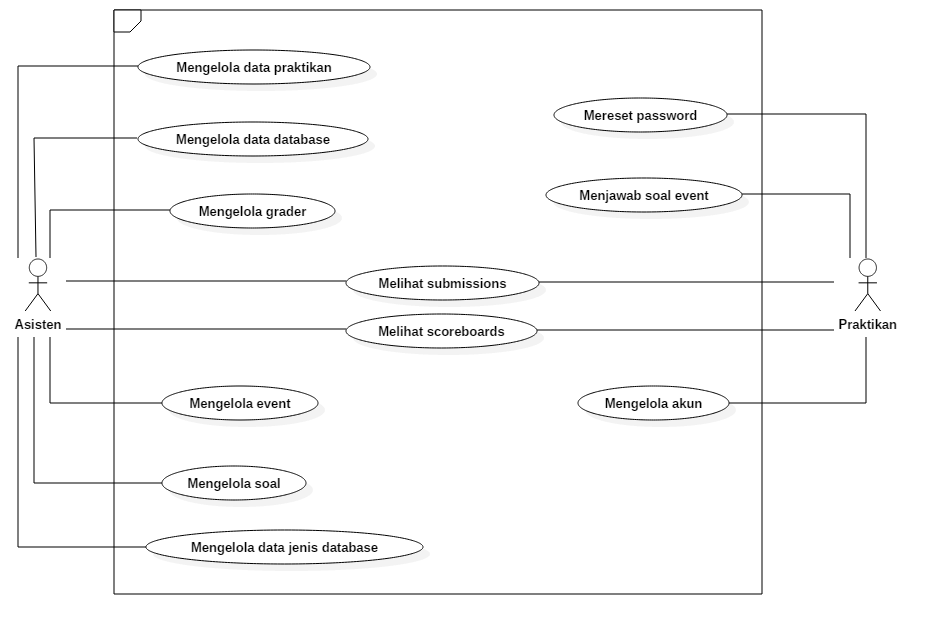
Analisis spesifikasi kebutuhan dalam aplikasi ini mencakup kebutuhan fungsional. Kebutuhan fungsional berisikan proses-proses yang dibutuhkan dalam sistem dan harus dijalankan. Kebutuhan fungsional didapatkan dari analisis proses bisnis yang telah dijelaskan sebelumnya. Kebutuhan fungsional sistem dideskripsikan dalam Tabel 3.1.

Tabel 3.1 Daftar Kebutuhan Fungsional Sistem

| Kode Kebutuhan | Kebutuhan Fungsional | Deskripsi |
| --- | --- | --- |
| F-001 | Mengelola data praktikan | Asisten dapat mengelola data praktikan masing-masing kelasnya |
| F-002 | Mengelola data *database* | Asisten dapat mengelola *database* yang digunakan untuk praktikum atau *tutorial*. |
| F-003 | Mengelola data jenis *database* | Asisten dapat mengelola jenis *database* yang digunakan untuk praktikum atau *tutorial*. |
| F-004 | Mengelola *grader* | Asisten dapat memulai atau memberhentikan *grader* tiap-tiap database. |
| F-005 | Mengelola *event* | Assiten dapat mengelola data praktikum dan *tutorial* |
| F-006 | Mengelola soal | Asisten dapat mengelola soal dari praktikum aatau *tutorial* |
| F-007 | Melihat *submissions* | Asisten dan praktikan dapat melihat jawaban dari praktikan yang telah di-*submit*. |
| F-008 | Melihat *scoreboards* | Asisten dan praktikan dapat melihat *scoreboard* dari praktikum |
| F-009 | Menjawab soal *event* | Praktikan dapat memasukkan *query* jawaban dari soal-soal praktikum atau *tutorial*. |
| F-010 | Mengelola akun | Praktikan dapat mengelola akunnya masing-masing. |
| F-011 | Me-*reset* password | Praktikan bisa melakukan permintaan pengubaan password apabila praktikan lupa password akunnya. |

### Analisis Aktor

Aktor mendefinisikan *stakeholder* yang terlibat dan berinteraksi langsung dengan sistem. *Stakeholder* ini bisa berupa manusia maupun sistem atau perangkat lunak yang lain. *Stakeholder* yang terdapat pada sistem ini hanya sebagai pengguna. Pengguna yang didefinisikan untuk perangkat lunak ini adalah asisten dan praktikan. Detail tugas dan hak akses pengguna dicantumkan pada Gambar 3.4.



Gambar 3.4 Diagram Kasus Penggunaan

## Perancangan Sistem

Tahap perancangan sistem dibagi menjadi beberapa bagian yaitu kasus penggunaan, perancangan arsitektur aplikasi, dan perancangan antarmuka aplikasi.

### Kasus Penggunaan

Kasus penggunaan yang dibutuhkan pada sistem sesuai dengan analisa yang telah dilakukan sebelumnya pada spesifikasi kebutuhan fungsional dan indentifikasi aktor. Diagram kasus penggunaan dapat dilihat pada Gambar 3.4 dan kode kasus penggunaan pada Tabel 3.2

Tabel 3.2 Keterangan Kode Kasus Penggunaan

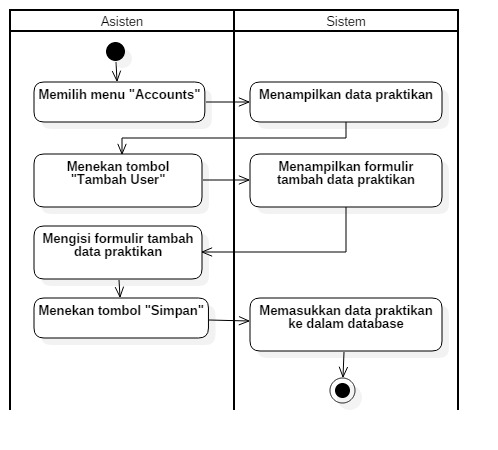
|  |  |
| --- | --- |
| Kode Kasus Penggunaan | Kasus Penggunaan |
| UC-001 | Mengelola data praktikan |
| UC-002 | Mengelola data *database* |
| UC-003 | Mengelola data jenis *database* |
| UC-004 | Mengelola *grader* |
| UC-005 | Mengelola *event* |
| UC-006 | Mengelola soal |
| UC-007 | Melihat *submissions* |
| UC-008 | Melihat *scoreboards* |
| UC-009 | Menjawab soal *event* |
| UC-010 | Mengelola akun |
| UC-011 | Me-*reset* password |

#### Kasus Penggunaan Mengelola Data Praktikan

Pada kasus penggunaan ini, asisten dapat mngelola data praktikan dari masing-masing kelasnya. Pengelolaan terdiri dari penambahan data praktikan, pengubahan data praktikan dan penghapusan data praktikan. Penambahan data praktikan bisa melalui formulir atau dengan mengunggah file excel.

Tabel 3.3 Spesifikasi Kasus Penggunaan Menambah Data Praktikan

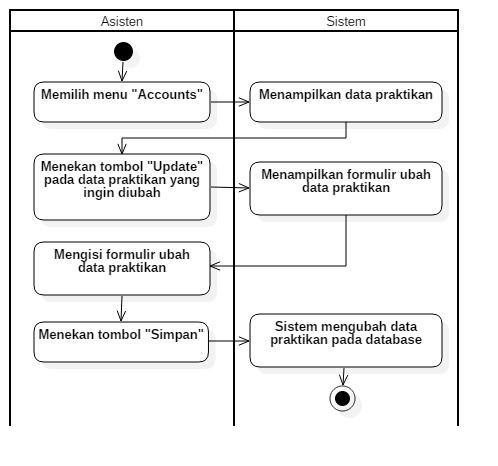
|  |  |
| --- | --- |
| Kode | UC-001.1 |
| Nama | **Menambah Data Praktikan** |
| Aktor | Asisten |
| Deskripsi | Asisten menambah data praktikan kelasnya |
| Tipe | Fungsional |
| Kondisi Awal | Asisten telah *login* ke dalam aplikasi. |
| Kondisi Akhir | Data praktikan telah ditambahkan pada database |
| Alur Kejadian Normal | |
| 1. Asisten memilih menu “Accounts” | |
| 1. Sistem menampilkan data praktikan | |
| 1. Asisten menekan tombol “Tambah User” | |
| 1. Sistem menampilkan formulir tambah data praktikan 2. Asisten mengisi formulir tambah data praktikan 3. Asisten menekan tombol simpan 4. Sistem memasukkan data praktikan ke dalam database | |
| Alur Kejadian Alternatif | |
| Tidak ada. | |



Gambar 3.5 Diagram Aktivitas Menambah Data Praktikan

Tabel 3.4 Spesifikasi Kasus Penggunaan Mengubah Data Praktikan

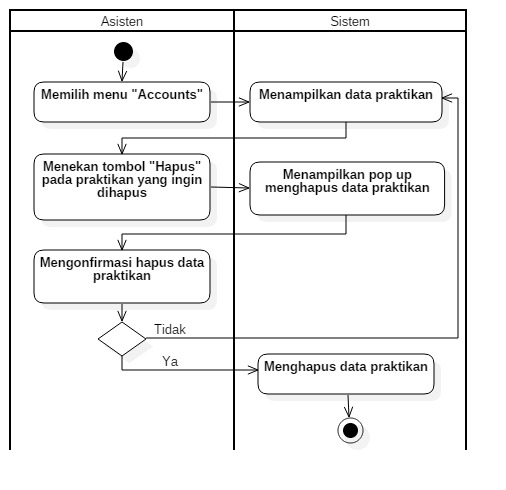
|  |  |
| --- | --- |
| Kode | UC-001.2 |
| Nama | **Mengubah Data Praktikan** |
| Aktor | Asisten |
| Deskripsi | Asisten mengubah data praktikan kelasnya |
| Tipe | Fungsional |
| Kondisi Awal | Asisten telah *login* ke dalam aplikasi. |
| Kondisi Akhir | Data praktikan telah diubah pada database |
| Alur Kejadian Normal | |
| 1. Asisten memilih menu “Accounts” | |
| 1. Sistem menampilkan data praktikan | |
| 1. Asisten menekan tombol “Update” pada data praktikan yang ingin diubah | |
| 1. Sistem menampilkan formulir ubah data praktikan 2. Asisten mengisi formulir ubah data praktikan 3. Asisten menekan tombol simpan 4. Sistem mengubah data praktikan pada database | |
| Alur Kejadian Alternatif | |
| Tidak ada. | |



Gambar 3.6 Diagram Aktivitas Mengubah Data Praktikan

Tabel 3.5 Spesifikasi Kasus Penggunaan Menghapus data praktikan

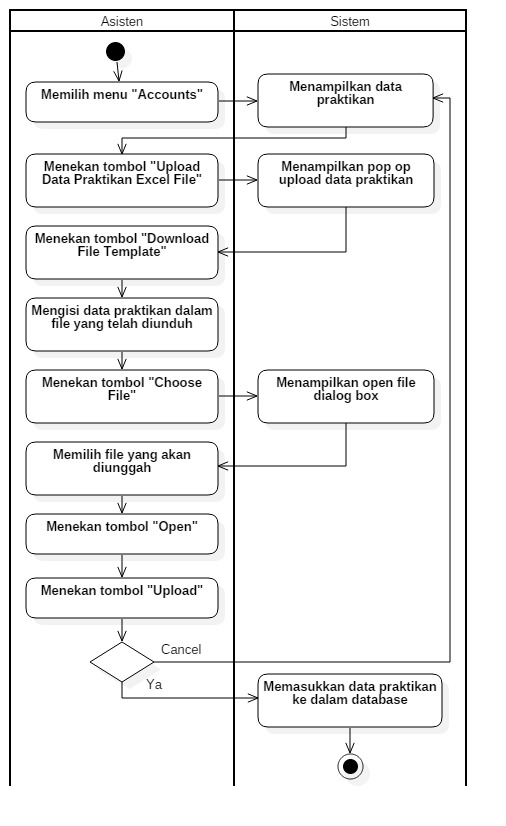
|  |  |
| --- | --- |
| Kode | UC-001.3 |
| Nama | **Menghapus Data Praktikan** |
| Aktor | Asisten |
| Deskripsi | Asisten menghapus data praktikan kelasnya |
| Tipe | Fungsional |
| Kondisi Awal | Asisten telah *login* ke dalam aplikasi. |
| Kondisi Akhir | Data praktikan telah dihapus pada database |
| Alur Kejadian Normal | |
| 1. Asisten memilih menu “Accounts”. | |
| 1. Sistem menampilkan data praktikan. | |
| 1. Asisten menekan tombol hapus pada praktikan yang ingin dihapus. | |
| 1. Sistem menampilkan *pop-up* menghapus data praktikan. 2. Asisten menekan tombol ya.   A.5 Asisten menekan tombol tidak.   1. Sistem menghapus data praktikan. | |
| Alur Kejadian Alternatif | |
| A.5. Asisten menekan tombol Tidak. | |
| 1. Kembali ke Alur Kejadian Normal nomor 2. | |



Gambar 3.7 Diagram Aktivitas Meghapus Data Praktikan

Tabel 3.6 Spesifikasi Kasus Menambah Data Praktikan dengan *File* Excel

|  |  |
| --- | --- |
| Kode | UC-001.4 |
| Nama | **Menambah Data Praktikan dengan *File* Excel** |
| Aktor | Asisten |
| Deskripsi | Asisten menambah data praktikan kelasnya dengan mengunggah file excel |
| Tipe | Fungsional |
| Kondisi Awal | Asisten telah *login* ke dalam aplikasi. |
| Kondisi Akhir | Data praktikan telah ditambahkan pada database |
| Alur Kejadian Normal | |
| 1. Asisten memilih menu “Accounts” | |
| 1. Sistem menampilkan data praktikan | |
| 1. Asisten menekan tombol “Upload Data Praktikan (Excel File)” | |
| 1. Sistem menampilkan *pop-up* Upload Data Praktikan 2. Asisten mengunduh *template file* Excel dengan menekan tombol “Download File Template” 3. Asisten mengisikan data-data praktikanya ke dalam *file* yang telah diunduh kemudian disimpan 4. Asisten menekan tombol “Choose File” 5. Sistem menampilkan *pop-up* untuk asisten memilih *file* yang akan diunggah 6. Asisten menekan tombol “Open” 7. Asisten menekan tombol “Upload”   A.10 Asisten menekan tombol “Cancel”  11. Sistem memasukkan data praktikan ke dalam database | |
| Alur Kejadian Alternatif | |
| A.10. Asisten menekan tombol “Cancel”. | |
| A.10.2 Kembali ke Alur Kejadian Normal nomor 2. | |



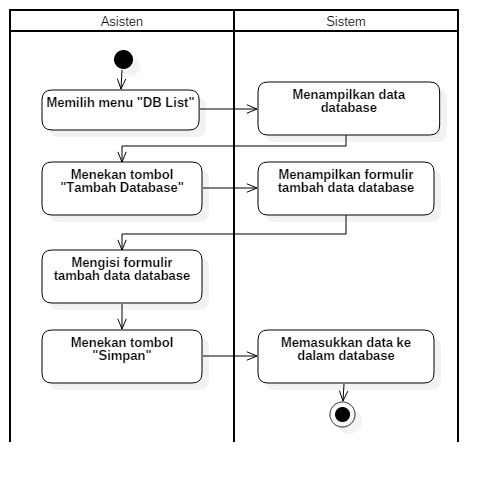
Gambar 3.8 Diagram Aktivitas Menambah Data Praktikan dengan *File* Excel

#### Kasus Penggunaan Mengelola Data *Database*

Pada kasus penggunaan ini, Asisten dapat mengelola data database yang akan digunakan untuk praktikum atau *tutorial*. Pengelola terdiri dari penambahan data *database*, pengubahan data *database*,penghapusan data *database*.

Tabel 3.7 Spesifikasi Kasus Penggunaan Menambah Data *Database*

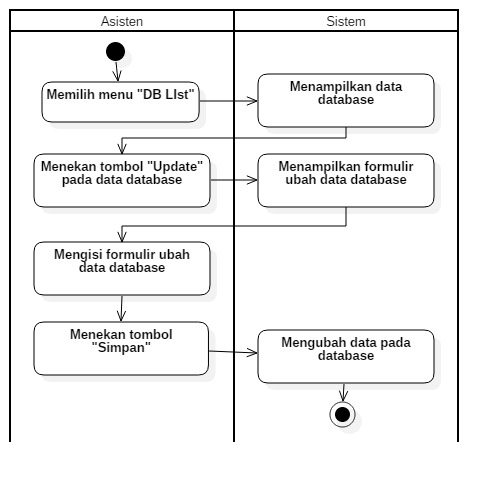
|  |  |
| --- | --- |
| Kode | UC-002.1 |
| Nama | **Menambah Data *Database*** |
| Aktor | Asisten |
| Deskripsi | Asisten menambah data *database* untuk praktikum atau *tutorial* kelasnya |
| Tipe | Fungsional |
| Kondisi Awal | Asisten telah *login* ke dalam aplikasi. |
| Kondisi Akhir | Data *database* telah ditambahkan pada *database* |
| Alur Kejadian Normal | |
| 1. Asisten memilih menu “DB List” | |
| 1. Sistem menampilkan data *database* | |
| 1. Asisten menekan tombol “Tambah *Database*” | |
| 1. Sistem menampilkan formulir tambah data *Database* 2. Asisten mengisi formulir tambah data *database* 3. Asisten menekan tombol simpan 4. Sistem memasukkan data *database* ke dalam *database* | |
| Alur Kejadian Alternatif | |
| Tidak ada. | |



Gambar 3.9 Diagram Aktivitas Menambah Data *Database*

Tabel 3.8 Spesifikasi Kasus Penggunaan Mengubah Data *Database*

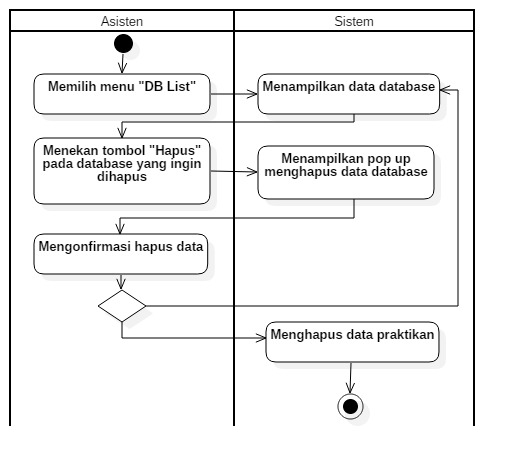
|  |  |
| --- | --- |
| Kode | UC-002.2 |
| Nama | **Mengubah Data *Database*** |
| Aktor | Asisten |
| Deskripsi | Asisten mengubah data *database* |
| Tipe | Fungsional |
| Kondisi Awal | Asisten telah *login* ke dalam aplikasi. |
| Kondisi Akhir | Data *database* telah diubah pada *database* |
| Alur Kejadian Normal | |
| 1. Asisten memilih menu “DB List” | |
| 1. Sistem menampilkan data *database* | |
| 1. Asisten menekan tombol “Update” pada data *database* yang ingin diubah | |
| 1. Sistem menampilkan formulir ubah data *database* 2. Asisten mengisi formulir ubah data *database* 3. Asisten menekan tombol simpan 4. Sistem mengubah data *database* pada *database* | |
| Alur Kejadian Alternatif | |
| Tidak ada. | |



Gambar 3.10 Diagram Aktivitas Mengubah Data *Database*

Tabel 3.9 Spesifikasi Kasus Penggunaan Menghapus Data *Database*

|  |  |
| --- | --- |
| Kode | UC-002.3 |
| Nama | **Menghapus Data *Database*** |
| Aktor | Asisten |
| Deskripsi | Asisten menghapus data *database* |
| Tipe | Fungsional |
| Kondisi Awal | Asisten telah *login* ke dalam aplikasi. |
| Kondisi Akhir | Data *database* telah dihapus pada *database* |
| Alur Kejadian Normal | |
| 1. Asisten memilih menu “DB List”. | |
| 1. Sistem menampilkan data *database*. | |
| 1. Asisten menekan tombol hapus pada *database* yang ingin dihapus. | |
| 1. Sistem menampilkan *pop-up* menghapus data *database*. 2. Asisten menekan tombol ya.   A.5 Asisten menekan tombol tidak.   1. Sistem menghapus data praktikan. | |
| Alur Kejadian Alternatif | |
| A.5. Asisten menekan tombol Tidak. | |
| 1. Kembali ke Alur Kejadian Normal nomor 2. | |



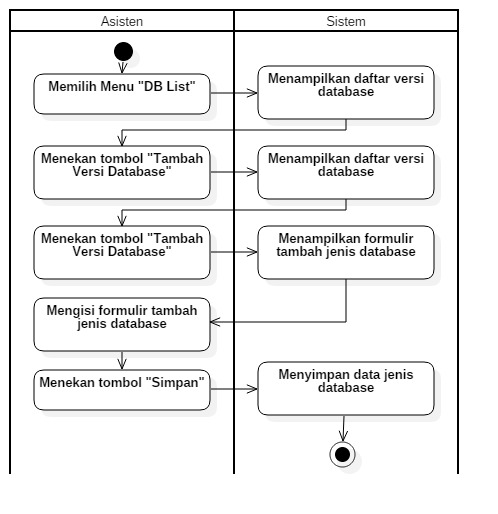
Gambar 3.11 Diagram Aktivitas Menghapus Data *Database*

#### Kasus Penggunaan Mengelola Jenis *Database*

Pada kasus penggunaan ini, asisten dapat mengelola jenis-jenis *database* yang digunakan pada sistem. Jenis *database* antara lain Oracle SQL, MySQL, PostgreSQL, dan lain-lain. Asisten diharuskan meng-*install plugin* untuk menghubungkan antara Python dengan *database* yang akan digunakan. Pengelolaan terdiri dari penambahan jenis *database* dan pengubahan jenis *database*.

Tabel 3.10 Spesifikasi Kasus Penggunaan Menambah Jenis *Database*

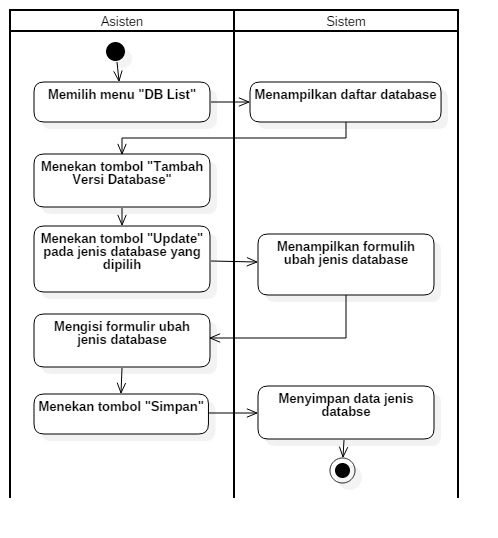
|  |  |
| --- | --- |
| Kode | UC-003.1 |
| Nama | **Menambah Jenis *Database*** |
| Aktor | Asisten |
| Deskripsi | Asisten menambah data jenis *database*. |
| Tipe | Fungsional |
| Kondisi Awal | Asisten telah *login*. |
| Kondisi Akhir | Jenis *database* telah ditambahkan. |
| Alur Kejadian Normal | |
| Asisten memilih menu “DB *List*”. | |
| Sistem menampilkan daftar *database*. | |
| Pelanggan menekan tombol “Tambah Versi *Database*”. | |
| Sistem menampilkan daftar versi *database*  Asisten menekan tombol “Tambah Versi *Database*”.  Sistem menampilkan formulir tambah jenis *database.*  Pelanggan mengisi formulir tambah jenis *database*.  Pelanggan menekan tombol simpan.  Sistem menyimpan data jenis database. | |
| Alur Kejadian Alternatif | |
| Tidak ada | |



Gambar 3.12 Diagram Aktivitas Menambah Jenis *Database*

Tabel 3.12 Spesifikasi Kasus Penggunaan Mengubah Data Jenis *Database*

|  |  |
| --- | --- |
| Kode | UC-003.2 |
| Nama | **Mengubah Jenis Database** |
| Aktor | Asisten |
| Deskripsi | Asisten mengubah data jenis *database*. |
| Tipe | Fungsional |
| Kondisi Awal | Pelanggan telah *login*. |
| Kondisi Akhir | Data jenis *database* telah diubah. |
| Alur Kejadian Normal | |
| Pelanggan memilih menu “DB *List*”. | |
| Sistem menampilkan daftar *database*. | |
| Asisten menekan tombol “Tambah Versi *Database*”. | |
| Sistem menampilkan daftar versi *database*  Asisten menekan tombol “*Update*” pada jenis *database* yang dipilih.  Sistem menampilkan formulir ubah jenis *database.*  Asisten mengisi formulir ubah jenis *database*.  Asisten menekan tombol simpan.  Sistem menyimpan data jenis database. | |
| Alur Kejadian Alternatif | |
| Tidak Ada | |



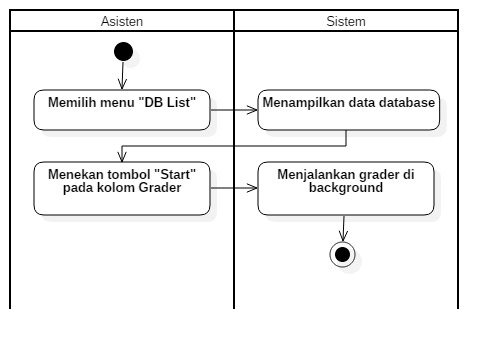
Gambar 3.13 Diagram Aktivitas Mengubah Jenis *Database*

#### Kasus Penggunaan Mengelola *Grader*

Pada kasus penggunaan ini, Asisten dapat mengelola *grader*. Grader digunakan untuk mengecek jawaban dari praktikan dan memberi nilai dari jawaban tersebut. Pengelolaan terdiri dari memulai grader dan menghentikan grader. *Grader* terdapat 2 jenis, yaitu untuk praktikum dan untuk *tutorial*.

Tabel 3.13 Spesifikasi Kasus Penggunaan Memulai *Grader*

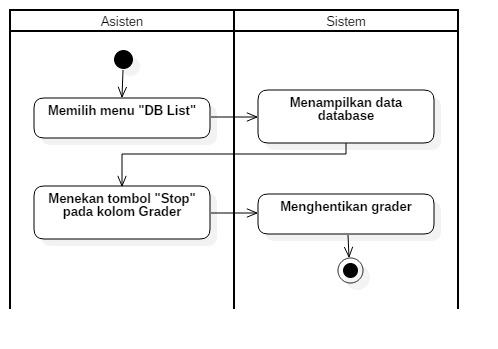
|  |  |
| --- | --- |
| Kode | UC-004.1 |
| Nama | **Memulai *Grader*** |
| Aktor | Asisten |
| Deskripsi | Asisten menjalankan grader apabila praktikum atau *tutorial* akan dimulai |
| Tipe | Fungsional |
| Kondisi Awal | Asisten telah *login* ke dalam aplikasi. |
| Kondisi Akhir | *Grader*  telah berjalan di *background* |
| Alur Kejadian Normal | |
| 1. Asisten memilih menu “DB List” | |
| 1. Sistem menampilkan data *database* | |
| 1. Asisten menekan tombol “*Start*” pada kolom *Grader* atau *Grader Tutorial* | |
| 1. Sistem menjalankan *grader* di *background* | |
| Alur Kejadian Alternatif | |
| Tidak ada. | |



Gambar 3.14 Diagram Aktivitas Memulai *Grader*

Tabel 3.13 Spesifikasi Kasus Penggunaan Mengubah Data Praktikan

|  |  |
| --- | --- |
| Kode | UC-004.2 |
| Nama | **Menghentikan *Grader*** |
| Aktor | Asisten |
| Deskripsi | Asisten menghentikan grader apabila praktikum atau *tutorial* telah selesai |
| Tipe | Fungsional |
| Kondisi Awal | Asisten telah *login* ke dalam aplikasi. |
| Kondisi Akhir | *Grader* telah berhenti |
| Alur Kejadian Normal | |
| 1. Asisten memilih menu “DB List” | |
| 1. Sistem menampilkan data *database* | |
| 1. Asisten menekan tombol “*Stop*” pada kolom *Grader* atau *Grader Tutorial* | |
| 1. Sistem mengehntikan *grader* | |
| Alur Kejadian Alternatif | |
| Tidak ada. | |



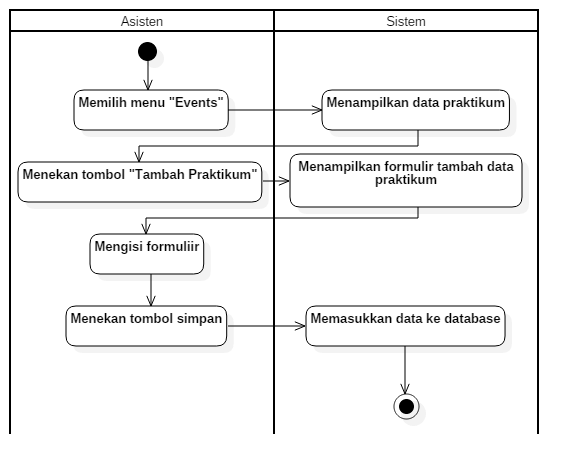
Gambar 3.15 Diagram Aktivitas Menghentikan *Grader*

#### Kasus Penggunaan Mengelola *Event*

Pada kasus penggunaan ini, asisten dapat mengelola data *event*. *Event* terdiri dari praktikum dan *tutorial*. Pengelolaan terdiri dari menambah praktikum, mengubah praktikum, menghapus praktikum, menambah *tutorial*, mengubah *tutorial*, menghapus *tutorial*, menambah kategori *tutorial*, mengubah kategori *tutorial*, dan menghapus kategori *tutorial*.

Tabel 3.14 Spesifikasi Kasus Penggunaan Menambah Praktikum

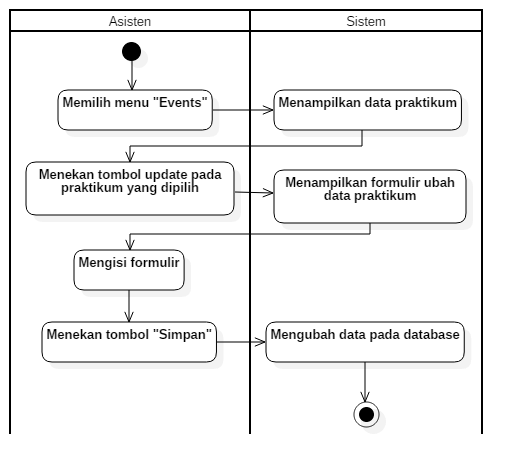
|  |  |
| --- | --- |
| Kode | UC-005.1 |
| Nama | **Menambah Praktikum** |
| Aktor | Asisten |
| Deskripsi | Asisten menambah data praktikum kelasnya |
| Tipe | Fungsional |
| Kondisi Awal | Asisten telah *login* ke dalam aplikasi. |
| Kondisi Akhir | Data praktikum telah ditambahkan pada *database* |
| Alur Kejadian Normal | |
| 1. Asisten memilih menu “*Events*” | |
| 1. Sistem menampilkan data praktikum | |
| 1. Asisten menekan tombol “Tambah *Event*” | |
| 1. Sistem menampilkan formulir tambah data praktikum 2. Asisten mengisi formulir tambah data praktikum 3. Asisten menekan tombol simpan 4. Sistem memasukkan data praktikumke dalam *database* | |
| Alur Kejadian Alternatif | |
| Tidak ada. | |



Gambar 3.16 Diagram Aktivitas Menambah Praktikum

Tabel 3.15 Spesifikasi Kasus Penggunaan Mengubah Praktikum

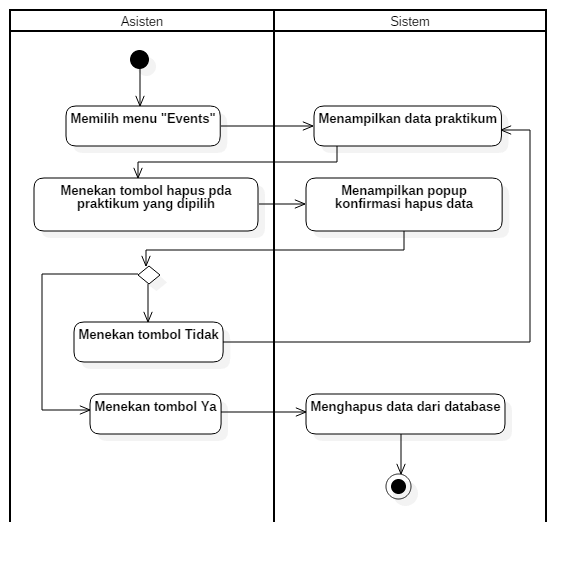
|  |  |
| --- | --- |
| Kode | UC-005.2 |
| Nama | **Mengubah Praktikum** |
| Aktor | Asisten |
| Deskripsi | Asisten mengubah data praktikum |
| Tipe | Fungsional |
| Kondisi Awal | Asisten telah *login* ke dalam aplikasi. |
| Kondisi Akhir | Data praktikum telah diubah pada *database* |
| Alur Kejadian Normal | |
| 1. Asisten memilih menu “*Events*” | |
| 1. Sistem menampilkan data praktikum | |
| 1. Asisten menekan tombol “Update” pada data praktikum yang ingin diubah | |
| 1. Sistem menampilkan formulir ubah data praktikum 2. Asisten mengisi formulir ubah data praktikum 3. Asisten menekan tombol simpan 4. Sistem mengubah data praktikum pada *database* | |
| Alur Kejadian Alternatif | |
| Tidak ada. | |



Gambar 3.17 Diagram Aktivitas Mengubah Praktikum

Tabel 3.16 Spesifikasi Kasus Penggunaan Menghapus Praktikum

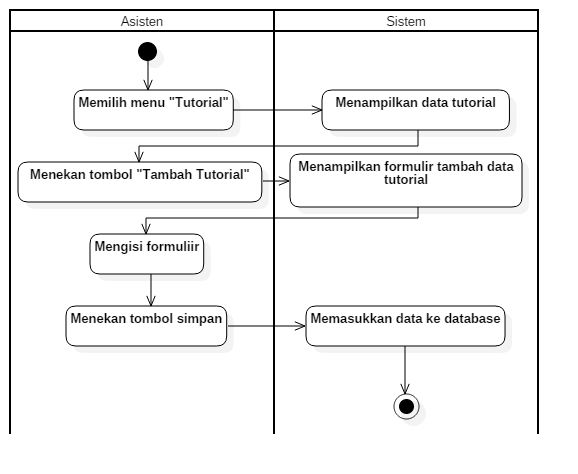
|  |  |
| --- | --- |
| Kode | UC-005.3 |
| Nama | **Menghapus Praktikum** |
| Aktor | Asisten |
| Deskripsi | Asisten menghapus data praktikum |
| Tipe | Fungsional |
| Kondisi Awal | Asisten telah *login* ke dalam aplikasi. |
| Kondisi Akhir | Data praktikumtelah dihapus pada *database* |
| Alur Kejadian Normal | |
| 1. Asisten memilih menu “*Events*”. | |
| 1. Sistem menampilkan data praktikum. | |
| 1. Asisten menekan tombol hapus pada praktikum yang ingin dihapus. | |
| 1. Sistem menampilkan *pop-up* menghapus data praktikum. 2. Asisten menekan tombol ya.   A.5 Asisten menekan tombol tidak.   1. Sistem menghapus data praktikum. | |
| Alur Kejadian Alternatif | |
| A.5. Asisten menekan tombol Tidak. | |
| 1. Kembali ke Alur Kejadian Normal nomor 2. | |



Gambar 3.18 Diagram Aktivitas Menghapus Praktikum

Tabel 3.17 Spesifikasi Kasus Penggunaan Menambah *Tutorial*

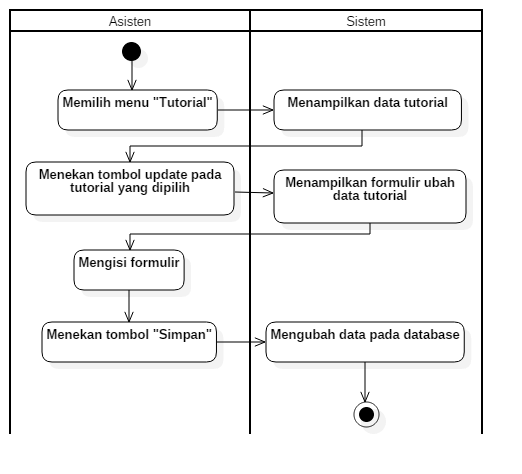
|  |  |
| --- | --- |
| Kode | UC-005.4 |
| Nama | **Menambah *Tutorial*** |
| Aktor | Asisten |
| Deskripsi | Asisten menambah data *tutorial* kelasnya |
| Tipe | Fungsional |
| Kondisi Awal | Asisten telah *login* ke dalam aplikasi. |
| Kondisi Akhir | Data *tutorial* telah ditambahkan pada *database* |
| Alur Kejadian Normal | |
| 1. Asisten memilih menu “*Tutorial*” | |
| 1. Sistem menampilkan data *tutorial* | |
| 1. Asisten menekan tombol “Tambah *Tutorial*” | |
| 1. Sistem menampilkan formulir tambah data *tutorial* 2. Asisten mengisi formulir tambah data *tutorial* 3. Asisten menekan tombol simpan 4. Sistem memasukkan data *tutorial* ke dalam *database* | |
| Alur Kejadian Alternatif | |
| Tidak ada. | |



Gambar 3.19 Diagram Aktivitas Menambah *Tutorial*

Tabel 3.18 Spesifikasi Kasus Penggunaan Mengubah *Tutorial*

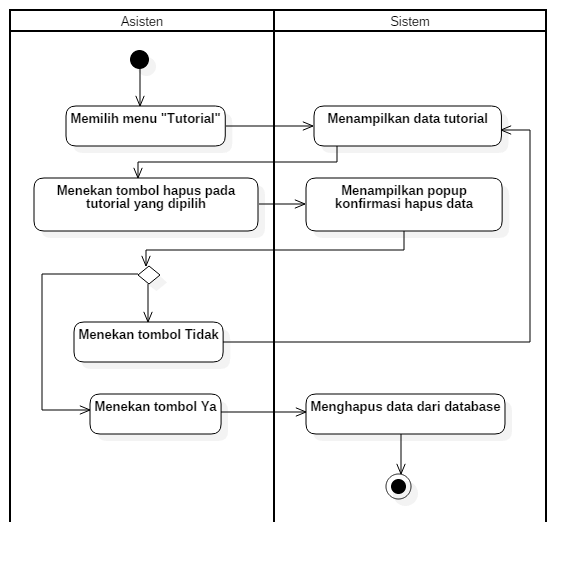
|  |  |
| --- | --- |
| Kode | UC-005.5 |
| Nama | **Mengubah *Tutorial*** |
| Aktor | Asisten |
| Deskripsi | Asisten mengubah data *tutorial* |
| Tipe | Fungsional |
| Kondisi Awal | Asisten telah *login* ke dalam aplikasi. |
| Kondisi Akhir | Data *tutorial* telah diubah pada *database* |
| Alur Kejadian Normal | |
| 1. Asisten memilih menu “*Tutorial*” | |
| 1. Sistem menampilkan data *tutorial* | |
| 1. Asisten menekan tombol “Update” pada data *tutorial* yang ingin diubah | |
| 1. Sistem menampilkan formulir ubah data *tutorial* 2. Asisten mengisi formulir ubah data *tutorial* 3. Asisten menekan tombol simpan 4. Sistem mengubah data *tutorial* pada *database* | |
| Alur Kejadian Alternatif | |
| Tidak ada. | |



Gambar 3.20 Diagram Aktivitas Mengubah *Tutorial*

Tabel 3.19 Spesifikasi Kasus Penggunaan Menghapus *Tutorial*

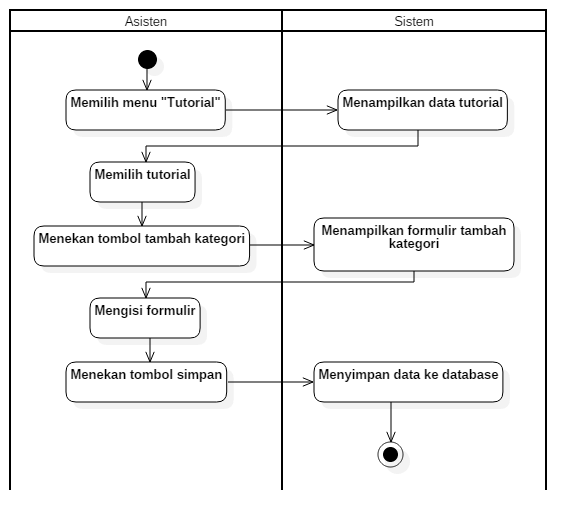
|  |  |
| --- | --- |
| Kode | UC-005.6 |
| Nama | **Menghapus *Tutorial*** |
| Aktor | Asisten |
| Deskripsi | Asisten menghapus data *tutorial* |
| Tipe | Fungsional |
| Kondisi Awal | Asisten telah *login* ke dalam aplikasi. |
| Kondisi Akhir | Data *tutorial* telah dihapus pada *database* |
| Alur Kejadian Normal | |
| 1. Asisten memilih menu “*Tutorial*”. | |
| 1. Sistem menampilkan data *tutorial*. | |
| 1. Asisten menekan tombol hapus pada *tutorial* yang ingin dihapus. | |
| 1. Sistem menampilkan *pop-up* menghapus data *tutorial*. 2. Asisten menekan tombol ya.   A.5 Asisten menekan tombol tidak.   1. Sistem menghapus data *tutorial*. | |
| Alur Kejadian Alternatif | |
| A.5. Asisten menekan tombol Tidak. | |
| A.5.2 Kembali ke Alur Kejadian Normal nomor 2. | |



Gambar 3.21 Diagram Aktivitas Menghapus *Tutorial*

Tabel 3.20 Spesifikasi Kasus Penggunaan Menambah Kategori *Tutorial*

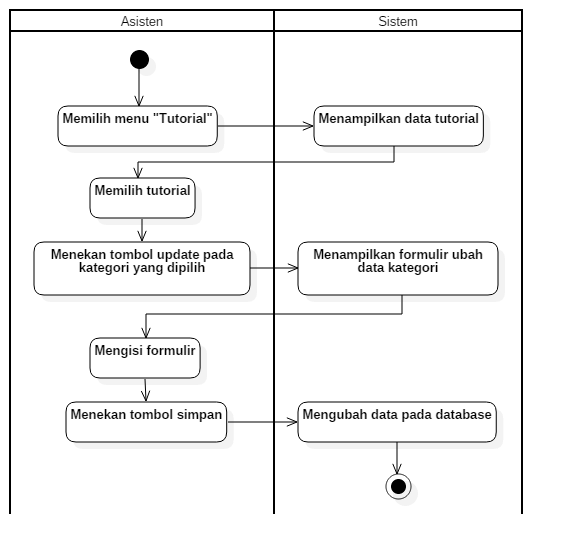
|  |  |
| --- | --- |
| Kode | UC-005.7 |
| Nama | **Menambah Kategori *Tutorial*** |
| Aktor | Asisten |
| Deskripsi | Asisten menambah data kategori untuk *tutorial* kelasnya |
| Tipe | Fungsional |
| Kondisi Awal | Asisten telah *login* ke dalam aplikasi. |
| Kondisi Akhir | Data kategori telah ditambahkan pada *database* |
| Alur Kejadian Normal | |
| 1. Asisten memilih menu “*Tutorial*” | |
| 1. Sistem menampilkan data *tutorial* 2. Asisten memilih salah satu *tutorial* | |
| 1. Asisten menekan tombol “Tambah Kategori” | |
| 1. Sistem menampilkan formulir tambah data kategori *tutorial* 2. Asisten mengisi formulir tambah data kategori *tutorial* 3. Asisten menekan tombol simpan 4. Sistem memasukkan data kategori *tutorial* ke dalam *database* | |
| Alur Kejadian Alternatif | |
| Tidak ada. | |



Gambar 3.22 Diagram Aktivitas Menambah Kategori *Tutorial*

Tabel 3.21 Spesifikasi Kasus Penggunaan Mengubah Kategori *Tutorial*

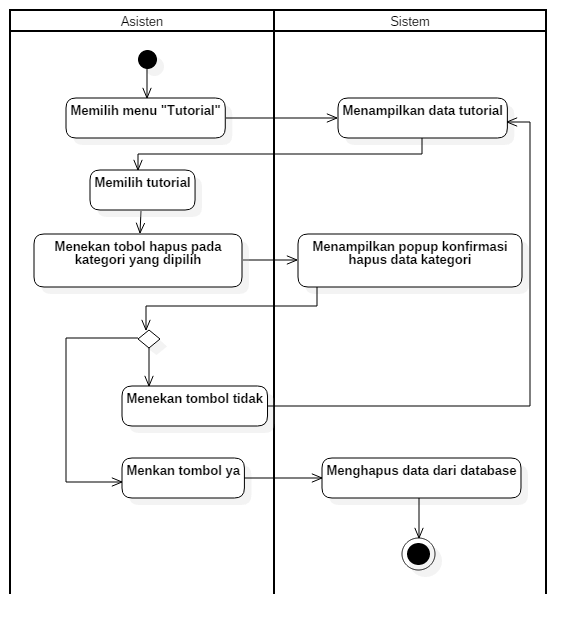
|  |  |
| --- | --- |
| Kode | UC-005.8 |
| Nama | **Mengubah Kategori *Tutorial*** |
| Aktor | Asisten |
| Deskripsi | Asisten mengubah data kategori *tutorial* |
| Tipe | Fungsional |
| Kondisi Awal | Asisten telah *login* ke dalam aplikasi. |
| Kondisi Akhir | Data kategori *tutorial* telah diubah pada *database* |
| Alur Kejadian Normal | |
| 1. Asisten memilih menu “*Tutorial*” | |
| 1. Sistem menampilkan data *tutorial* 2. Asisten memilih salah satu *tutorial* | |
| 1. Asisten menekan tombol “Update” pada data kategori *tutorial* yang ingin diubah | |
| 1. Sistem menampilkan formulir ubah data kategori *tutorial* 2. Asisten mengisi formulir ubah data kategori *tutorial* 3. Asisten menekan tombol simpan 4. Sistem mengubah data kategori *tutorial* pada *database* | |
| Alur Kejadian Alternatif | |
| Tidak ada. | |



Gambar 3.23 Diagram Aktivitas Mengubah Kategori *Tutorial*

Tabel 3.22 Spesifikasi Kasus Penggunaan Menghapus Kategori *Tutorial*

|  |  |
| --- | --- |
| Kode | UC-005.9 |
| Nama | **Menghapus Kategori *Tutorial*** |
| Aktor | Asisten |
| Deskripsi | Asisten menghapus data kategori *tutorial* |
| Tipe | Fungsional |
| Kondisi Awal | Asisten telah *login* ke dalam aplikasi. |
| Kondisi Akhir | Data kategori *tutorial* telah dihapus pada *database* |
| Alur Kejadian Normal | |
| 1. Asisten memilih menu “*Tutorial*”. | |
| 1. Sistem menampilkan data *tutorial*. 2. Asisten memilih salah satu *tutorial* | |
| 1. Asisten menekan tombol hapus pada kategori *tutorial* yang ingin dihapus. | |
| 1. Sistem menampilkan *pop-up* menghapus data kategori *tutorial*. 2. Asisten menekan tombol ya.   A.6 Asisten menekan tombol tidak.   1. Sistem menghapus data kategori *tutorial*. | |
| Alur Kejadian Alternatif | |
| A.6. Asisten menekan tombol Tidak. | |
| A.6.2 Kembali ke Alur Kejadian Normal nomor 2. | |



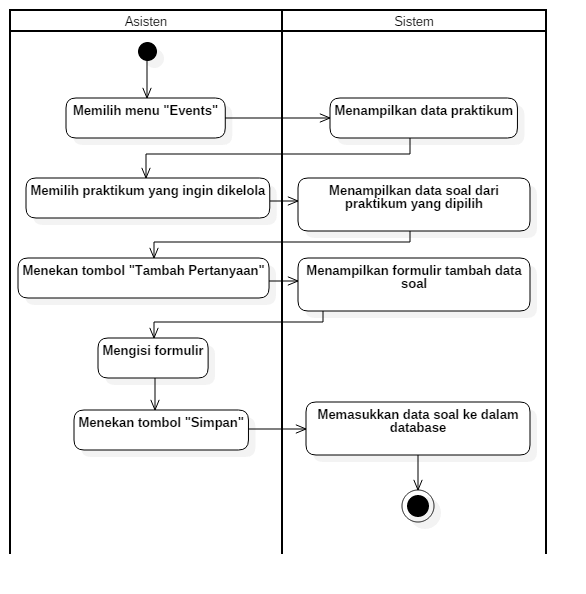
Gambar 3.24 Diagram Aktivitas Menghapus Kategori *Tutorial*

#### Kasus Penggunaan Mengelola Soal

Pada kasus penggunaan ini, asisten dapat mengelola soal untuk praktikum dan *tutorial*. Soal yang dibuat harus memiliki query jawaban yang benar untuk kunci jawaban dari masing-masing soal. Pengelolaan soal terdiri dari menambah soal praktikum, mengubah soal praktikum, menghapus soal praktikum, menambah soal *tutorial*, mengubah soal *tutorial*, dan menghapus soal *tutorial*.

Tabel 3.23 Spesifikasi Kasus Penggunaan Menambah Soal Praktikum

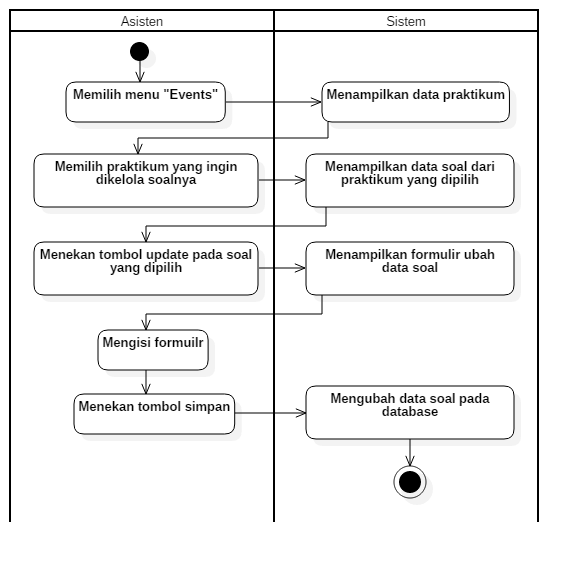
|  |  |
| --- | --- |
| Kode | UC-006.1 |
| Nama | **Menambah Soal Praktikum** |
| Aktor | Asisten |
| Deskripsi | Asisten menambah data soal untuk praktikum kelasnya |
| Tipe | Fungsional |
| Kondisi Awal | Asisten telah *login* ke dalam aplikasi. |
| Kondisi Akhir | Data soal praktikum telah ditambahkan pada *database* |
| Alur Kejadian Normal | |
| 1. Asisten memilih menu “*Events*” | |
| 1. Sistem menampilkan data praktikum | |
| 1. Asisten memilih praktikum yang ingin dikelola soalnya 2. Sistem menampilkan data soal dari praktikum yang dipilih 3. Asisten menekan tombol “Tambah Pertanyaan” | |
| 1. Sistem menampilkan formulir tambah data soal praktikum 2. Asisten mengisi formulir tambah data soal praktikum 3. Asisten menekan tombol simpan 4. Sistem memasukkan data soal praktikumke dalam *database* | |
| Alur Kejadian Alternatif | |
| Tidak ada. | |



Gambar 3.25 Diagram Aktivitas Menambah Soal Praktikum

Tabel 3.24 Spesifikasi Kasus Penggunaan Mengubah Soal Praktikum

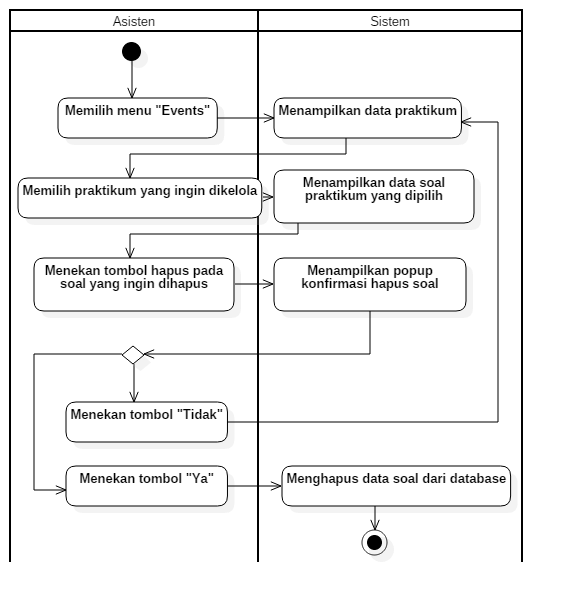
|  |  |
| --- | --- |
| Kode | UC-006.2 |
| Nama | **Mengubah Soal Praktikum** |
| Aktor | Asisten |
| Deskripsi | Asisten mengubah data soal praktikum |
| Tipe | Fungsional |
| Kondisi Awal | Asisten telah *login* ke dalam aplikasi. |
| Kondisi Akhir | Data soal praktikum telah diubah pada *database* |
| Alur Kejadian Normal | |
| 1. Asisten memilih menu “*Events*” | |
| 1. Sistem menampilkan data praktikum | |
| 1. Asisten memilih praktikum yang ingin dikelola soalnya 2. Sistem menampilkan data soal dari praktikum yang dipilih 3. Asisten menekan tombol “*Update*” pada soal yang dipilih | |
| 1. Sistem menampilkan formulir ubah data soal praktikum 2. Asisten mengisi formulir ubah data soal praktikum 3. Asisten menekan tombol simpan 4. Sistem mengubah data soal praktikumpada *database* | |
| Alur Kejadian Alternatif | |
| Tidak ada. | |



Gambar 3.26 Diagram Aktivitas Mengubah Soal Praktikum

Tabel 3.25 Spesifikasi Kasus Penggunaan Menghapus Soal Praktikum

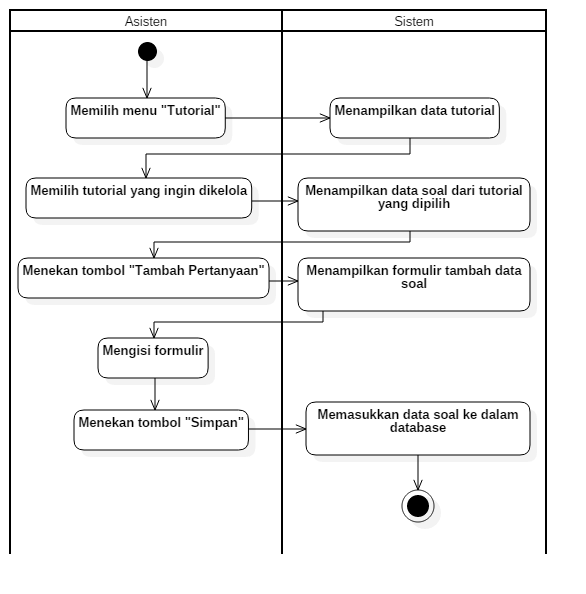
|  |  |
| --- | --- |
| Kode | UC-006.3 |
| Nama | **Menghapus Soal Praktikum** |
| Aktor | Asisten |
| Deskripsi | Asisten menghapus data soal praktikum |
| Tipe | Fungsional |
| Kondisi Awal | Asisten telah *login* ke dalam aplikasi. |
| Kondisi Akhir | Data soal praktikumtelah dihapus pada *database* |
| Alur Kejadian Normal | |
| 1. Asisten memilih menu “*Events*”. | |
| 1. Sistem menampilkan data praktikum. 2. Asisten memilih praktikum yang ingin dikelola soalnya 3. Sistem menampilkan data soal dari praktikum yang dipilih | |
| 1. Asisten menekan tombol hapus pada soal yang ingin dihapus. | |
| 1. Sistem menampilkan *pop-up* menghapus data soal praktikum. 2. Asisten menekan tombol ya.   A.7 Asisten menekan tombol tidak.   1. Sistem menghapus data soal praktikum. | |
| Alur Kejadian Alternatif | |
| A.7. Asisten menekan tombol Tidak. | |
| A.7.2 Kembali ke Alur Kejadian Normal nomor 2. | |



Gambar 3.27 Diagram Aktivitas Menghapus Soal Praktikum

Tabel 3.26 Spesifikasi Kasus Penggunaan Menambah Soal *Tutorial*

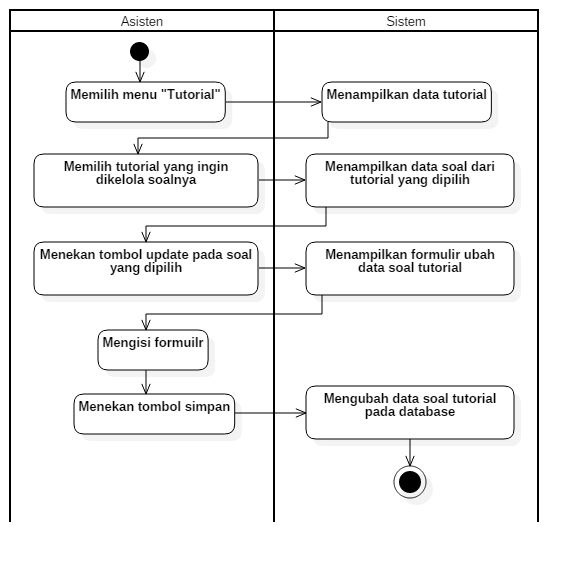
|  |  |
| --- | --- |
| Kode | UC-006.4 |
| Nama | **Menambah Soal *Tutorial*** |
| Aktor | Asisten |
| Deskripsi | Asisten menambah data soal *tutorial* |
| Tipe | Fungsional |
| Kondisi Awal | Asisten telah *login* ke dalam aplikasi. |
| Kondisi Akhir | Data *tutorial* telah ditambahkan pada *database* |
| Alur Kejadian Normal | |
| 1. Asisten memilih menu “*Tutorial*” | |
| 1. Sistem menampilkan data *tutorial* | |
| 1. Asisten memilih *tutorial* yang ingin dikelola soalnya 2. Sistem menampilkan data soal dari *tutorial* yang dipilih 3. Asisten menekan tombol “Tambah Pertanyaan” | |
| 1. Sistem menampilkan formulir tambah data soal *tutorial* 2. Asisten mengisi formulir tambah data soal *tutorial* 3. Asisten menekan tombol simpan 4. Sistem memasukkan data soal *tutorial* ke dalam *database* | |
| Alur Kejadian Alternatif | |
| Tidak ada. | |



Gambar 3.28 Diagram Aktivitas Menambah Soal *Tutorial*

Tabel 3.27 Spesifikasi Kasus Penggunaan Mengubah Soal *Tutorial*

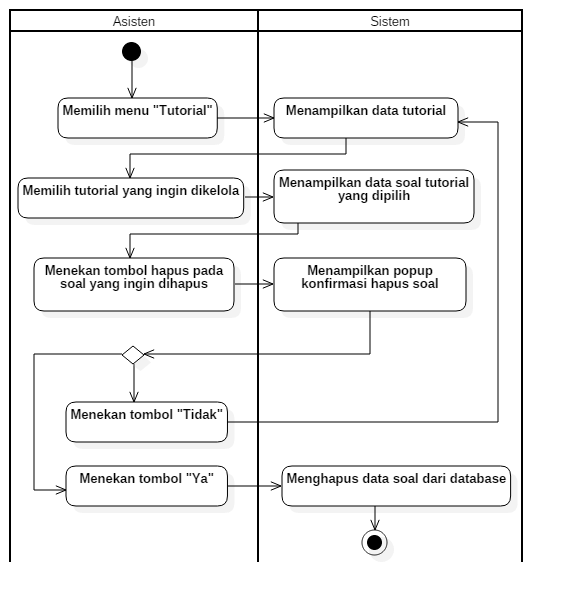
|  |  |
| --- | --- |
| Kode | UC-006.5 |
| Nama | **Mengubah Soal *Tutorial*** |
| Aktor | Asisten |
| Deskripsi | Asisten mengubah data soal *tutorial* |
| Tipe | Fungsional |
| Kondisi Awal | Asisten telah *login* ke dalam aplikasi. |
| Kondisi Akhir | Data soal *tutorial* telah diubah pada *database* |
| Alur Kejadian Normal | |
| 1. Asisten memilih menu “*Tutorial*” | |
| 1. Sistem menampilkan data *tutorial* | |
| 1. Asisten memilih *tutorial* yang ingin dikelola soalnya 2. Sistem menampilkan data soal dari *tutorial* yang dipilih 3. Asisten menekan tombol “*Update*” pada soal yang dipilih | |
| 1. Sistem menampilkan formulir ubah data soal *tutorial* 2. Asisten mengisi formulir ubah data soal *tutorial* 3. Asisten menekan tombol simpan 4. Sistem mengubah data soal *tutorial* pada *database* | |
| Alur Kejadian Alternatif | |
| Tidak ada. | |



Gambar 3.29 Diagram Aktivitas Mengubah Soal *Tutorial*

Tabel 3.28 Spesifikasi Kasus Penggunaan Menghapus Soal *Tutorial*

|  |  |
| --- | --- |
| Kode | UC-006.6 |
| Nama | **Menghapus Soal *Tutorial*** |
| Aktor | Asisten |
| Deskripsi | Asisten menghapus data soal *tutorial* |
| Tipe | Fungsional |
| Kondisi Awal | Asisten telah *login* ke dalam aplikasi. |
| Kondisi Akhir | Data soal *tutorial* telah dihapus pada *database* |
| Alur Kejadian Normal | |
| 1. Asisten memilih menu “*Tutorial*”. | |
| 1. Sistem menampilkan data *tutorial*. 2. Asisten memilih *tutorial* yang ingin dikelola soalnya 3. Sistem menampilkan data soal dari *tutorial* yang dipilih | |
| 1. Asisten menekan tombol hapus pada soal *tutorial* yang ingin dihapus. | |
| 1. Sistem menampilkan *pop-up* konfirmasi menghapus data soal *tutorial*. 2. Asisten menekan tombol ya.   A.7 Asisten menekan tombol tidak.   1. Sistem menghapus data soal *tutorial*. | |
| Alur Kejadian Alternatif | |
| A.7. Asisten menekan tombol Tidak. | |
| A.7.2 Kembali ke Alur Kejadian Normal nomor 2. | |



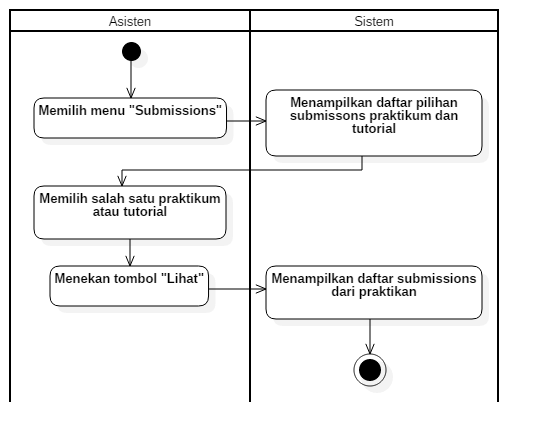
Gambar 3.30 Diagram Aktivitas Menghapus Soal Tutorial

#### Kasus Penggunaan Melihat *Submissions*

Pada kasus penggunaan ini, asisten dan praktikan dapat melihat jawaban-jawaban yang telah di-*submit* pada praktikum atau *tutorial* tertentu.

Tabel 3.29 Spesifikasi Kasus Penggunaan Melihat *Submissions* untuk Asisten

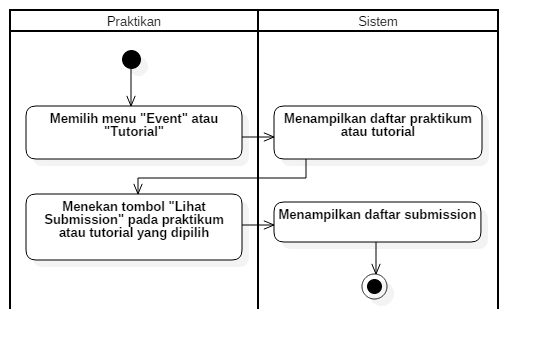
|  |  |
| --- | --- |
| Kode | UC-007.1 |
| Nama | **Melihat *Submissions* untuk Asisten** |
| Aktor | Asisten |
| Deskripsi | Asisten melihat jawaban yang telah di-*submit* oleh praktikan. |
| Tipe | Fungsional |
| Kondisi Awal | Asisten atau praktikan telah *login*. |
| Kondisi Akhir | Daftar *submissions* ditampilkan. |
| Alur Kejadian Normal | |
| Asisten memilih menu “*Submissions*”. | |
| Sistem menampilkan daftarpilihan *submissions* praktikum dan *tutorial*. | |
| Asisten memilih praktikum atau *tutorial*. | |
| Asisten menekan tombol “Lihat”.  Sistem menampilkan daftar *submissions* dari praktikum atau *tutorial* yang dipilih. | |
| Alur Kejadian Alternatif | |
| Tidak ada | |



Gambar 3.31 Diagram Aktivitas Melihat *Submissions* untuk Asisten

Tabel 3.30 Spesifikasi Kasus Penggunaan Melihat *Submissions* untuk Praktikan

|  |  |
| --- | --- |
| Kode | UC-007.2 |
| Nama | **Melihat *Submissions* untuk Praktikan** |
| Aktor | Praktikan |
| Deskripsi | Praktikan melihat jawaban yang telah di-*submit*  olehnya. |
| Tipe | Fungsional |
| Kondisi Awal | Praktikan telah *login*. |
| Kondisi Akhir | Daftar *submissions* ditampilkan. |
| Alur Kejadian Normal | |
| Praktikan memilih menu “*Events*” atau “*Tutorial*”. | |
| Sistem menampilkan daftarpraktikum atau *tutorial* | |
| Praktikan menekan tombol “Lihat *submission*” pada praktikum atau *tutorial* yang dipilih | |
| Sistem menampilkan daftar *submissions* dari praktikum atau *tutorial* yang dipilih. | |
| Alur Kejadian Alternatif | |
| Tidak Ada | |



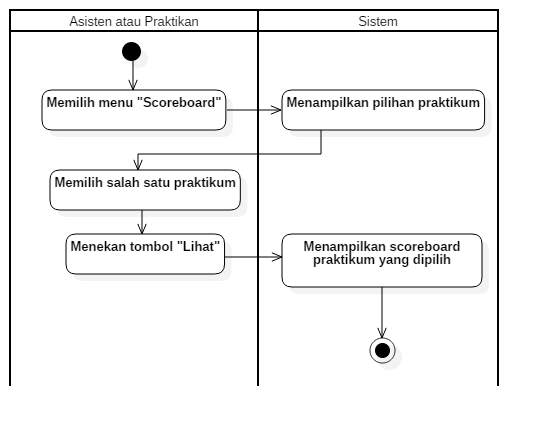
Gambar 3.32 Diagram Aktivitas Melihat *Submissions* untuk Praktikan

#### Kasus Penggunaan Melihat *Scoreboard*

Pada kasus penggunaan ini, asisten dan praktikan bisa melihat *scoreboard* dari suatu praktikum. Scoreboard tidak tersedia untuk *tutorial*.

Tabel 3.31 Spesifikasi Kasus Penggunaan Melihat *Scoreboard*

|  |  |
| --- | --- |
| Kode | UC-008 |
| Nama | **Melihat Scoreboard** |
| Aktor | Asisten dan Praktikan |
| Deskripsi | Asisten atau praktikan bisa melihat *scoreboard* dari suatu praktikum. |
| Tipe | Fungsional |
| Kondisi Awal | Asisten atau praktikan telah *login*. |
| Kondisi Akhir | *Scoreboard* suatu praktikum ditampilkan. |
| Alur Kejadian Normal | |
| Asisten atau praktikan memilih menu “*Scoreboard*”. | |
| Sistem menampilkan pilihan praktikum | |
| Asisten atau praktikan memilih praktikum yang ingin dilihat *scoreboard*-nya.  Asisten atau praktikan menekan tombol “Lihat”.  Sistem menampilkan *scoreboard* dari praktikum yang dipilih. | |
|  | |
| Alur Kejadian Alternatif | |
| Tidak ada. | |



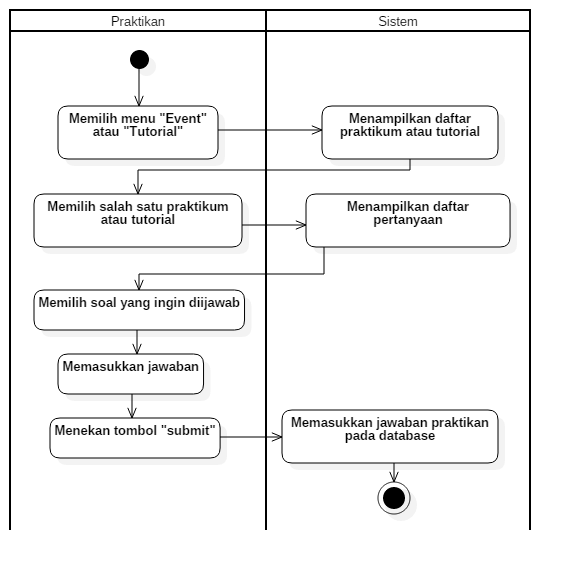
Gambar 3.33 Diagram Aktivitas Melihat Scoreboard

#### Kasus Penggunaan Menjawab Soal *Event*

Pada kasus penggunaan ini, praktikan dapat menjawab soal dari praktikum dan *tutorial* kelasnya masing-masing. Praktikan hanya bisa menjawab soal selama *event* tersebut masih aktif.

Tabel 3.32 Spesifikasi Kasus Penggunaan Menjawab Soal *Event*

|  |  |
| --- | --- |
| Kode | UC-009 |
| Nama | **Menjawab Soal *Event*** |
| Aktor | Praktikan |
| Deskripsi | Praktikan dapat menjawab soal-soal pada praktikum atau *tutorial*. |
| Tipe | Fungsional |
| Kondisi Awal | Praktikan telah *login*. |
| Kondisi Akhir | Jawaban dari praktikan telah masuk pada database. |
| Alur Kejadian Normal | |
| 1. Praktikan memilih menu “*Events*” atau “*Tutorial*”. | |
| 1. Sistem menampilkan daftar praktikum atau *tutorial*. | |
| 1. Praktikan memilih praktikum atau *tutorial*. | |
| 1. Sistem menampilkan daftar pertanyaan dari praktikum atatu *tutorial*  yang dipilih. 2. Praktikan memilih soal yang ingin dijawab. 3. Praktikan memasukkan jawaban. 4. Praktikan menekan tombol “*Submit*”. 5. Sistem memasukkan jawaban pada *database*. | |
| Alur Kejadian Alternatif | |
| Tidak ada. | |



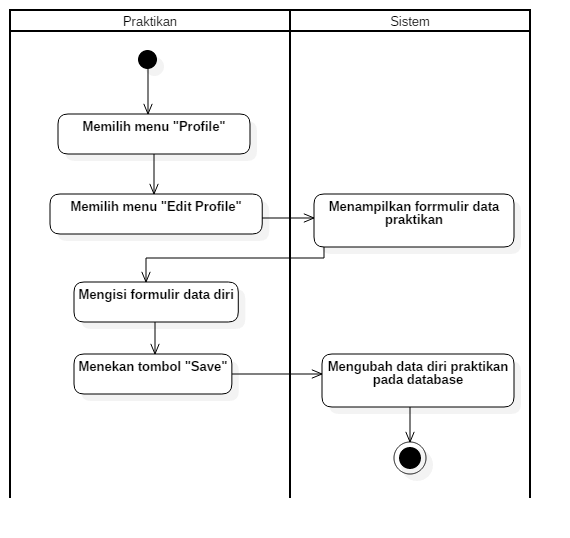
Gambar 3.34 Diagram Aktivitas Menjawab Soal Event

#### Kasus Penggunaan Mengelola Akun

Pada kasus penggunaan ini, praktikan dapat mengelola akunnya masing-masing. Pengelolaan akun terdiri dari mengubah data diri praktikan dan mengganti password dari akunnya.

Tabel 3.33 Spesifikasi Kasus Penggunaan Mengubah Data Diri Praktikan

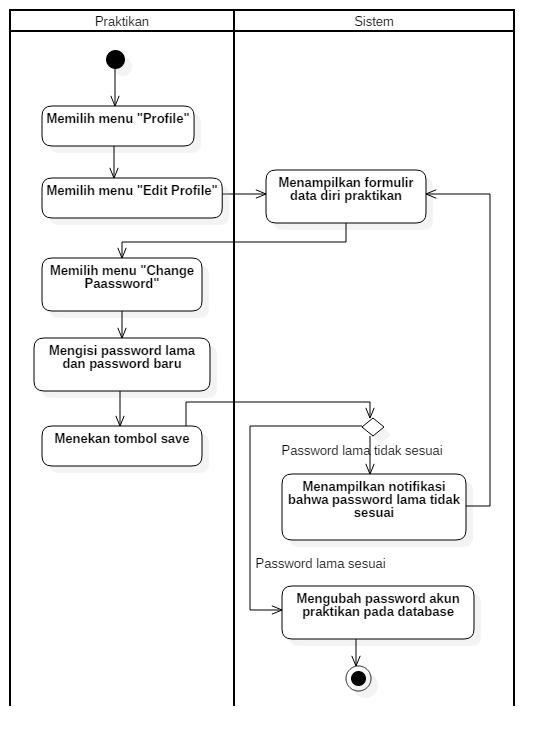
|  |  |
| --- | --- |
| Kode | UC-010.1 |
| Nama | **Mengubah Data Diri Praktikan** |
| Aktor | Praktikan |
| Deskripsi | Praktikan dapat mengubah data diri dari akunnya. |
| Tipe | Fungsional |
| Kondisi Awal | Praktikan telah *login*. |
| Kondisi Akhir | Data diri praktikan telah diubah pada *database*. |
| Alur Kejadian Normal | |
| 1. Praktikan memilih menu “*Profile*”. | |
| 1. Praktikan memilih menu “*Edit Profile*” | |
| 1. Sistem menampilkan formulir data diri praktikan. | |
| 1. Praktikan mengisi formulir data diri. 2. Praktikan menekan tombol “*Save*”. 3. Sistem mengubah data diri praktikan pada database. | |
| Alur Kejadian Alternatif | |
| Tidak ada | |



Gambar 3.35 Diagram Aktivitas Mengubah Data Diri Praktikan

Tabel 3.34 Spesifikasi Kasus Penggunaan Mengubah *Password*

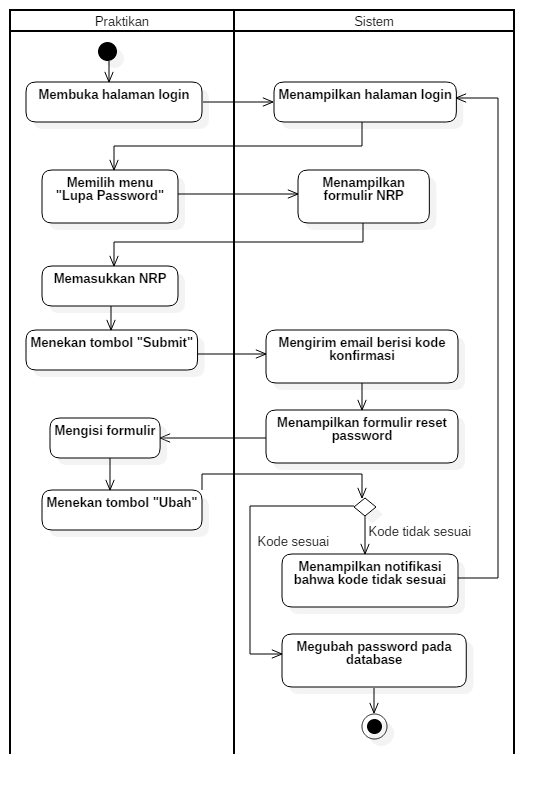
|  |  |
| --- | --- |
| Kode | UC-010.2 |
| Nama | **Mengubah *Password*** |
| Aktor | Praktikan |
| Deskripsi | Praktikan dapat mengubah *password* dari akunnya. |
| Tipe | Fungsional |
| Kondisi Awal | Praktikan telah *login*. |
| Kondisi Akhir | *Password* dari akun praktikan telah diubah pada *database*. |
| Alur Kejadian Normal | |
| 1. Praktikan memilih menu “*Profile*”. | |
| 1. Praktikan memilih menu “*Edit Profile*” | |
| 1. Sistem menampilkan formulir data diri praktikan. 2. Praktikan memilih menu “*Change Password*” | |
| 1. Praktikan memasukkan *password* lama dan *password* baru. 2. Praktikan menekan tombol “*Save*”.   A.6 *Password* lama tidak sesuai   1. Sistem mengubah *password* dari akun praktikan pada *database*. | |
| Alur Kejadian Alternatif | |
| A.6. *Password* lama tidak sesuai. | |
| A.6.2 Sistem menampilkan notifikasi bahwa *password* lama tidak sesuai  A.6.3 Kembali ke Alur Kejadian Normal nomor 3 | |



Gambar 3.36 Diagram Aktivitas Mengubah *Password*

#### Kasus Penggunaan Me-*reset Password*

Pada kasus penggunaan ini, praktikan dapat mengubah *password-*nya apabila praktikan lupa *password* dari akun tersebut. Kode dari *reset* password akan dikirim via *e-mail* dan akan dikirim ke *e-mail* praktikan tersebut.



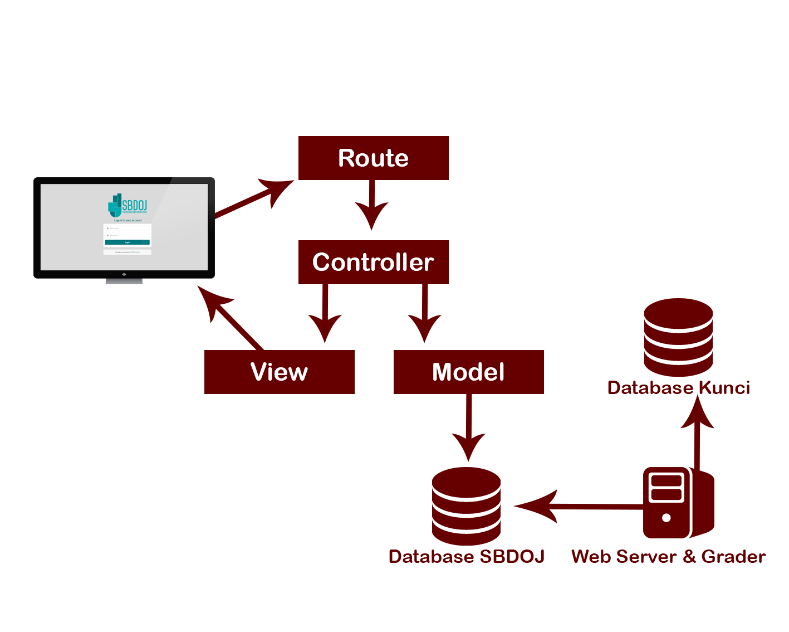
Gambar 3.37 Diagram Aktivitas Me-*reset Password*

Tabel 3.35 Spesifikasi Kasus Me-*reset Password*

|  |  |
| --- | --- |
| Kode | UC-011 |
| Nama | **Me-*reset Password*** |
| Aktor | Praktikan |
| Deskripsi | Praktikan me-*reset* password akunnya*.* |
| Tipe | Fungsional |
| Kondisi Awal | Praktikan lupa *password* dari akunnya. |
| Kondisi Akhir | *Password* dari akun praktikan telah diubah pada *database*. |
| Alur Kejadian Normal | |
| 1. Praktikan membuka halaman *login* dari sistem. 2. Sistem menampilkan halaman *login*. | |
| 1. Praktikan memilih menu “Lupa *password*”. | |
| 1. Sistem menampilkan formulir untuk praktikan memasukkan NRP. 2. Praktikan memasukkan NRP. | |
| 1. Praktikan menekan tombol “*Submit*”. 2. Sistem mengirim *e-mail* yang berisi kode untuk konfirmasi *reset password*. 3. Sistem menampilkan formulir *reset password*. 4. Praktikan mengisi formulir beserta kode yang telah dikirim. 5. Praktikan menekan tombol “Ubah”   A.10. Kode *reset password* tidak sesuai.   1. Sistem mengubah *password* dari akun praktikan pada *database.* | |
| Alur Kejadian Alternatif | |
| A.10. Kode *reset password* tidak sesuai. | |
| A.10.1 Sistem menampilkan notifikasi bahwa kode yang dimasukkan tidak sesuai.  A.10.2 Kembali ke Alur Kejadian Normal nomor 2. | |

### Perancangan Arsitektur Aplikasi

Aplikasi yang dibangun adalah aplikasi berbasis *web*. Oleh karena itu dibutuhkan *web server*, *grader* dan *database* kunci untuk memenuhi kebutuhan sistem.



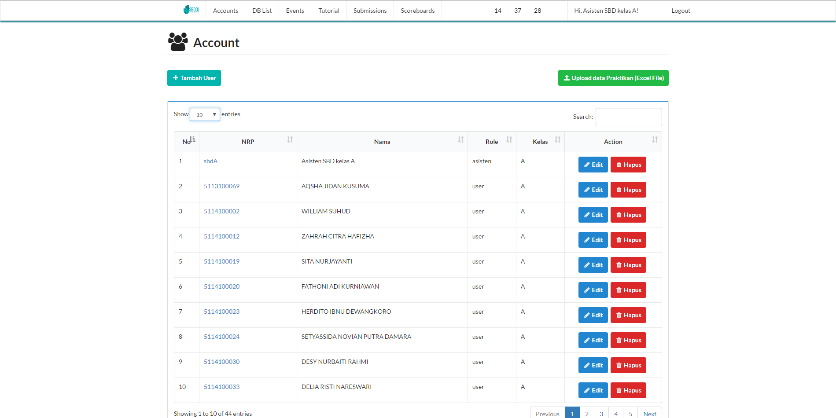
Gambar 3.38 Arsitektur Aplikasi

### Perancangan Antarmuka

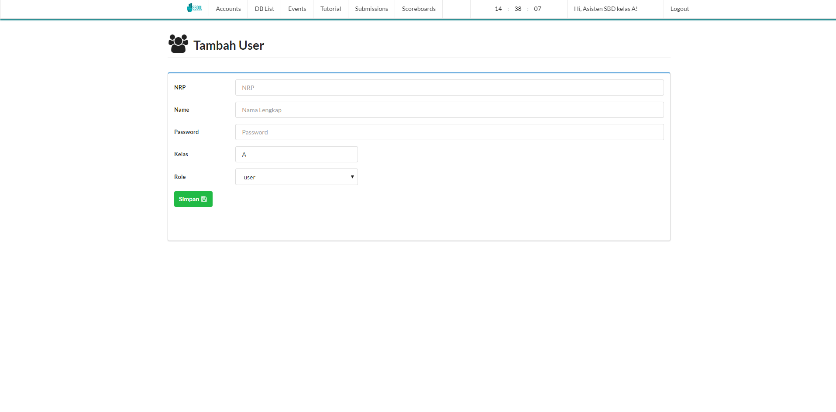
Perancangan antarmuka merupakan hal yang penting dalam melakukan perancangan aplikasi. Antarmuka pengguna yang berhubungan langsung dengan aktor harus memiliki kemudahan-kemudahan dan tampilan yang rapi dan menarik bagi penggunanya.

#### Rancangan Antarmuka Mengelola Data Praktikan

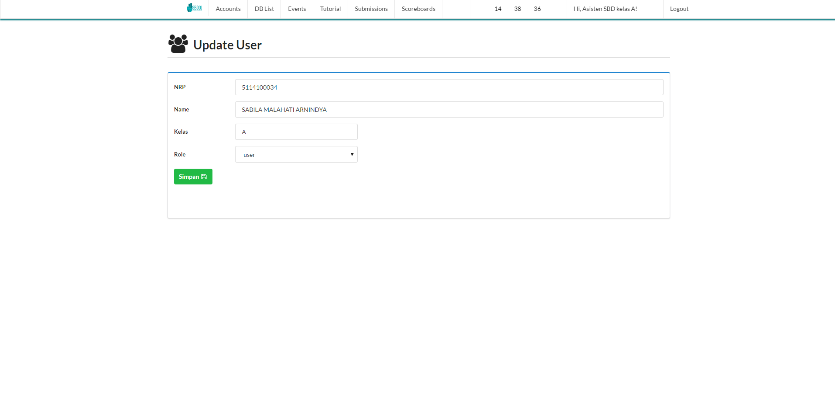
Rancangan antarmuka ini digunakan untuk kasus penggunaan mengelola data praktikan. Pada halama mengelola data praktikan terdapat beberapa bagian, yaitu tabel untuk menampilkan data praktikan, tombol tambah data, tombol upload data menggunakan excel, tombol ubah dan hapus data. Rancangan antarmuka untuk mengelola data praktikan dapat dilihat pada **Gambar 3.39**. Sedangkan untuk menambah data praktikan, akan ditampilkan formulir untuk mengisi data praktikan dengan isi nama, NRP, *password* dan kelas seperti pada **Gambar 3.40**. Untuk mengubah data praktikan, isian formulir sama seperti saat menambah data praktikan dan ditunjukkan oleh **Gambar 3.41**. Untuk proses menghapus data praktikan, berupa *pop-up* konfirmasi penghapusan saat asisten menekan tombol hapus pada salah satu data praktikan. Proses ini bisa dilihat pada **Gambar 3.42**. Selain bisa menambah data dengan form, asisten juga bisa menambah data praktikan menggunakan file excel. *Template* file excel bisa diunduh pada tombol di rancangan antarmuka “*Upload File Exce*l” yang dapat dilihat pada **Gambar 3.43.**



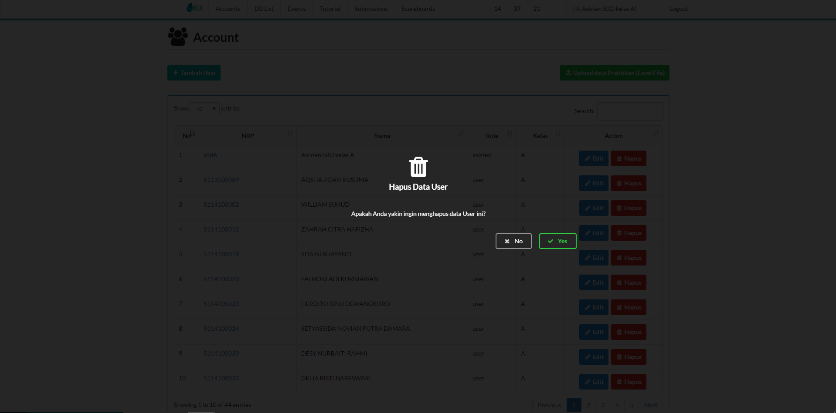
Gambar 3.39 Rancangan Antarmuka Mengelola Data Praktikan



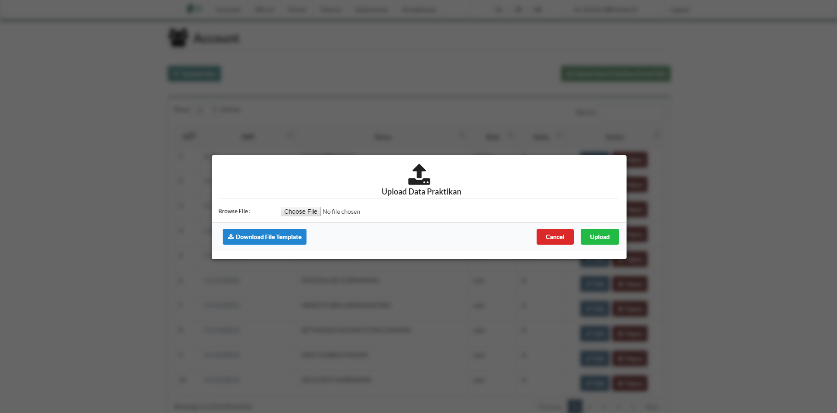
Gambar 3.39 Rancangan Antarmuka Menambah Data Praktikan



Gambar 3.39 Rancangan Antarmuka Mengubah Data Praktikan



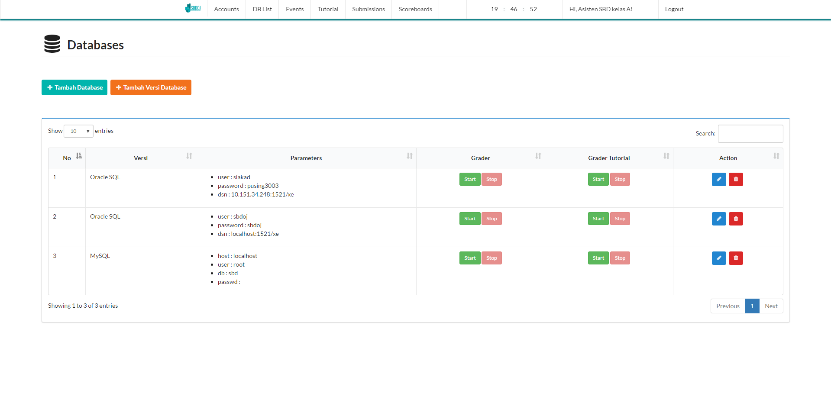
Gambar 3.39 Rancangan Antarmuka Menghapus Data Praktikan



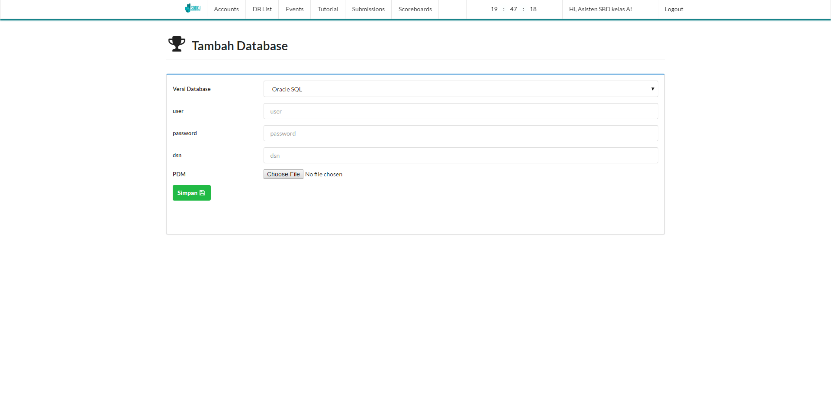
Gambar 3.39 Rancangan Antarmuka *Upload File Execel*

#### Rancangan Antarmuka Mengelola *Database*

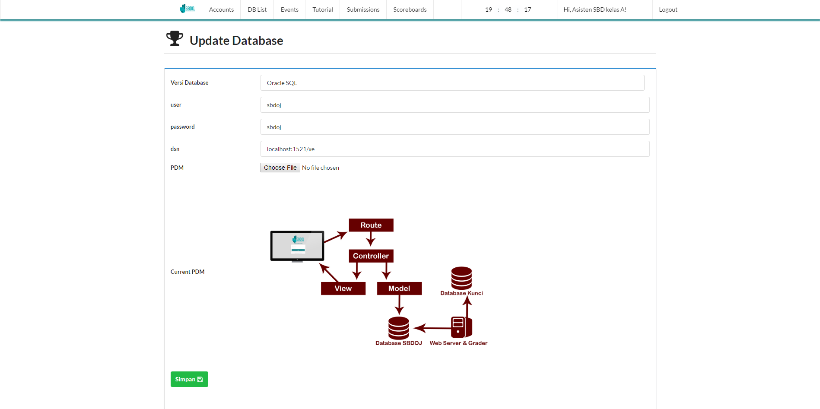
Rancangan antarmuka ini digunakan untuk kasus penggunaan mengelola *database*. Pada antarmuka tersebut terdapat halaman yang menampilkan daftar *database* yang dapat digunakan untuk praktikum maupun *tutorial*. Pada halaman tersebut juga digunakan untuk mengelola *database*. Pengelolaan *database* terdiri dari menambah, mengubah, menghapus database, menambah versi database dan mengelola *grader*. Antarmuka tersebut ditunjukkan oleh **Gambar 3.40**. Halaman untuk menambah *database* berisi formulir yang isinya bergantung dari parameter yang dibutuhkan oleh suatu versi *database*. Setiap *database* diharuskan mempunyai PDM. Halaman ini dapat dilihat pada **Gambar 3.41**. Sedangkan halaman untuk mengubah data *database* juga berupa formulir yang bergantung pada versi *database* yang dapat dilihat pada **Gambar 3.42**. Untuk rancangan antarmuka menghapus data *database* adalah berupa *pop-up* yang digunakan untuk konfirmasi setelah asisten menekan tombol *delete*. Rancangan antarmuka ditunjukkan oleh **Gambar 3.43**.



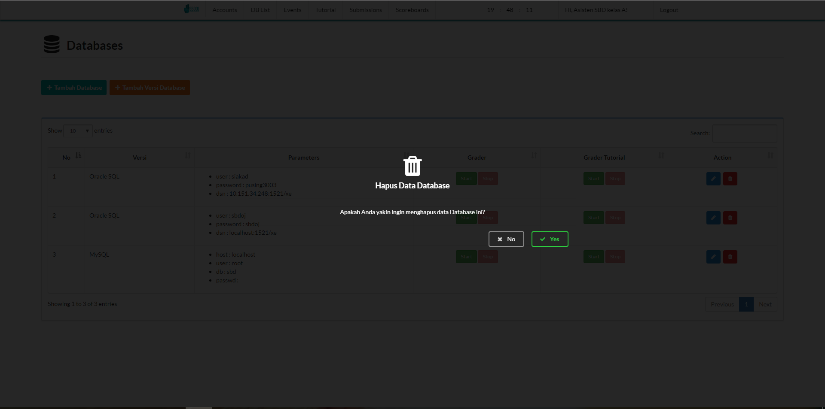
Gambar 3.40 Rancangan Antarmuka Mengelola *Database*



Gambar 3.41 Rancangan Antarmuka Menambah *Database*



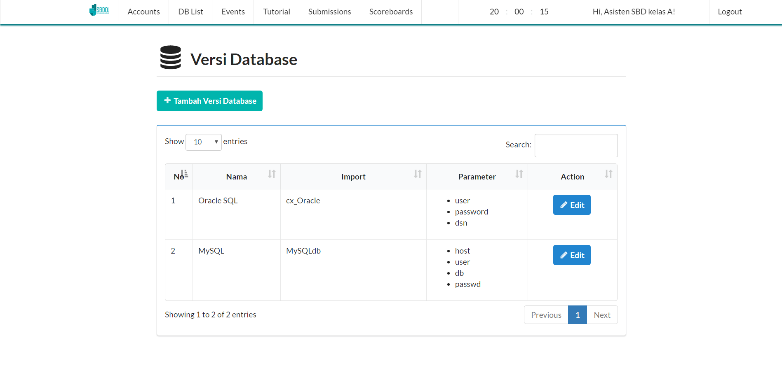
Gambar 3.42 Rancangan Antarmuka Mengubah *Database*



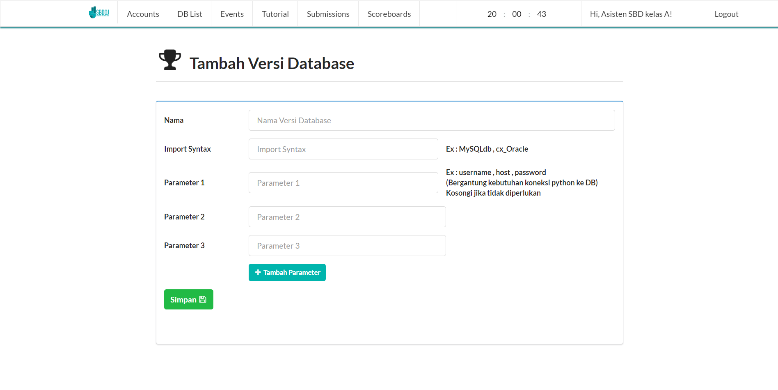
Gambar 3.43 Rancangan Antarmuka Menghapus *Database*

#### Rancangan Antarmuka Mengelola Jenis *Database*

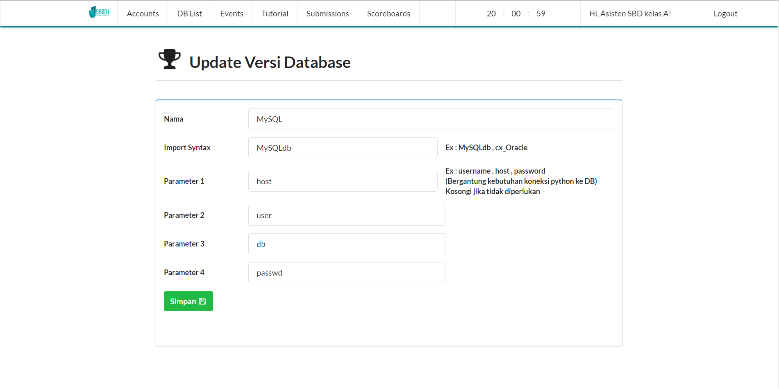
Rancangan antarmuka ini digunakan untuk kasus penggunaan mengelola jenis *database*. Pada halaman pengelolaan jenis *database* terdapat beberapa bagian, antara lain tabel untuk menampilkan daftar jenis *database*, tombol tambah dan ubah jenis *database*. Racangan antarmuka tersebut ditunjukkan oleh **Gambar 3.44**. Sedangkan untuk menambah jenis *database*, terdapat form yang dinamis bergantung dari kebutuhan parameter suatu jenis *database* yang ditunjukkan oleh **Gambar 3.45**. Pada saat mengubah jenis *database*, asisten tidak dapat menambah parameter lagi, jadi parameter tetap seperti saat membuat jenis *database* dan dapat dilihat pada **Gambar 3.46**.



Gambar 3.44 Rancangan Antarmuka Mengelola Jenis *Database*



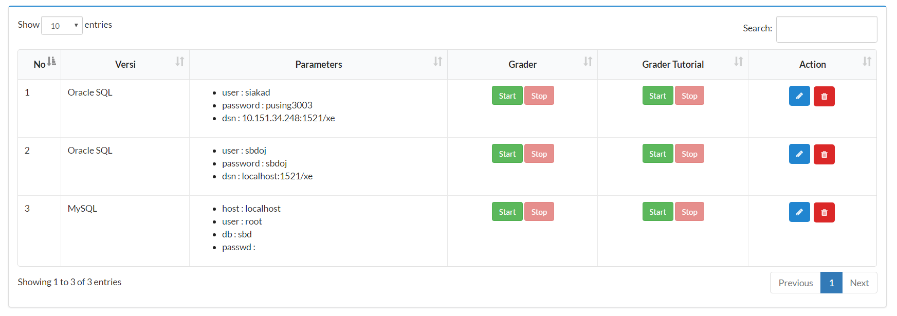
Gambar 3.45 Rancangan Antarmuka Menambah Jenis *Database*



Gambar 3.46 Rancangan Antarmuka Mengubah Jenis *Database*

#### Rancangan Antarmuka Mengelola *Grader*

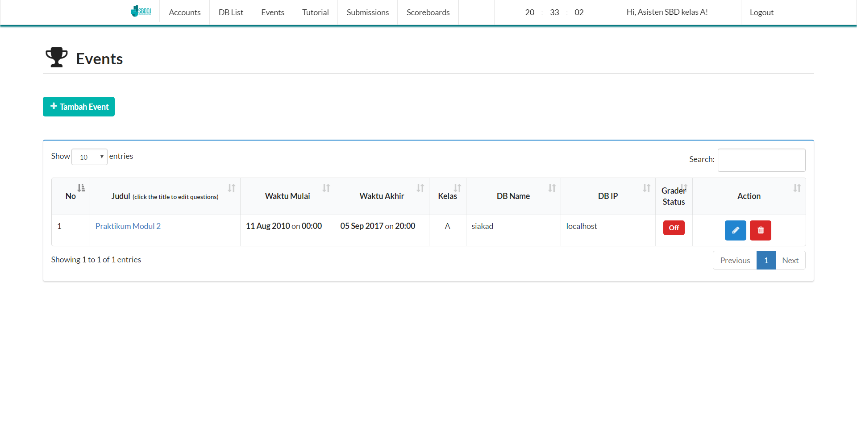
Rancangan antarmuka ini digunakan untuk kasus penggunaan mengelola g*rader*. Pengelolaan *grader* terdiri dari memulai dan mengehentikan *grader*. Setiap *database* memiliki 1 *grader* untuk praktikum dan 1 *grader* untuk *tutorial*. Rancangan antarmuka pengelolaan *grader* terdiri dari 2 tombol yaitu tombol *Start* dan *Stop* yang terletak di sebelah masing-masing *database*. Halaman tersebut dapat dilihat pada **Gambar 3.47**.



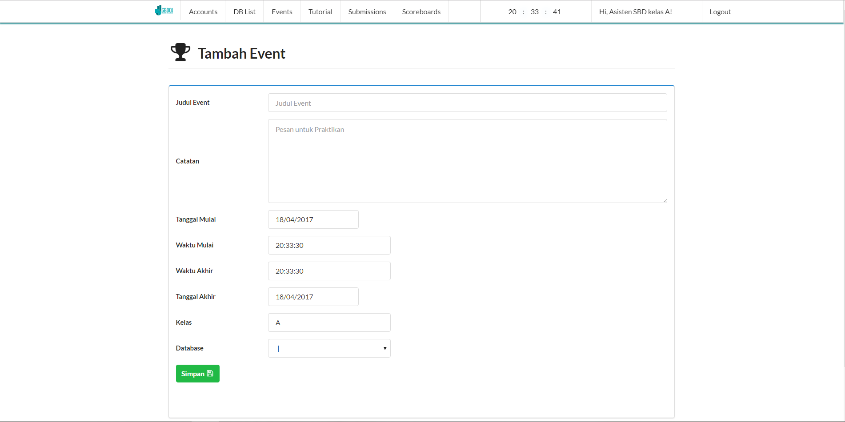
Gambar 3.47 Rancangan Antarmuka Mengelola *Grader*

#### Rancangan Antarmuka Mengelola *Event*

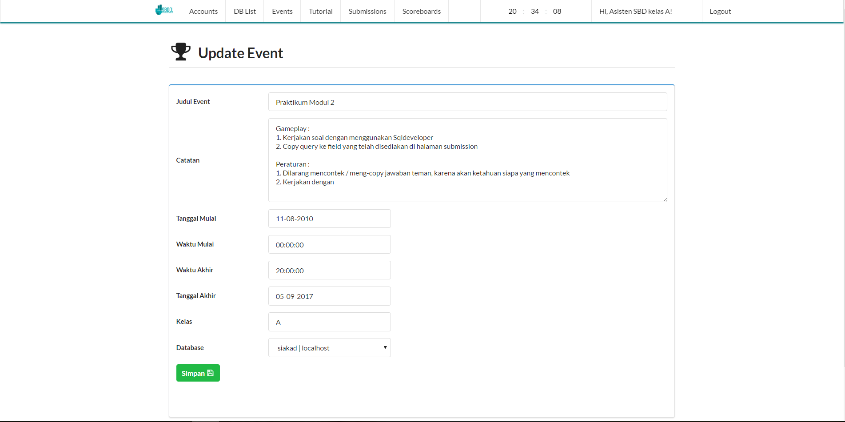
Rancangan antarmuka ini digunakan untuk kasus penggunaan mengelola *event*. Rancangan antarmuka pengelolaan *event* terdiri dari tabel daftar praktikum atau *tutorial* yang berisi judul, waktu mulai, waktu akhir, kelas, nama database yang dipakai,status *grader* serta tombol untuk mengubah dan menghapus *event*. Selain itu juga terdapat tombol untuk menambah *event*. Rancangan antarmuka tersebut dapat dilihat pada **Gambar 3.48**. Sedangkan halaman untuk menambah *event*  terdiri dari sebuah formulir yang berisi judul, catatan / keterangan, tanggal mulai, waktu mulai, waku akhir, tanggal akhir, kelas dan *database* yang digunakan. Halaman ini ditunjukkan oleh **Gambar 3.49**. Sedangkan untuk mengubah data *event*, terdapat halaman yang berisi formulir yang sama dengan formulir saat membuat *event* dan dapat dilihat pada **Gambar 3.50**. Untuk menghapus data *event*, digunakan *pop-up* untuk konfirmasi penghapusan seperti pada **Gambar 3.51**. Untuk halaman pengelolaan kategori *tutorial*, dapat dilihat pada **Gambar 3.52**. Pengelolaan kategori berupa tabel yang berisi nama kategori dan tombol ubah serta hapus data kategori. Untuk menambah kategori, digunkna formulir yang berisi nama kategori dan penjelasan dari kategori tersebut yang ditunjukkan oleh **Gambar 3.53**.



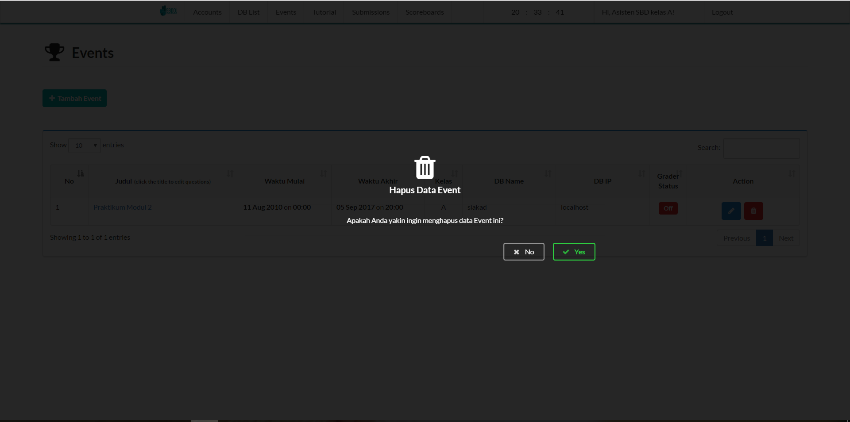
Gambar 3.48 Rancangan Antarmuka Mengelola *Event*



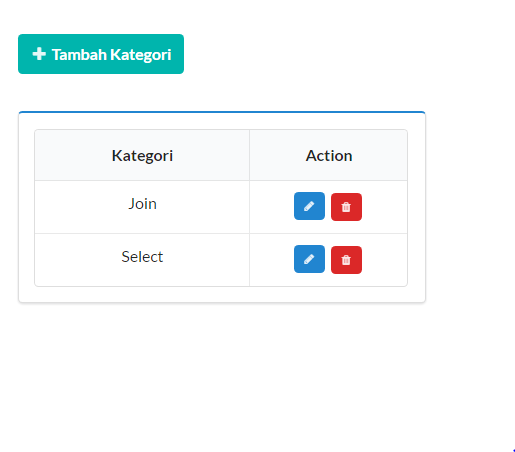
Gambar 3.49 Rancangan Antarmuka Menambah *Event*



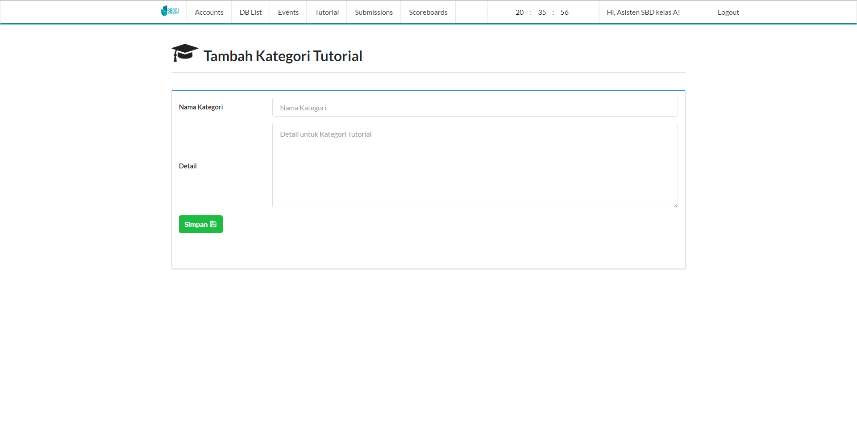
Gambar 3.50 Rancangan Antarmuka Mengubah *Event*



Gambar 3.51 Rancangan Antarmuka Menghapus *Event*



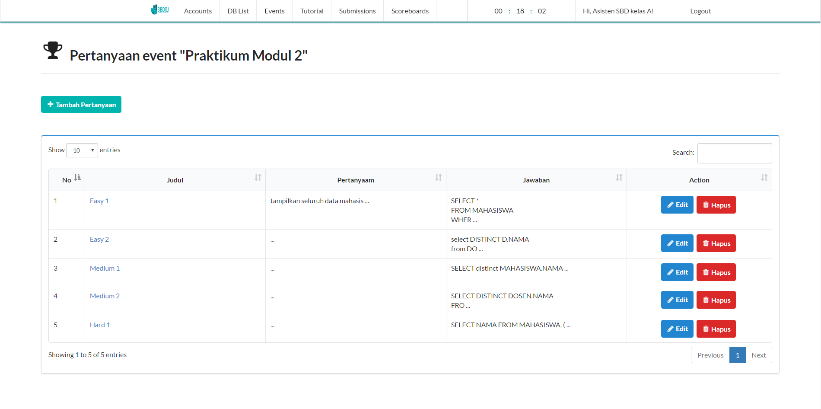
Gambar 3.52 Rancangan Antarmuka Mengelola Kategori *Tutorial*



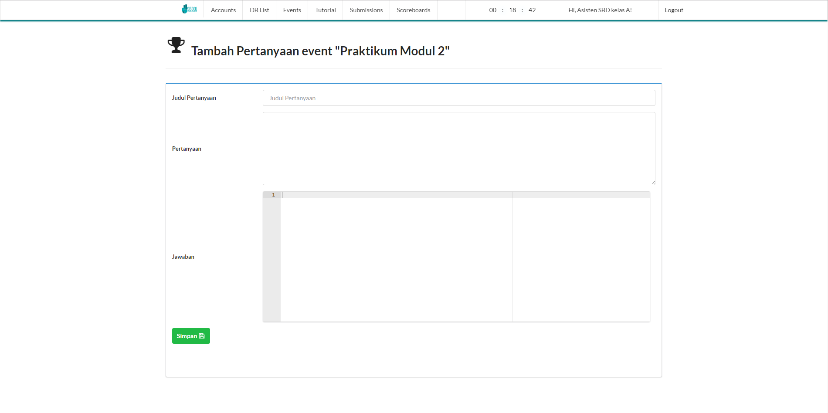
Gambar 3.48 Rancangan Antarmuka Menambah Kategori *Tutorial*

#### Rancangan Antarmuka Mengelola Soal

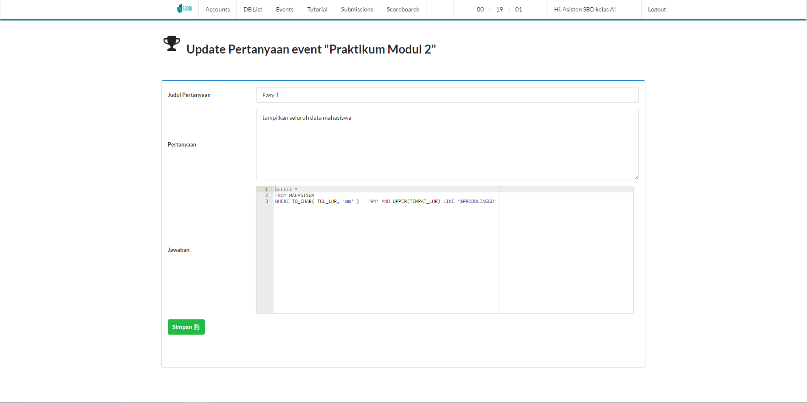
Rancangan antarmuka ini digunakan untuk kasus penggunaan mengelola soal praktikum atau *tutorial*. Pada antarmuka tersebut terdapat beberapa halaman. Halaman mengelola soal terdapat beberapa bagian, yaitu tombol untuk menambah soal, daftar soal yang ditampilkan dalam bentuk tabel, dan tombol untuk menghapus dan mengubah data soal. Rancangan antarmuka untuk mengelola soal dapat dilihat pada Gambar ….. . Halaman tambah data soal menampilkan formulir yang berisi judul soal, pertanyaan, dan jawaban dari soal yang dapat dilihat pada Gambar …. . Untuk halaman mengubah data soal, menampilkan formulir untuk mengubah data soal dengan isi sama seperti formulir tambah soal dan dapat dilihat pada Gambar …. . Sedangkan antarmuka untuk menghapus data soal berupa *pop-up* seperti pada Gambar …. .



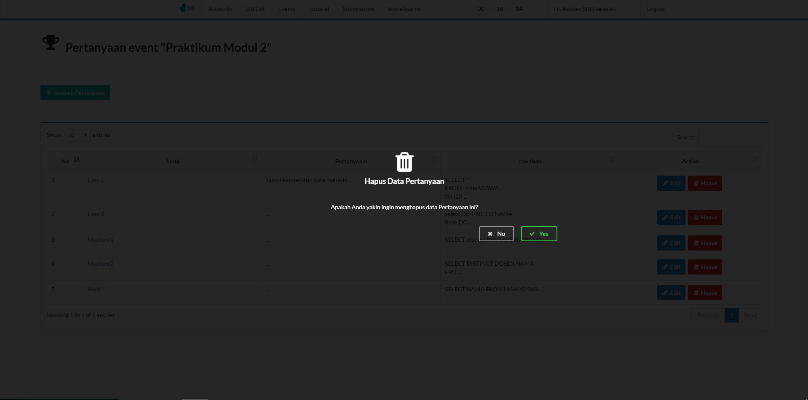
Gambar 3.42 Rancangan Antarmuka Mengelola Soal



Gambar 3.43 Rancangan Antarmuka Tambah Soal



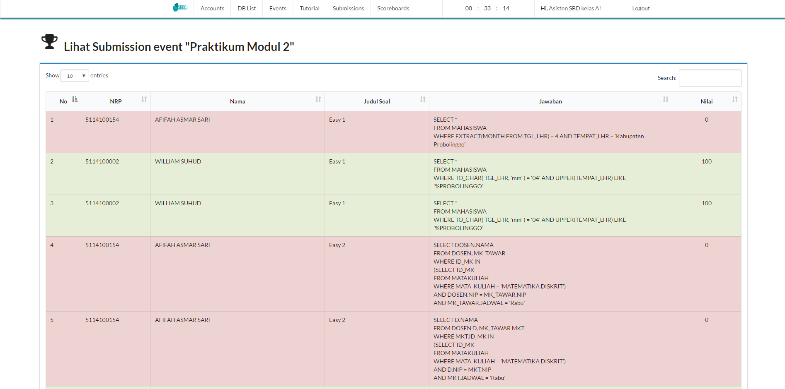
Gambar 3.44 Rancangan Antarmuka Ubah Soal



Gambar 3.45 Rancangan Antarmuka Hapus Soal

#### Rancangan Antarmuka Melihat *Submissions*

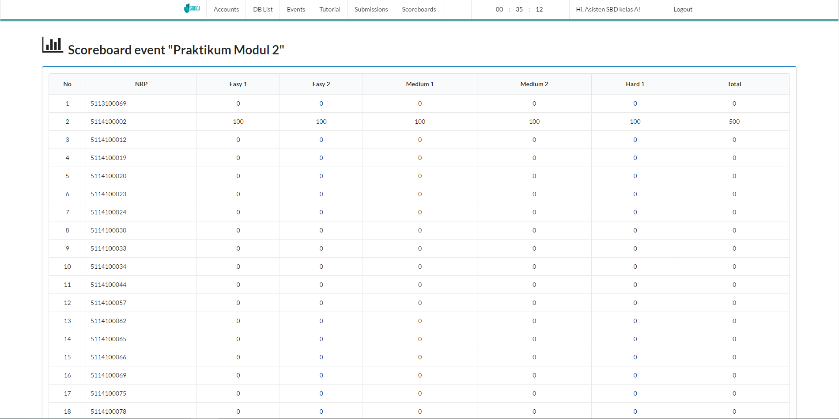
Rancangan antarmuka ini digunakan untuk kasus penggunaan melihat *submissions*. Halaman *submissions* menampilkan tabel dengan isi daftar *submissions* dari praktikan dalam suatu praktikum atau *tutorial*. Terdapat pembeda pada tabel tersebut, yaitu apabila jawaban salah maka akan diberi latar warna merah sedangkan untuk jawaban benar diberi warna hijau. Rancangan antarmuka *submissions* dapat dilihat pada Gambar …. .



Gambar 3.45 Rancangan Antarmuka Daftar *Submissions*

#### Rancangan Antarmuka Melihat *Scoreboard*

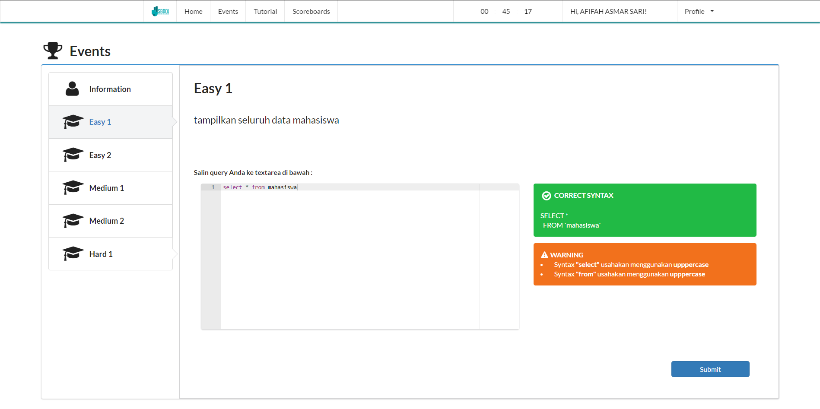
Rancangan antarmuka ini digunakan untuk kasus penggunaan melihat *scoreboard*. Halaman *scoreboard* menampilkan tabel dengan isi *scoreoard* dari suatu praktikum. Pada tabel tersebut terdapat nilai dari masing-masing praktikan pada masing-masing soal dan juga terdapat total nilai dari tiap praktikan. Rancangan antarmuka *scoreboard* dapat dilihat pada Gambar …. .



Gambar 3.48 Rancangan Antarmuka *Scoreboard*

#### Rancangan Antarmuka Menjawab Soal *Event*

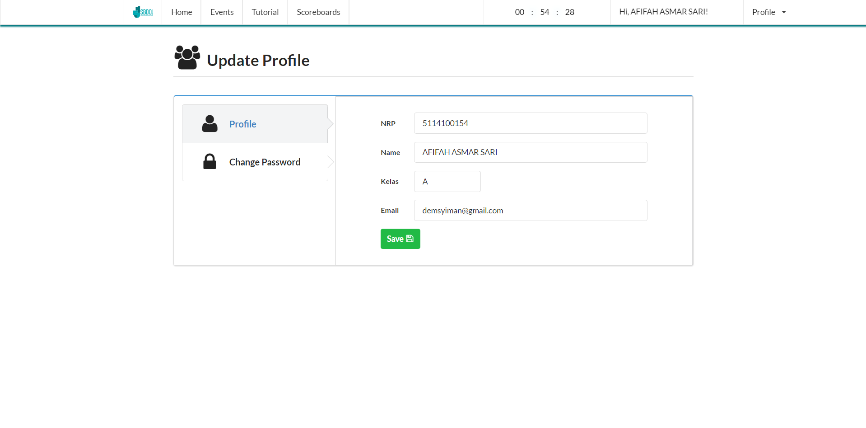
Rancangan antarmuka ini digunakan untuk kasus penggunaan menjawab soal praktikum dan *tutorial*. Pada antarmuka tersebut soal ditampilkan dalam bentuk daftar tabulasi dan pada masing-masing soal terdapat *field* untuk memasukkan jawaban praktikan. Pada masing-masing soal juga terdapat SQL Parser untuk mengecek benar atau tidaknya *syntax* sql dari praktikan. Rancangan antarmuka menjawab soal *event* dapat dilihat pada Gambar …. .



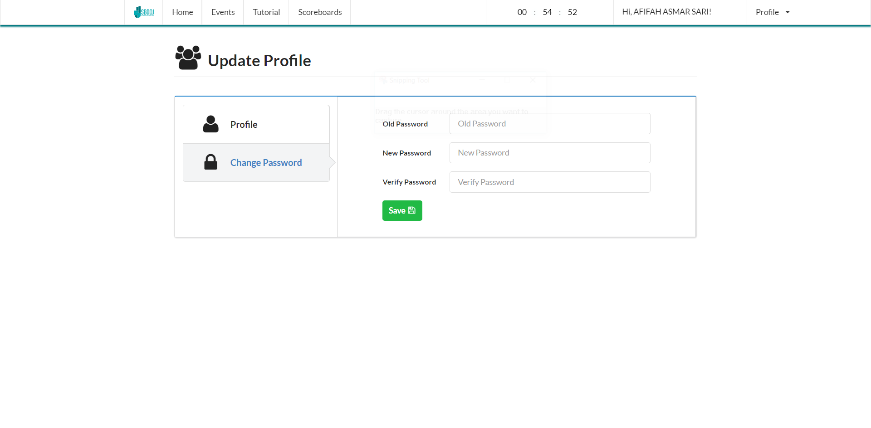
Gambar 3.50 Rancangan Antarmuka Daftar Pembayaran

#### Rancangan Antarmuka Mengelola Akun

Rancangan antarmuka ini digunakan untuk kasus penggunaan mengelola akun parktikan. Pada antarmuka tersebut terdapat beberapa menu. Menu pertama adalah untuk mengubah data diri praktikan. Menu ini menampilkan formulir denga nisi NRP, nama, kelas dan juga email dari praktikan yang dapat dilihat pada Gambar … . Sedangkan menu selanjutnya adalah mengubah password. Praktikan dapat mengubah *password*-nya dengan mengisi formulir pada halaman ini. Formulir tersebut berisi *password* lama, *password* baru dan konfirmasi *password* baru. Menu ini dapat dilihat pada Gambar … .



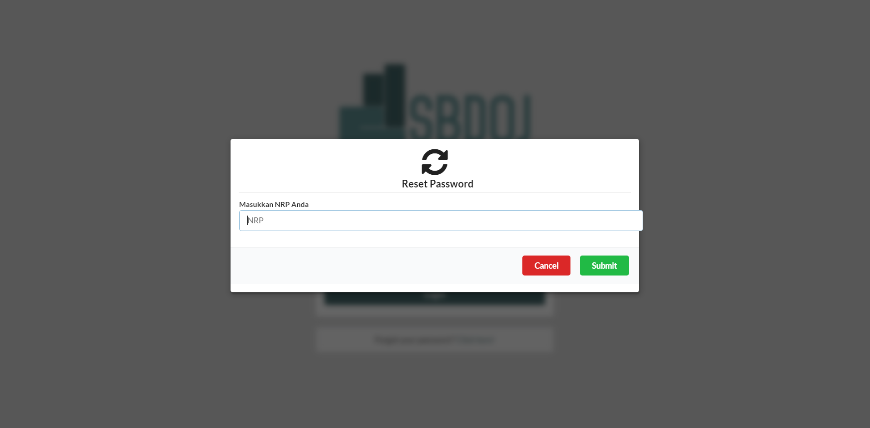
Gambar 3.52 Rancangan Antarmuka Mengubah Data Diri Praktikan



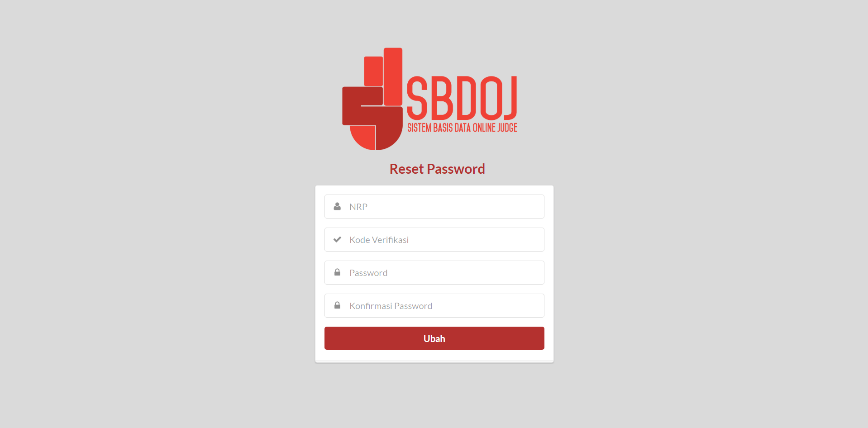
Gambar 3.53 Rancangan Antarmuka Mengubah *Password*

#### Rancangan Antarmuka Me-*reset Password*

Rancangan antarmuka ini digunakan untuk kasus penggunaan me-*reset password*. Pada antarmuka tersebut terdapat *pop-up* untuk praktikan memasukkan NRP yang dapat dilihat pada Gambar … . Setelah praktikan mengirim NRP, maka akan diarahkan ke halaman *reset password* yang menampilkan formulir dengan isi NRP, kode verifikasi, *password* baru dan konfirmasi *password* baru. Antarmuka tersebut dapat dilihat pada Gambar … .



Gambar 3.54 Rancangan Antarmuka Memasukkan NRP



Gambar 3.55 Rancangan Antarmuka *Reset Password*

# BAB IV IMPLEMENTASI

## Lingkungan Implementasi

## Implementasi Perangakat Lunak

## Implementasi Antarmuka Pengguna

# BAB V PENGUJIAN DAN EVALUASI

## Lingkungan Pelaksanaan Pengujian

## Skenario Pengujian

## Evaluasi Pengujian

# BAB VI KESIMPULAN DAN SARAN

## Kesimpulan

## Saran

# DAFTAR PUSTAKA

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | [1] | A. M. Revilla, S. Manzoor dan R. Liu, Competitive Learning in Informatics: The UVa Online Judge Experience, Vilnius, 2008. | | [2] | R. Ramakrishnan dan G. Johannes, Database Management System Second Edition, 1999. | | [3] | R. Ramakrishnan dan G. Johannes, Database Management System Second Edition, 1999. | | [4] | A. Rohman, “Mengenal Framework "Laravel",” Ilmuti, Tangerang, 2014. | | [5] | Y. Purdianto, “Pengenalan PHP,” [Online]. Available: http://yadicucuklauk.blogspot.co.id/2012/01/pengenalan-php.html. [Diakses 15 May 2016]. | | [6] | D. N. Prabawa, “Pengertian Oracle,” 23 November 2014. [Online]. Available: http://dibyanp.blogspot.co.id/2014/11/pengertian-oracle.html. [Diakses 15 May 2016]. | | [7] | M. Pratama, M. Komarudin dan H. Fitriawan, “Rancang Bangun Sistem Informasi Laboratorium Teknik Elektro Terpadu Universitas Lampung Berbasis Web,” *Jurnal Rekayasa dan Teknologi Elektro,* vol. 8, p. September, 2014. | |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |