­­­­­­PEMBANGUNAN SISTEM APLIKASI PENGELOLAAN SOAL MENGGUNAKAN METODE PROTOTYPING (STUDI KASUS : FAKULTAS ILMU KOMPUTER UNIVERSITAS BRAWIJAYA)

SKRIPSI

Disusun oleh:

Abul A’la Alghifari

NIM: 145150200111112



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2018

DAFTAR ISI

[DAFTAR ISI ii](#_Toc512317552)

[DAFTAR TABEL iv](#_Toc512317553)

[DAFTAR GAMBAR v](#_Toc512317554)

[DAFTAR LAMPIRAN vi](#_Toc512317555)

[BAB 1 PENDAHULUAN 1](#_Toc512317556)

[1.1 Latar belakang 1](#_Toc512317557)

[1.2 Rumusan masalah 2](#_Toc512317558)

[1.3 Tujuan 2](#_Toc512317559)

[1.4 Manfaat 3](#_Toc512317560)

[1.5 Batasan masalah 3](#_Toc512317561)

[1.6 Sistematika pembahasan 3](#_Toc512317562)

[BAB 2 LANDASAN KEPUSTAKAAN 5](#_Toc512317563)

[2.1 Bank Soal 5](#_Toc512317564)

[2.2 Siklus Hidup Pengembangan Sistem 6](#_Toc512317565)

[2.3 Model *Prototyping* 7](#_Toc512317566)

[2.4 UML (*Unified Modeling Language*) 9](#_Toc512317567)

[2.5 Teori Pengujian 12](#_Toc512317568)

[2.5.1 White Box Testing 12](#_Toc512317569)

[2.5.2 Black Box Testing 12](#_Toc512317570)

[BAB 3 METODOLOGI 14](#_Toc512317571)

[3.1 Studi Literatur 15](#_Toc512317572)

[3.2 Analisis Kebutuhan 15](#_Toc512317573)

[3.3 Perancangan Sistem 15](#_Toc512317574)

[3.4 Implementasi Sistem 16](#_Toc512317575)

[3.5 Pengujian dan Analisis 16](#_Toc512317576)

[3.6 Kesimpulan dan Saran 16](#_Toc512317577)

[BAB 4 ANALISIS KEBUTUHAN 17](#_Toc512317578)

[4.1 Analisis Kebutuhan Sistem 17](#_Toc512317579)

[4.2 Gambaran Umum Aplikasi 17](#_Toc512317580)

[4.3 Identifikasi Aktor 17](#_Toc512317581)

[4.4 Kebutuhan Fungsional Sistem 18](#_Toc512317582)

[4.5 *Diagram* Use Case 23](#_Toc512317583)

[4.6 *Use Case Scenario* 24](#_Toc512317584)

[DAFTAR PUSTAKA 39](#_Toc512317585)

DAFTAR TABEL

**No table of figures entries found.**

DAFTAR GAMBAR

[Gambar 2‑1 Model *Prototyping* 8](#_Toc503950989)

[Gambar 2‑2 *Use Case* Diagram 10](#_Toc503950990)

[Gambar 2‑3 *Sequence* Diagram 11](#_Toc503950991)

[Gambar 2‑4 *Class* Diagram 12](#_Toc503950992)

[Gambar 3‑1 Diagram Metodologi Penelitian 14](file:///C:\Users\abula\Dropbox\SKRIPSI\mine\proposal0.docx#_Toc503950993)

[Gambar 3‑2 Gambaran Implementasi Sistem 16](file:///C:\Users\abula\Dropbox\SKRIPSI\mine\proposal0.docx#_Toc503950994)

DAFTAR LAMPIRAN

**No table of contents entries found.**

# PENDAHULUAN

## Latar belakang

Sebagai pengajar, dosen memiliki tanggung jawab yang besar untuk menyampaikan materi dan menilai apakah materi yang disampaikan dapat diterima dengan baik oleh mahasiswanya. Salah satu ciri keberhasilan dalam mengajar adalah mahasiswa mampu mendapatkan prestasi yang tinggi, prestasi yang dimaksud adalah penguasaan materi atau keterampilan suatu mata kuliah oleh mahasiswa (H.M. Farid Nasution, 2016). Ujian merupakan salah satu metode yang digunakan oleh dosen untuk mengetahui tingkat pemahaman dari mahasiswanya, namun soal ujian yang diberikan tidak dapat dibuat dengan sembarangan dan tanpa memandang kualitas soal. Bank soal dapat menjadi solusi dalam menjaga kualitas soal dan memudahkan dosen untuk menyiapkan ujian dengan mudah, cepat dan efisien (Dhien Astrini, 2016).

Fakultas Ilmu Komputer menerapkan adanya bank soal untuk memudahkan dosennya dalam pembuatan soal UTS dan UAS yang berkualitas. Setiap mata kuliah memiliki sebuah *team teaching* yang terdiri dari dosen yang mengajar mata kuliah tersebut, dimana tim tersebut bertugas untuk mengevaluasi dan menyiapkan soal ujian. Namun pada kenyataannya masih terdapat masalah yang dirasakan oleh *team teaching* Fakultas Ilmu Komputer dalam pengelolaan bank soal. Masalah utama pada pengelolaan bank soal adalah sulitnya mengumpulkan dosen yang tergabung dalam *team teaching* dalam satu waktu dan tempat. Masalah ini disebabkan karena setiap dosen memiliki jadwal mengajar, penelitian, dan rapat yang berbeda-beda. Sehingga seringkali pertemuan *team teaching* dihadiri hanya oleh beberapa dosen. Dampak dari masalah ini adalah kurang maksimalnya *team teaching* dalam menjaga kualitas soal, karena hanya beberapa dosen yang terlibat dalam evaluasi.

Masalah kedua adalah keterlambatan anggota *team teaching* untuk mengumpulkan soal kepada ketua *team teaching*. Keterlambatan ini dapat terjadi karena anggota lupa terhadap tugas yang diberikan, atau ketua yang lupa untuk mengingatkan tugas kepada anggota. Selain adanya faktor lupa, keterlambatan dapat terjadi karena ada kebingungan dalam penentuan prioritas dari tugas-tugas yang diberikan sehingga menghambat tugas utama yang harus diselesaikan (Eko Rosnanda Putra, 2013). Keterlambatan ini akan menghambat proses lainnya dan berujung pada keterlambatan pada penyerahan soal kepada panitia ujian.

Masalah selanjutnya adalah terdapat perbedaan format dalam pengumpulan soal oleh anggota kepada ketua. Akibat perbedaan format ini ketua *team teaching* kesulitan dalam mengumpulkan soal-soal yang telah dibuat, karena harus memindahkan dari suatu format ke format yang lain. Masalah ini secara tidak langsung berhubungan dengan masalah kedua, karena ketua *team teaching* membutuhkan waktu lebih untuk mengumpulkan soal dalam 1 format sehingga masalah ini menjadi salah satu faktor penyebab keterlambatan. Masalah ini terjadi karena setiap anggota memiliki kenyamanan sendiri dalam pembuatan soal, sehingga terdapat perbedaan format.

Masalah-masalah yang terdapat pada pengelolaan soal ini secara tidak langsung juga berdampak kepada efisiensi dan efektivitas kerja dari dosen. Bila efisiensi dan efektivitas kerja menurun maka produktivitas kerja akan menurun sehingga hanya akan membuang tenaga tanpa hasil yang maksimal (Mardjan Dunggio, 2013). Efisiensi dan efektivitas kerja memiliki manfaat yang besar bagi organisasai dan pegawainya, bagi suatu organisasi dengan adanya efisiensi dan efektivitas kerja akan menjamin kelancaraan pelaksanaan tugas, bagi pegawai dapat meningkatkan kesadaran kerja untuk mewujudkan tujuan yang diharapkan (Firanda Raintung, 2016).

Maka berdasarkan masalah yang telah dijelaskan dibangunlah sistem aplikasi pengelolaan soal. Aplikasi ini bertujuan untuk memudahkan dosen dalam melakukan koordinasi, membuat semua dosen dalam tim dapat berpartisipasi, memberi pengingat dalam proses pembuatan soal, dan menghemat waktu dosen agar tidak terbuang. Aplikasi ini dapat membantu dosen untuk membuat soal sesuai dengan format yang ditentukan, ketua *team teaching* juga dapat mengatur pembagian soal yang akan ditugaskan kepada anggotanya, dan memudahkan ketua tim untuk dapat mengumpulkan dan menyaring soal yang telah dibuat. Dengan dibuatnya aplikasi ini diharapkan dapat membantu dosen dalam pengumpulan soal dan juga dapat meningkatkan efisiensi dan efektivitas kerja dosen.

## Rumusan masalah

Berdasarkan uraian latar belakang di atas, didapatkan rumusan masalah dalam membangun sistem aplikasi pengelolaan soal menggunakan metode prototyping adalah sebagai berikut:

1. Bagaimanakah analisis dan spesifikasi kebutuhan dalam membangun sistem aplikasi pengelolaan soal dengan metode *prototyping*?
2. Bagaimana perancangan dan impelementasi sistem aplikasi pengelolaan soal dapat menjadi solusi dari masalah yang ada?
3. Bagaimana hasil pengujian sistem aplikasi pengelolaan soal dengan metode *prototyping* memenuhi kebutuhan?

## Tujuan

Bagian ini berisi tujuan yang ingin dicapai dari skripsi ini. Tujuan yang ditulis harus dapat memberikan arah pada capaian penelitian. Tujuan ini dapat terdiri dari beberapa butir yang masing-masing harus dituliskan dalam kalimat pernyataan yang sederhana dan jelas, sesuai dengan masalah penelitian dan hasil yang ingin dicapai.

1. Menganalisa dan menyusun spesifikasi kebutuhan sistem aplikasi pengelolaan soal dengan metode *prototyping*
2. Merancang dan mengimplementasi sistem aplikasi pengelolaan soal untuk menyelesaikan masalah yang ada.
3. Menguji sistem aplikasi pengelolaan soal dengan metode *prototyping* memenuhi kebutuhan

## Manfaat

1. Untuk penulis, memberikan pengetahuan dan pengalaman dalam mengembangkan aplikasi.
2. Untuk *team teaching*, membantu memudahkan pekerjaan mereka dalam mengevaluasi dan mengumpulkan soal, sehingga dapat menghemat waktu dan tenaga.

## Batasan masalah

Batasan masalah pada pembangunan sistem aplikasi pengelolaan soal ini adalah sebagai berikut :

1. Sistem aplikasi pengelolaan soal hanya cocok digunakan pada Fakultas Ilmu Komputer Universitas Brawijaya.
2. Sistem aplikasi pengelolaan soal membutuhkan browser (aplikasi pendukung) untuk menjalankannya, karena aplikasi berbasis website.
3. Aplikasi membutuhkan koneksi internet untuk dapat dijalankan.

## Sistematika pembahasan

Untuk mencapai tujuan yang diharapkan maka sistematika penulisan yang disusun dalam skripsi ini adalah :

1. Bab I PENDAHULUAN

Pada bab ini akan membahas mengenai latar belakang, identifikasi masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, ruang lingkup, sistemika pembahsan dan jadwal penelitian.

1. Bab II LANDASAN KEPUSTAKAAN

Pada bab ini akan menguraiakan kajian pustaka, dasar-dasar teori yang melandasi penulisan dan penelitian.

1. Bab III METODE PENELITIAN

Pada bab ini menjelaskan mengenai kajian pustaka, analisis kebutuhan, wawancara, pengumpulan data dan pengelolaan data.

1. Bab IV ANALISIS KEBUTUHAN

Pada bab ini akan dijelaskan mengenai analisis kebutuhan untuk membangun sistem aplikasi pengelolaan soal.

1. Bab V PERANCANGAN DAN IMPLEMENTASI

Pada bab ini akan dijelaskan mengenai perancangan sistem yang digunakan dalam pembangunan sistem aplikasi pengelolaan soal, bab ini juga menjelaskan gambaran sistem dan deskripsi sistem hasil analisa kebutuhan dan perancangan yang diimplementasikan ke dalam program komputer.

1. Bab VI PENGUJIAN DAN ANALISIS

Pada bab ini memuat tentang analisa dan hasil pengujian dari penelitian yang telah dilakukan.

1. Bab VII PENUTUP

Pada bab ini berisi tentang kesimpulan dan saran yang diperoleh dari peneltian serta pemecahan masalah yang dilakukan untuk pengembangan lebih lanjut.

# LANDASAN KEPUSTAKAAN

## Kajian Pustaka

Pembahasan pada kajian pustaka adalah tentang penelitian-penelitian yang sebelumnya telah dilakukan yang menjadi refrensi untuk penelitian ini, berikut kajian pustaka yang digunakan :

Penelitian yang dilakukan oleh Ikfan Febriyana. dengan judul Pengembangan Aplikasi Bank Soal Matematika Berbasis Web Untuk Mendukung Proses Evaluasi dan Belajar Mandiri Siswa Di SMA Negeri 1 Sleman. Pada penelitian ini mengembangkan sistem aplikasi bank soal secara khusus untuk mata pelajaran Matematika yang terintegrasi dengan *Computerized Adaptive Testing* (CAT) dalam mendukung proses evaluasi dan belajar mandiri siswa.

Penelitian yang dilakukan oleh M. Redo Alfendo dan Sudji M. dengan judul Analisis Kualitas Butir Soal Teori Kejuruan Kelas X Teknik Pemesinan SMK Muhammadiyah 3 Yogyakarta. Penelitian ini melakukan analisis kualitas butir soal yang mana teknik yang digunakan dalam menganalisis data menggunakan bantuan aplikasi ANATES V.4.0.9.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Nama Penulis, Tahun, dan Judul | Persamaan | Perbedaan | |
| Penelitian Terdahulu | Rencana Penelitian |
| 1 | Ikfan Febriyana, 2015, Pengembangan Aplikasi Bank Soal Matematika Berbasis Web Untuk Mendukung Proses Evaluasi dan Belajar Mandiri Siswa Di SMA Negeri 1 Sleman. | Penggunaan teknologi *web* dalam mengembangkan aplikasi bank soal | Berfokus pada memberikan kemudahan dalam pembuatan bank soal pada 1 mata pelajaran yakni Matematka dan terintegrasi dengan sistem *Computerized Adaptive Testing* (CAT) untuk proses belajar mandiri siswa. | Berfokus pada memberikan kemudahan pada tim pengajar dalam pembuatan bank soal pada beberapa mata kuliah. |
| 2 | M. Redo Alfendo dan Sudji M, 2017, Analisis Kualitas Butir Soal Teori Kejuruan Kelas X Teknik Pemesinan SMK Muhammadiyah 3 Yogyakarta | Analisis soal berdasarkan validitas, reliabilitas, tingkat kesukaran, daya pembeda dan efektivitas pengecoh untuk menjaga kualitas soal pada bank soal. | Proses analisis soal dibantu dengan aplikasi ANATES V.4.0.9 | Proses analisis soal melalui diskusi antar tim pengajar pada masing-masing butir soal sesuai dengan pedoman pembuatan soal yang digunakan. |

## Bank Soal

Menurut Choppin (dalam J. Umar, 1999) bank soal adalah kumpulan butir tes yang terorganisasi dan dikelompokkan dalam jumlah tertentu berdasarkan karakteristik butir tes. Bank soal tidak hanya terfokus dalam pengumpulan soal, tapi juga fokus dalam pemantauan dan penyimpanan sesuai informasi terkait, sehingga memudahkan dalam merakit soal-soal (Thorndike, 1982).

Manfaat dari bank soal dikategorikan dalam 4 kategori oleh Choppin (dalam J. Umar, 1999). Kategori pertama adalah kategori ekonomi, dengan adanya bank soal butir-butir soal dapat digunakan secara berulang. Kategori kedua adalah fleksibilitas, yang mana panjang tes dapat disesuaikan dengan kebutuhan. Kategori konsistensi adalah kategori ketiga dimana dengan adanya bank soal, tes parallel dapat dikembangkan, dan hasilnya dapat dibandingkan. Kategori keempat merupakan kategori keamanan, dengan adanya bank soal dalam penyusunan tes maka akan mengurangi kemungkinan untuk terjadi kebocoran soal.

Dalam pembuatan bank soal, soal yang dimasukkan harus memiliki standar sehingga kualitas bank soal akan terjaga. Menurut Timothy W. Bothell (2001) terdapat 14 aturan dalam pembuatan soal pilihan ganda, aturan tersebut menjelaskan bagaimana pengajar membuat soal pilihan ganda yang baik dan proporsional bagi perserta ujian. Pada aturan tersebut Bothell memerhatikan semua aspek, mulai dari bagaimana membuat soal yang tepat, hingga bagaimana membuat opsi yang mengoceh yang baik.

## Siklus Hidup Pengembangan Sistem

Siklus hidup pengembangan sistem merupakan proses dalam mengembangkan sistem, dimana proses tersebut dibagi dalam beberapa fase. Menurut Naresh Kumar (2013), fase yang terdapat pada siklus hidup pengembangan sistem adalah:

Analisis Kebutuhan

Analisis kebutuhan merupakan fase awal dalam siklus hidup pengembangan sistem. Tujuannya adalah untuk mengerti kebutuhan dari klien dan untuk membuat dokumen secara tepat. Hasil dari Analisis kebutuhan adalah *Software Requirement Spesification* (SRS).

Perancangan

Perancangan adalah langkah awal untuk merancang solusi dari dari masalah yang ada. Fase ini adalah fase yang paling kreatif dalam siklus hidup pengembangan sistem. Hasil akhir dari fase ini adalah *Software Design Document* (SDD).

Implementasi

Pada fase ini *Software Design Document* (SDD) diimplementasi dalam kode menggunakan bahasa pemrograman. Fase ini adalah fase logika dari siklus hidup pengembangan sistem, hasil dari fase ini adalah kode program.

Pengujian

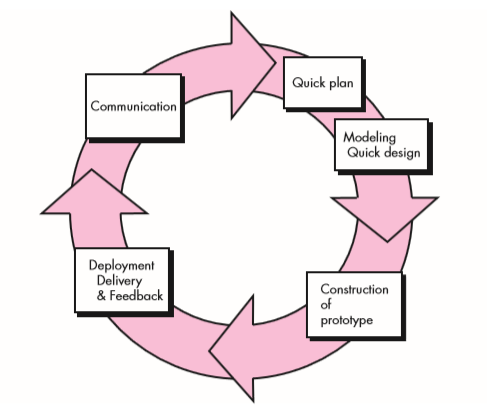
Fase pengujian adalah fase terpenting, pengujian yang efektif akan menghasilkan perangkat lunak berkualitas, kepuasan pelanggan, biaya perawatan yang lebih rendah, dan hasil yang lebih akurat.

Perawatan

Fase ini dimulai setelah produk diluncurkan, jika ada kesalahan yang terjadi atau butuhnya perubahan maka akan dikerjakan pada fase ini.

## Model *Prototyping*

Menurut Roger S. Pressman (2010), seringkali klien tidak mendefinisikan kebutuhannya secara detail sehingga pengembang seringkali salah mengartikan kebutuhan klien, interaksi mesin dan manusia juga seringkali membuat pengembang kesulitan dalam mengembangkan sistem. Model *Prototyping* menjadi solusi untuk masalah tersebut, pengembang membuat *prototype* untuk membantu komunikasi antara klien dan pengembang agar kebutuhan klien dapat tersampaikan dengan baik dan kedua aktor tersebut memiliki banyangan jelas terhadap sistem yang akan dibuat.



Gambar 2‑1 Model *Prototyping*

(Sumber: Pressman, 2010)

Alur dalam model prototyping dapat dilihat dari diagram alir diatas, dengan langkah – langkah sebagai berikut :

Komunikasi

Komunikasi adalah tahap awal yang penting dalam model *prototyping*, karena dalam tahap ini klien mengidentifikasi kebutuhan awal dari suatu perangkat lunak, yang mana akan menjadi dasar pembuatan *prototype*.

Perencanaan dan pemodelan secara sederhana

Kebutuhan dari proses komunikasi akan digunakan pada tahap ini untuk membuat rencana iterasi dan pembuatan model *prototype* yang terfokus pada kebutuhan klien dan akan digunakan pada tahap selanjutnya.

Pembuatan *prototype*

Setelah membuat model *prototype* maka *prototype* akan dibuat pada tahap ini berdasarkan model yang telah dibuat, *prototype* berfokus pada interface dari kebutuhan klien.

Ditunjukkan dan dievaluasi oleh klien

Pada tahap ini *prototype* akan ditunjukkan pada klien dan akan dievalusi oleh mereka, sehingga jika masih ada hal yang belum sesuai dengan keinginan mereka maka akan di kumpulkan dan akan dilakukan iterasi dari proses komunikasi kembali, iterasi akan berhenti jika kebutuhan klien telah terpenuhi sepenuhnya.

Menurut Naresh Kumar (2013), model *prototyping* ini memiliki keuntungan dan kerugian sebagai berikut:

Keuntungan:

1. Klien dapat terlibat langsung dalam pengembangan.
2. Memudahkan pengembang dalam mendapatkan kebutuhan yang jelas dari klien.
3. Mengurangi resiko kegagalan, karena risiko dapat diidentifikasi sejak dini.
4. Klien tidak perlu menunggu lama untuk pengerjaan sistem.

Kerugian

1. Membuang waktu dan uang untuk membangun *prototype*, jika ternyata klien tidak puas dengan hasilnya.
2. Banyaknya perubahan yang diberikan klien dapat mengganggu irama tim pengembang.
3. Butuh waktu lebih banyak.

Pada penelitian ini model *prototyping* digunakan untuk memudahkan peneliti dalam menggali kebutuhan terhadap pemangku kepentingan. Penggunaan model *prototyping* diharapkan peneliti mampu untuk menjaring seluruh kebutuhan dan memperkecil kemungkinan untuk terjadinya ambiguitas dalam penggalian kebutuhan. Peneliti menggali kebutuhan dengan wawancara, dan dari kebutuhan yang didapatkan peneliti membuat *prototype* sehingga pemangku kepentingan memiliki gambaran jelas seperti apa sistem yang akan dibuat.

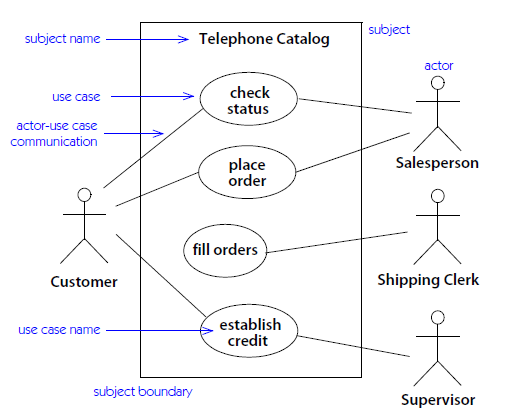
Model *prototyping* dalam penelitian ini juga mempermudah peneliti dan pemangku kepentingan dalam hal merevisi kebutuhan, sehingga memberikan kejelasan bagi peneliti dalam proses perancangan dan implementasi. Proses revisi kebutuhan dapat dilakukan berulang kali oleh pemangku kepentingan hingga kebutuhan yang diinginkan tercapai. Setelah kebutuhan yang diinginkan telah tercapai peniliti dapat melanjutkan proses perancangan dan implementasi dari kebutuhan yang telah didapatkan.

## UML (*Unified Modeling Language*)

*Unified Modeling Language* adalah pemodelan visual yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasi sistem perangkat lunak (James Rumbaugh, 2004). UML terdiri dari beberapa jenis diagram, yaitu:

*Use Case* Diagram

*Use case* berguna untuk mengetahui bagaimana sebuah sistem harus bekerja. *Use case*  membantu untuk membangun skenario proses dari suatu sistem, *Use case* merupakan solusi terbaik untuk menganalisis sistem berorientasi objek (IBM, 2018).



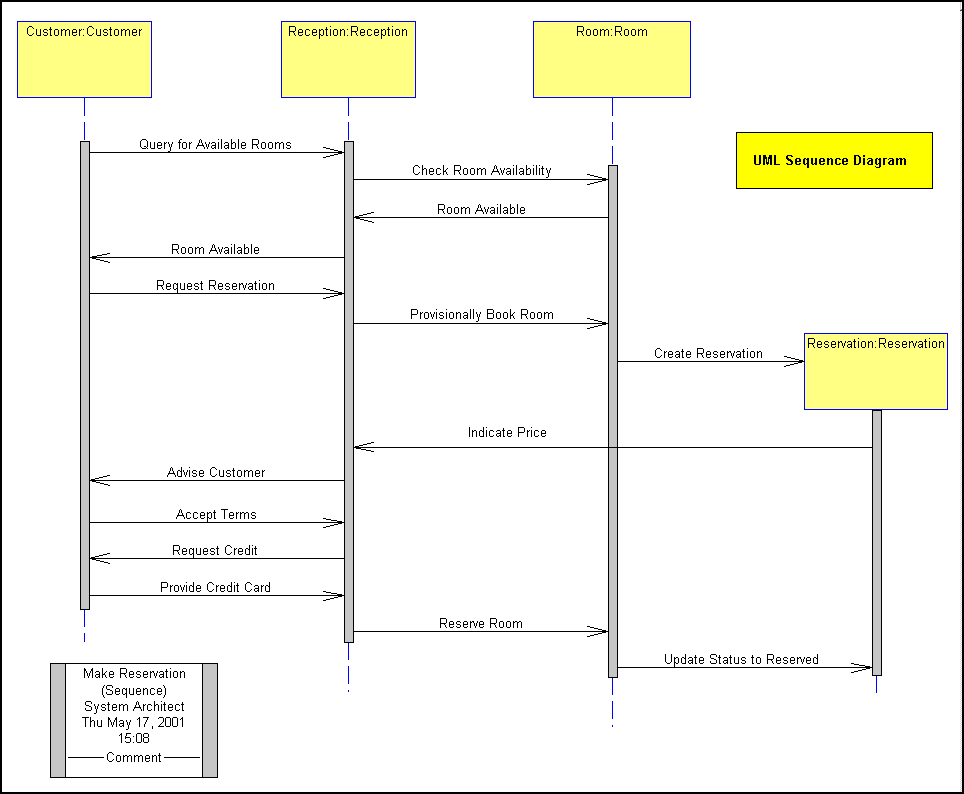
Gambar 2‑2 *Use Case* Diagram

(Sumber: *Unified Modeling Language Reference Manual, The (2nd Edition)* , 2010)

*Use case* Diagram memodelkan fungsionalitas sistem terhadap aktor yang berinteraksi dengan sistem dari sudut pandang tertentu. Salah satu tujuan dari *Use case* Diagram adalah untuk mendapatkan aktor dan *use case* apa saja yang ada pada suatu sistem, dan juga interaksi antara aktor dan *use case* yang ada (James Rumbaugh, 2004).

*Sequence* Diagram

*Sequence* Diagram digunakan untuk memodelkan rincian dari skenario sistem. Diagram ini menjelaskan interaksi pertukaran pesan antar objek untuk menghasilkan hasil yang diinginkan (IBM, 2018).



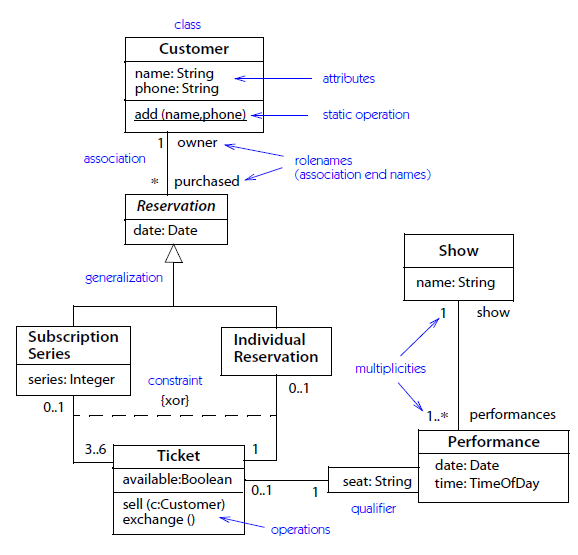
Gambar 2‑3 *Sequence* Diagram

(Sumber: IBM , 2018)

*Sequence* Diagram menunjukkan objek yang terlibat dalam skenario dengan garis vertikal putus-putus yang disebut *lifelines*. Garis horizontal antar objek menunjukkan pertukaran pesan antar objek, kronologi pertukaran pesan dapat dilihat dari atas diagram ke bawah.

*Class* Diagram

*Class* Diagram adalah Diagram statis analisis utama untuk sebuah sistem, dimana struktur kelas dari sistem ditentukan dan menentukan hubungan antar kelas dan turunannya (IBM, 2018).



Gambar 2‑4 *Class* Diagram

(Sumber: *Unified Modeling Language Reference Manual, The (2nd Edition)* , 2010)

## Teori Pengujian

### White Box Testing

White box testing, atau biasa disebut glass-box testing adalah tes untuk menguji desain dan kasus uji yang menggunakan struktur control. Menggunakan white box testing maka akan diperoleh uji kasus yang menjamin semua jalur independen dalam sebuah modul telah selesai dieksekusi setidaknya sekali, melaksanakan semua keputusan logis entah itu benar atau salah, mengeksekusi semua perulangan pada batas tertentu, memeriksa struktur data unutk memastikan validitasnya (Pressman, 2010).

### Black Box Testing

Black box testing atau biasa disebut behavioral testing berfokus dalam kebutuhan fungsional dari perangkat lunak. Tenik black box testing memungkinkan kita untuk memberikan inputan yang mana akan digunakan untuk menguji suatu fungsional. Black box testing bukan alternatif dari white box testing, namun black box testing merupkan pelengkap dimana memungkinkan untuk mengungkapkan kesalahan yang tidak terungkap di white box testing (Pressman, 2010).

# METODOLOGI

Metodologi penelitian, dari kata metode yang berarti cara atau langkah untuk melakukan suatu hal. Logos yang berarti ilmu atau pengetahuan, sehingga arti dari metodologi adalah cara untuk melakukan suatu hal untuk mencapai tujuan menggunakan ilmu pengetahuan. Sedangkan penelitian sendiri merupakan pemikiran yang sistematis untuk memecahkan masalah dengan fakta-fakta yang ada. Penelitian yang akan dilakukan disini merupakan jenis penelitian studi kasus, yang mana pada penelitian ini mempelajari individu atau kelompok tertentu yang mengalami kasus/ masalah, dimana hasil dari penelitian ini adalah solusi dari kasus atau masalah yang dihadapi. Tahapan metodologi penelitian ini dijelaskan pada diagram dibawah ini :

Mulai

Studi Literatur

Analisis Kebutuhan

Perancangan sistem

Implementasi Sistem

Pengujian dan Analisis

Kesimpulan

Selesai

Gambar 3‑1 Diagram Metodologi Penelitian

Gambar diagram diatas menjelaskan tahapan yang dilakukan oleh peneliti dalam pembangunan sistem aplikasi pengelolaan soal mulai dari survei, pengumpulan data, pengolahan data, perancangan sistem, implementasi, pengujian, dan mendapatkan kesimpulan.

## Studi Literatur

Studi literatur menjelaskan dasar teori yang digunakan didapat dari penelitian yang pernah dilakukan sebelumnya, buku, *ebook*, *journal,* dan beberapa literatur dari internet untuk menunjang penulisantugas akhir. Berikut teori-teori pendukung tersebut meliputi:

1. Pengertian Bank Soal
2. Siklus Hidup Pengembangan Sistem
3. Model *Prototyping*
4. UML (*Unified Modeling Language)*
5. Teori Pengujian

## Analisis Kebutuhan

Analisis kebutuhan dibutuhkan untuk menggali kebutuhan yang akan digunakan dalam proses perancangan. Kebutuhan dari penelitian akan didapatkan dari hasil wawancara terhadap narasumber dari ketua *team teaching* mata kuliah Rekayasa Perangkat Lunak Bayu Priyambadha, S.Kom, M.Kom dan ketua *team teaching* mata kuliah Analisis Perancangan Sistem Fajar Pradana, S.ST, M.Eng. Proses wawancara dilakukan dengan memberi pertanyaan tentang sistem yang sedang berjalan, masalah yang dihadapi terhadap sistem, dan berdiskusi tentang solusi yang akan dibuat. Setelah mendapatkan kebutuhan maka didapatkan kebutuhan yang akan menjadi dasar dalam proses perancangan.

## Perancangan Sistem

Perancangan sistem dikerjakan berdasarkan kebutuhan yang didapatkan dari analisis kebutuhan. Dalam penelitian ini perancangan dilakukan dengan metode *Object Oriented Design*, dimana suatu masalah akan dimodelkan menurut konsep sekitar dunia nyata dan diorganisir sebagai kumpulan objek yang memiliki struktur data dan perilakunya. Proses perancangan sistem dan *prototype* ini dapat dilakukan setelah kebutuhan dari *stackholder* telah disepakati, sehingga dapat masuk ke proses perancangan dan *prototyping*.

Proses perancangan dan *protoyping* dapat dikatakan selesai apabila *stackholder* tidak memiliki revisi berupa tambahan atau pengurangan dari *prototype* yang telah diajukan. Jika *stackholder* masih memiliki revisi maka pengembang diharuskan untuk mengulang proses analisa kebutuhan, sehingga mendapatkan kebutuhan yang sesuai dilanjutkan ke proses perancangan dan membuat *prototype* yang baru untuk di presentasikan kembali kepada *stackholder*.

Hasil dari tahap ini akan menghasilkan rancangan sistem berupa:

*Sequence* diagram

*Class* diagram

*Prototype* sistem

## Implementasi Sistem

Proses implementasi dapat dilakukan setelah *stackholder* menyetujui *prototype* yang telah dibuat. Implementasi dilakukan menggunakan *framework* Codeigniter. Implementasi akan dilakukan sesuai perancangan yang telah dibuat. Penggunaan aplikasi desain seperti photoshop juga diperlukan sebagai media desain untuk menyiapkan aset gambar yang diperlukan pada sistem.

Gambar 3‑2 Gambaran Implementasi Sistem

Sistem

Database

## Pengujian dan Analisis

Pengujian dilakukan untuk melihat apakah sistem yang dibuat telah sesuai dengan kebutuhan dan dapat menyelesaikan masalah yang ada. Terdapat 2 tahap pengujian yakni untuk menguji fungsional sistem itu sendiri menggunakan *black box testing* dan *white box testing*. Pada *white box testing* digunakan teknik *cyclomatic complexity* untuk mencoba kemungkinan jalur dari suatu fungsionalitas. Hasil pengujian akan digunakan untuk mengevaluasi sistem dan mengoptimalkan sistem.

## Kesimpulan dan Saran

Kesimpulan didapatkan setelah semua proses dilakukan, kesimpulan bertujuan untuk menjawab masalah yang telah dirumuskan pada rumusan masalah. Selain kesimpulan saran dibutuhkan untuk memperbaiki kesalahan – kesalahan yang terjadi pada penelitian dan memberikan pandangan serta pertimbangan untuk pengembangan sistem.

# ANALISIS KEBUTUHAN

Dalam analisis kebutuhan yang dilakukan pada bab ini menjelaskaan gambaran umum sistem yang akan dibangun, aktor yang akan menggunakan aplikasi dan kebutuhan dalam membangun sistem.

## Analisis Kebutuhan Sistem

Analisis kebutuhan bertujuan untuk menentukan kebutuhan umum sistem untuk membangun sistem aplikasi pengelolaan soal. Penggalian kebutuhan dilakukan dengan wawancara terhadap ketua *team teaching* mata kuliah Rekayasa Perangkat Lunak Bayu Priyambadha, S.Kom, M.Kom dan ketua *team teaching* mata kuliah Analisis Perancangan Sistem Fajar Pradana, S.ST, M.Eng. Hasil dari wawancara terdapat beberapa masalah yang dirasakan terhadap sistem yang sudah ada, masalah yang dirasakan berupa kesulitan dalam mengumpulkan dosen dalam satu waktu dan tempat, keterlambatan anggota *team teaching* dalam mengumpulkan soal, dan sering adanya perbedaan format dalam pengumpulan soal. Dari masalah yang telah dijabarkan maka didapatkan kebutuhan sebagai solusi dari masalah berupa sistem yang harus mampu untuk memfasilitasi *team teaching* untuk mendiskusikan soal yang dibuat, mampu memberikan *reminder* untuk mengatasi keterlambatan, dan memberikan kemudahan pada *team teaching* untuk mengumpulkan soal yang dibuat dan memfilternya. Selain kebutuhan juga didapatkan aktor yang berinteraksi dengan sistem sesuai dari kebutuhan yang ada, aktor yang didapatkan berupa Koordinator KJFD (Kelompok Jabatan Fungsional Dosen) yang fungsinya dapat membuat dan merubah anggota *team teaching*, Ketua *team teaching* yang bertugas membuat penugasan dan merevisi soal, dan anggota *team teaching* yang bertugas dalam pembuatan soal dan membantu dalam proses revisi soal.

## Gambaran Umum Aplikasi

Aplikasi pengelola soal ini adalah aplikasi yang bertujuan untuk mempermudah *team teaching*  dalam membuat, mengumpulkan dan menjaga standar soal yang dibuat. Menggunakan aplikasi ini ketua *team teaching* dapat menentukan bab dan kuota anggota yang dibutuhkan dalam hal pembagian tugas dan anggota *team teaching* akan memilih bab yang telah disediakan dan membuat soal sesuai bab yang dipilih. Ketua *team teaching* akan menerima soal yang telah dibuat anggotanya dan memilah soal dan menghasilkan draf soal yang akan mendapat masukan dari anggota untuk direvisi oleh ketua *team teaching* sesuai masukan yang ada. Setelah direvisi oleh ketua *team teaching* didapatkan soal yang sesuai standar dan siap untuk diberikan kepada panitia ujian sesuai format yang dibutuhkan.

## Identifikasi Aktor

Aktor merupakan seseorang ataupun sistem yang dapat berinteraksi dengan sistem. Adapun aktor dalam aplikasi ini ditujukkan pada tabel 4.1.

**Tabel 4.1 Identifikasi Aktor**

|  |  |
| --- | --- |
| **Aktor** | **Deskripsi** |
| Pengguna | Merupakan aktor umum yang belum dikenali oleh sistem. |
| Koordinator KJFD (Kelompok Jabatan Fungsional Dosen) | Merupakan pengguna yang telah dikenali sistem sebagai koordinator yang mengatur dosen dalam beberapa mata kuliah tertentu, berfungsi untuk menentukan ketua dan anggota *team teaching*. |
| Ketua *team teaching* | Merupakan pengguna yang telah dikenali sistem sebagai ketua dari suatu tim pada mata kuliah tertentu, dimana bertugas untuk mengatur timnya untuk membuat soal dan mengevaluasi pembelajaran. |
| Anggota *team teaching* | Merupakan pengguna yang telah dikenali sistem sebagai dosen yang mengajar mata kuliah tertentu dan bekerja sama dengan ketua *team teaching* untuk membuat soal dan mengevaluasi pembelajaran. |

## Kebutuhan Fungsional Sistem

Kebutuhan fungsional sistem merupakan fungsionalitas atau layanan yang harus tersedia dalam sistem, kebutuhan ini bergantung pada sistem yang akan dikembangkan (Richard R F S, 2014). Kebutuhan fungsional harus tersedia sebagai solusi dari masalah yang telah ditentukan sebelumnya. Kebutuhan fungsional didapatkan dengan wawancara terhadap 2 ketua *team teaching* dari 2 mata kuliah yang berbeda yakni Analisis dan Perancangan Sistem dan Rekayasa Perangkat Lunak. Adapun kebutuhan fungsional sistem ditunjukkan pada Tabel 4.2 dan spesifikasinya ditunjukan pada Tabel 4.3. Setiap Kebutuhan akan diberikan kode SRS-TT-F-X untuk kebutuhan fungsional. SRS merupakan *System Requirement Spesification*, TT merupakan singkatan dari *Team Teaching*, F untuk kebutuhan fungsional, dan X menunjukkan nomor dari definisi kebutuhan utama.

**Tabel 4.2 Kebutuhan Fungsional Sistem**

Pengguna

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Kode Kebutuhan** | **Deskripsi** | **Nama Use Case** |
| 1 | SRS-TT-F-1 | Sistem harus mampu menyediakan fungsi *register* agar pengguna dapat mendaftarkan diri agar dapat dikenali oleh sistem dan memiliki hak akses. | *Register* |
| 2 | SRS-TT-F-2 | Sistem harus mampu menyediakan fungsi *login* sehingga pengguna dapat dikenali oleh sitem dan mendapatkan hak akses sesuai otoritasnya. | *Login* |

Koordinator KJFD

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Kode Kebutuhan** | **Deskripsi** | **Nama Use Case** |
| 1 | SRS-TT-F-1 | Sistem harus mampu menyediakan fungsi *register* agar pengguna dapat mendaftarkan diri agar dapat dikenali oleh sistem dan memiliki hak akses. | *Register* |
| 2 | SRS-TT-F-2 | Sistem harus mampu menyediakan fungsi *login* sehingga pengguna dapat dikenali oleh sitem dan mendapatkan hak akses sesuai otoritasnya. | *Login* |
| 3 | SRS-TT-F-3 | Sistem harus mampu menyediakan fungsi *logout* agar pengguna yang telah masuk kedalam sistem dapat keluar. | *Logout* |
| 4 | SRS-TT-F-4 | Sistem harus mampu menambahkan *team teaching*. | Menambah *team teaching* |
| 5 | SRS-TT-F-5 | Sistem harus mampu menampilkan daftar *team teaching* yang telah dibuat. | Melihat *team teaching* |
| 6 | SRS-TT-F-7 | Sistem harus mampu mengubah data pada *team teaching* yang telah dibuat. | Mengubah *team teaching* |
| 7 | SRS-TT-F-8 | Sistem harus mampu menghapus *team teaching* yang telah dibuat. | Menghapus *team teaching* |

Ketua *team teaching*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Kode Kebutuhan** | **Deskripsi** | **Nama Use Case** |
| 1 | SRS-TT-F-1 | Sistem harus mampu menyediakan fungsi *register* agar pengguna dapat mendaftarkan diri agar dapat dikenali oleh sistem dan memiliki hak akses. | *Register* |
| 2 | SRS-TT-F-2 | Sistem harus mampu menyediakan fungsi *login* sehingga pengguna dapat dikenali oleh sitem dan mendapatkan hak akses sesuai otoritasnya. | *Login* |
| 3 | SRS-TT-F-3 | Sistem harus mampu menyediakan fungsi *logout* agar pengguna yang telah masuk kedalam sistem dapat keluar. | *Logout* |
| 4 | SRS-TT-F-8 | Sistem harus mampu menyediakan fungsi untuk membuat penugasan yang dibuat oleh Ketua *team teaching*. | Membuat penugasan |
| 5 | SRS-TT-F-9 | Sistem harus mampu menampilkan penugasan yang telah dibuat. | Melihat penugasan |
| 6 | SRS-TT-F-10 | Sistem harus mampu mengubah data pada penugasan yang telah dibuat. | Mengubah penugasan |
| 7 | SRS-TT-F-11 | Sistem harus mampu menghapus penugasanyang telah dibuat. | Menghapus penugasan |
| 8 | SRS-TT-F-12 | Sistem harus mampu mengirim *reminder* kepada dosen yang tergabung dalam penugasan. | Mengirim *reminder* |
| 9 | SRS-TT-F-13 | Sistem harus mampu menampilkan kumpulan soal yang telah dibuat oleh anggota penugasan. | Melihat kumpulan soal |
| 10 | SRS-TT-F-14 | Sistem harus mampu memfilter soal dari kumpulan soal sesuai keinginan ketua *team teaching* dan menghasilkan draf soal. | Memfilter kumpulan soal |
| 11 | SRS-TT-F-15 | Sistem harus mampu menyediakan fungsi revisi soal untuk mengizinkan ketua *team teaching* merevisi soal sesuai masukan dari anggota penugasan. | Merevisi soal |
| 12 | SRS-TT-F-16 | Sistem haris mampu menyediakan fungsi untuk mengizinkan ketua *team teaching* mengacak soal dan/atau jawaban. | Mengacak soal |
| 13 | SRS-TT-F-17 | Sistem harus mampu menyimpan soal. | Menyimpan soal |
| 14 | SRS-TT-F-18 | Sistem harus mampu menyediakan fungsi memilih materi dalam penugasan untuk menentukan materi dalam pembuatan soal. | Memilih materi |
| 15 | SRS-TT-F-19 | Sistem harus mampu menyediakan fungsi membuat soal untuk mempermudah anggota penugasan membuat soal pada sistem. | Membuat soal |
| 16 | SRS-TT-F-20 | Sistem harus mampu menyediakan fungsi yang mengizinkan anggota penugasan memberi komentar terhadap draf soal. | Memberi komentar |

Anggota *team teaching*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Kode Kebutuhan** | **Deskripsi** | **Nama Use Case** |
| 1 | SRS-TT-F-1 | Sistem harus mampu menyediakan fungsi *register* agar pengguna dapat mendaftarkan diri agar dapat dikenali oleh sistem dan memiliki hak akses. | *Register* |
| 2 | SRS-TT-F-2 | Sistem harus mampu menyediakan fungsi *login* sehingga pengguna dapat dikenali oleh sitem dan mendapatkan hak akses sesuai otoritasnya. | *Login* |
| 3 | SRS-TT-F-3 | Sistem harus mampu menyediakan fungsi *logout* agar pengguna yang telah masuk kedalam sistem dapat keluar. | *Logout* |
| 4 | SRS-TT-F-18 | Sistem harus mampu menyediakan fungsi memilih materi dalam penugasan untuk menentukan materi dalam pembuatan soal. | Memilih materi |
| 5 | SRS-TT-F-19 | Sistem harus mampu menyediakan fungsi membuat soal untuk mempermudah anggota penugasan membuat soal pada sistem. | Membuat soal |
| 6 | SRS-TT-F-20 | Sistem harus mampu menyediakan fungsi yang mengizinkan anggota penugasan memberi komentar terhadap draf soal. | Memberi komentar |

**Tabel 4.3 Spesifikasi Kebutuhan**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Kode Kebutuhan** | **Kode Spesifikasi** | **Deskripsi** |
| 1 | SRS-TT-F-1 | SRS-TT-F-1-1 | Sistem menyediakan form *register* untuk menyimpan data berupa nama, *e-mail*, NIP, *username* dan *password*. |
| 2 | SRS-TT-F-2 | SRS-TT-F-2-1 | Sistem menyediakan *form* *login* berupa *username* dan *password*. |
| 3 | SRS-TT-F-4 | SRS-TT-F-4-1 | Sistem menyediakan *form* tambah *team teaching* untuk menyimpan data berupa berupa nama mata kuliah, nama ketua *team teaching*, dan nama anggota *team teaching*. |
| SRS-TT-F-4-2 | Sistem mampu mengidentifikasi *team teaching* sudah pernah dibuat atau belum. |
| 4 | SRS-TT-F-5 | SRS-TT-F-5-1 | Sistem menampilkan data berupa nama mata kuliah, nama ketua *team teaching*, dan nama anggota *team taching*. |
| 5 | SRS-TT-F-6 | SRS-TT-F-6-1 | Sistem menampilkan *form* ubah *team teaching* dengan data berupa nama mata kuliah, nama ketua *team teaching*, dan nama anggota *team taching* yang telah terisi dengan data ketika proses pembuatan. |
| 6 | SRS-TT-F-8 | SRS-TT-F-8-1 | Sistem menyediakan *form* tambah penugasan untuk menyimpan data berupa materi, kuota, dan batas waktu pengumpulan. |
| 7 | SRS-TT-F-9 | SRS-TT-F-9-1 | Sistem menampilkan data berupa berupa materi, kuota, dan batas waktu pengumpulan. |
| 8 | SRS-TT-F-10 | SRS-TT-F-10-1 | Sistem menampilkan *form* ubah penugasan dengan data berupa materi, kuota, dan batas waktu pengumpulan yang telah terisi dengan data ketika proses pembuatan. |
| 9 | SRS-TT-F-12 | SRS-TT-F-12-1 | Sistem mengirimkan *reminder* melalui *e-mail*. |
| SRS-TT-F-12-2 | Sistem mengirim *reminder* kepada anggota penugasan yang belum memenuhi tanggung jawabnya. |
| 10 | SRS-TT-F-13 | SRS-TT-F-13-1 | Sistem menampilkan data soal berupa berupa soal, tingkat kesulitan dan estimasi waktu pengerjaan setiap soal. |
| 11 | SRS-TT-F-19 | SRS-TT-F-19-1 | Sistem menampilkan *form* untuk membuat soal berupa soal, opsi jawaban, estimasi waktu pengerjaan, dan tingkat kesulitan. |

## Kebutuhan Non-Fungsional Sistem

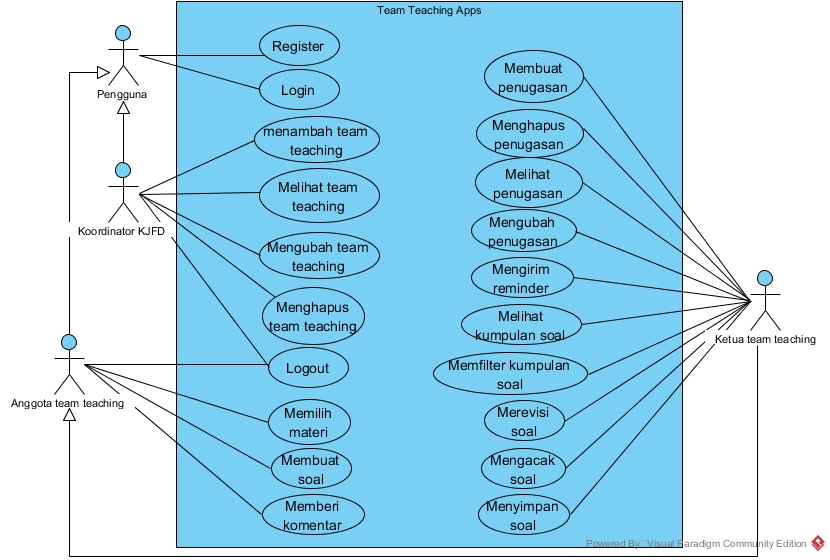
Kebutuhan non-fungsional adalah batasan pada sebuah layanan atau fungsionalitas yang disediakan oleh sistem, kebutuhan non-fungsional biasanya diaplikasikan pada sistem secara keseluruhan, bukan dikenakan pada masingmasing fitur atau layanan (Sommerville, 2011). Setiap Kebutuhan akan diberikan kode SRS-TT-NF-X untuk kebutuhan non-fungsional. SRS merupakan *System Requirement Spesification*, TT merupakan singkatan dari *Team Teaching*, NF untuk kebutuhan non fungsional dan X menunjukkan nomor dari definisi kebutuhan utama.

**Tabel 4.4 Kebutuhan Non Fungsional Sistem**

|  |  |  |
| --- | --- | --- |
| **No.** | **Kode Kebutuhan Sistem** | **Deskripsi Kabutuhan** |
| 1 | SRS-TT-NF-1 | *Security*, sistem akan menggunakan enkripsi md5 untuk keamanan *password* penggunanya. |

## *Diagram* Use Case

Diagram *use case* berisi sejumlah aksi yang dilakukan oleh aktor kepada sistem untuk mencapai tujuan tertentu. Aktor dalam diagram *use case* bisa berupa manusia atau sistem eksternal yang berinteraksi dengan sistem yang dimaksud. Diagram *use case* ditunjukkan dalam Gambar 4.1.



Gambar 4.1 Diagram *Use Case* Sistem

## *Use Case Scenario*

*Use case scenario* adalah penjabaran dari *use case diagram* yang telah digambarkan. Pada *use case scenario* proses penjabaran dilakukan dengan menjelaskan nama *use case*, kode *use case*, aktor sebagai pengguna *use case*, kondisi awal yang harus dipenuhi sebelum menggunakan *use case* (*pre condition)*, alur kerja *use* case untuk mencapai tujuan (*main flow*), alur alternatif jika *main flow* tidak sesuai harapan (*alternative flow*)*,* dan Hasil akhir setelah aktor menjalankan *main flow* (*post condition*). Skenario *use case* akan ditunjukkan oleh Tabel 4.4 hingga Tabel 4.7.

**Tabel 4.5 Skenario *Use Case Login***

|  |  |
| --- | --- |
| Nama *Use Case* | *Register* |
| Kode Kebutuhan Terkait | SRS-TT-F-1 |
| Aktor | Pengguna |
| Tujuan | Pengguna dapat mendaftarkan diri untuk dapat menggunakan fungsi utama sistem |
| *Pre Condition* | Halaman login telah dibuka oleh pengguna |
| *Main flow* | 1. Pengguna menekan tombol *register* 2. Sistem menampilkan *form* pendaftaran berisi nama, *e-mail*, NIP, *username* dan *password* 3. Pengguna mengisi *form* dengan lengkap dan menekan tombol *register* 4. Sistem menyimpan data yang telah diisi pengguna |
| *Post Condition* | Pendfataran diri berhasil dilakukan oleh pengguna |
| *Alternative Flow* | Jika username atau *e-mail* telah digunakan, maka sistem akan menampilkan pesan “username atau *e-mail* telah digunakan” |

**Tabel 4.6 Skenario *Use Case Login***

|  |  |
| --- | --- |
| Nama *Use Case* | *Login* |
| Kode Kebutuhan Terkait | SRS-TT-F-2 |
| Aktor | Pengguna |
| Tujuan | Mengenali pengguna dan memberikan hak akses sesuai otoritas pengguna |
| *Pre Condition* | Halaman login telah dibuka oleh pengguna |
| *Main flow* | 1. Pengguna memasukkan u*sername* dan *password* dan menekan tombol *login* 2. Sistem melakukan pencocokan terhadap *username* dan password yang ada di *database*, membuat *cookies*,dan menampilkan halaman *dashboard* |
| *Post Condition* | Sistem berhasil mengenali pengguna |
| *Alternative Flow* | Jika *username* atau *password* tidak diisi proses login tidak dapat dilanjutkan dan sistem meminta untuk mengisi kolom yang kosong  Jika *username* atau *password* maka akan muncul pesan “Maaf, *Username* atau *Password* anda salah, Silahkan diisi kembali dengan tepat.” |

**Tabel 4.7 Skenario *Use Case Logout***

|  |  |
| --- | --- |
| Nama *Use Case* | *Logout* |
| Kode Kebutuhan Terkait | SRS-TT-F-3 |
| Aktor | Koordinator KJFD, ketua, dan anggota *team teaching* |
| Tujuan | Aktor dapat keluar dari sistem |
| *Pre Condition* | Proses login telah dilakukan aktor |
| *Main flow* | 1. Aktor menekan tombol logout 2. Sistem akan menghapus *cookies* dan kembali ke halaman *login* |
| *Post Condition* | Aktor berhasil keluar dari sistem |
| *Alternative Flow* | - |

**Tabel 4.8 Skenario *Use Case* Menambah *Team Teaching***

|  |  |
| --- | --- |
| Nama *Use Case* | Menambah *Team Teaching* |
| Kode Kebutuhan Terkait | SRS-TT-F-4 |
| Aktor | Koordinator KJFD |
| Tujuan | Koordinator KJFD dapat membuat *team teaching* baru |
| *Pre Condition* | Proses login telah dilakukan Koordinator KJFD |
| *Main flow* | 1. Koordinator KJFD memilih menu ‘*team teaching*’ 2. Sistem akan menampilkan halaman *team teaching* 3. Koordinator KJFD menekan tombol tambah 4. Sistem akan menampilkan halaman tambah *team teaching* berupa form 5. Koordinator KJFD mengisi form berupa nama mata kuliah, nama ketua *team teaching*, nama anggota *team taching* dan menekan tombol tambah 6. Sistem akan menyimpan data yang telah dimasukkan oleh Koordinator KJFD pada *database* dan menampilkan halaman *team teaching* dengan pesan ‘berhasil menambah data’ |
| *Post Condition* | team teaching baru berhasil dibuat oleh Koordinator KJFD |
| *Alternative Flow* | * 1. Jika *team teaching* mata kuliah telah dibuat sebelumnya maka akan muncul pesan ‘gagal menambah data, data sudah pernah dibuat’ |

**Tabel 4.9 Skenario *Use Case* Melihat *Team Teaching***

|  |  |
| --- | --- |
| Nama *Use Case* | Melihat *Team Teaching* |
| Kode Kebutuhan Terkait | SRS-TT-F-5 |
| Aktor | Koordinator KJFD |
| Tujuan | Koordinator KJFD dapat melihat *team teaching* yang telah dibuat |
| *Pre Condition* | Proses login telah dilakukan Koordinator KJFD |
| *Main flow* | 1. Koordinator KJFD memilih menu ‘*team teaching*’ 2. Sistem akan menampilkan halaman *team teaching* |
| *Post Condition* | Daftar *team teaching* dilihat oleh Koordinator KJFD |
| *Alternative Flow* | 1. Jika belum ada data *team teaching* yang dibuat akan memunculkan pesan ‘data belum tersedia, silahkan tambahkan data’ |

**Tabel 4.10 Skenario *Use Case* Mengubah *Team Teaching***

|  |  |
| --- | --- |
| Nama *Use Case* | Mengubah*Team Teaching* |
| Kode Kebutuhan Terkait | SRS-TT-F-6 |
| Aktor | Koordinator KJFD |
| Tujuan | Koordinator KJFD dapat mengubah data *team teaching* yang telah dibuat |
| *Pre Condition* | Proses login telah dilakukan Koordinator KJFD |
| *Main flow* | 1. Koordinator KJFD memilih menu ‘*team teaching*’ 2. Sistem akan menampilkan halaman *team teaching* 3. Koordinator KJFD menekan tombol ‘*edit*’ pada data *team teaching* yang ingin dirubah 4. Sistem akan menampilkan halaman e*dit* *team teaching* berupa form yang telah terisi dengan data sebelumnya 5. Koordinator KJFD dapat mengubah data nama mata kuliah, nama ketua *team teaching*, nama anggota *team taching* dan menekan tombol simpan 6. Sistem akan menyimpan data yang telah dirubah oleh Koordinator KJFD pada *database* dan menampilkan halaman *team teaching* dengan pesan ‘berhasil mengubah data’ |
| *Post Condition* | Data *team teaching* berhasil diubah oleh Koordinator KJFD |
| *Alternative Flow* | 1. Jika tidak ada perubahan yang dilakukan akan muncul pesan ‘tidak ada data yang dirubah’ |

**Tabel 4.11 Skenario *Use Case* Menghapus *Team Teaching***

|  |  |
| --- | --- |
| Nama *Use Case* | Menghapus *Team Teaching* |
| Kode Kebutuhan Terkait | SRS-TT-F-7 |
| Aktor | Koordinator KJFD |
| Tujuan | Koordinator KJFD dapat menghapus data *team teaching* yang telah dibuat |
| *Pre Condition* | Proses login telah dilakukan Koordinator KJFD |
| *Main flow* | 1. Koordinator KJFD memilih menu ‘*team teaching*’ 2. Sistem akan menampilkan halaman *team teaching* 3. Koordinator KJFD menekan tombol ‘hapus’ pada data *team teaching* yang ingin dihapus 4. Sistem akan memunculkan *pop-up* untuk konfirmasi penghapusan 5. Koordinator KJFD menekan tombol ‘Ya’ untuk melakukan penghapusan data 6. Sistem akan menghapus data *team teaching* yang dipilih |
| *Post Condition* | Data *team teaching* berhasildihapus oleh Koordinator KJFD |
| *Alternative Flow* | 1. Jika Koordinator KJFD menekan tombol ‘Tidak’ maka *pop-up* akan tertutup dan data tidak dihapus |

**Tabel 4.12 Skenario *Use Case* Membuat Penugasan**

|  |  |
| --- | --- |
| Nama *Use Case* | Membuat Penugasan |
| Kode Kebutuhan Terkait | SRS-TT-F-8 |
| Aktor | Ketua *Team Teaching* |
| Tujuan | Ketua *Team Teaching* dapat membuat penugasan baru untuk pembagian tugas pembuatan soal |
| *Pre Condition* | Proses login telah dilakukan ketua *team teaching* |
| *Main flow* | 1. Ketua *Team Teaching* memilih menu ‘penugasan’ 2. Sistem akan menampilkan halaman penugasan 3. Ketua *Team Teaching* menekan tombol tambah 4. Sistem akan menampilkan halaman tambah penugasan berupa form 5. Ketua *Team Teaching* mengisi form berupa materi, kuota, batas waktu pengumpulan dan menekan tombol tambah 6. Sistem akan menyimpan data yang telah dimasukkan oleh Ketua *Team Teaching* pada *database* dan menampilkan halaman penugasan dengan pesan ‘berhasil menambah data’ |
| *Post Condition* | Penugasan baru berhasil dibuat oleh Ketua *Team Teaching* |
| *Alternative Flow* | - |

**Tabel 4.13 Skenario *Use Case* Melihat Penugasan**

|  |  |
| --- | --- |
| Nama *Use Case* | Melihat Penugasan |
| Kode Kebutuhan Terkait | SRS-TT-F-9 |
| Aktor | Ketua *Team Teaching* |
| Tujuan | Ketua *Team Teaching* dapat melihat penugasan penugasan yang telah dibuat |
| *Pre Condition* | Proses login telah dilakukan ketua *team teaching* |
| *Main flow* | 1. Ketua *Team Teaching* memilih menu ‘penugasan’ 2. Sistem akan menampilkan halaman penugasan |
| *Post Condition* | Daftar penugasan berhasil dilihat oleh Ketua *Team Teaching* |
| *Alternative Flow* | 1. Jika belum ada data penugasanyang dibuat akan memunculkan pesan ‘data belum tersedia, silahkan tambahkan data’ |

**Tabel 4.14 Skenario *Use Case* Mengubah Penugasan**

|  |  |
| --- | --- |
| Nama *Use Case* | Mengubah Penugasan |
| Kode Kebutuhan Terkait | SRS-TT-F-10 |
| Aktor | Ketua *Team Teaching* |
| Tujuan | Ketua *Team Teaching* dapat mengubah data penugasanyang telah dibuat |
| *Pre Condition* | Proses login telah dilakukan ketua *team teaching* |
| *Main flow* | 1. Ketua *Team Teaching* memilih menu ‘penugasan’ 2. Sistem akan menampilkan halaman penugasan 3. Ketua *Team Teaching* menekan tombol ‘*edit*’ pada data penugasanyang ingin dirubah 4. Sistem akan menampilkan halaman e*dit* penugasanberupa form yang telah terisi dengan data sebelumnya 5. Ketua *Team Teaching* dapat mengubah data materi, kuota, batas waktu pengumpulan dan menekan tombol simpan 6. Sistem akan menyimpan data yang telah dirubah oleh Ketua *Team Teaching* pada *database* dan menampilkan halaman penugasan dengan pesan ‘berhasil mengubah data’ |
| *Post Condition* | Data penugasan berhasil diubah oleh Ketua *Team Teaching* |
| *Alternative Flow* | 1. Jika tidak ada perubahan yang dilakukan akan muncul pesan ‘tidak ada data yang dirubah’ |

**Tabel 4.15 Skenario *Use Case* Menghapus Penugasan**

|  |  |
| --- | --- |
| Nama *Use Case* | Menghapus Penugasan |
| Kode Kebutuhan Terkait | SRS-TT-F-11 |
| Aktor | Ketua *Team Teaching* |
| Tujuan | Ketua *Team Teaching* dapat menghapus data penugasanyang telah dibuat |
| *Pre Condition* | Proses login telah dilakukan ketua *team teaching* |
| *Main flow* | 1. Ketua *Team Teaching* memilih menu ‘penugasan’ 2. Sistem akan menampilkan halaman penugasan 3. Ketua *Team Teaching* menekan tombol ‘hapus’ pada data penugasanyang ingin dihapus 4. Sistem akan memunculkan *pop-up* untuk konfirmasi penghapusan 5. Ketua *Team Teaching* menekan tombol ‘Ya’ untuk melakukan penghapusan data 6. Sistem akan menghapus data penugasan yang dipilih |
| *Post Condition* | Data penugasan berhasil dihapus oleh Ketua *Team Teaching* |
| *Alternative Flow* | 1. Jika Ketua *Team Teaching* menekan tombol ‘Tidak’ maka *pop-up* akan tertutup dan data tidak dihapus |

**Tabel 4.16 Skenario *Use Case* Mengirim *Reminder***

|  |  |
| --- | --- |
| Nama *Use Case* | Mengirim *Reminder* |
| Kode Kebutuhan Terkait | SRS-TT-F-12 |
| Aktor | Ketua *Team Teaching* |
| Tujuan | Ketua *Team Teaching* dapat mengirimkan *reminder* kepada anggota penugasan untuk mengingatkan tenggat penugasan |
| *Pre Condition* | Proses login telah dilakukan ketua *team teaching* |
| *Main flow* | 1. Ketua *Team Teaching* memilih menu ‘penugasan’ 2. Sistem akan menampilkan halaman penugasan 3. Ketua *Team Teaching* menekan tombol ‘kirim *reminder*’ 4. Sistem akan memunculkan *pop-up* untuk konfirmasi pengiriman 5. Ketua *Team Teaching* menekan tombol ‘Ya’ untuk melakukan pengiriman *reminder* 6. Sistem akan mengirim *reminder* kepada *team teaching* yang ada dalam penugasan |
| *Post Condition* | *Reminder* berhasil dikirim oleh Ketua *Team Teaching* |
| *Alternative Flow* | 1. Jika Ketua *Team Teaching* menekan tombol ‘Tidak’ maka *pop-up* akan tertutup dan *reminder* tidak dikirim |

**Tabel 4.17 Skenario *Use Case* Melihat Kumpulan Soal**

|  |  |
| --- | --- |
| Nama *Use Case* | Melihat Kumpulan Soal |
| Kode Kebutuhan Terkait | SRS-TT-F-13 |
| Aktor | Ketua *Team Teaching* |
| Tujuan | Ketua *Team Teaching* dapat melihat soal yang telah dibuat oleh anggotanya |
| *Pre Condition* | Proses login telah dilakukan ketua *team teaching* |
| *Main flow* | 1. Ketua *Team Teaching* memilih menu ‘kumpulan soal’ 2. Sistem akan menampilkan halaman kumpulan soal 3. Ketua *Team Teaching* memilih kumpulan soal dari daftar penugasan yang pernah dibuat 4. Sistem akan menampilkan kumpulan soal pada penugasan tertentu yang dipilih oleh Ketua *Team Teaching* |
| *Post Condition* | Kumpulan soal berhasil dilihat Ketua *Team Teaching* |
| *Alternative Flow* | 1. Jika belum ada data kumpulan soal akan memunculkan pesan ‘data belum tersedia, silahkan buat penugasan’ |

**Tabel 4.18 Skenario *Use Case* Memfilter Kumpulan Soal**

|  |  |
| --- | --- |
| Nama *Use Case* | Memfilter Kumpulan Soal |
| Kode Kebutuhan Terkait | SRS-TT-F-14 |
| Aktor | Ketua *Team Teaching* |
| Tujuan | Ketua *Team Teaching* dapat memfilter soal dari kumpulan soal |
| *Pre Condition* | Proses login telah dilakukan ketua *team teaching* |
| *Main flow* | 1. Ketua *Team Teaching* memilih menu ‘kumpulan soal’ 2. Sistem akan menampilkan halaman kumpulan soal 3. Ketua *Team Teaching* memilih kuota soal pada tingkat kesulitan mudah, sedang, dan sulit untuk memfilter soal yang dibutuhkan 4. Sistem akan menampilkan soal hasil filter dan menghasilkan draf soal |
| *Post Condition* | Kumpulan soal berhasil difilter oleh Ketua *Team Teaching* |
| *Alternative Flow* | - |

**Tabel 4.19 Skenario *Use Case* Merevisi Soal**

|  |  |
| --- | --- |
| Nama *Use Case* | Merevisi Soal |
| Kode Kebutuhan Terkait | SRS-TT-F-15 |
| Aktor | Ketua *Team Teaching* |
| Tujuan | Ketua *Team Teaching* dapat merevisi soal dari draf soal yang telah diberi masukan |
| *Pre Condition* | Proses login telah dilakukan ketua *team teaching* |
| *Main flow* | 1. Ketua *Team Teaching* memilih menu ‘Draf Soal’ 2. Sistem akan menampilkan halaman draf soal 3. Ketua *Team Teaching* memilih draf soal dari penugasan yang ingin direvisi dengan menekan tombol revisi 4. Sistem akan menampilkan draf soal yang dipilih 5. Ketua *Team Teaching* merevisi soal sesuai masukan yang diberikan oleh anggota penugasan dan menyimpan revisi dengan menekan tombol simpan 6. Sistem menyimpan hasil revisi soal, menampilkan halaman draf soal dan menampilkan pesan ‘berhasil merevisi soal’ |
| *Post Condition* | Draf soal berhasil direvisi oleh Ketua *Team Teaching* |
| *Alternative Flow* | - |

**Tabel 4.20 Skenario *Use Case* Mengacak Soal**

|  |  |
| --- | --- |
| Nama *Use Case* | Mengacak Soal |
| Kode Kebutuhan Terkait | SRS-TT-F-16 |
| Aktor | Ketua *Team Teaching* |
| Tujuan | Ketua *Team Teaching* dapat mengacak urutan soal dan urutan jawaban |
| *Pre Condition* | Proses login telah dilakukan ketua *team teaching* |
| *Main flow* | 1. Ketua *Team Teaching* memilih menu ‘Draf Soal’ 2. Sistem akan menampilkan halaman draf soal 3. Ketua *Team Teaching* memilih draf soal dari penugasan yang ingin diacak 4. Sistem akan menampilkan draf soal yang dipilih 5. Ketua *Team Teaching* memilih ingin mengacak urutan soal saja, urutan jawaban saja, atau urutan soal dan jawaban secara bersamaan, kemudian menekan tombol acak 6. Sistem akan mengacak sesuai pilihan Ketua *Team Teaching* dan menampilkan draf soal |
| *Post Condition* | Draf soal berhasil diacak oleh Ketua *Team Teaching* |
| *Alternative Flow* | - |

**Tabel 4.21 Skenario *Use Case* Menyimpan Soal**

|  |  |
| --- | --- |
| Nama *Use Case* | Menyimpan Soal |
| Kode Kebutuhan Terkait | SRS-TT-F-17 |
| Aktor | Ketua *Team Teaching* |
| Tujuan | Ketua *Team Teaching* menyimpan soal dalam format docx |
| *Pre Condition* | Proses login telah dilakukan ketua *team teaching* |
| *Main flow* | 1. Ketua *Team Teaching* memilih menu ‘Draf soal’ 2. Sistem akan menampilkan halaman draf soal 3. Ketua *Team Teaching* memilih draf soal dari penugasan yang ingin disimpan dengan menekan tombol simpan 4. Sistem akan menyimpan draf soal yang dipilih dengan format docx |
| *Post Condition* | Soal berhasil disimpan oleh Ketua *Team Teaching* |
| *Alternative Flow* | - |

**Tabel 4.22 Skenario *Use Case* Memilih Materi**

|  |  |
| --- | --- |
| Nama *Use Case* | Memilih materi |
| Kode Kebutuhan Terkait | SRS-TT-F-18 |
| Aktor | Ketua dan Anggota *Team Teaching* |
| Tujuan | Ketua dan Anggota *Team Teaching* memilih materi penugasan yang telah tersedia |
| *Pre Condition* | Proses login telah dilakukan aktor |
| *Main flow* | 1. Ketua dan Anggota *Team Teaching* memilih menu ‘penugasan’ 2. Sistem menampilkan halaman penugasan 3. Ketua dan Anggota *Team Teaching* memilih materi yang akan dikerjakan dengan menekan tombol ‘*Join*’ 4. Sistem akan menyimpan data |
| *Post Condition* | Materi penugasan berhasil dipilih oleh aktor |
| *Alternative Flow* | - |

**Tabel 4.23 Skenario *Use Case* Membuat Soal**

|  |  |
| --- | --- |
| Nama *Use Case* | Membuat soal |
| Kode Kebutuhan Terkait | SRS-TT-F-19 |
| Aktor | Ketua dan Anggota *Team Teaching* |
| Tujuan | Ketua dan Anggota *Team Teaching* membuat soal dari materi yang telah dipilih |
| *Pre Condition* | Proses login telah dilakukan aktor |
| *Main flow* | 1. Ketua dan Anggota *Team Teaching* memilih menu ‘penugasan’ 2. Sistem menampilkan halaman penugasan 3. Ketua dan Anggota *Team Teaching* memilih penugasan yang ingin dikerakan 4. Sistem menampilkan *form* membuat soal berupa soal, opsi jawaban, estimasi waktu pengerjaan, dan tingkat kesulitan. 5. Ketua dan Anggota *Team Teaching* mengisi form untuk membuat soal dan menekan tombol submit 6. Sistem akan menyimpan data soal yang dibuat |
| *Post Condition* | Soal berhasil dibuat oleh aktor |
| *Alternative Flow* | - |

**Tabel 4.24 Skenario *Use Case* Memberi Komentar**

|  |  |
| --- | --- |
| Nama *Use Case* | Memberi komentar |
| Kode Kebutuhan Terkait | SRS-TT-F-20 |
| Aktor | Ketua dan Anggota *Team Teaching* |
| Tujuan | Ketua dan Anggota *Team Teaching* memberi komentar terhadap draf soal |
| *Pre Condition* | Proses login telah dilakukan aktor |
| *Main flow* | 1. Ketua dan Anggota *Team Teaching* memilih menu ‘penugasan’ 2. Sistem menampilkan halaman penugasan 3. Ketua dan Anggota *Team Teaching* memilih draf soal yang akan dikomentari 4. Sistem menampilkan draf soal yang dipilih 5. Ketua dan Anggota *Team Teaching* memberi komentar pada draf soal dan menekan tombol *submit* 6. Sistem menyimpan komentar pada *database* |
| *Post Condition* | Komentar berhasil diberikan oleh aktor pada draf soal |
| *Alternative Flow* | - |

DAFTAR PUSTAKA

Nasution, H.M. Farid 2016. *Hubungan Metode Mengajar Dosen, Keterampilan Belajar, Sarana Belajar dan Lingkungan Belajar dengan Prestasi Belajar Mahasiswa*.

Astrini, D., Elvira, M. 2015. *Model Bank Soal Berbasis Web Untuk Meningkatkan Kualitas Pendidikan di Era Masyarakat Ekonomi Asean*.

Putra, E.R. 2013. *Pengaruh Disiplin Kerja dan Lingkungan Kerja Terhadap Kinerja Karyawan di Workshop PT Dunia Marine Internusa Pekanbaru*.

Mardjan, D. 2013. S*emangat dan Disiplin Kerja Terhadap Produktivitas Kerja Karyawan Pada PT. Jasa Raharja (persero) Cabang Sulawesi Utara*.

Firanda, R. 2016. *Pentingnya Efisiensi Kerja Dalam Meningkatkan Kinerja Karyawan Pada PT. Kimia Farma Tranding and Distribution*.

Thorndike, R.L. 1982. *Applied Psychometrics*. Boston: Houghton Mifflin.

Umar, J. (1999). Item Banking. Dalam Masters, G.N. dan Keeves, J.P. (Ed). Advances in measurement in educational research and assessment. New York: Pergamon.

Kumar, N., Zadgaonkar, A.S. & Shukla, A. 2013. *Evolving a New Software Development Life Cycle Model SDLC-2013 with Client Satisfaction*.

Rumbaugh, J., Jacobson, I., Booch, G. 2004. *Unified Modeling Language Reference Manual, The (2nd Edition)*. Pearson Higher Education.