

**PEMBANGUNAN SISTEM PENGELOLAAN SOAL (STUDI
KASUS: FAKULTAS ILMU KOMPUTER UNIVERSITAS
BRAWIJAYA)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Abul A'la Alghifari

NIM: 145150200111112



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

PEMBANGUNAN SISTEM PENGELOLAAN SOAL (STUDI KASUS: FAKULTAS ILMU
KOMPUTER UNIVERSITAS BRAWIJAYA)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh :
Abul A'la Alghifari
NIM: 145150200111112

Skripsi ini telah diuji dan dinyatakan lulus pada
19 Desember 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Bayu Priyambadha, S.Kom, M.Kom

NIP. 19820909 200812 1 004

Dosen Pembimbing II



Fajar Pradana, S.ST, M.Eng

NIP. 19871121 201504 1 004

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T, M.T, Ph.D

NIP. 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh oranglain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 19 Desember 2018




Abul A'la Alghifari

NIM: 14515020011112

KATA PENGANTAR

Rasa syukur penulis panjatkan kepada Allah SWT karena atas segala limpahan kasih dan sayang-Nya penulis dapat menyelesaikan skripsi yang berjudul “PEMBANGUNAN SISTEM PENGELOLAAN SOAL (STUDI KASUS: FAKULTAS ILMU KOMPUTER UNIVERSITAS BRAWIJAYA)”. Pada pengantar ini, penulis ingin berterimakasih kepada pihak-pihak yang telah membantu penulis dalam penyusunan skripsi ini, diantaranya kepada:

1. Kedua orang tua tercinta, yang telah memberikan semangat, kasih sayang, dan dukungan yang selalu diberikan dengan tulus kepada penulis.
2. Saudara kandung, saudara ipar, dan calon keponakan penulis yang telah memberikan dukungan moril dan perhatian kepada penulis
3. Bapak Bayu Priyambadha, S.Kom, M.Kom selaku Dosen Pembimbing I skripsi dan Bapak Fajar Pradana, S.ST, M.Eng selaku Dosen Pembimbing II skripsi, yang telah mengarahkan, membina, dan membimbing saya sehingga dapat menyelesaikan skripsi ini.
4. Para Dosen Fakultas Ilmu Komputer Universitas Brawijaya yang telah bersedia membagi ilmu kepada penulis beserta Staff Akademik yang telah memberikan bantuan akademik selama proses menempuh studi hingga penyelesaian skripsi.
5. Teman-teman dari TPOC yang selalu memberi bantuan, motivasi, dan kebersamaan selama ini.
6. Teman-teman teknik informatika 2014 atas bantuan, motivasi dan kebersamaan selama ini sehingga penulis dapat menyelesaikan skripsi dengan baik.
7. Semua pihak yang tidak semuanya bisa dituliskan disini yang terlibat secara langsung maupun tidak langsung dalam proses pengerjaan skripsi maupun sebagai pemberi semangat dan motivasi.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 19 Desember 2018

Penulis

Abul.alghifari@gmail.com

ABSTRAK

Pengelolaan soal merupakan suatu proses yang penting untuk dilakukan dalam proses belajar mengajar, proses pengelolaan soal dilakukan oleh pengajar, dan dengan adanya pengelolaan soal yang baik pengajar akan dapat menjaga kualitas soal. Pada Fakultas Ilmu Komputer Universitas Brawijaya pengelolaan soal dilakukan oleh sebuah tim dosen atau biasa disebut dengan *team teaching*. Saat ini, penerapan pengelolaan soal yang telah berjalan di Fakultas Ilmu Komputer Universitas Brawijaya memiliki beberapa permasalahan, masalah pertama adalah sulitnya mengumpulkan dosen pada satu tempat dan satu waktu. Masalah kedua adalah keterlambatan dari dosen untuk mengumpulkan soal yang telah ditugaskan. Masalah ketiga adalah terdapatnya perbedaan format pengumpulan soal antara satu dosen dengan dosen lainnya, sehingga menyulitkan ketua tim untuk mengkompilasi soal-soal yang dikumpulkan. Berdasarkan masalah tersebut dilakukan penelitian yang dimaksudkan untuk membangun sistem yang mampu untuk mempermudah pembagian penugasan antar dosen, memberikan *reminder* kepada dosen tentang tanggungan penugasannya, dan mempermudah proses kompilasi dan evaluasi soal. Sistem ini diharapkan mampu mengatasi masalah-masalah yang ada dan membantu meningkatkan efisiensi kerja dosen. Sistem ini telah melewati pengujian unit dengan menggunakan *whitebox testing*, pengujian integrasi dan pengujian validasi dengan menggunakan *blackbox testing* yang menghasilkan nilai 100% valid, dan pengujian kompatibilitas dimana hasilnya sistem dapat berjalan pada 8 *browsers* berbeda.

Kata kunci: soal, rekayasa perangkat lunak, fakultas ilmu komputer

ABSTRACT

Questions management is important process to be carried out in teaching and learning process, questions management is carried out by teacher, and with good questions management the teacher will be able to maintain the quality of the questions. At the Faculty of Computer Science Brawijaya University, questions management is carried out by lecturer team or commonly referred to as team teaching. At present, the implementation of questions management that has been running has several problems, the first problem is the difficulty of gathering lecturers in one place at a same time. The second problem is the delay from the lecturer to collect the question that have been assigned. The third problem is the difference in the format of collected questions between one lecturer and another lecturer, that making it difficult for the team leader to combine the collected questions. Based on that problems, research was conducted which was intended to build a system that was able to facilitate the assignment distribution between lecturers, provide reminder to the lecturer about the assignment dependents, and facilitate the process of compiling and evaluating questions. This system is expected to be able to overcome existing problems and help improve the work efficiency of lecturers. This system has passed unit testing using whitebox testing, integration testing and validation testing using blackbox testing which results in a 100% valid value, and compatibility testing where the results of the system can run on 8 different browsers.

Keywords: *questions, software engineering, faculty of computer science*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xv
DAFTAR LAMPIRAN	xvii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	3
1.5 Batasan masalah.....	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Rekayasa Perangkat Lunak	5
2.2.1 Software Development Life Cycle (SDLC).....	6
2.2.2 Pendekatan Berorientasi Objek	7
2.2.3 Pemodelan Berorientasi Objek	7
2.2.4 Teori Pengujian	8
2.3 <i>Business Process Model and Notation</i> (BPMN)	9
2.4 Soal.....	10
2.4.1 Soal Pilihan Ganda.....	10
2.4.2 Aturan Membuat Soal Pilihan Ganda.....	10
2.5 Bank Soal.....	12
2.6 Codeigniter.....	12
2.7 PHPWord.....	13

BAB 3 METODOLOGI	14
3.1 Studi Literatur	14
3.2 Analisis Kebutuhan	15
3.3 Perancangan Sistem.....	16
3.4 Implementasi Sistem	17
3.5 Pengujian Sistem.....	17
3.6 Kesimpulan dan Saran	18
BAB 4 ANALISIS KEBUTUHAN	19
4.1 Elisitasi Kebutuhan.....	19
4.2 Identifikasi Aktor	23
4.3 Kebutuhan Fungsional Sistem	23
4.4 Kebutuhan Nonfungsional Sistem	30
4.5 Validasi dan Verifikasi	31
4.6 Diagram <i>Use Case</i>	33
4.7 <i>Use Case Scenario</i>	33
BAB 5 PERANCANGAN DAN IMPLEMENTASI SISTEM	53
5.1 Perancangan Sistem.....	53
5.1.1 Perancangan Arsitektur.....	53
5.1.2 Perancangan Komponen.....	58
5.1.3 Perancangan Data	60
5.1.4 Perancangan Antarmuka.....	60
5.2 Implementasi Sistem	64
5.2.1 Spesifikasi Sistem	64
5.2.2 Implementasi Kode Program	65
5.2.3 Implementasi Basis Data	67
5.2.4 Implementasi Antarmuka	68
BAB 6 Pengujian	70
6.1 Pengujian Unit.....	70
6.1.2 Pengujian Unit <i>Method</i> “proses_gabung_penugasan” class “C_penugasan”.....	70
6.1.3 Pengujian Unit <i>Method</i> “kirim_reminder” class “C_penugasan”.....	72

6.1.4 Pengujian Unit <i>Method</i> “proses_buat_soal” class “C_penugasan”	74
6.2 Pengujian Integrasi	76
6.3 Pengujian Validasi	80
6.4 Pengujian <i>Compatibility</i>	106
BAB 7 PENUTUP	108
7.1 Kesimpulan	108
7.2 Saran	109
DAFTAR PUSTAKA	110
LAMPIRAN A HASIL WAWANCARA	112
LAMPIRAN B KUESIONER	116



DAFTAR TABEL

Tabel 2.1 Kajian Pustaka	5
Tabel 4.3 Identifikasi Aktor	23
Tabel 4.4 Kebutuhan Fungsional Sistem	24
Tabel 4.5 Kebutuhan Nonfungsional Sistem	31
Tabel 4.6 Skenario <i>Use Case Register</i>	34
Tabel 4.7 Skenario <i>Use Case Login</i>	34
Tabel 4.8 Skenario <i>Use Case Logout</i>	35
Tabel 4.9 Skenario <i>Use Case</i> Melihat Dosen	35
Tabel 4.10 Skenario <i>Use Case</i> Aktivasi Dosen	35
Tabel 4.11 Skenario <i>Use Case</i> Mengubah Data Dosen	36
Tabel 4.12 Skenario <i>Use Case</i> Menghapus Dosen	36
Tabel 4.13 Skenario <i>Use Case</i> Menambah KJFD	37
Tabel 4.14 Skenario <i>Use Case</i> Melihat KJFD	37
Tabel 4.15 Skenario <i>Use Case</i> Mengubah KJFD	37
Tabel 4.16 Skenario <i>Use Case</i> Menghapus KJFD	38
Tabel 4.17 Skenario <i>Use Case</i> Menambah Mata Kuliah	38
Tabel 4.18 Skenario <i>Use Case</i> Melihat Mata Kuliah	39
Tabel 4.19 Skenario <i>Use Case</i> Mengubah Mata Kuliah	39
Tabel 4.20 Skenario <i>Use Case</i> Menghapus Mata Kuliah	39
Tabel 4.21 Skenario <i>Use Case</i> Melihat <i>Team Teaching</i>	40
Tabel 4.22 Skenario <i>Use Case</i> Mengubah <i>Team Teaching</i>	40
Tabel 4.23 Skenario <i>Use Case</i> Menambah Materi	41
Tabel 4.24 Skenario <i>Use Case</i> Melihat Materi	41
Tabel 4.25 Skenario <i>Use Case</i> Mengubah Materi	41
Tabel 4.26 Skenario <i>Use Case</i> Menghapus Materi	42
Tabel 4.27 Skenario <i>Use Case</i> Menambah Penugasan	42
Tabel 4.28 Skenario <i>Use Case</i> Melihat Penugasan	43
Tabel 4.29 Skenario <i>Use Case</i> Melihat Progres Penugasan	43
Tabel 4.30 Skenario <i>Use Case</i> Mengubah Penugasan	43
Tabel 4.31 Skenario <i>Use Case</i> Menghapus Penugasan	44

Tabel 4.32 Skenario <i>Use Case</i> Mengatur Anggota Materi Penugasan	44
Tabel 4.33 Skenario <i>Use Case</i> Gabung Materi Penugasan	45
Tabel 4.34 Skenario <i>Use Case</i> Keluar Materi Penugasan.....	45
Tabel 4.35 Skenario <i>Use Case</i> Membuat Soal.....	46
Tabel 4.36 Skenario <i>Use Case</i> Mengubah Soal	46
Tabel 4.37 Skenario <i>Use Case</i> Submit Soal	47
Tabel 4.38 Skenario <i>Use Case</i> Mempercepat Evaluasi	47
Tabel 4.39 Skenario <i>Use Case</i> Mengirim <i>Reminder</i>	48
Tabel 4.40 Skenario <i>Use Case</i> Memberi Komentar	48
Tabel 4.41 Skenario <i>Use Case</i> Menghapus Komentar	48
Tabel 4.42 Skenario <i>Use Case</i> Mengevaluasi Soal	49
Tabel 4.43 Skenario <i>Use Case</i> Menutup Evaluasi	49
Tabel 4.44 Skenario <i>Use Case</i> Review Soal	50
Tabel 4.45 Skenario <i>Use Case</i> Cetak Soal Acak.....	50
Tabel 4.46 Skenario <i>Use Case</i> Cetak Soal Pilih.....	51
Tabel 4.47 Skenario <i>Use Case</i> Melihat Riwayat Cetak Soal	51
Tabel 4.48 Skenario <i>Use Case</i> Evaluasi Bank Soal.....	52
Tabel 5.1 Perancangan Komponen Method Gabung Penugasan	58
Tabel 5.2 Perancangan Komponen Method Kirim Reminder	59
Tabel 5.3 Perancangan Komponen Method Buat Soal	59
Tabel 5.4 Penjelasan antarmuka detail penugasan	61
Tabel 5.5 Penjelasan antarmuka evaluasi penugasan	62
Tabel 5.6 Penjelasan antarmuka membuat soal.....	63
Tabel 5.7 Spesifikasi Perangkat Keras	64
Tabel 5.8 Spesifikasi Perangkat Lunak	65
Tabel 5.9 Sistem Operasi.....	65
Tabel 5.10 Implementasi <i>Method</i> Gabung Penugasan	65
Tabel 5.11 Implementasi <i>Method</i> Kirim Reminder.....	66
Tabel 5.12 Implementasi <i>Method</i> Buat Soal.....	66
Tabel 6.1 Hasil pengujian unit method “proses_gabung_penugasan” class “C_penugasan”.....	72
Tabel 6.2 Hasil pengujian unit method “kirim_reminder” class “C_penugasan” .	74

Tabel 6.3 Hasil pengujian unit method “proses_buat_soal” class “C_penugasan”	75
Tabel 6.4 Identifikasi dan Perancangan Pengujian Integrasi	76
Tabel 6.5 Hasil Pengujian Integrasi Nomor 1	76
Tabel 6.6 Hasil Pengujian Integrasi Nomor 2	77
Tabel 6.7 Hasil Pengujian Integrasi Nomor 3	79
Tabel 6.8 Hasil Pengujian Validasi <i>Register</i>	80
Tabel 6.9 Hasil Pengujian Validasi <i>Register</i> Alternatif 1	80
Tabel 6.10 Hasil Pengujian Validasi <i>Register</i> Alternatif 2	81
Tabel 6.11 Hasil Pengujian Validasi <i>Register</i> Alternatif 3	81
Tabel 6.12 Hasil Pengujian Validasi <i>Login</i>	81
Tabel 6.13 Hasil Pengujian Validasi <i>Login</i> Alternatif 1.....	82
Tabel 6.14 Hasil Pengujian Validasi <i>Login</i> Alternatif 2.....	82
Tabel 6.15 Hasil Pengujian Validasi <i>Login</i> Alternatif 3.....	83
Tabel 6.16 Hasil Pengujian Validasi <i>Logout</i>	83
Tabel 6.17 Hasil Pengujian Validasi Melihat Dosen	83
Tabel 6.18 Hasil Pengujian Validasi Aktivasi Dosen	83
Tabel 6.19 Hasil Pengujian Validasi Mengubah Data Dosen.....	84
Tabel 6.20 Hasil Pengujian Validasi Mengubah Data Dosen Alternatif 1	84
Tabel 6.21 Hasil Pengujian Validasi Mengubah Data Dosen Alternatif 2	85
Tabel 6.22 Hasil Pengujian Validasi Mengubah Data Dosen Alternatif 3	85
Tabel 6.23 Hasil Pengujian Validasi Menghapus Dosen.....	86
Tabel 6.24 Hasil Pengujian Validasi Menghapus Dosen Alternatif 1	86
Tabel 6.25 Hasil Pengujian Validasi Menambah KJFD.....	86
Tabel 6.26 Hasil Pengujian Validasi Menambah KJFD Alternatif 1	87
Tabel 6.27 Hasil Pengujian Validasi Melihat KJFD.....	87
Tabel 6.28 Hasil Pengujian Validasi Melihat KJFD Alternatif 1	87
Tabel 6.29 Hasil Pengujian Validasi Mengubah KJFD.....	87
Tabel 6.30 Hasil Pengujian Validasi Mengubah KJFD Alternatif 1	88
Tabel 6.31 Hasil Pengujian Validasi Menghapus KJFD	88
Tabel 6.32 Hasil Pengujian Validasi Menghapus KJFD Alternatif 1.....	89
Tabel 6.33 Hasil Pengujian Validasi Menambah Mata Kuliah.....	89



Tabel 6.34 Hasil Pengujian Validasi Menambah Mata Kuliah Alternatif 1	89
Tabel 6.35 Hasil Pengujian Validasi Melihat Mata Kuliah	90
Tabel 6.36 Hasil Pengujian Validasi Melihat Mata Kuliah Alternatif 1.....	90
Tabel 6.37 Hasil Pengujian Validasi Mengubah Mata Kuliah.....	90
Tabel 6.38 Hasil Pengujian Validasi Mengubah Mata Kuliah Alternatif 1.....	91
Tabel 6.39 Hasil Pengujian Validasi Menghapus Mata Kuliah	91
Tabel 6.40 Hasil Pengujian Validasi Menghapus Mata Kuliah Alternatif 1	91
Tabel 6.41 Hasil Pengujian Validasi Melihat <i>Team Teaching</i>	92
Tabel 6.42 Hasil Pengujian Validasi Melihat <i>Team Teaching</i> Alternatif 1	92
Tabel 6.43 Hasil Pengujian Validasi Mengubah <i>Team Teaching</i>	92
Tabel 6.44 Hasil Pengujian Validasi Menambah Materi	93
Tabel 6.45 Hasil Pengujian Validasi Melihat Materi	93
Tabel 6.46 Hasil Pengujian Validasi Melihat Materi Alternatif 1	93
Tabel 6.47 Hasil Pengujian Validasi Mengubah Materi	94
Tabel 6.48 Hasil Pengujian Validasi Menghapus Materi.....	94
Tabel 6.49 Hasil Pengujian Validasi Menghapus Materi Alternatif 1.....	94
Tabel 6.50 Hasil Pengujian Validasi Menambah Penugasan.....	95
Tabel 6.51 Hasil Pengujian Validasi Melihat Penugasan.....	95
Tabel 6.52 Hasil Pengujian Validasi Melihat Penugasan Alternatif 1	95
Tabel 6.53 Hasil Pengujian Validasi Melihat Progres Penugasan	96
Tabel 6.54 Hasil Pengujian Validasi Mengubah Penugasan.....	96
Tabel 6.55 Hasil Pengujian Validasi Menghapus Penugasan	96
Tabel 6.56 Hasil Pengujian Validasi Menghapus Penugasan Alternatif 1	97
Tabel 6.57 Hasil Pengujian Validasi Mengatur Anggota Materi Penugasan.....	97
Tabel 6.58 Hasil Pengujian Validasi Mengatur Anggota Materi Penugasan Alternatif 1.....	97
Tabel 6.59 Hasil Pengujian Validasi Gabung Materi Penugasan.....	98
Tabel 6.60 Hasil Pengujian Validasi Gabung Materi Penugasan Alternatif 1	98
Tabel 6.61 Hasil Pengujian Validasi Keluar Materi Penugasan	98
Tabel 6.62 Hasil Pengujian Validasi Keluar Materi Penugasan Alternatif 1.....	99
Tabel 6.63 Hasil Pengujian Validasi Membuat Soal	99
Tabel 6.64 Hasil Pengujian Validasi Mengubah Soal.....	100



Tabel 6.65 Hasil Pengujian Validasi Submit Soal.....	100
Tabel 6.66 Hasil Pengujian Validasi Submit Soal Alternatif 1	100
Tabel 6.67 Hasil Pengujian Validasi Mempercepat Evaluasi.....	101
Tabel 6.68 Hasil Pengujian Validasi Mempercepat Evaluasi Alternatif 1	101
Tabel 6.69 Hasil Pengujian Validasi Mengirim Reminder	101
Tabel 6.70 Hasil Pengujian Validasi Mengirim Reminder Alternatif 1	102
Tabel 6.71 Hasil Pengujian Validasi Memberi Komentar.....	102
Tabel 6.72 Hasil Pengujian Validasi Menghapus Komentar.....	102
Tabel 6.73 Hasil Pengujian Validasi Menghapus Komentar Alternatif 1	103
Tabel 6.74 Hasil Pengujian Validasi Mengevaluasi Soal.....	103
Tabel 6.75 Hasil Pengujian Validasi Menutup Evaluasi.....	103
Tabel 6.76 Hasil Pengujian Validasi Menutup Evaluasi Alternatif 1	104
Tabel 6.77 Hasil Pengujian Validasi Review Soal	104
Tabel 6.78 Hasil Pengujian Validasi Cetak Soal Acak	104
Tabel 6.79 Hasil Pengujian Validasi Cetak Soal Pilih	105
Tabel 6.80 Hasil Pengujian Melihat Riwayat Cetak Soal	105
Tabel 6.81 Hasil Pengujian Validasi Melihat Riwayat Cetak Soal Alternatif 1	105
Tabel 6.82 Hasil Pengujian Validasi Evaluasi Bank Soal	106
Tabel 6.83 Pengujian <i>Compatibility</i>	106

DAFTAR GAMBAR

Gambar 2.1 <i>Waterfall Model</i>	6
Gambar 3.1 Diagram Metodologi Penelitian	14
Gambar 4.1 <i>Business Process Model and Notation</i> Pengelolaan Soal Fakultas Ilmu komputer.....	19
Gambar 4.2 <i>Business Process Model and Notation</i> dari <i>Subprocess</i> Mengatur KJFD	20
Gambar 4.3 <i>Business Process Model and Notation</i> dari <i>Subprocess</i> Mengatur <i>Team Teaching</i>	20
Gambar 4.4 <i>Business Process Model and Notation</i> dari <i>Subprocess</i> Mengatur Penugasan	21
Gambar 4.5 <i>Business Process Model and Notation</i> dari <i>Subprocess</i> Mengerjakan Penugasan	21
Gambar 4.6 <i>Business Process Model and Notation</i> dari <i>Subprocess</i> Mengevaluasi Penugasan	22
Gambar 4.7 <i>Mockup Login</i>	31
Gambar 4.8 <i>Mockup</i> membuat penugasan	31
Gambar 4.9 <i>Mockup</i> memilih materi penugasan	32
Gambar 4.10 <i>Mockup</i> membuat soal penugasan	32
Gambar 4.11 <i>Mockup</i> memberi komentar	32
Gambar 4.12 Diagram <i>Use Case</i> Sistem	33
Gambar 5.1 <i>Sequence Diagram</i> Gabung Materi Penugasan.....	53
Gambar 5.2 <i>Sequence Diagram</i> Mengirim <i>Reminder</i>	54
Gambar 5.3 <i>Sequence Diagram</i> Membuat Soal	55
Gambar 5.4 <i>Class diagram</i>	57
Gambar 5.5 <i>Conceptual Data Model</i>	60
Gambar 5.6 Perancangan antarmuka detail penugasan	60
Gambar 5.7 Perancangan antarmuka evaluasi penugasan.....	62
Gambar 5.8 Perancangan antarmuka membuat soal	63
Gambar 5.9 Implementasi basis data	67
Gambar 5.10 Implementasi antarmuka detail penugasan	68
Gambar 5.11 Implementasi antarmuka evaluasi penugasan	68
Gambar 5.12 Implementasi antarmuka membuat soal.....	69

Gambar 6.1 Flow graph method “proses_gabung_penugasan” class “C_penugasan” 71

Gambar 6.2 Flow graph method “kirim_reminder” class “C_penugasan” 73

Gambar 6.3 Flow graph method “proses_buat_soal” class “C_penugasan” 75

Gambar 6.4 Hasil Pengujian *Compatibility*..... 106



DAFTAR LAMPIRAN

LAMPIRAN A HASIL WAWANCARA.....	112
A.1 Hasil Wawancara Narasumber 1	112
A.2 Hasil Wawancara Narasumber 2	114
LAMPIRAN B KUESIONER	116



BAB 1 PENDAHULUAN

1.1 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. Hal ini juga didukung dari hasil penyebaran kuesioner terkait masalah yang dihadapi oleh dosen dalam pengelolaan bank soal pada Lampiran B. 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 organisasi 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 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.

1.2 Rumusan masalah

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

1. Bagaimana hasil analisis dan spesifikasi kebutuhan dalam membangun sistem pengelolaan soal?
2. Bagaimana hasil perancangan dan implementasi sistem pengelolaan soal dapat menjadi solusi dari masalah yang ada?
3. Bagaimana hasil pengujian sistem pengelolaan soal memenuhi kebutuhan?

1.3 Tujuan

Tujuan yang akan dicapai dalam penelitian membangun sistem pengelolaan soal adalah:

1. Menganalisis dan menyusun spesifikasi kebutuhan sistem pengelolaan soal berbasis *web*.
2. Merancang dan mengimplementasi sistem pengelolaan soal untuk menyelesaikan masalah yang ada.
3. Menguji sistem pengelolaan soal berbasis *web* memenuhi kebutuhan

1.4 Manfaat

Manfaat yang didapatkan dari penelitian membangun sistem pengelolaan soal adalah:

1. Membantu memudahkan ketua *team teaching* dalam mengatur penugasan, evaluasi, dan mengkompilasi soal.
2. Mempermudah dosen dalam pembuatan soal dan menghemat waktu dan tenaga.

1.5 Batasan masalah

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

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

1.6 Sistematika pembahasan

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

BAB 1 : PENDAHULUAN

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

BAB 2 : LANDASAN KEPUSTAKAAN

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

BAB 3 : METODE PENELITIAN

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

BAB 4 : ANALISIS KEBUTUHAN

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

BAB 5 : PERANCANGAN DAN IMPLEMENTASI

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

BAB 6 : PENGUJIAN DAN ANALISIS

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

BAB 7 : PENUTUP

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



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Pembahasan pada kajian pustaka adalah tentang penelitian-penelitian yang sebelumnya telah dilakukan yang menjadi referensi untuk penelitian ini, kajian pustaka yang digunakan akan dijelaskan pada Tabel 2.1 yang menjelaskan tentang judul penelitian, nama penulis, dan persamaan serta perbedaan dari penelitian tersebut dengan penelitian ini.

Tabel 2.1 Kajian Pustaka

No	Nama Penulis, Tahun, dan Judul	Persamaan	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
1	Heri Retnawati dan Samsul Hadi, 2014, Sistem Bank Soal Daerah Terkalibrasi Untuk Menyongsong Era Desentralisasi.	Penggunaan bank soal dalam memudahkan proses pembuatan soal dan evaluasi soal.	Cakupan penelitian merupakan guru pada SMP dan SMA sederajat Proses evaluasi soal dilakukan oleh guru yang membuat soal tersebut.	Cakupan penelitian adalah dosen pada Fakultas Ilmu Komputer Proses evaluasi dilakukan oleh tim dengan memberikan masukan pada soal yang telah dibuat.
2	Siti Mutmainah, 2017, Studi Kelayakan Pengembangan Aplikasi Bank Soal Dalam Rumah Belajar.	Penggunaan teknologi <i>web</i> dalam mengembangkan aplikasi bank soal	Keluaran aplikasi berupa evaluasi dari soal yang telah dikerjakan didalam aplikasi.	Keluaran berupa soal-soal yang diambil dari bank soal dalam format .docx

2.2 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak merupakan ilmu yang mempelajari terkait dengan segala aspek produksi dari suatu perangkat lunak, dimulai dari tahap awal yakni spesifikasi sistem hingga cara untuk dapat mempertahankan sistem setelah digunakan (Sommerville, 2011). Pada definisi tersebut terdapat dua kalimat penting yaitu:

1. *Engineering discipline*

Perekayasa membuat alat atau sistem berjalan sesuai fungsinya, dengan menerapkan teori, metode, dan peralatan pendukung yang digunakan secara

selektif untuk mendapatkan solusi dari masalah yang ada, bahkan jika tidak terdapat teori ataupun metode yang mendukung.

2. *All aspects of software production*

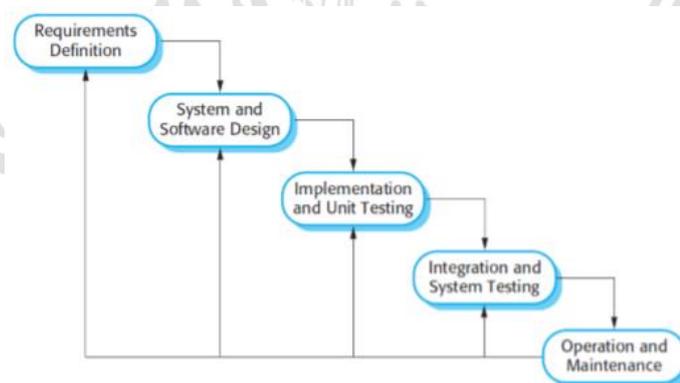
Rekayasa perangkat lunak tidak hanya berfokus pada tahapan teknis pengembangan perangkat lunak. Proses manajemen, pengembangan alat, metode, dan teori juga perlu diperhatikan untuk mendukung produksi perangkat lunak.

2.2.1 Software Development Life Cycle (SDLC)

Pada proses pengembangan suatu perangkat lunak terdapat siklus yang menjelaskan tahapan-tahapan dalam penembangan perangkat lunak, siklus ini biasa disebut *software development life cycle* (SLDC) atau *software process model*. SDLC merupakan metode pengembangan perangkat lunak secara sistematis yang bertujuan untuk meningkatkan kemungkinan menyelesaikan proyek dalam batas waktu tertentu (Deepti Singh, 2015). Terdapat beberapa model SDLC yaitu *waterfall model*, *prototyping model*, *V model*, *inceregmental model*, *RAD model*, *agile model*, *iterative model*, dan *spiral model*.

2.2.1.1 Waterfall Model

Waterfall model atau biasanya disebut dengan siklus hidup pengembangan klasik, yang mana alur yang dilakukan dalam siklus dilakukan secara sistematis dan sesuai urutan. Syarat dari penggunaan *waterfall model* dalam mengembangkan perangkat lunak adalah dengan pendefinisian kebutuhan yang baik dan stabil (Pressman, 2010). Pada penelitian ini digunakan *waterfall model* karena *stakeholder* telah mampu mendefinisikan kebutuhannya dengan baik. Proses yang ada pada *waterfall model* dimulai dari analisis kebutuhan, setelah proses analisis kebutuhan dilakukan maka akan dilakukan perancangan sistem dan implementasi sistem. Setelah diimplementasikan sistem akan diuji, dan proses terakhir adalah proses perawatan sistem. Diagram dari *waterfall model* dapat dilihat pada Gambar 2.1.



Gambar 2.1 Waterfall Model

Sumber : (Sommerville, 2011)

Berikut ini merupakan tahapan dari pengembangan dengan *waterfall model*:

1. Requirement Definition

Pada tahap ini akan dilakukan analisis kebutuhan, dimana kebutuhan akan didefinisikan oleh *stakeholder*. Proses analisis kebutuhan sangat penting untuk dilakukan dengan baik dan benar, karena akan menjadi dasar dari pembangunan sistem. Setelah mendefinisikan kebutuhan, akan dilakukan spesifikasi kebutuhan, untuk memperjelas kebutuhan yang didapatkan.

2. System and Software Design

Pada tahap ini akan dilakukan perancangan dari perangkat lunak yang akan dibangun. Proses perancangan dilakukan berdasarkan kebutuhan yang telah didapatkan pada tahap sebelumnya dan berdasarkan pendekatan yang dilakukan.

3. Implementation and Unit Testing

Pada tahap ini akan dilakukan implementasi dari perancangan perangkat lunak yang telah dibuat. Bersamaan dengan proses implementasi, dilakukan pula pengujian unit, dimana unit terkecil akan diuji apakah dapat berjalan sesuai harapan.

4. Integration and System Testing

Pada tahap ini akan dilakukan integrasi terhadap unit-unit yang telah teruji pada pengujian testing dari proses sebelumnya. Integrasi ini dilakukan untuk meminimalisir kesalahan sebelum pada pengujian sistem. Pengujian sistem dilakukan untuk melihat apakah sistem dapat berjalan sesuai kebutuhan.

5. Operation and Maintenance

Pada tahap ini, perangkat lunak telah digunakan sesuai kebutuhan oleh *stakeholder*. Maintenance dilakukan untuk memperbaiki error yang tidak ditemukan ketika tahap pengujian.

2.2.2 Pendekatan Berorientasi Objek

Pada pendekatan berorientasi objek proses dekomposisi atau penyederhanaan dari masalah yang ada dilakukan berdasarkan objek – objek yang ada dalam sistem. Pendekatan ini memiliki perkembangan yang cukup pesat, hal ini disebabkan karena pendekatan ini memiliki *reusability* yang cukup tinggi diantara pendekatan lainnya. Kelebihan *reusability* yang cukup tinggi membuat sistem yang dikembangkan dengan pendekatan ini dapat digunakan untuk diimplementasikan pada sistem lain yang dikembangkan (A.S. & Shalahuddin, 2015).

2.2.3 Pemodelan Berorientasi Objek

Pemodelan merupakan proses menyederhanakan realita melalui gambaran yang dituangkan dalam pemetaan dengan atura tertentu, sehingga memberikan kemudahan untuk dapat memahaminya (A.S. & Shalahuddin, 2015). Tujuan utama dari pemodelan adalah memudahkan *stakeholder* dalam memahami solusi yang

akan dikerjakan, selain itu dengan adanya pemodelan resiko kegagalan yang mungkin terjadi dapat diminimalisasi. Pada penelitian ini *Unified Modelling Language* (UML) digunakan dalam proses pemodelan. UML adalah sebuah bahasa visual yang digunakan untuk memodelkan perangkat lunak yang akan dikembangkan, UML memiliki kemampuan untuk dapat menspesifikasikan, menggambarkan, membangun, dan mendokumentasi dari sistem perangkat lunak (A.S. & Shalahuddin, 2015). Terdapat tiga diagram yang digunakan dalam proses pemodelan pada pengembangan perangkat lunak penelitian ini, yaitu:

1. *Use Case Diagram*

Use case diagram adalah pemodelan yang berfokus pada kelakuan (*behavior*) dari sistem yang akan dibuat. *Use case diagram* menjelaskan interaksi antara aktor dengan sistem, sehingga dapat diketahui fungsi-fungsi yang ada dalam sebuah sistem dan siapa saja aktor yang berhak menggunakan fungsi-fungsi tersebut.

2. *Sequence Diagram*

Sequence diagram menggambarkan komunikasi antar objek pada *use case*, dengan mendeskripsikan *lifetime* dan pertukaran pesan antar objek. Pembuatan *sequence diagram* membutuhkan objek-objek yang telah didefinisikan pada suatu *use case* beserta metode-metode yang dimiliki.

3. *Class Diagram*

Class diagram menggambarkan struktur sistem dalam pendefinisian kelas-kelas yang akan dibuat dalam membangun sistem. Tujuan pembuatan *class diagram* adalah untuk menjadi pedoman bagi *programmer* dalam melakukan implementasi sehingga antara dokumen dan implementasi dapat sinkron.

2.2.4 Teori Pengujian

Pengujian perangkat lunak merupakan sebuah proses dari eksekusi program yang memiliki tujuan utama adalah menemukan error, kegagalan, yang ditujukan untuk secara intens menghilangkan kecacatan pada suatu perangkat lunak (Runeson, 2011).

2.2.4.1 Unit Testing

Unit testing merupakan pengujian yang berfokus pada usaha dalam melakukan verifikasi pada unit terkecil dalam suatu perangkat lunak yakni modul atau komponen perangkat lunak. Pada *unit testing* terdapat jalur dari komponen untuk diuji yang bertujuan untuk mengungkap kesalahan dalam batas komponen. Proses yang dibutuhkan dalam melakukan *unit testing* adalah membuat *test case* yang berisi tahapan dalam melakukan pengujian (Pressman, 2010).

2.2.4.2 Integration Testing

Integration testing merupakan teknik untuk membangun arsitektur dari perangkat lunak secara sistematis dan disaat bersamaan melakukan pengujian untuk mengungkap kesalahan. Tujuan dari pengujian ini untuk menggunakan komponen yang telah diuji dalam *unit testing* dan membangun struktur program

sesuai desain (Pressman, 2010). Menurut Pressman terdapat 2 pendekatan pada *integration testing* yakni *incremental integration* dan *non-incremental integration*.

Non-incremental integration merupakan teknik dalam membangun perangkat lunak dengan pendekatan “big bang” dimana semua komponen/unit yang telah diuji digabungkan terlebih dahulu baru dilakukan pengujian. Namun dalam penggunaan pendekatan big bang hasil yang diberikan sering kali tidak diharapkan, ketika ditemukan satu kesalahan pada suatu komponen/unit akan sangat sulit melakukan koreksi karena luasnya batasan yang ada. *Incremental integration* merupakan kebalikan dari pendekatan big bang, dimana perangkat lunak dibangun dan diuji sedikit demi sedikit sehingga kesalahan yang ditemukan lebih mudah untuk dikoreksi (Pressman, 2010).

2.2.4.3 Validation Testing

Pengujian validasi dilakukan pada tahap akhir, dimana perangkat lunak telah dijadikan satu dan telah dilakukan pengujian integrasi sehingga kesalahan yang ditemukan telah diminimalisir. Pada pengujian validasi tidak ada lagi perbedaan antara perangkat lunak konvensional, perangkat lunak berorientasi objek, dan webapps. Pengujian ini berfokus pada tindakan dari pengguna terhadap sistem dan keluaran yang dihasilkan oleh sistem, validasi dianggap sukses bila hasil keluaran sistem sesuai dengan harapan pengguna (Pressman, 2010).

2.2.4.4 Compatibility Testing

Compatibility testing merupakan pengujian untuk mengetahui masalah-masalah yang terjadi ketika sistem dijalankan pada beberapa perangkat layar, sistem operasi atau browser yang berbeda (Pressman, 2010). Pada penelitian ini sistem dijalankan pada browser sehingga *compatibility testing* dibutuhkan untuk melihat apakah terdapat masalah ketika sistem berjalan di browser lainnya. Masalah yang dapat muncul pada sistem berbasis *web* adalah masalah tata letak dari tampilan yang seharusnya, perubahan *font*, atau tidak berjalannya suatu fungsi tertentu karena browser yang digunakan tidak mendukung *library* tertentu. Tujuan dari pengujian ini adalah untuk meminimalisir masalah yang terjadi sehingga tidak mengganggu kenyamanan pengguna. Pada penelitian ini, *tool Sortsite* digunakan sebagai sarana untuk melakukan *compatibility testing*.

2.3 Business Process Model and Notation (BPMN)

BPMN adalah suatu notasi grafis yang menggambarkan langkah-langkah yang ada dalam suatu proses bisnis. BPMN menggambarkan proses bisnis mulai dari awal proses hingga akhir proses menggunakan notasi-notasi yang secara khusus dirancang untuk mengkoordinasi urutan proses dan pesan berdasarkan aktor yang terkait dalam proses. Meskipun terlihat mirip namun terdapat perbedaan antara UML dan BPMN. Pada UML digunakan pendekatan berorientasi objek untuk pemodelan perangkat lunak, namun pada BPMN melakukan pendekatan berorientasi proses untuk pemodelan sistem. Selain itu UML dalam penggunaannya lebih fokus pada desain perangkat lunak, sedangkan BPMN terfokus pada proses bisnisnya (BPMN, 2018).

2.4 Soal

Soal adalah suatu hal yang memerlukan jawaban atau masalah yang harus dipecahkan (Kamus Besar Bahasa Indonesia, 2018). Menjawab soal dapat dilakukan menjawab soal dengan berupa tulisan atau memberikan tanda untuk menjawab. Pada tes tertulis soal diklasifikasikan menjadi dua, yakni soal dengan memberikan jawaban dengan tertulis dan soal dengan memberi jawaban dengan memilih opsi jawaban yang telah disediakan. Dalam pembuatan soal terdapat kaidah-kaidah penulisan dari segi konstruksi, materi, dan bahasa (Kementerian Pendidikan dan Kebudayaan, 2017).

2.4.1 Soal Pilihan Ganda

Soal pilihan ganda adalah soal yang jawabannya dipilih dari opsi jawaban yang tersedia, dimana soal pilihan ganda terdiri dari soal (*stem*) dan pilihan jawaban (*option*) yang mana pilihan jawaban terdiri jawaban benar dan jawaban pengecoh (*distractor*) (Kementerian Pendidikan dan Kebudayaan, 2017). Menurut Jay Parkes dan Dawn Zimmaro (2016) tidak ada jumlah optimal dalam menentukan banyaknya pilihan jawaban pada suatu soal pilihan ganda, karena semua itu tergantung dari jawaban pengecoh yang masuk akal (*plausible*) yang mampu didapatkan oleh pembuat soal. Dalam penggunaannya soal pilihan ganda memiliki kelebihan yakni hasil didapatkan dengan mudah dan cepat, ruang lingkup materi lebih luas, dan tepat digunakan untuk ujian berskala besar. Kekurangan dari penggunaan soal pilihan ganda adalah perlunya waktu dalam penyusunan soal, sulitnya membuat jawaban pengecoh, dan adanya peluang menebak jawaban.

2.4.2 Aturan Membuat Soal Pilihan Ganda

Dalam membuat soal pilihan ganda terdapat aturan-aturan yang harus dipatuhi demi mendapatkan soal yang baik dan jelas. Brigham Young University (2001) mengeluarkan 14 aturan dalam penulisan soal pilihan ganda, aturan tersebut berupa:

1. Menggunakan opsi pengecoh yang masuk akal

Buatlah opsi pengecoh yang masuk akal dimana akan membuat murid berpikir berulang kali sebelum memutuskan jawaban. Opsi pengecoh yang baik adalah opsi yang sejalan dengan konten yang sedang dibahas. Pembuatan opsi pengecoh yang baik juga dapat dilakukan dengan menggunakan jawaban pada ujian sebelumnya untuk membuat opsi pengecoh yang lebih baik.

2. Menggunakan kalimat pertanyaan

Para ahli sepakat dalam membuat soal pilihan ganda lebih baik menggunakan kalimat pertanyaan yang baik daripada menggunakan kalimat pernyataan yang tidak lengkap.

3. Memaksa untuk berpikir keras

Menggunakan pertanyaan yang dapat membuat murid untuk mengingat kembali konsep awal, aturan, dan fakta pada konteks kehidupan nyata. Kunci

dalam membuat pertanyaan ini adalah dengan memberikan fakta-fakta yang ada, dimana murid dapat memahami fakta tersebut dan menggunakannya dalam menjawab pertanyaan.

4. Mengusahakan panjang opsi jawaban sama

Dalam membuat soal pilihan ganda yang baik perlu dihindari untuk membuat pilihan jawaban benar terkadang panjang dan terkadang pendek.

5. Menempatkan jawaban benar dengan seimbang

Penempatan jawaban benar dalam urutan opsi jawaban juga menentukan baik tidaknya suatu soal pilihan ganda. Jawaban benar harus ditempatkan secara baik dan seimbang tidak hanya ditempatkan disatu tempat sehingga membuat murid dapat menjawab meskipun sebenarnya tidak memahami pertanyaan.

6. Menggunakan tata bahasa yang baik dan benar

Penggunaan tata bahasa yang tepat dan tidak berbelit-belit, pemilihan kata juga diharapkan tidak ambigu yang bertujuan untuk menghindari persepsi yang berbeda.

7. Hindari memberi petunjuk jawaban pada soal

Hindari memberikan jawaban suatu soal dengan memberi petunjuk pada soal lain dalam tes tersebut.

8. Hindari menggunakan kalimat tanya negatif

Penggunaan kalimat tanya negatif akan membuat murid mengetahui opsi yang salah tanpa mengetahui jawaban yang benar.

9. Menyediakan hanya satu jawaban benar

Sebuah soal pilihan ganda harusnya hanya memiliki 1 jawaban benar dimana opsi jawabannya tidak tumpang tindih kebenarannya.

10. Memberikan instruksi yang jelas

Sebuah soal pilihan ganda harus memberikan instruksi dengan jelas agar fakta-fakta dapat tersampaikan kepada murid dan murid dapat berpikir kritis dalam menjawab soal.

11. Memberi penjelasan yang jelas tentang masalah pada pertanyaan

Suatu soal pilihan ganda yang baik adalah soal yang masalahnya dijelaskan pada pertanyaan, bukan soal dimana murid dapat mengetahui inti masalah dengan membaca opsi jawaban.

12. Hindari opsi jawaban “semua benar”

Penggunaan opsi jawaban dimana semua jawaban benar tidak membantu proses belajar murid, karena murid dapat menjawab pertanyaan hanya dengan mengetahui bahwa 2 opsi jawaban lainnya adalah benar.

13. Hindari opsi jawaban “tidak ada jawaban yang benar”

Penggunaan opsi jawaban dimana tidak ada jawaban benar akan membuat pengajar tidak dapat menilai apakah murid benar memahami materi yang diujikan.

14. Jangan menggunakan soal pilihan ganda ketika tipe soal lainnya lebih sesuai

Soal pilihan ganda tidak sesuai digunakan dalam kondisi dimana opsi pengecoh yang masuk akal terbatas atau jawaban dari soal membutuhkan kreatifitas murid dalam menjawabnya.

2.5 Bank Soal

Menurut Sumardiyono dan Wiworo (2011) bank soal adalah kumpulan butir soal yang telah teruji dengan baik secara teoritis ataupun empiris dan mengandung informasi penting sehingga bisa digunakan dengan mudah digunakan dalam menyusun sebuah instrumen penilaian atau tes. Dalam penggunaannya bank soal bermanfaat bagi pengajar untuk melakukan penilaian, karena dalam mendapatkan penilaian dibutuhkan sebuah instrumen tes. Bank soal dapat mempermudah pengajar dalam menyusun instrumen tes karena pengajar hanya perlu mengambil butir-butir soal yang ada di bank soal. Selain mempermudah pembuatan instrumen tes, bank soal juga membantu pengajar dalam menjamin kualitas soal dari instrumen yang dibuat (Mutmainah S., 2017).

Terdapat beberapa karakteristik bank soal yang perlu dipahami, karakteristik pertama setiap butir soal pada bank soal adalah butir soal yang teruji, teruji yang dimaksud adalah soal tersebut dianggap layak dimana soal baik dalam hal validitas, bahasa, keterukuran, dan efisiensi. Karakteristik kedua, setiap butir soal dilengkapi dengan informasi berguna, informasi yang dimaksud merupakan informasi berupa materi, tingkat kesulitan, dan lainnya dari butir soal terkait. Karakteristik terakhir, soal-soal pada bank soal dibangun secara terstruktur, bank soal bukan hanya kumpulan soal semata, namun kumpulan soal yang diorganisir dan disusun secara sistematis sehingga memudahkan pengajar dalam menyusun instrumen tes dengan tujuan mengukur kompetensi tertentu (Sumardiyono dan Wiworo, 2011).

2.6 Codeigniter

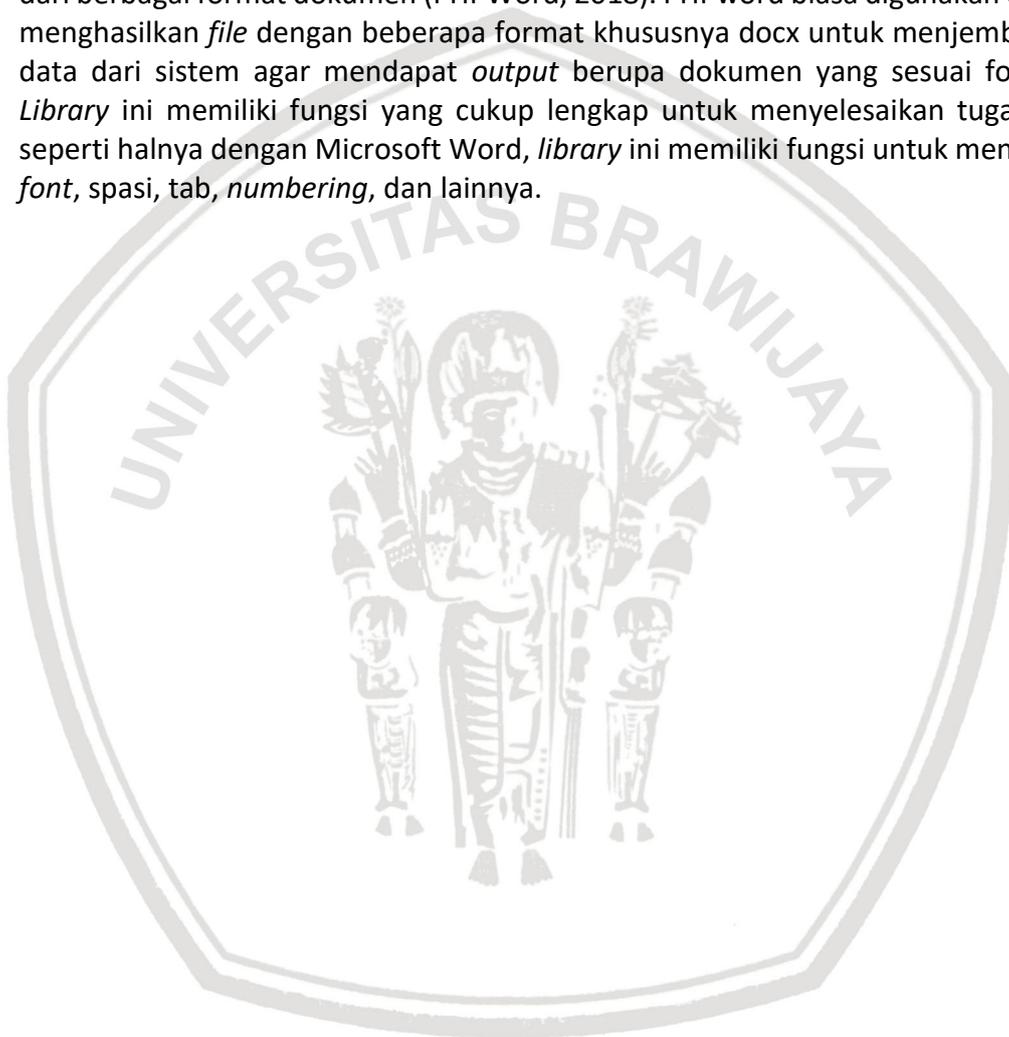
Codeigniter merupakan suatu *framework* pengembangan aplikasi untuk orang-orang yang membangun *website* menggunakan PHP. Tujuannya adalah untuk mempermudah dan mempercepat dalam pengembangan *website* dibandingkan dengan menulis kode program dari awal. Codeigniter menyediakan banyak *library* untuk kebutuhan membangun *website*. Berikut merupakan beberapa fitur yang disediakan oleh Codeigniter: (Codeigniter, 2018)

1. *Model-View-Controller based system*
2. *Form and data validation*
3. *Email sending class*
4. *Error logging*

Codeigniter memiliki beberapa kelebihan dibandingkan dengan *framework* PHP lainnya, dimana Codeigniter ringan dan cepat dibandingkan *framework* lainnya. Penggunaan Codeigniter untuk orang yang awam dipermudah dengan adanya dokumentasi yang lengkap dari Codeigniter yang memberikan “*user-guide*” dalam file installernya.

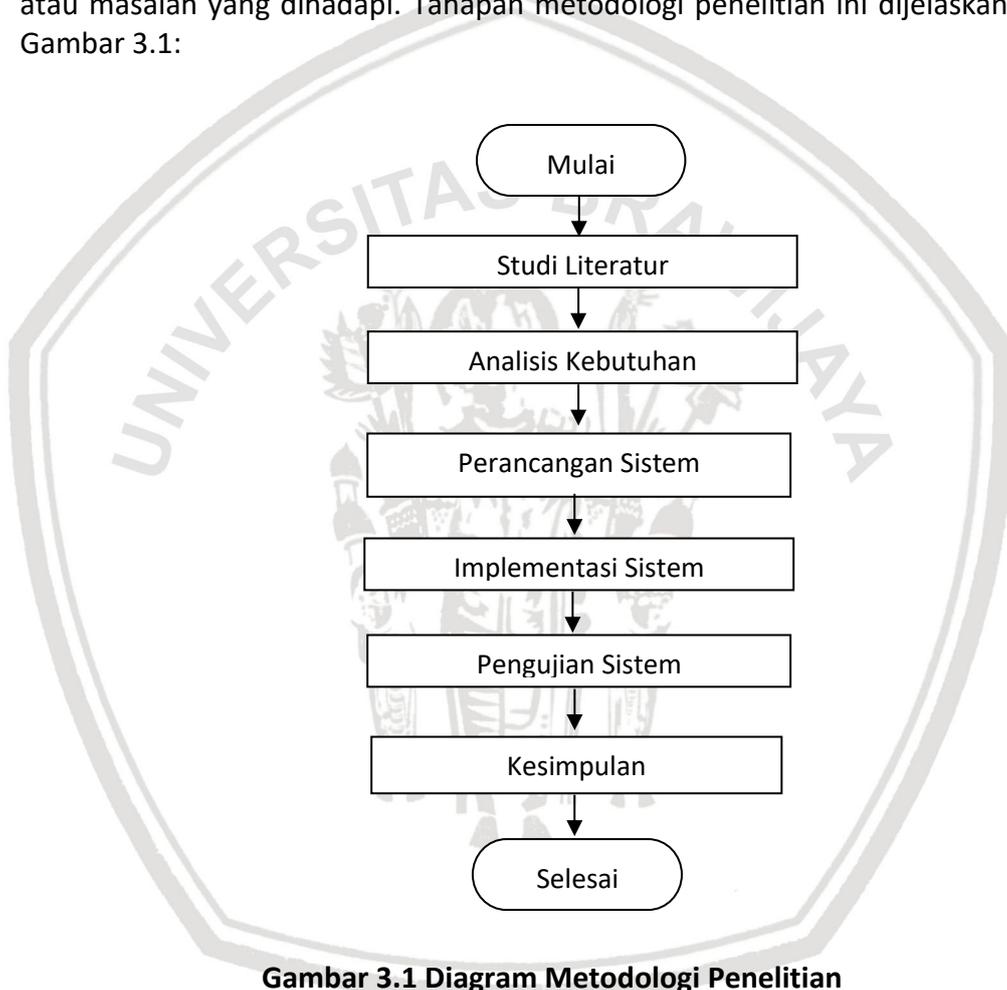
2.7 PHPWord

PHPword merupakan sebuah *library* yang dibuat dengan hanya menggunakan bahasa PHP yang menyediakan kelas-kelas untuk menulis dan membaca dokumen dari berbagai format dokumen (PHPWord, 2018). PHPword biasa digunakan untuk menghasilkan *file* dengan beberapa format khususnya docx untuk menjembatani data dari sistem agar mendapat *output* berupa dokumen yang sesuai format. *Library* ini memiliki fungsi yang cukup lengkap untuk menyelesaikan tugasnya, seperti halnya dengan Microsoft Word, *library* ini memiliki fungsi untuk mengatur *font*, spasi, tab, *numbering*, dan lainnya.



BAB 3 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 Gambar 3.1:



Gambar 3.1 Diagram Metodologi Penelitian

Pada Gambar 3.1 menjelaskan tahapan yang dilakukan oleh dalam pembangunan sistem pengelolaan soal mulai dari pencarian studi literatur, melakukan analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan membuat kesimpulan.

3.1 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 penulisan tugas akhir. Berikut teori-teori pendukung tersebut meliputi:

1. Kajian pustaka dari penelitian sebelumnya
2. *Software Development Life Cycle*
3. *Waterfall model*
4. *Unit Testing*
5. *Integration Testing*
6. *Validation Testing*
7. Teori soal pilihan ganda
8. Aturan pembuatan soal pilihan ganda
9. Bank soal
10. Codeigniter
11. PHPWord

3.2 Analisis Kebutuhan

Analisis kebutuhan dibutuhkan untuk menggali kebutuhan yang akan digunakan dalam proses perancangan. Metode dalam analisis kebutuhan pada penelitian ini menggunakan *Object Oriented Analysis* (OOA) dengan memodelkan kebutuhan menggunakan bahasa *Unified Modelling Language* (UML). Terdapat proses yang perlu dilakukan dalam melakukan analisis kebutuhan yaitu:

1. Elisitasi kebutuhan

Penggalan kebutuhan pada penelitian ini dilakukan dengan wawancara kepada pemangku kepentingan. Wawancara dilakukan dengan tujuan agar pengembang dapat menggali kebutuhan lebih detail dan mampu memahami masalah dengan lebih baik. Wawancara dilakukan kepada 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. Wawancara dilakukan dengan memberikan satu set pertanyaan tentang permasalahan yang ada dan solusi yang diharapkan, jawaban dari pertanyaan wawancara akan digunakan untuk menentukan solusi dan kebutuhan.

2. Spesifikasi kebutuhan

Kebutuhan umum yang telah didapatkan dari fase elisitasi kebutuhan dijelaskan secara mendetail. Identifikasi aktor juga dilakukan untuk mengetahui siapa yang akan menggunakan sistem nantinya. Kebutuhan yang telah dispesifikasi akan dimodelkan menggunakan *use case diagram* dan *use case scenario*.

3. Validasi dan verifikasi kebutuhan

Tahap validasi dan verifikasi dilakukan agar kebutuhan yang telah didapatkan dan didefinisikan sudah benar, akurat dan lengkap. Proses validasi dan verifikasi dilakukan dengan membuat *prototype* agar membantu pengembang dalam memahami kebutuhan dan memastikan bahwa kebutuhan yang telah didefinisikan dan dispesifikkan adalah benar.

4. Manajemen kebutuhan

Dalam menjaga konsistensi dari kebutuhan yang didapatkan maka setiap kebutuhan memerlukan kode unik yang membedakan antar satu kebutuhan dengan kebutuhan lainnya. Kode tersebut akan dibagi berdasarkan kebutuhan fungsional dan nonfungsional, kode unik yang digunakan pada sistem pengelolaan soal adalah SRS-TT-F-1 untuk kebutuhan fungsional dan SRS-TT-NF-1 untuk kebutuhan nonfungsional.

3.3 Perancangan Sistem

Perancangan sistem dikerjakan berdasarkan kebutuhan yang telah didefinisikan *stackholder* pada 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. Pada tahap perancangan sistem, spesifikasi kebutuhan akan dimodelkan dengan menggunakan *Unified Modeling Language* (UML). Adapun hasil dari tahap ini berupa:

1. Perancangan arsitektur

Pada perancangan arsitektur dilakukan pemodelan berupa *sequence diagram* dan *class diagram*. *Sequence diagram* merupakan pemodelan yang menggambarkan interaksi antar objek. Pada penelitian ini dibuat tiga *sequence diagram* yang merupakan fungsionalitas utama. *Class diagram* merupakan pemodelan yang menggambarkan relasi antar *class* pada sistem, dimana setiap *class* memiliki atribut dan oprasinya.

2. Perancangan komponen

Pada perancangan komponen dituliskan algoritme dari tiga fungsionalitas utama dari sistem yang dibangun. Penulisan algoritme dari tiga fungsionalitas tersebut dituliskan dalam *pseudocode*.

3. Perancangan data

Pada perancangan data digunakan *Conceptual Data Model* (CDM) untuk memodelkan relasi data pada database yang akan dibuat. CDM ini nantinya akan diimplementasikan menjadi *Physical Data Model* (PDM) pada tahap implementasi.

4. Perancangan antarmuka

Pada perancangan antarmuka, rancangan antarmuka sistem akan dibuat berupa *wireframe* dengan komponen-komponen pada antarmuka sistem. Penjelasan pada setiap komponen pada *wireframe* diberikan untuk menghindari kesalahan pada tahap implementasi.

3.4 Implementasi Sistem

Pada tahap implementasi sistem, rancangan dan algoritme yang telah dibuat pada tahap perancangan sistem akan diimplementasikan dalam bahasa program. Bahasa pemrograman yang digunakan dalam pengembangan sistem pada penelitian ini adalah PHP dengan menggunakan kerangka kerja *Codeigniter*. Adapun proses pada tahap implementasi sistem adalah sebagai berikut:

1. Implementasi kode program

Implementasi kode program dilakukan berdasarkan perancangan arsitektur dan komponen yang telah dilakukan, yang merujuk pada *sequence diagram*, *class diagram*, dan *pseudocode*.

2. Implementasi basis data

Implementasi basis data dilakukan berdasarkan perancangan data yang telah dikerjakan, yang merujuk pada *conceptual data model* (CDM) diimplementasikan menggunakan MySQL.

3. Implementasi antarmuka

Implementasi antarmuka dilakukan berdasarkan perancangan antarmuka yang telah dibuat, yang merujuk pada *wireframe* dan penjelasan komponen antarmuka. Pada implementasinya digunakan bahasa pemrograman HTML, CSS, dan PHP dimana digunakan *Codeigniter* sebagai *framework* PHP dan *Bootstrap* sebagai *framework* CSS.

3.5 Pengujian Sistem

Pengujian dilakukan untuk melihat apakah sistem yang dibuat telah sesuai dengan kebutuhan dan berjalan sesuai harapan. Pengujian dilakukan untuk meminimalisir atau menghilangkan masalah yang masih ada pada sistem yang telah dibangun. Menurut Pressman (2010) ada beberapa tahapan dalam strategi pengujian, yaitu:

1. Pengujian unit

Pengujian unit dilakukan dengan tujuan untuk menguji suatu komponen dari suatu sistem yang telah dibuat. Komponen yang diuji dapat berupa *class* atau objek berdasarkan perancangan komponen. Metode pengujian *basis path testing* digunakan untuk melakukan teknik pengujian *whitebox*.

2. Pengujian integrasi

Pada pengujian integrasi ini digunakan strategi *top-down integration* dimana pengujian dilakukan dari modul tertinggi dan turun ke modul-modul dibawahnya. Pengujian integrasi dilakukan menggunakan teknik *blackbox testing*.

3. Pengujian validasi

Pada pengujian validasi digunakan teknik *blackbox testing* dalam pengujiannya. Pengujian ini bertujuan untuk menguji kesesuaian kebutuhan yang

telah didefinisikan pada sistem yang telah dibangun. Pengujian blackbox testing dilakukan dengan hanya melihat kondisi masukan dan keluaran yang dihasilkan sistem.

4. Pengujian kesesuaian

Pengujian kesesuaian atau *compability testing* adalah pengujian yang menguji kesesuaian sistem untuk dapat berjalan pada beberapa platform, yang mana dalam penelitian ini sistem berjalan pada perambah web sehingga perlu diuji apakah sistem dapat berjalan pada beberapa perambah web yang ada. Pengujian dilakukan dengan bantuan *website powermapping*, untuk mengeahun apakah terdapat bagian yang tidak sesuai antar satu perambah web dengan yang lainnya.

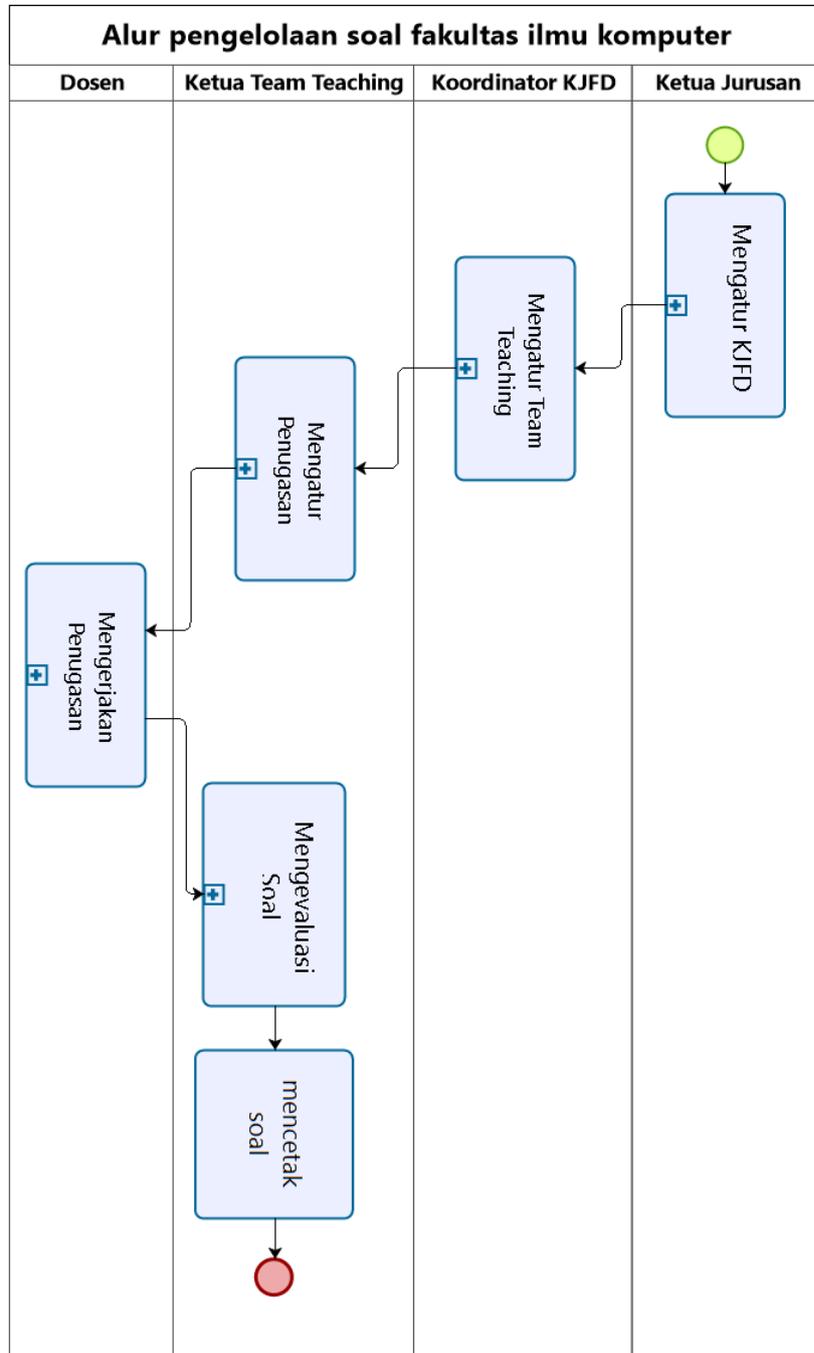
3.6 Kesimpulan dan Saran

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.

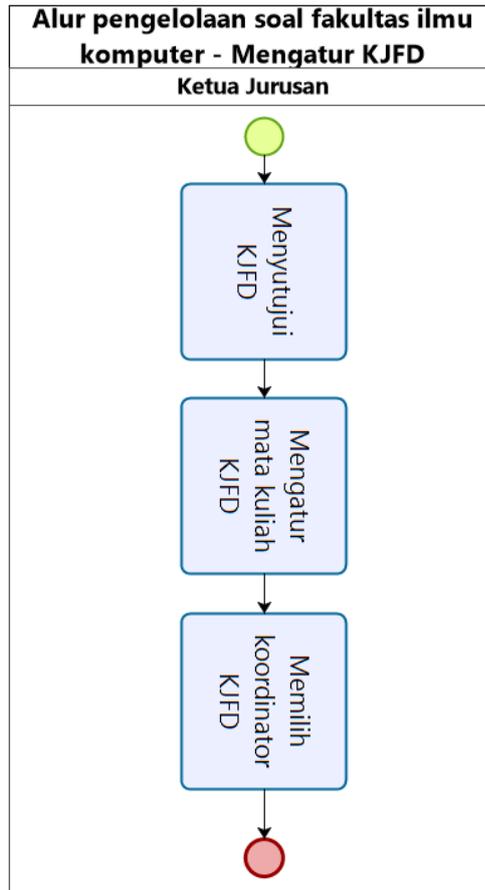


BAB 4 ANALISIS KEBUTUHAN

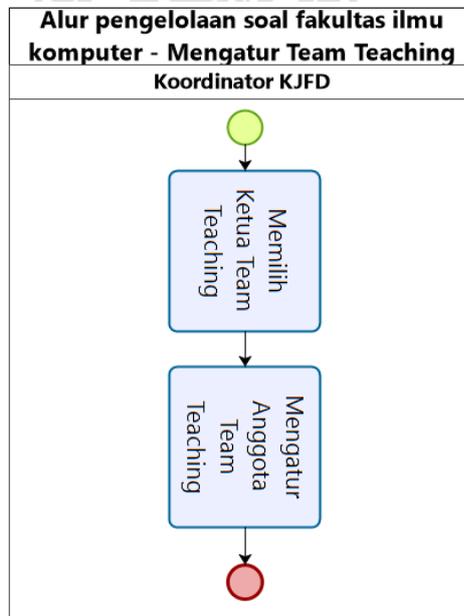
4.1 Elisitasi Kebutuhan



Gambar 4.1 *Business Process Model and Notation* Pengelolaan Soal Fakultas Ilmu komputer

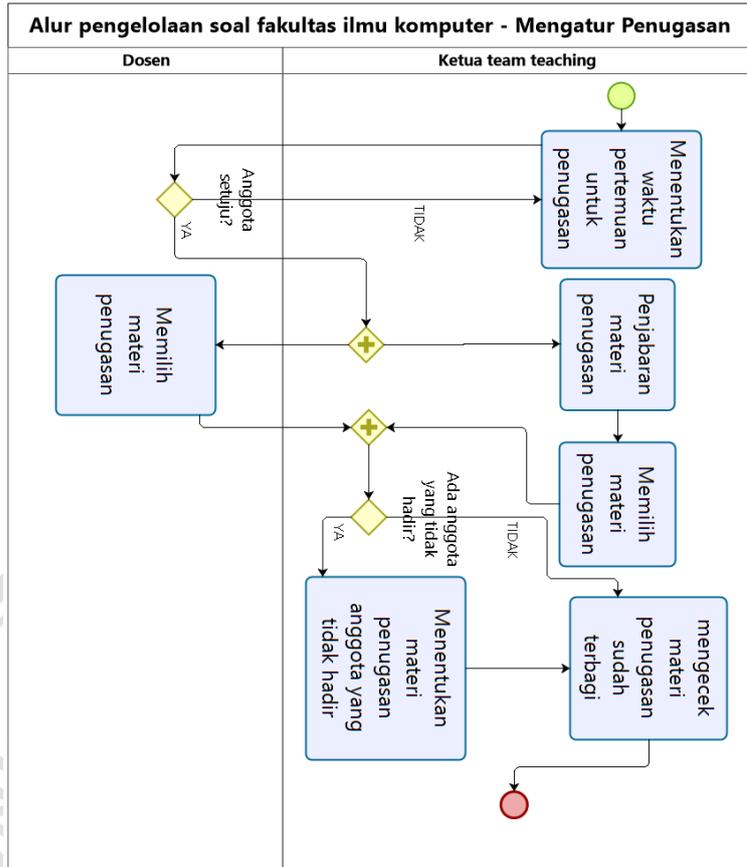


Gambar 4.2 Business Process Model and Notation dari Subprocess Mengatur KJFD

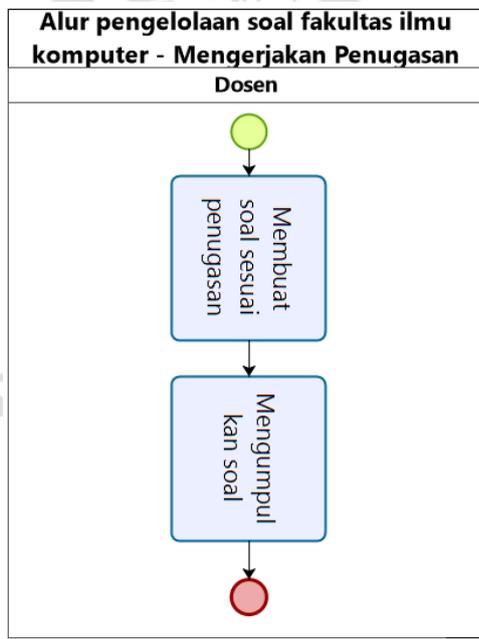


Gambar 4.3 Business Process Model and Notation dari Subprocess Mengatur Team Teaching



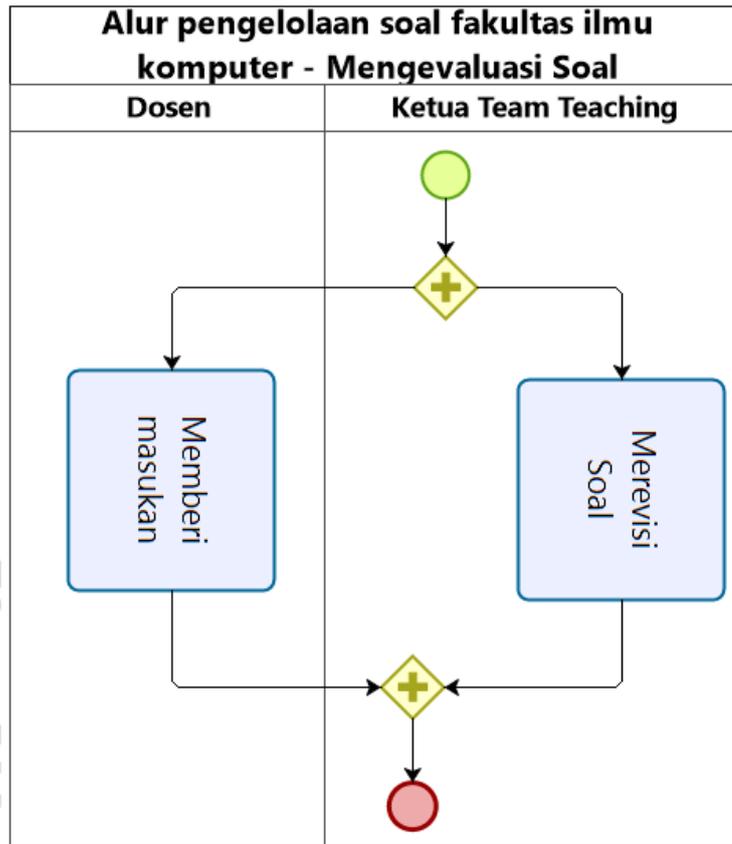


Gambar 4.4 Business Process Model and Notation dari Subprocess Mengatur Penugasan



Gambar 4.5 Business Process Model and Notation dari Subprocess Mengerjakan Penugasan





Gambar 4.6 Business Process Model and Notation dari Subprocess Mengevaluasi Penugasan

Analisis kebutuhan bertujuan untuk mendapatkan kebutuhan umum sistem untuk membangun sistem pengelolaan soal. Penggalan kebutuhan dilakukan dengan wawancara terhadap ketua *team teaching* mata kuliah Rekayasa Perangkat Lunak dan ketua *team teaching* mata kuliah Analisis Perancangan Sistem. Hasil dari wawancara terdapat beberapa masalah yang dirasakan terhadap sistem yang sudah ada, dapat dilihat pada Gambar 4.1 – Gambar 4.6 yang merupakan alur pengelolaan soal yang diterapkan pada Fakultas Ilmu Komputer universitas Brawijaya. Terdapat beberapa masalah yang dirasakan, berupa kesulitan dalam mengumpulkan dosen dalam satu waktu dan tempat, keterlambatan anggota penugasan dalam mengumpulkan soal, dan sering adanya perbedaan format dalam pengumpulan soal. Dari masalah yang telah dijabarkan maka didapatkan solusi 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 diidentifikasi 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 evaluasi soal.

4.2 Identifikasi Aktor

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

Tabel 4.1 Identifikasi Aktor

Aktor	Deskripsi
Pengguna	Merupakan aktor umum yang belum dikenali oleh sistem.
Ketua Jurusan	Merupakan pengguna yang telah dikenali sistem sebagai ketua jurusan yang melakukan verifikasi akun dosen yang mendaftarkan diri, mengatur kelompok jabatan fungsional dosen dan mengatur mata kuliah.
Koordinator KJFD (Kelompok Jabatan Fungsional Dosen)	Merupakan pengguna yang telah dikenali sistem sebagai koordinator yang berfungsi untuk menentukan ketua dan anggota <i>team teaching</i> .
Ketua <i>team teaching</i>	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.
Dosen	Merupakan pengguna yang telah dikenali sistem sebagai dosen, sebelum dosen dapat menjalankan fungsinya dengan baik kajar perlu untuk melakukan proses aktivasi sebagai bentuk verifikasi. Dosen bekerja sama dengan ketua <i>team teaching</i> untuk membuat soal dan mengevaluasi soal.

4.3 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 dan spesifikasinya ditunjukkan pada Tabel 4.2. 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

No.	Kode	Deskripsi Kebutuhan	Nama Use Case
1	SRS-TT-F-1	Sistem harus mampu menyimpan data pengguna yang mendaftarkan diri.	<i>Register</i>
1.1	SRS-TT-F-1-1	Sistem menyediakan form <i>register</i> berupa nama, NIP, <i>e-mail</i> , <i>username</i> dan <i>password</i> .	
2	SRS-TT-F-2	Sistem harus mampu mengenali pengguna yang masuk kedalam sistem dan memberikan hak akses sesuai otoritasnya.	<i>Login</i>
2.1	SRS-TT-F-2-1	Sistem menyediakan <i>form login</i> berupa <i>username</i> dan <i>password</i> .	
3	SRS-TT-F-3	Sistem harus mampu menghapus <i>session</i> dari pengguna yang berhasil <i>login</i> .	<i>Logout</i>
4	SRS-TT-F-4	Sistem harus mampu menampilkan data dosen yang telah mendaftar.	Melihat dosen
4.1	SRS-TT-F-4-1	Sistem menampilkan data berupa NIP, nama, <i>e-mail</i> , <i>username</i> , keminatan <i>mayor</i> , keminatan <i>minor</i> , dan jabatan.	
5	SRS-TT-F-5	Sistem harus mampu mengaktifkan dosen baru yang terdaftar sehingga dapat masuk kedalam sistem.	Aktivasi dosen
5.1	SRS-TT-F-5-1	Sistem menyediakan form <i>register</i> berupa nama, keminatan <i>mayor</i> , dan keminatan <i>minor</i> .	
5.2	SRS-TT-F-5-2	Setiap dosen hanya memiliki satu keminatan <i>mayor</i> , namun dapat memiliki beberapa keminatan <i>mayor</i> .	
6	SRS-TT-F-6	Sistem harus mampu mengubah data dosen aktif.	Mengubah data dosen
6.1	SRS-TT-F-6-1	Sistem menampilkan <i>form</i> ubah data dosen berupa NIP, nama, <i>e-mail</i> , <i>username</i> yang telah terisi sesuai data saat itu.	
7	SRS-TT-F-7	Sistem harus mampu menghapus dosen yang telah terdaftar.	Menghapus dosen
7.1	SRS-TT-F-7-1	Sistem menampilkan kotak konfirmasi penghapusan dosen	
8	SRS-TT-F-8	Sistem harus mampu menambah KJFD	Menambah KJFD

8.1	SRS-TT-F-8-1	Sistem menyediakan form tambah KJFD berupa nama KJFD dan dosen sebagai Koordinator KJFD.	
8.2	SRS-TT-F-8-2	Setiap dosen berhak untuk menjadi koordinator KJFD, namun seorang dosen hanya dapat menjadi salah satu koordinator dari KJFD yang ada.	
9	SRS-TT-F-9	Sistem harus mampu menampilkan data KJFD yang telah ditambahkan.	Melihat KJFD
9.1	SRS-TT-F-9-1	Sistem menampilkan data berupa nama KJFD dan nama Koordinator KJFD.	
10	SRS-TT-F-10	Sistem harus mampu mengubah data KJFD yang telah ditambahkan.	Mengubah KJFD
10.1	SRS-TT-F-10-1	Sistem menampilkan <i>form</i> ubah KJFD berupa nama KJFD dan dosen sebagai Koordinator KJFD.	
10.2	SRS-TT-F-10-2	Setiap dosen berhak untuk menjadi koordinator KJFD, namun seorang dosen hanya dapat menjadi salah satu koordinator dari KJFD yang ada.	
11	SRS-TT-F-11	Sistem harus mampu menghapus KJFD yang telah ditambahkan.	Menghapus KJFD
11.1	SRS-TT-F-11-1	Sistem menampilkan kotak konfirmasi penghapusan KJFD.	
12	SRS-TT-F-12	Sistem harus mampu menambah mata kuliah.	Menambah mata kuliah
12.1	SRS-TT-F-12-1	Sistem menyediakan form tambah mata kuliah berupa kode mata kuliah, nama mata kuliah, dan KJFD dari mata kuliah tersebut.	
13	SRS-TT-F-13	Sistem harus mampu menampilkan data mata kuliah yang telah ditambahkan.	Melihat mata kuliah
13.1	SRS-TT-F-13-1	Sistem menampilkan data berupa kode mata kuliah, nama mata kuliah, dan KJFD dari mata kuliah.	
14	SRS-TT-F-14	Sistem harus mampu mengubah data mata kuliah yang telah ditambahkan.	Mengubah mata kuliah
14.1	SRS-TT-F-14-1	Sistem menampilkan form ubah mata kuliah berupa kode mata kuliah, nama mata kuliah, dan KJFD dari mata kuliah.	
15	SRS-TT-F-15	Sistem harus mampu menghapus mata kuliah yang telah ditambahkan.	Menghapus mata kuliah

15.1	SRS-TT-F-15-1	Sistem menampilkan kotak konfirmasi penghapusan mata kuliah.	
16	SRS-TT-F-16	Sistem harus mampu menampilkan daftar <i>team teaching</i> dari mata kuliah yang ada sesuai dengan KJFD yang dikoordinatori.	Melihat <i>team teaching</i>
16.1	SRS-TT-F-16-1	Sistem menampilkan data berupa nama mata kuliah, nama ketua <i>team teaching</i> , dan nama anggota <i>team teaching</i> .	
17	SRS-TT-F-17	Sistem harus mampu mengubah data pada <i>team teaching</i> .	Mengubah <i>team teaching</i>
17.1	SRS-TT-F-17-1	Sistem menampilkan <i>form</i> ubah <i>team teaching</i> dengan data berupa nama ketua <i>team teaching</i> dan nama anggota <i>team teaching</i> .	
17.2	SRS-TT-F-17-2	Ketua dan anggota <i>team teaching</i> dipilih berdasarkan keminatan mayor dan minor dosen yang sesuai dengan KJFD dari mata kuliah tersebut.	
17.3	SRS-TT-F-17-3	Setiap <i>team teaching</i> hanya memiliki satu ketua dan dapat memiliki beberapa anggota.	
18	SRS-TT-F-18	Sistem harus mampu menambah materi dari <i>team teaching</i> yang diketuai.	Menambah materi
18.1	SRS-TT-F-18-1	Sistem menyediakan form tambah materi berupa nama materi dan bab dari materi yang akan ditambahkan.	
19	SRS-TT-F-19	Sistem harus mampu menampilkan data materi yang telah ditambahkan.	Melihat materi
19.1	SRS-TT-F-19-1	Sistem menampilkan data berupa nama materi dan bab dari materi.	
20	SRS-TT-F-20	Sistem harus mampu mengubah data materi yang telah ditambahkan.	Mengubah materi
20.1	SRS-TT-F-20-1	Sistem menampilkan form ubah materi berupa nama materi dan bab dari materi.	
21	SRS-TT-F-21	Sistem harus mampu menghapus materi yang telah ditambahkan.	Menghapus materi
21.1	SRS-TT-F-21-1	Sistem menampilkan kotak konfirmasi penghapusan materi.	
22	SRS-TT-F-22	Sistem harus mampu menambah penugasan untuk pembuatan soal.	Menambah penugasan

22.1	SRS-TT-F-22-1	Sistem menyediakan <i>form</i> tambah penugasan berupa judul penugasan, materi, kuota, kebutuhan soal pada tiap tingkat kesulitan, batas waktu pengambilan penugasan, dan batas waktu penugasan.	
23	SRS-TT-F-23	Sistem harus mampu menampilkan penugasan yang telah dibuat.	Melihat penugasan
23.1	SRS-TT-F-23-1	Sistem menampilkan data kepada ketua <i>team teaching</i> berupa nama penugasan, jumlah materi, batas pengambilan, dan batas penugasan.	
23.2	SRS-TT-F-23-2	Sistem menampilkan data kepada anggota <i>team teaching</i> berupa nama <i>team teaching</i> penugasan, nama penugasan, jumlah materi, batas pengambilan, dan batas penugasan.	
24	SRS-TT-F-24	Sistem harus mampu menampilkan progres penugasan yang telah dibuat.	Melihat progres penugasan
24.1	SRS-TT-F-24-1	Sistem menampilkan data kepada berupa nama dosen, jumlah materi yang diambil, soal yang telah dikerjakan, dan status pengerjaan.	
25	SRS-TT-F-25	Sistem harus mampu mengubah data pada penugasan yang telah dibuat.	Mengubah penugasan
25.1	SRS-TT-F-25-1	Sistem menampilkan <i>form</i> ubah penugasan dengan data berupa judul penugasan, materi, kuota, kebutuhan soal pada tiap tingkat kesulitan dan batas waktu penugasan yang telah terisi sesuai data saat itu.	
26	SRS-TT-F-26	Sistem harus mampu menghapus penugasan yang telah dibuat.	Menghapus penugasan
26.1	SRS-TT-F-26-1	Sistem menampilkan kotak konfirmasi penghapusan penugasan.	
27	SRS-TT-F-27	Sistem harus mampu mengatur anggota materi penugasan	Mengatur anggota materi penugasan
27.1	SRS-TT-F-27-1	Sistem memberikan hak kepada ketua <i>team teaching</i> untuk mengatur anggota materi penugasan, dengan memasukkan atau mengeluarkan dosen dari materi penugasan untuk	

		menyeimbangkan pengambilan materi.	
27.2	SRS-TT-F-27-2	Ketua <i>team teaching</i> dapat memasukkan dosen selama kuota belum terpenuhi.	
28	SRS-TT-F-28	Sistem harus menyediakan fungsi gabung materi pilihan dalam penugasan untuk menentukan materi dalam pembuatan soal.	Gabung materi penugasan
28.1	SRS-TT-F-28-1	Sistem memberikan hak bergabung kepada dosen dalam menentukan materi.	
28.2	SRS-TT-F-28-2	Fungsi gabung materi penugasan dapat digunakan jika kuota dari materi yang akan diambil belum penuh.	
29	SRS-TT-F-29	Sistem harus menyediakan fungsi keluar materi pilihan dalam penugasan untuk keluar dari materi yang telah dipilih sebelumnya.	Keluar materi penugasan
29.2	SRS-TT-F-29-2	Sistem memberikan hak kepada dosen untuk keluar dari materi.	
30	SRS-TT-F-30	Sistem harus menyediakan fungsi membuat soal untuk mempermudah anggota penugasan dalam membuat soal pada sistem.	Membuat soal
30.1	SRS-TT-F-30-1	Sistem menampilkan <i>form</i> untuk membuat soal berupa soal, opsi jawaban, dan estimasi waktu pengerjaan yang terbagi berdasar tingkat kesulitan yang telah diatur.	
30.2	SRS-TT-F-30-2	Pembuatan soal dapat dilakukan jika <i>team teaching</i> telah memilih materi yang akan dikerjakan dan batas waktu telah dilewati.	
31	SRS-TT-F-31	Sistem harus mampu mengubah data soal yang telah dibuat.	Mengubah soal
31.1	SRS-TT-F-31-1	Sistem menampilkan <i>form</i> berupa soal, opsi jawaban, dan estimasi waktu pengerjaan yang terbagi berdasar tingkat kesulitan yang telah diatur dan telah terisi sesuai data saat itu.	
31.2	SRS-TT-F-31-2	Sistem memberikan anggota penugasan hak untuk mengubah soal selama masa penugasan.	
31.3	SRS-TT-F-31-3	Sistem memberikan ketua <i>team teaching</i> hak untuk mengubah soal setelah proses evaluasi penugasan.	

32	SRS-TT-F-32	Sistem harus mampu menyediakan fungsi <i>submit</i> sehingga anggota penugasan yang telah selesai dalam pembuatan soal dapat diketahui oleh ketua <i>team teaching</i> .	Submit soal
33	SRS-TT-F-33	Sistem harus mampu menyediakan fungsi mempercepat evaluasi.	Mempercepat evaluasi
33.1	SRS-TT-F-33-1	Sistem memberikan hak mempercepat evaluasi kepada ketua <i>team teaching</i> ketika semua anggota penugasan telah melakukan <i>submit</i> soal sebelum batas waktu penugasan.	
34	SRS-TT-F-34	Sistem harus mampu mengirim <i>reminder</i> kepada dosen yang tergabung dalam penugasan.	Mengirim <i>reminder</i>
34.1	SRS-TT-F-34-1	Sistem mengirimkan <i>reminder</i> melalui <i>e-mail</i> anggota penugasan.	
34.2	SRS-TT-F-34-2	Sistem mengirim <i>reminder</i> kepada anggota penugasan yang belum memenuhi tanggung jawabnya.	
35	SRS-TT-F-35	Sistem harus mampu menyediakan fungsi yang mengizinkan anggota penugasan memberi masukan terhadap soal pada penugasan sebagai proses evaluasi penugasan.	Memberi komentar
36	SRS-TT-F-36	Sistem harus mampu menghapus komentar.	Menghapus komentar
36.1	SRS-TT-F-36-1	Menghapus komentar hanya dapat dilakukan oleh dosen yang memberikan komentar.	
36.2	SRS-TT-F-36-2	Ketua <i>team teaching</i> penugasan dapat menghapus komentar dosen lain.	
37	SRS-TT-F-37	Sistem harus mampu mengubah soal sesuai hasil evaluasi.	Mengevaluasi soal
37.1	SRS-TT-F-37-1	Sistem memberikan hak pada ketua <i>team teaching</i> untuk mengubah soal penugasan sesuai komentar yang telah diberikan.	
38	SRS-TT-F-38	Sistem harus mampu menyediakan fungsi tutup evaluasi.	Tutup evaluasi
38.1	SRS-TT-F-38-1	Sistem menyediakan fungsi tutup evaluasi jika waktu pengerjaan penugasan telah selesai dan masuk waktu evaluasi.	
39	SRS-TT-F-39	Sistem harus mampu memberikan <i>review</i> soal sebelum mencetak soal.	Review soal

39.1	SRS-TT-F-39-1	Pada review soal terdapat <i>form</i> berupa tahun ajaran, mata kuliah, nama dosen, banyak kode soal yang akan dicetak, jenis <i>font</i> yang digunakan dalam mencetak soal, estimasi waktu pengerjaan, sifat ujian, dan ukuran font.	
40	SRS-TT-F-40	Sistem harus mampu menghasilkan soal yang telah dievaluasi, secara acak, dalam format 'docx'.	Cetak soal acak
40.1	SRS-TT-F-40-1	Sebelum sistem mencetak soal Sistem harus menampilkan review soal yang akan dicetak.	
40.2	SRS-TT-F-40-2	Sistem dapat menghasilkan soal secara acak, berdasarkan kebutuhan soal secara persentase pada tiap materi yang diatur ketua <i>team teaching</i> .	
41	SRS-TT-F-41	Sistem harus mampu menghasilkan soal yang telah dievaluasi dan dipilih, dalam format 'docx'.	Cetak soal pilih
41.1	SRS-TT-F-41-1	Sebelum sistem mencetak soal Sistem harus menampilkan review soal yang akan dicetak.	
41.1	SRS-TT-F-41-2	Sistem dapat menghasilkan soal sesuai soal yang dipilih oleh ketua <i>team teaching</i> .	
42	SRS-TT-F-42	Sistem dapat menampilkan riwayat cetak soal yang pernah dilakukan.	Melihat riwayat cetak soal
42.1	SRS-TT-F-42-1	Sistem menampilkan riwayat cetak soal kepada ketua <i>team teaching</i> .	
42.2	SRS-TT-F-42-2	Sistem menampilkan riwayat mata kuliah sesuai dengan <i>team teaching</i> yang dipegang.	
43	SRS-TT-F-43	Sistem harus mampu mengevaluasi soal yang telah siap cetak	Evaluasi bank soal
43.1	SRS-TT-F-43-1	Pada evaluasi dapat dilakukan perubahan pada soal dan dapat merubah status soal untuk dapat digunakan dalam cetak soal atau tidak	

4.4 Kebutuhan Nonfungsional Sistem

Kebutuhan nonfungsional adalah batasan pada sebuah layanan atau fungsionalitas yang disediakan oleh sistem, kebutuhan nonfungsional biasanya diaplikasikan pada sistem secara keseluruhan, bukan dikenakan pada masing-masing fitur atau layanan (Sommerville, 2011). Setiap Kebutuhan akan diberikan kode SRS-TT-NF-X untuk kebutuhan nonfungsional. SRS merupakan

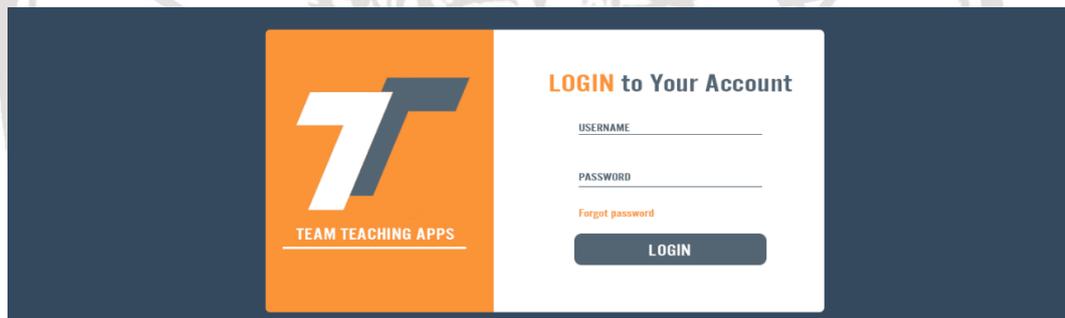
System Requirement Spesification, TT merupakan singkatan dari *Team Teaching*, NF untuk kebutuhan nonfungsional dan X menunjukkan nomor dari definisi kebutuhan utama. Kebutuhan nonfungsional dapat dilihat pada Tabel 4.3.

Tabel 4.3 Kebutuhan Nonfungsional Sistem

No.	Kode Kebutuhan Sistem	Deskripsi Kebutuhan
1	SRS-TT-NF-1	<i>Compatibility</i> , sistem dapat berjalan dengan baik pada beberapa browser seperti Mozilla Firefox, Google Chrome, Internet Explorer, UC Browser, Opera, dan Safari.

4.5 Validasi dan Verifikasi

Pada proses analisis kebutuhan dilakukan proses validasi dan verifikasi, untuk mengetahui apakah sistem yang kita bangun adalah benar. Proses validasi dan verifikasi dilakukan dengan menggunakan teknik *prototyping* dimana dilakukan pembuatan *prototype* dari kebutuhan yang telah didapatkan. Teknik ini dipilih karena dapat mempermudah *stakeholder* untuk mendapatkan gambaran sistem yang akan dibuat, sehingga meminimalisir kesalahan yang terjadi. *Prototype* yang dibuat untuk proses validasi dan verifikasi dapat dilihat pada Gambar 4.7 – Gambar 4.11.



Gambar 4.7 Mockup Login

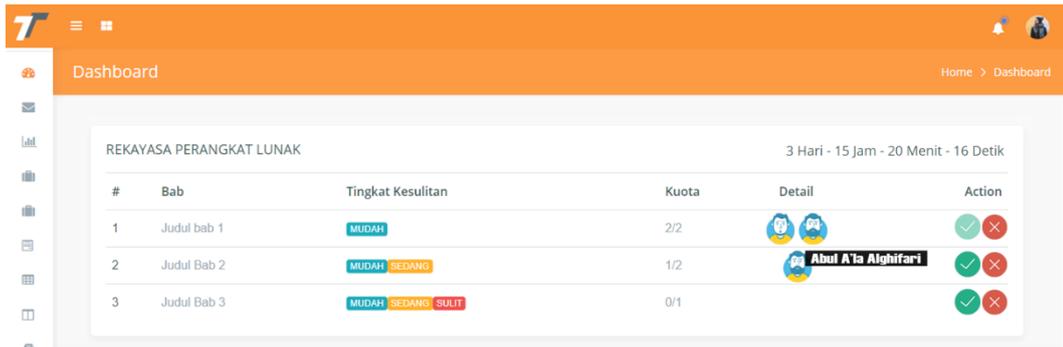
Pada Gambar 4.7 merupakan mockup dari kebutuhan *stakeholder* untuk login.



Gambar 4.8 Mockup membuat penugasan

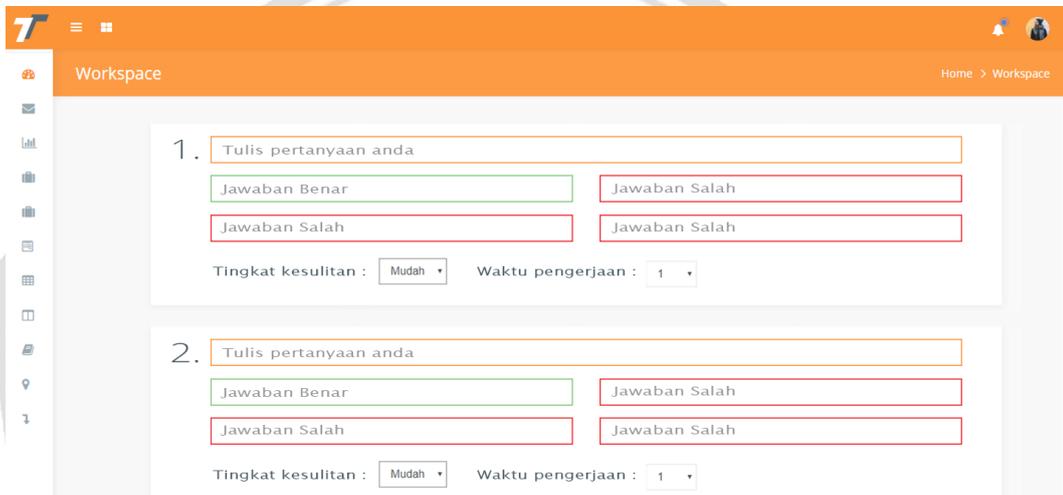
Pada Gambar 4.8 merupakan mockup dari kebutuhan *stakeholder* dalam membuat penugasan.





Gambar 4.9 Mockup memilih materi penugasan

Pada Gambar 4.9 merupakan *mockup* dari kebutuhan *stakeholder* dalam memilih materi penugasan.



Gambar 4.10 Mockup membuat soal penugasan

Pada Gambar 4.10 merupakan *mockup* dari kebutuhan *stakeholder* dalam membuat soal penugasan.



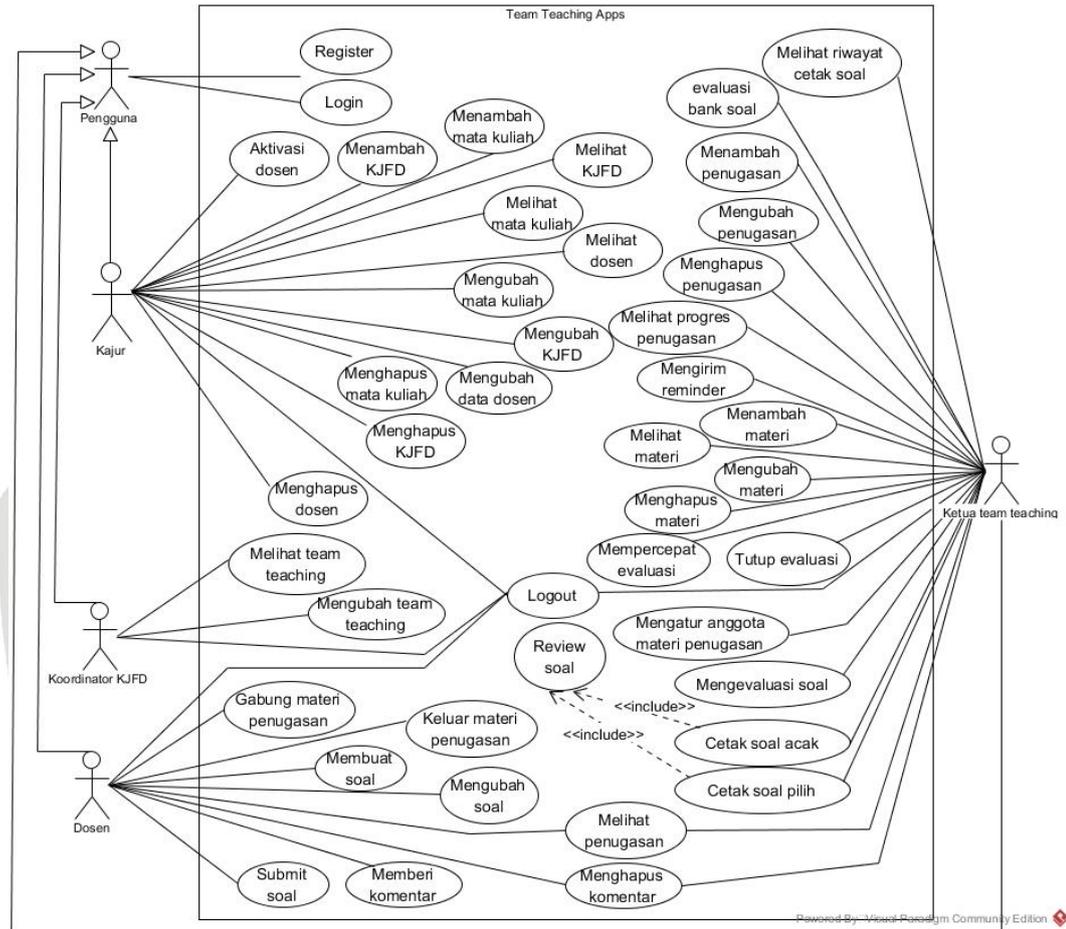
Gambar 4.11 Mockup memberi komentar

Pada Gambar 4.11 merupakan *mockup* dari kebutuhan *stakeholder* dalam memberi komentar.



4.6 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* dibuat berdasarkan kebutuhan pada tahap analisis kebutuhan, diagram ini dapat dilihat pada Gambar 4.12.



Gambar 4.12 Diagram Use Case Sistem

4.7 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*). *Use case scenario* akan ditunjukkan pada Tabel 4.6 hingga Tabel 4.48.



Tabel 4.4 Skenario Use Case Register

Nama Use Case	Register
Kode Kebutuhan Terkait	SRS-TT-F-1
Aktor	Pengguna
Tujuan	Use case ini berfungsi untuk dapat mendaftarkan diri untuk dapat menggunakan fungsi utama sistem
Pre Condition	Aktor telah membuka halaman "login"
Main flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol <i>register</i> 2. Sistem menampilkan <i>form</i> pendaftaran berisi nama, <i>e-mail</i>, NIP, <i>username</i> dan <i>password</i> 3. Pengguna mengisi <i>form</i> dengan lengkap dan menekan tombol <i>register</i> 4. Sistem menyimpan data yang telah diisi pengguna, menampilkan halaman "login" dan pesan sukses mendaftar
Post Condition	Pendaftaran diri berhasil dilakukan oleh aktor
Alternative Flow	<ol style="list-style-type: none"> 1. Jika terdapat bagian form yang belum diisi akan menampilkan peringatan untuk mengisi. 2. Jika <i>username</i> telah digunakan, maka sistem akan menampilkan pesan "username telah digunakan" 3. Jika <i>e-mail</i> telah digunakan, maka sistem akan menampilkan pesan "<i>e-mail</i> telah digunakan"

Tabel 4.5 Skenario Use Case Login

Nama Use Case	Login
Kode Kebutuhan Terkait	SRS-TT-F-2
Aktor	Pengguna
Tujuan	Use case ini berfungsi untuk mengenali pengguna dan memberikan hak akses sesuai otoritas pengguna
Pre Condition	Aktor telah membuka halaman "login"
Main flow	<ol style="list-style-type: none"> 1. Pengguna memasukkan <i>username</i> dan <i>password</i> dan menekan tombol <i>login</i> 2. Sistem melakukan verifikasi terhadap <i>username</i> dan <i>password</i> yang ada di <i>database</i>, membuat <i>session</i>, dan menampilkan halaman "<i>dashboard</i>"
Post Condition	Sistem berhasil mengenali aktor
Alternative Flow	<ol style="list-style-type: none"> 1. Jika terdapat bagian form yang belum diisi akan menampilkan peringatan untuk mengisi. 2. Jika akun belum diaktifkan maka akan muncul pesan "Akun anda tidak aktif, silahkan hubungi ketua jurusan" 3. Jika <i>username</i> atau <i>password</i> tidak sesuai maka akan muncul pesan "Maaf, <i>Username</i> atau <i>Password</i> anda salah, Silahkan diisi kembali dengan tepat."

Tabel 4.6 Skenario Use Case Logout

Nama Use Case	Logout
Kode Kebutuhan Terkait	SRS-TT-F-3
Aktor	Ketua jurusan, Koordinator KJFD, ketua <i>team teaching</i> , dan dosen
Tujuan	Use case ini berfungsi untuk keluar dari sistem
Pre Condition	Aktor telah melakukan <i>login</i>
Main flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol logout 2. Sistem akan menghapus <i>session</i> dan kembali ke halaman "<i>login</i>"
Post Condition	Aktor berhasil keluar dari sistem
Alternative Flow	-

Tabel 4.7 Skenario Use Case Melihat Dosen

Nama Use Case	Melihat dosen
Kode Kebutuhan Terkait	SRS-TT-F-4
Aktor	Ketua jurusan
Tujuan	Use case ini berfungsi untuk melihat daftar dosen yang telah mendaftar
Pre Condition	Aktor telah <i>login</i> dan diidentifikasi sebagai ketua jurusan
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih menu "pengaturan dosen" 2. Sistem akan menampilkan daftar dosen yang telah mendaftar.
Post Condition	Daftar dosen dilihat oleh aktor
Alternative Flow	-

Tabel 4.8 Skenario Use Case Aktivasi Dosen

Nama Use Case	Aktivasi dosen
Kode Kebutuhan Terkait	SRS-TT-F-5
Aktor	Ketua jurusan
Tujuan	Use case ini berfungsi untuk melakukan aktivasi pada akun dosen yang telah mendaftar
Pre Condition	Aktor telah membuka halaman "pengaturan dosen"
Main flow	<ol style="list-style-type: none"> 1. Aktor melihat daftar dosen dan menekan tombol "Aktifkan" pada akun dosen yang ingin di aktivasi 2. Sistem akan menampilkan <i>form</i> aktivasi dosen berupa nama dosen, keminatan mayor, dan keminatan minor 3. Aktor memilih keminatan mayor dan minor dosen yang akan diaktivasi 4. Sistem melakukan <i>update</i> data sesuai akun yang dipilih dan menampilkan halaman "pengaturan dosen" dengan pesan sukses sukses melakukan aktivasi

<i>Post Condition</i>	Akun dosen berhasil diaktifkan oleh aktor
<i>Alternative Flow</i>	-

Tabel 4.9 Skenario Use Case Mengubah Data Dosen

Nama Use Case	Mengubah data dosen
Kode Kebutuhan Terkait	SRS-TT-F-6
Aktor	Ketua jurusan
Tujuan	Use case ini berfungsi untuk mengubah data dosen
<i>Pre Condition</i>	Aktor telah membuka halaman "Edit Dosen"
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor merubah data dosen berupa nama, NIP, <i>e-mail</i>, <i>username</i>, keminatan mayor, dan keminatan minor pada form "edit dosen" dan menekan tombol simpan 2. Sistem akan melakukan <i>update</i> data sesuai akun dosen yang ingin dirubah dan menampilkan halaman "pengaturan dosen" dengan pesan "berhasil mengubah data dosen"
<i>Post Condition</i>	Data dosen dirubah oleh aktor
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika terdapat bagian form yang belum diisi akan menampilkan peringatan untuk mengisi. 2. Jika <i>username</i> telah digunakan, maka sistem akan menampilkan pesan "username telah digunakan" 3. Jika <i>e-mail</i> telah digunakan, maka sistem akan menampilkan pesan "<i>e-mail</i> telah digunakan"

Tabel 4.10 Skenario Use Case Menghapus Dosen

Nama Use Case	Menghapus dosen
Kode Kebutuhan Terkait	SRS-TT-F-7
Aktor	Ketua jurusan
Tujuan	Use case ini berfungsi untuk menghapus akun dosen yang telah dibuat
<i>Pre Condition</i>	Aktor telah membuka halaman "pengaturan dosen"
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor melihat daftar dosen dan menekan tombol "hapus" pada akun dosen yang ingin dihapus 2. Sistem akan memunculkan <i>pop-up</i> untuk konfirmasi penghapusan 3. Aktor menekan tombol "Ya" untuk melakukan penghapusan data 4. Sistem akan menghapus data dosen yang dipilih dan menampilkan kembali halaman "Pengaturan dosen" dengan pesan "Berhasil menghapus data"
<i>Post Condition</i>	Data dosen berhasil dihapus oleh aktor

<i>Alternative Flow</i>	1. Jika aktor menekan tombol “Tidak” maka <i>pop-up</i> akan tertutup dan data tidak dihapus
-------------------------	--

Tabel 4.11 Skenario Use Case Menambah KJFD

Nama Use Case	Menambah KJFD
Kode Kebutuhan Terkait	SRS-TT-F-8
Aktor	Ketua jurusan
Tujuan	Use case ini berfungsi untuk menambah KJFD
Pre Condition	Aktor telah membuka halaman “tambah KJFD”
Main flow	<ol style="list-style-type: none"> 1. Aktor mengisi form berupa nama KJFD dan memilih dosen sebagai Koordinator KJFD kemudian menekan tombol simpan 2. Sistem akan menyimpan data yang telah dimasukkan oleh aktor pada <i>database</i> dan menampilkan halaman “KJFD” dengan pesan “berhasil menambah data”
Post Condition	KJFD baru berhasil ditambah oleh aktor
Alternative Flow	<ol style="list-style-type: none"> 1. Jika nama KJFD telah digunakan sebelumnya maka akan muncul pesan “gagal menambah data, nama KJFD telah digunakan”

Tabel 4.12 Skenario Use Case Melihat KJFD

Nama Use Case	Melihat KJFD
Kode Kebutuhan Terkait	SRS-TT-F-9
Aktor	Ketua jurusan
Tujuan	Use case ini berfungsi untuk melihat daftar KJFD yang telah dibuat
Pre Condition	Aktor telah <i>login</i> dan diidentifikasi sebagai ketua jurusan
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih menu “KJFD” 2. Sistem akan menampilkan daftar KJFD yang telah dibuat.
Post Condition	Daftar KJFD dilihat oleh aktor
Alternative Flow	<ol style="list-style-type: none"> 1. Jika belum ada data KJFD yang dibuat maka akan muncul pesan “data belum tersedia”

Tabel 4.13 Skenario Use Case Mengubah KJFD

Nama Use Case	Mengubah KJFD
Kode Kebutuhan Terkait	SRS-TT-F-10
Aktor	Ketua jurusan
Tujuan	Use case ini berfungsi untuk mengubah data KJFD
Pre Condition	Aktor telah membuka halaman “Edit KJFD”

<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor merubah data KJFD berupa nama KJFD dan dosen sebagai Koordinator KJFD pada form “<i>edit KJFD</i>” dan menekan tombol simpan 2. Sistem akan melakukan <i>update</i> data sesuai KJFD yang ingin dirubah dan menampilkan halaman “KJFD” dengan pesan “berhasil mengubah data KJFD”
<i>Post Condition</i>	Data KJFD dirubah oleh aktor
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika nama KJFD telah digunakan sebelumnya maka akan muncul pesan “gagal mengubah data, nama KJFD telah digunakan”

Tabel 4.14 Skenario Use Case Menghapus KJFD

Nama Use Case	Menghapus KJFD
Kode Kebutuhan Terkait	SRS-TT-F-11
Aktor	Ketua jurusan
Tujuan	Use case ini berfungsi untuk menghapus KJFD yang telah dibuat
<i>Pre Condition</i>	Aktor telah membuka halaman “KJFD”
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor melihat daftar KJFD dan menekan tombol “hapus” pada KJFD yang ingin dihapus 2. Sistem akan memunculkan <i>pop-up</i> untuk konfirmasi penghapusan 3. Aktor menekan tombol “Ya” untuk melakukan penghapusan data 4. Sistem akan menghapus data KJFD yang dipilih dan menampilkan halaman “KJFD” dengan pesan “Berhasil menghapus data”
<i>Post Condition</i>	Data KJFD berhasil dihapus oleh aktor
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika aktor menekan tombol “Tidak” maka <i>pop-up</i> akan tertutup dan data tidak dihapus

Tabel 4.15 Skenario Use Case Menambah Mata Kuliah

Nama Use Case	Menambah mata kuliah
Kode Kebutuhan Terkait	SRS-TT-F-12
Aktor	Ketua jurusan
Tujuan	Use case ini berfungsi untuk menambah mata kuliah
<i>Pre Condition</i>	Aktor telah membuka halaman “Tambah mata kuliah”
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor mengisi form berupa kode mata kuliah, nama mata kuliah, dan KJFD dari mata kuliah tersebut dan menekan tombol simpan 2. Sistem akan menyimpan data yang telah dimasukkan oleh aktor pada <i>database</i> dan

	menampilkan halaman “mata kuliah” dengan pesan “berhasil menambah data”
<i>Post Condition</i>	Mata kuliah baru berhasil ditambah oleh aktor
<i>Alternative Flow</i>	1. Jika kode mata kuliah telah digunakan sebelumnya maka akan muncul pesan “Maaf kode mata kuliah telah digunakan”

Tabel 4.16 Skenario Use Case Melihat Mata Kuliah

Nama Use Case	Melihat mata kuliah
Kode Kebutuhan Terkait	SRS-TT-F-13
Aktor	Ketua jurusan
Tujuan	Use case ini berfungsi untuk melihat daftar mata kuliah yang telah dibuat
<i>Pre Condition</i>	Aktor telah login dan diidentifikasi sebagai ketua jurusan
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu “mata kuliah” 2. Sistem akan menampilkan daftar mata kuliah yang telah dibuat.
<i>Post Condition</i>	Daftar mata kuliah dilihat oleh aktor
<i>Alternative Flow</i>	1. Jika belum ada data mata kuliah yang dibuat maka akan muncul pesan “data belum tersedia”

Tabel 4.17 Skenario Use Case Mengubah Mata Kuliah

Nama Use Case	Mengubah mata kuliah
Kode Kebutuhan Terkait	SRS-TT-F-14
Aktor	Ketua jurusan
Tujuan	Use case ini berfungsi untuk mengubah data mata kuliah
<i>Pre Condition</i>	Aktor telah membuka halaman “Edit mata kuliah”
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor merubah data mata kuliah berupa kode mata kuliah, nama mata kuliah, dan KJFD pada form “edit mata kuliah” dan menekan tombol simpan 2. Sistem akan melakukan <i>update</i> data sesuai mata kuliah yang ingin dirubah dan menampilkan pesan “berhasil mengubah data mata kuliah”
<i>Post Condition</i>	Data mata kuliah dirubah oleh aktor
<i>Alternative Flow</i>	1. Jika kode mata kuliah telah digunakan sebelumnya maka akan muncul pesan “Maaf kode mata kuliah telah digunakan”

Tabel 4.18 Skenario Use Case Menghapus Mata Kuliah

Nama Use Case	Menghapus Mata kuliah
Kode Kebutuhan Terkait	SRS-TT-F-15

Aktor	Ketua jurusan
Tujuan	<i>Use case</i> ini berfungsi untuk menghapus mata kuliah yang telah dibuat
<i>Pre Condition</i>	Aktor telah membuka halaman “mata kuliah”
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor melihat daftar mata kuliah dan menekan tombol “hapus” pada mata kuliah yang ingin dihapus 2. Sistem akan memunculkan <i>pop-up</i> untuk konfirmasi penghapusan 3. Aktor menekan tombol “Ya” untuk melakukan penghapusan data 4. Sistem akan menghapus data mata kuliah yang dipilih dan menampilkan halaman “mata kuliah” dengan pesan “Berhasil menghapus data”
<i>Post Condition</i>	Data mata kuliah berhasil dihapus oleh aktor
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika aktor menekan tombol “Tidak” maka <i>pop-up</i> akan tertutup dan data tidak dihapus

Tabel 4.19 Skenario Use Case Melihat Team Teaching

Nama Use Case	Melihat <i>Team Teaching</i>
Kode Kebutuhan Terkait	SRS-TT-F-16
Aktor	Koordinator KJFD
Tujuan	<i>Use case</i> ini berfungsi untuk melihat daftar <i>team teaching</i> yang telah dibuat
<i>Pre Condition</i>	Aktor telah login dan diidentifikasi sebagai Koordinator KJFD
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu “<i>team teaching</i>” 2. Sistem akan menampilkan daftar <i>team teaching</i> yang ada sesuai dari mata kuliah yang telah dibuat.
<i>Post Condition</i>	Daftar <i>team teaching</i> dilihat oleh aktor
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika belum ada data mata kuliah yang dibuat maka akan muncul pesan “data belum tersedia”

Tabel 4.20 Skenario Use Case Mengubah Team Teaching

Nama Use Case	Mengubah <i>Team Teaching</i>
Kode Kebutuhan Terkait	SRS-TT-F-17
Aktor	Koordinator KJFD
Tujuan	<i>Use case</i> ini berfungsi untuk mengubah data <i>team teaching</i>
<i>Pre Condition</i>	Aktor telah membuka halaman “ <i>Edit team teaching</i> ”
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor merubah data <i>team teaching</i> berupa nama ketua <i>team teaching</i> dan nama anggota <i>team</i>

	<p><i>teaching</i> pada form “<i>edit</i> mata kuliah” dan menekan tombol simpan</p> <p>2. Sistem akan melakukan <i>update</i> data sesuai <i>team teaching</i> yang ingin dirubah dan menampilkan pesan “berhasil mengubah data <i>team teaching</i>”</p>
<i>Post Condition</i>	Data <i>team teaching</i> dirubah oleh aktor
<i>Alternative Flow</i>	-

Tabel 4.21 Skenario Use Case Menambah Materi

Nama Use Case	Menambah materi
Kode Kebutuhan Terkait	SRS-TT-F-18
Aktor	Ketua <i>team teaching</i>
Tujuan	Use case ini berfungsi untuk menambah materi
<i>Pre Condition</i>	Aktor telah membuka halaman “Tambah Materi”
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor mengisi form berupa nama materi dan bab kemudian menekan tombol simpan 2. Sistem akan menyimpan data yang telah dimasukkan oleh aktor pada <i>database</i> dan menampilkan halaman “Atur Materi” dengan pesan “berhasil menambah data”
<i>Post Condition</i>	Materi baru berhasil dibuat oleh aktor
<i>Alternative Flow</i>	-

Tabel 4.22 Skenario Use Case Melihat Materi

Nama Use Case	Menambah penugasan
Kode Kebutuhan Terkait	SRS-TT-F-19
Aktor	Ketua <i>team teaching</i>
Tujuan	Use case ini berfungsi untuk melihat daftar materi yang telah dibuat
<i>Pre Condition</i>	Aktor telah login dan diidentifikasi sebagai ketua <i>team teaching</i>
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu “atur materi” 2. Sistem akan menampilkan daftar materi yang telah dibuat.
<i>Post Condition</i>	Daftar materi dilihat oleh aktor
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika belum ada data mata kuliah yang dibuat maka akan muncul pesan “data belum tersedia”

Tabel 4.23 Skenario Use Case Mengubah Materi

Nama Use Case	Mengubah materi
Kode Kebutuhan Terkait	SRS-TT-F-20
Aktor	Ketua <i>team teaching</i>

Tujuan	<i>Use case</i> ini berfungsi untuk mengubah materi
<i>Pre Condition</i>	Aktor telah membuka halaman " <i>Edit materi</i> "
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor merubah data materi berupa nama materi dan bab materi pada form "<i>edit materi</i>" dan menekan tombol simpan 2. Sistem akan melakukan <i>update</i> data sesuai materi yang ingin dirubah dan menampilkan pesan "berhasil mengubah data materi"
<i>Post Condition</i>	Data materi dirubah oleh aktor
<i>Alternative Flow</i>	-

Tabel 4.24 Skenario *Use Case* Menghapus Materi

Nama <i>Use Case</i>	Menghapus Materi
Kode Kebutuhan Terkait	SRS-TT-F-21
Aktor	Ketua <i>team teaching</i>
Tujuan	<i>Use case</i> ini berfungsi untuk menghapus materi yang telah dibuat
<i>Pre Condition</i>	Aktor telah membuka halaman " <i>atur materi</i> "
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor melihat daftar materi dan menekan tombol "<i>hapus</i>" pada materi yang ingin dihapus 2. Sistem akan memunculkan <i>pop-up</i> untuk konfirmasi penghapusan 3. Aktor menekan tombol "<i>Ya</i>" untuk melakukan penghapusan data 4. Sistem akan menghapus data materi yang dipilih dan menampilkan halaman "<i>Atur materi</i>" dengan pesan "<i>Berhasil menghapus data</i>"
<i>Post Condition</i>	Data materi berhasil dihapus oleh aktor
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika aktor menekan tombol "<i>Tidak</i>" maka <i>pop-up</i> akan tertutup dan data tidak dihapus

Tabel 4.25 Skenario *Use Case* Menambah Penugasan

Nama <i>Use Case</i>	Menambah penugasan
Kode Kebutuhan Terkait	SRS-TT-F-22
Aktor	Ketua <i>team teaching</i>
Tujuan	<i>Use case</i> ini berfungsi untuk menambah Penugasan
<i>Pre Condition</i>	Aktor telah membuka halaman " <i>Tambah penugasan</i> "
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor mengisi form berupa judul penugasan, materi, kuota, kebutuhan soal pada tiap tingkat kesulitan, batas waktu pengambilan materi dan batas waktu penugasan dan menekan tombol simpan

	2. Sistem akan menyimpan data yang telah dimasukkan oleh aktor pada <i>database</i> dan menampilkan halaman “Atur Penugasan” dengan pesan “berhasil menambah data”
<i>Post Condition</i>	Penugasan baru berhasil dibuat oleh aktor
<i>Alternative Flow</i>	-

Tabel 4.26 Skenario Use Case Melihat Penugasan

Nama Use Case	Melihat penugasan
Kode Kebutuhan Terkait	SRS-TT-F-23
Aktor	Ketua <i>team teaching</i> dan dosen
Tujuan	Use case ini berfungsi untuk melihat daftar penugasan yang telah dibuat
<i>Pre Condition</i>	Aktor telah login dan diidentifikasi sebagai Ketua <i>team teaching</i> atau anggota <i>team teaching</i>
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu “Penugasan” 2. Sistem akan menampilkan daftar penugasan yang telah dibuat.
<i>Post Condition</i>	Daftar penugasan dilihat oleh aktor
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika belum ada data penugasan yang dibuat maka akan muncul pesan “data belum tersedia”

Tabel 4.27 Skenario Use Case Melihat Progres Penugasan

Nama Use Case	Melihat progres penugasan
Kode Kebutuhan Terkait	SRS-TT-F-24
Aktor	Ketua <i>team teaching</i>
Tujuan	Use case ini berfungsi untuk melihat progres penugasan yang telah diinginkan
<i>Pre Condition</i>	Aktor telah membuka halaman “atur penugasan”
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor melihat daftar penugasan dan menekan tombol “lihat progres” pada penugasan yang diinginkan 2. Sistem akan menampilkan data progres penugasan yang telah dibuat.
<i>Post Condition</i>	Progres penugasan dilihat oleh aktor
<i>Alternative Flow</i>	-

Tabel 4.28 Skenario Use Case Mengubah Penugasan

Nama Use Case	Mengubah Penugasan
Kode Kebutuhan Terkait	SRS-TT-F-25
Aktor	Ketua <i>team teaching</i>
Tujuan	Use case ini berfungsi untuk mengubah data KJFD

<i>Pre Condition</i>	Aktor telah membuka halaman “ <i>Edit Penugasan</i> ”
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor merubah data penugasan berupa judul penugasan, materi, kuota, kebutuhan soal pada tiap tingkat kesulitan, batas waktu pengambilan materi dan batas waktu penugasan pada form “<i>edit penugasan</i>” dan menekan tombol simpan 2. Sistem akan melakukan <i>update</i> data sesuai penugasan yang ingin dirubah dan menampilkan pesan “berhasil mengubah data penugasan”
<i>Post Condition</i>	Data penugasan dirubah oleh aktor
<i>Alternative Flow</i>	-

Tabel 4.29 Skenario Use Case Menghapus Penugasan

Nama Use Case	Menghapus Penugasan
Kode Kebutuhan Terkait	SRS-TT-F-26
Aktor	Ketua <i>team teaching</i>
Tujuan	Use case ini berfungsi untuk menghapus penugasan yang telah dibuat
<i>Pre Condition</i>	Aktor telah membuka halaman “ <i>Atur penugasan</i> ”
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor melihat daftar penugasan dan menekan tombol “hapus” pada penugasan yang ingin dihapus 2. Sistem akan memunculkan <i>pop-up</i> untuk konfirmasi penghapusan 3. Aktor menekan tombol “Ya” untuk melakukan penghapusan data 4. Sistem akan menghapus data penugasan yang dipilih dan menampilkan halaman “<i>Atur penugasan</i>” dengan pesan “Berhasil menghapus data”
<i>Post Condition</i>	Data penugasan berhasil dihapus oleh aktor
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika aktor menekan tombol “Tidak” maka <i>pop-up</i> akan tertutup dan data tidak dihapus

Tabel 4.30 Skenario Use Case Mengatur Anggota Materi Penugasan

Nama Use Case	Mengatur anggota materi penugasan
Kode Kebutuhan Terkait	SRS-TT-F-27
Aktor	Ketua <i>team teaching</i>
Tujuan	Use case ini berfungsi untuk mengatur anggota materi penugasan yang telah tersedia
<i>Pre Condition</i>	Aktor telah membuka halaman “ <i>Detail penugasan</i> ”
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih anggotanya untuk dimasukkan atau dikeluarkan dari materi dengan menekan tombol “gabung” atau “keluar”.

	<ol style="list-style-type: none"> 2. Sistem akan memunculkan <i>pop-up</i> untuk konfirmasi tindakan aktor 3. Aktor menekan tombol 'Ya' untuk menjalankan pilihan aktor 4. Sistem akan memperbarui anggota materi penugasan dan menampilkan pesan sukses
<i>Post Condition</i>	Anggota materi penugasan berhasil diatur oleh aktor
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika aktor menekan tombol 'Tidak' maka <i>pop-up</i> akan tertutup dan batal menjalankan pilihan aktor

Tabel 4.31 Skenario Use Case Gabung Materi Penugasan

Nama Use Case	Gabung materi penugasan
Kode Kebutuhan Terkait	SRS-TT-F-28
Aktor	dosen
Tujuan	Use case ini berfungsi untuk memilih materi penugasan yang telah tersedia
<i>Pre Condition</i>	Aktor telah membuka halaman "Detail penugasan"
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol "gabung" pada materi yang ingin diambil 2. Sistem akan memunculkan <i>pop-up</i> untuk konfirmasi bergabung 3. Aktor menekan tombol "Ya" untuk bergabung pada materi 4. Sistem akan menambahkan aktor pada materi penugasan dan menampilkan pesan "sukses bergabung"
<i>Post Condition</i>	Materi penugasan berhasil dipilih oleh aktor
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika aktor menekan tombol "Tidak" maka <i>pop-up</i> akan tertutup dan batal untuk bergabung 2. Jika kuota pada materi telah terpenuhi maka akan menampilkan pesan "gagal bergabung" 3. Jika aktor telah tergabung pada materi penugasan yang dipilih akan menampilkan pesan "Sudah tergabung pada materi tersebut"

Tabel 4.32 Skenario Use Case Keluar Materi Penugasan

Nama Use Case	Keluar materi penugasan
Kode Kebutuhan Terkait	SRS-TT-F-29
Aktor	dosen
Tujuan	Use case ini berfungsi untuk keluar dari materi penugasan yang telah dipilih
<i>Pre Condition</i>	Aktor telah membuka halaman "Detail penugasan"
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol "keluar" pada materi yang ingin ditolak untuk diambil

	<ol style="list-style-type: none"> 2. Sistem akan memunculkan <i>pop-up</i> untuk konfirmasi keluar 3. Aktor menekan tombol 'Ya' untuk keluar dari materi 4. Sistem akan menghapus aktor dari anggota materi penugasan yang dipilih dan menampilkan pesan "sukses batal bergabung"
<i>Post Condition</i>	Materi penugasan berhasil ditolak oleh aktor
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika aktor menekan tombol 'Tidak' maka <i>pop-up</i> akan tertutup dan batal untuk keluar

Tabel 4.33 Skenario Use Case Membuat Soal

Nama Use Case	Membuat soal
Kode Kebutuhan Terkait	SRS-TT-F-30
Aktor	Ketua <i>team teaching</i> dan dosen
Tujuan	<i>Use case</i> ini berfungsi untuk membuat soal dari materi yang telah dipilih
<i>Pre Condition</i>	Aktor telah membuka halaman "Buat soal"
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Ketua dan Anggota <i>Team Teaching</i> mengisi form berupa soal, opsi jawaban, dan estimasi waktu pengerjaan kemudian menekan tombol simpan 2. Sistem akan menyimpan data soal yang dibuat menampilkan halaman "detail penugasan" dengan pesan "berhasil membuat soal"
<i>Post Condition</i>	Soal berhasil dibuat oleh aktor
<i>Alternative Flow</i>	-

Tabel 4.34 Skenario Use Case Mengubah Soal

Nama Use Case	Mengubah Soal
Kode Kebutuhan Terkait	SRS-TT-F-31
Aktor	Ketua <i>team teaching</i> dan dosen
Tujuan	<i>Use case</i> ini berfungsi untuk mengubah soal
<i>Pre Condition</i>	Aktor telah membuka halaman "Edit soal"
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor merubah soal berupa soal, opsi jawaban, dan estimasi waktu pengerjaan pada form "edit soal" dan menekan tombol simpan 2. Sistem akan melakukan <i>update</i> data sesuai soal yang ingin dirubah dan menampilkan halaman "Detail penugasan" dengan pesan "berhasil mengubah soal"
<i>Post Condition</i>	Soal dirubah oleh aktor
<i>Alternative Flow</i>	-

Tabel 4.35 Skenario Use Case Submit Soal

Nama Use Case	Submit soal
Kode Kebutuhan Terkait	SRS-TT-F-32
Aktor	Ketua <i>team teaching</i> dan dosen
Tujuan	Use case ini berfungsi untuk melakukan <i>submit</i> dari soal yang telah dianggap selesai dikerjakan
Pre Condition	Aktor telah membuka halaman “detail penugasan”
Main flow	<ol style="list-style-type: none"> 1. Aktor melihat daftar materi dan menekan tombol “submit” pada materi yang ingin di-<i>submit</i> 2. Sistem akan memunculkan <i>pop-up</i> untuk konfirmasi submit 3. Aktor menekan tombol “Ya” untuk melakukan submit soal 4. Sistem akan <i>submit</i> soal dari materi yang dipilih dan menampilkan halaman “detail penugasan” dengan pesan “Berhasil submit”
Post Condition	Soal pada materi berhasil di- <i>submit</i> oleh aktor
Alternative Flow	<ol style="list-style-type: none"> 1. Jika aktor menekan tombol “Tidak” maka <i>pop-up</i> akan tertutup dan <i>submit</i> batal dilakukan

Tabel 4.36 Skenario Use Case Mempercepat Evaluasi

Nama Use Case	Mempercepat evaluasi
Kode Kebutuhan Terkait	SRS-TT-F-33
Aktor	Ketua <i>Team Teaching</i>
Tujuan	Use case ini berfungsi untuk mempercepat batas waktu pengerjaan sehingga dapat memulai evaluasi lebih awal
Pre Condition	Aktor telah membuka halaman “Lihat progres”
Main flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol “Evaluasi sekarang” 2. Sistem akan memunculkan <i>pop-up</i> untuk konfirmasi mempercepat evaluasi 3. Aktor menekan tombol ‘Ya’ untuk melakukan mempercepat evaluasi 4. Sistem akan mempercepat batas waktu pengerjaan dan menampilkan halaman “Lihat progres” dan pesan sukses mempercepat evaluasi
Post Condition	Evaluasi berhasil dipercepat oleh aktor
Alternative Flow	<ol style="list-style-type: none"> 1. Jika aktor menekan tombol ‘Tidak’ maka <i>pop-up</i> akan tertutup dan batas waktu pengerjaan tidak berubah

Tabel 4.37 Skenario Use Case Mengirim Reminder

Nama Use Case	Mengirim <i>Reminder</i>
Kode Kebutuhan Terkait	SRS-TT-F-34
Aktor	Ketua <i>Team Teaching</i>
Tujuan	<i>Use case</i> ini berfungsi untuk mengirimkan <i>reminder</i> kepada anggota penugasan untuk mengingatkan tanggung jawab penugasan
<i>Pre Condition</i>	Aktor telah membuka halaman “Lihat progres”
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol ‘<i> kirim reminder</i>’ 2. Sistem akan memunculkan <i> pop-up</i> untuk konfirmasi pengiriman 3. Aktor menekan tombol ‘Ya’ untuk melakukan pengiriman <i> reminder</i> 4. Sistem akan mengirim <i> reminder</i> kepada anggota penugasan dan menampilkan halaman “Lihat progres” dan pesan sukses mengirim <i> reminder</i>
<i>Post Condition</i>	<i>Reminder</i> berhasil dikirim oleh aktor
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika aktor menekan tombol ‘Tidak’ maka <i> pop-up</i> akan tertutup dan <i> reminder</i> tidak dikirim 2. Jika anggota penugasan telah menyelesaikan semua tanggung jawab penugasan maka akan muncul pesan “tidak ada <i> reminder</i> yang dikirimkan”

Tabel 4.38 Skenario Use Case Memberi Komentar

Nama Use Case	Memberi komentar
Kode Kebutuhan Terkait	SRS-TT-F-35
Aktor	Ketua <i> team teaching</i> dan Anggota <i> team teaching</i>
Tujuan	<i>Use case</i> ini berfungsi untuk memberi komentar terhadap soal dari penugasan
<i>Pre Condition</i>	Aktor telah membuka halaman “ <i> evaluasi penugasan</i> ”
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol “<i> komentar</i>” 2. Sistem menampilkan <i> form</i> komentar 3. Aktor memilih kategori komentar dan isi komentar kemudian menekan tombol “<i> tambah</i>” 4. Sistem menyimpan komentar pada <i> database</i> dan menampilkan pesan sukses menambah komentar
<i>Post Condition</i>	Komentar berhasil ditambah oleh aktor
<i>Alternative Flow</i>	-

Tabel 4.39 Skenario Use Case Menghapus Komentar

Nama Use Case	Menghapus Komentar
Kode Kebutuhan Terkait	SRS-TT-F-36

Aktor	Ketua <i>team teaching</i> dan dosen
Tujuan	<i>Use case</i> ini berfungsi untuk menghapus komentar yang telah dibuat
<i>Pre Condition</i>	Aktor telah membuka halaman “evaluasi penugasan”
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor melihat komentar yang telah dibuat dan menekan tombol “X” 2. Sistem akan memunculkan <i>pop-up</i> untuk konfirmasi penghapusan 3. Aktor menekan tombol “Ya” untuk melakukan penghapusan komentar 4. Sistem akan menghapus komentar yang dipilih dan menampilkan halaman “Evaluasi penugasan” dengan pesan “Berhasil menghapus komentar”
<i>Post Condition</i>	Komentar berhasil dihapus oleh aktor
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika aktor menekan tombol “Tidak” maka <i>pop-up</i> akan tertutup dan data tidak dihapus

Tabel 4.40 Skenario Use Case Mengevaluasi Soal

Nama <i>Use Case</i>	Mengevaluasi Soal
Kode Kebutuhan Terkait	SRS-TT-F-37
Aktor	Ketua <i>team teaching</i>
Tujuan	<i>Use case</i> ini berfungsi untuk mengevaluasi soal yang telah dibuat berdasarkan komentar yang telah diberikan
<i>Pre Condition</i>	Aktor telah membuka halaman “edit evaluasi”
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor merubah soal berupa soal, opsi jawaban, dan estimasi waktu pengerjaan pada form “<i>edit evaluasi</i>” dan menekan tombol simpan 2. Sistem akan melakukan <i>update</i> data soal dan menampilkan halaman “<i>edit evaluasi</i>” dengan pesan “berhasil mengevaluasi soal”
<i>Post Condition</i>	Soal berhasil dievaluasi oleh aktor
<i>Alternative Flow</i>	-

Tabel 4.41 Skenario Use Case Menutup Evaluasi

Nama <i>Use Case</i>	Menutup evaluasi
Kode Kebutuhan Terkait	SRS-TT-F-38
Aktor	Ketua <i>team teaching</i>
Tujuan	<i>Use case</i> ini berfungsi untuk menutup evaluasi penugasan sehingga soal dari penugasan dapat dicetak
<i>Pre Condition</i>	Aktor telah membuka halaman “atur penugasan”
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor melihat daftar penugasan dan menekan tombol “tutup evaluasi”

	<ol style="list-style-type: none"> 2. Sistem akan memunculkan <i>pop-up</i> untuk konfirmasi penutupan evaluasi 3. Aktor menekan tombol “Ya” untuk melakukan penutupan evaluasi 4. Sistem akan menutup evaluasi dari penugasan yang dipilih dan menampilkan halaman “atur penugasan” dengan pesan “Berhasil tutup evaluasi”
<i>Post Condition</i>	Evaluasi penugasan berhasil ditutup oleh aktor
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika aktor menekan tombol ‘Tidak’ maka <i>pop-up</i> akan tertutup dan batal menutup evaluasi

Tabel 4.42 Skenario Use Case Review Soal

Nama Use Case	Review soal
Kode Kebutuhan Terkait	SRS-TT-F-39
Aktor	Ketua <i>team teaching</i>
Tujuan	Use case ini berfungsi untuk me-review soal sebelum dicetak
<i>Pre Condition</i>	Aktor telah membuka halaman “cetak soal”
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor memasukkan persentase soal atau aktor memilih soal sesuai kebutuhan dan menekan tombol “review” 2. Sistem mengolah data soal dan menampilkan halaman “review soal” 3. Aktor mengatur tahun ajaran, mata kuliah, nama dosen, banyak kode soal yang akan dicetak, jenis <i>font</i> yang digunakan dalam mencetak soal, estimasi waktu pengerjaan, sifat ujian, dan ukuran font kemudian menekan tombol “cetak”
<i>Post Condition</i>	Soal berhasil dicetak oleh aktor
<i>Alternative Flow</i>	-

Tabel 4.43 Skenario Use Case Cetak Soal Acak

Nama Use Case	Cetak soal acak
Kode Kebutuhan Terkait	SRS-TT-F-40
Aktor	Ketua <i>team teaching</i>
Tujuan	Use case ini berfungsi untuk mencetak soal yang telah dievaluasi secara acak
<i>Pre Condition</i>	Aktor telah membuka halaman “cetak soal”
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor menentukan jumlah soal dan memasukkan persentase dari soal yang diinginkan pada <i>form</i> cetak soal sesuai materi dan tingkat kesulitan yang tersedia dan kemudian menekan tombol “review”

	<ol style="list-style-type: none"> 2. Sistem akan menampilkan <i>review</i> soal sebelum dicetak 3. Aktor dapat mengubah atribut soal sesuai yang diharapkan dan menekan tombol “cetak” untuk lanjut mencetak 4. Sistem akan mencetak sesuai pengaturan <i>review</i> yang ditampilkan sebelumnya
<i>Post Condition</i>	Soal acak berhasil dicetak oleh aktor
<i>Alternative Flow</i>	-

Tabel 4.44 Skenario Use Case Cetak Soal Pilih

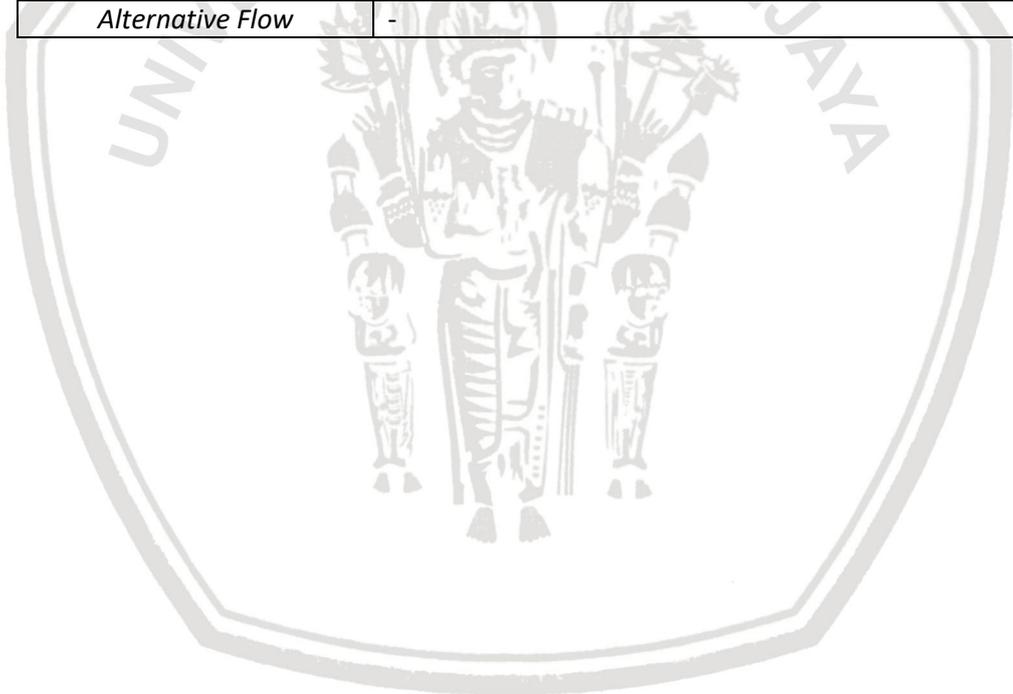
Nama Use Case	Cetak soal pilih
Kode Kebutuhan Terkait	SRS-TT-F-41
Aktor	Ketua <i>team teaching</i>
Tujuan	<i>Use case</i> ini berfungsi untuk mencetak soal yang telah dievaluasi dengan memilihnya secara manual
<i>Pre Condition</i>	Aktor telah membuka halaman “pilih soal”
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih soal dari materi yang ada sesuai kebutuhan dan kemudian menekan tombol “review” 2. Sistem akan menampilkan <i>review</i> soal sebelum dicetak 3. Aktor dapat mengubah atribut soal sesuai yang diharapkan dan menekan tombol “cetak” untuk lanjut mencetak 4. Sistem akan mencetak sesuai pengaturan <i>review</i> yang ditampilkan sebelumnya
<i>Post Condition</i>	Soal pilihan berhasil dicetak oleh aktor
<i>Alternative Flow</i>	-

Tabel 4.45 Skenario Use Case Melihat Riwayat Cetak Soal

Nama Use Case	Melihat riwayat cetak soal
Kode Kebutuhan Terkait	SRS-TT-F-42
Aktor	Ketua <i>team teaching</i>
Tujuan	<i>Use case</i> ini berfungsi untuk melihat riwayat cetak soal yang telah dilakukan
<i>Pre Condition</i>	Aktor telah login dan diidentifikasi sebagai ketua <i>team teaching</i>
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu “riwayat cetak soal” 2. Sistem akan menampilkan data riwayat cetak soal yang telah dilakukan
<i>Post Condition</i>	Data riwayat cetak soal dilihat oleh aktor
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika belum ada data riwayat cetak soal maka akan muncul pesan “data belum tersedia”

Tabel 4.46 Skenario *Use Case* Evaluasi Bank Soal

Nama <i>Use Case</i>	Evaluasi Bank soal
Kode Kebutuhan Terkait	SRS-TT-F-43
Aktor	Ketua <i>team teaching</i>
Tujuan	<i>Use case</i> ini berfungsi untuk mengevaluasi soal yang telah siap untuk dicetak
<i>Pre Condition</i>	Aktor telah membuka halaman “dashboard”
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih materi yang ingin dievaluasi dari grafik atau menekan tombol “evaluasi” yang ada diatas grafik 2. Sistem akan menampilkan halaman “evaluasi bank soal” 3. Aktor dapat mengubah atribut soal dan dapat mengubah status soal dari digunakan menjadi tidak digunakan dan menekan tombol simpan 4. Sistem akan menyimpan perubahan yang dilakukan aktor dan mengembalikan ke halaman “evaluasi bank soal” dengan pesan sukses
<i>Post Condition</i>	Soal berhasil dievaluasi oleh aktor
<i>Alternative Flow</i>	-



BAB 5 PERANCANGAN DAN IMPLEMENTASI SISTEM

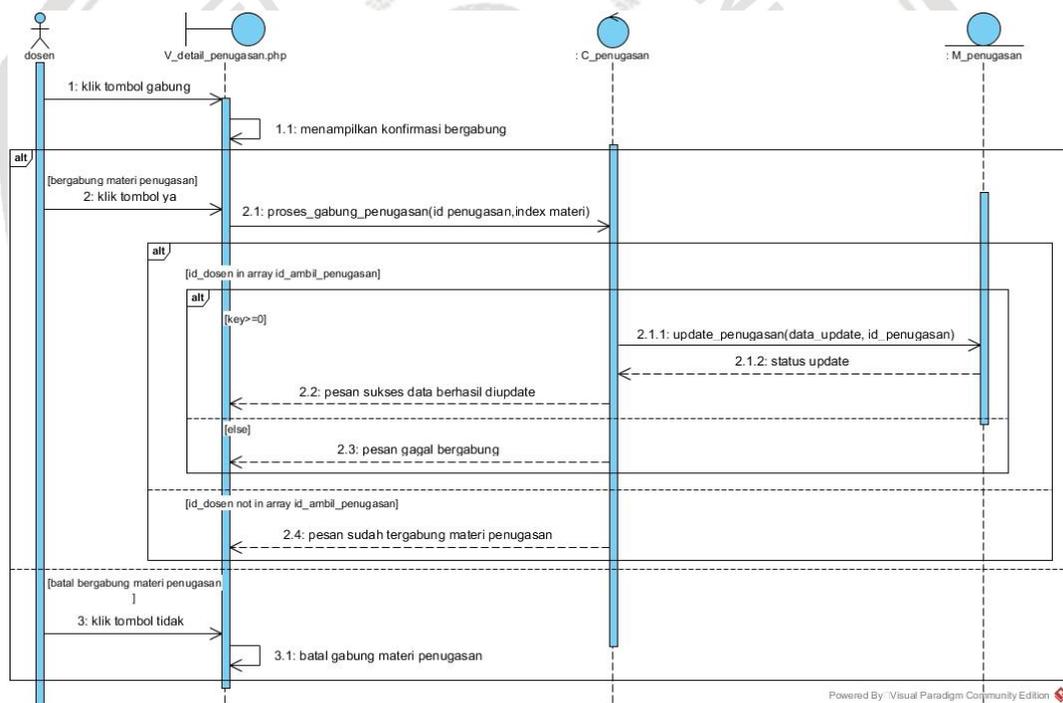
5.1 Perancangan Sistem

Setelah melakukan proses analisis kebutuhan, proses perancangan sistem dilakukan dengan memodelkan hasil dari analisis kebutuhan. Pada proses perancangan sistem pengelolaan soal terdapat empat bagian perancangan, yaitu perancangan arsitektur, perancangan komponen, perancangan data, dan perancangan antarmuka.

5.1.1 Perancangan Arsitektur

Perancangan arsitektur memberikan penjelasan tentang *sequence diagram* dan *class diagram*. *Sequence diagram* menggambarkan kelakuan antar objek dengan *lifetime* dan *message* yang dikirim atau diterima antar objek. Pada bagian ini akan dijelaskan *sequence diagram* dari tiga fitur utama sistem pengelolaan soal yaitu, gabung materi penugasan, mengirim *reminder*, dan membuat soal.

5.1.1.1 Sequence Diagram Gabung Materi Penugasan

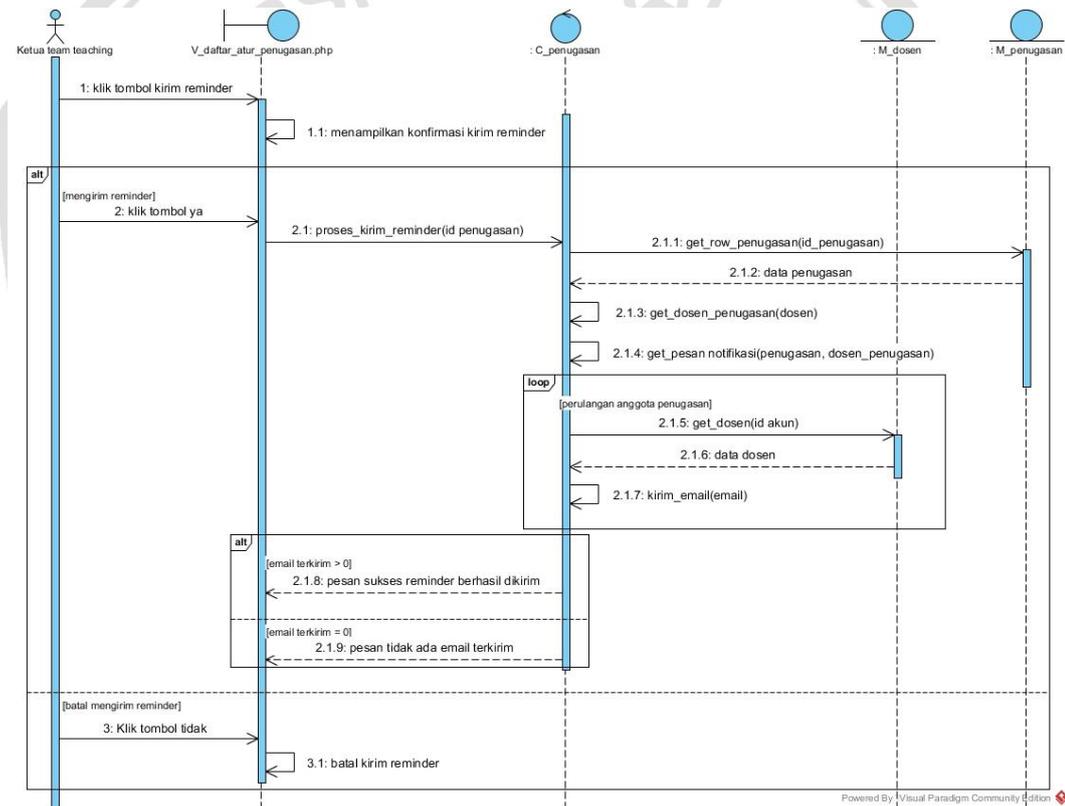


Gambar 5.1 Sequence Diagram Gabung Materi Penugasan

Pada Gambar 5.1 merupakan *sequence diagram use case* gabung materi penugasan, pada *sequence diagram* tersebut dijelaskan bagaimana aktor yakni dosen sebagai anggota *team teaching* dapat bergabung dalam materi pada suatu penugasan. Terdapat sebuah *boundary* yang menjembatani aktor dengan *controller* yaitu *boundary* dengan nama *v_detail_penugasan.php*, adanya objek *controller* yaitu *controller* dengan nama *C_penugasan*, dan sebuah objek *entity* dengan nama *M_penugasan*. Untuk dapat bergabung pada sebuah materi, aktor

menekan tombol gabung pada materi yang diinginkan, yang kemudian *boundary* `v_detail_penugasan.php` akan menampilkan *pop-up* untuk konfirmasi bergabung, jika aktor menyetujui dengan memilih 'ya' maka *boundary* `v_detail_penugasan.php` akan memanggil *method* `proses_gabung_penugasan` dengan 2 parameter yakni id penugasan dari materi tersebut dan index dari materi tersebut dari materi penugasan yang telah dilakukan proses *unserialize* dan menjadi *array*. Terdapat kondisi jika `id_dosen` yang akan bergabung telah terdapat pada *array* maka akan menampilkan notifikasi bahwa dosen telah bergabung dalam materi. Namun jika kondisi tidak terpenuhi maka akan terdapat variabel *key* dimana jika *key* lebih dari atau sama dengan nol maka *controller* `c_penugasan` akan memanggil *method* `update_penugasan` dengan parameter data yang akan diperbarui dan id penugasan sebagai penunjuk data yang akan diperbarui. Model akan memberikan kembalian berupa status *update* berhasil dilakukan, maka *controller* `c_penugasan` akan mengirim pesan ke *view* bahwa aktor berhasil bergabung, dan menampilkan notifikasi berhasil.

5.1.1.2 Sequence Diagram Mengirim Reminder



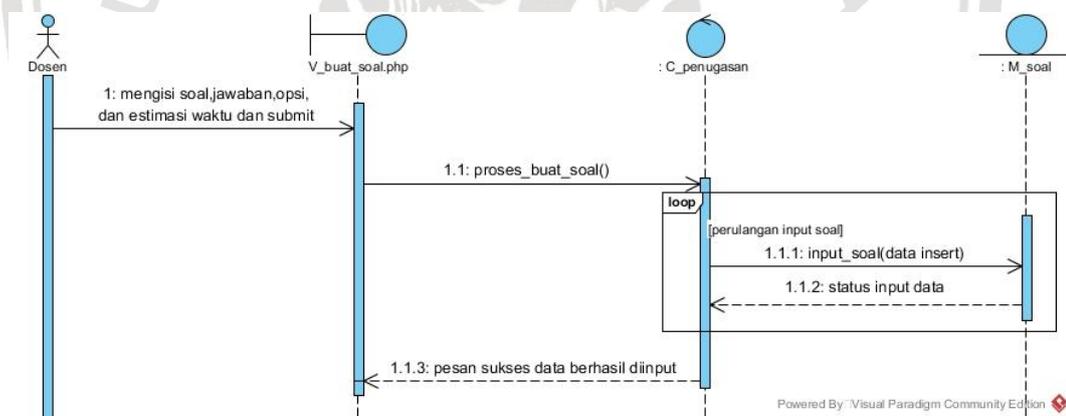
Gambar 5.2 Sequence Diagram Mengirim Reminder

Pada gambar 5.2 merupakan *sequence use case* mengirim *reminder*, pada *sequence diagram* tersebut dijelaskan bagaimana aktor yakni ketua *team teaching* dapat mengirimkan *reminder* kepada anggota penugasannya untuk menyelesaikan penugasan. Terdapat sebuah *boundary* dengan nama `v_progres_penugasan.php` yang menjembatani komunikasi antara aktor dengan sebuah objek *controller* dengan nama `c_penugasan` dimana nantinya akan



membutuhkan objek *entity* M_penugasan dan M_dosen untuk mendapatkan data yang dibutuhkan. Untuk dapat mengirimkan *reminder*, aktor perlu menekan tombol “*kirim reminder*” pada *v_progres_penugasan.php*, kemudian *boundary* *v_progres_penugasan.php* meminta konfirmasi dan jika aktor mengonfirmasi untuk melakukan pengiriman *reminder* maka *v_progres_penugasan.php* memanggil *method* *proses_kirim_reminder* dengan parameter *id_penugasan*. Menggunakan *id_penugasan* *controller* *c_penugasan* memanggil *get_row_penugasan* dan mendapatkan data penugasan dan mendapat *id* anggota penugasan. Dimana *id* anggota ini digunakan untuk mendapatkan *id* dosen yang berada pada penugasan tersebut melalui *method* *get_dosen_penugasan* dengan parameter *id* anggota yang telah didapatkan sebelumnya. Kemudian dibutuhkan *method* *get_pesanan_notifikasi* dengan parameter penugasan dan dosen penugasan untuk mendapatkan pesan yang akan dikirim untuk tiap dosen penugasan. Kemudian melalui *id* anggota penugasan didapatkan *email* dengan memanggil *method* *get_dosen* dan dengan *email* yang didapatkan proses mengirim *email* sebagai *reminder* dilakukan dengan memanggil *method* *kirim_email*, dan proses ini diulangi hingga semua anggota penugasan mendapatkan *reminder* berupa *email*. Setelah proses pengiriman *email* selesai maka akan terdapat pesan sukses mengirimkan email pada *v_daftar_atur_penugasan.php* jika *email* yang dikirim lebih dari nol, namun tidak ada *email* yang dikirim maka akan menampilkan pesan tidak ada *email* yang dikirim.

5.1.1.3 Sequence Diagram Membuat Soal



Gambar 5.3 Sequence Diagram Membuat Soal

Pada gambar 5.3 merupakan *sequence* diagram use case membuat soal, pada *sequence* diagram tersebut dijelaskan bagaimana aktor yakni dosen dapat membuat soal didalam sistem. Terdapat sebuah *boundary* dengan nama *v_buat_soal.php* yang menjembatani komunikasi dengan sebuah *controller* dengan nama *c_penugasan* dimana nantinya akan memasukkan data melalui objek *entity* dengan nama *M_soal*. Aktor dapat membuat soal pada sistem dengan mengisi *form* yang ada, pengisian yang perlu dilakukan yakni soal, jawaban, opsi, dan estimasi waktu pengerjaan, kemudian aktor dapat menyimpannya dengan menekan tombol simpan. Setelah menekan tombol simpan *v_buat_soal.php* memanggil *method* *proses_buat_soal* pada *controller* *c_penugasan*, untuk

melakukan proses penyimpanan soal `c_penugasan` memanggil model dengan *method* `input_soal` data yang akan disimpan. Terjadi iterasi pada proses penyimpanan soal, karena data soal, jawaban, opsi dan estimasi waktu tidak hanya ada satu, sehingga iterasi dibutuhkan agar beberapa soal dapat dimasukkan dalam waktu yang bersamaan. Setelah semua proses penyimpanan selesai maka `c_penugasan` memberi kembalian berupa pesan sukses.

5.1.1.4 Class Diagram

Class diagram menggambarkan struktur dari sistem, penggambaran diambil dari segi pendefinisian kelas-kelas yang akan dibuat dalam membangun sistem. *Class diagram* dapat dilihat pada Gambar 5.4.



5.1.2 Perancangan Komponen

Dalam perancangan komponen ini akan menjelaskan proses yang terjadi pada beberapa bagian dari sub-sistem untuk menjalankan fungsionalitas sistem. Perancangan algoritme yang dibuat digunakan sebagai acuan dalam membuat kode program pada tahap implementasi. Dalam perancangan komponen ini dipilih tiga algoritme utama yang mewakili tiga fitur utama dalam sistem pengelolaan soal yaitu *method* “gabung_penugasan” dari *class* “C_penugasan”, *method* “kirim_reminder” dari *class* “C_penugasan”, dan *method* “buat_soal” dari *class* “C_penugasan”.

5.1.2.1 Perancangan Komponen *Method* “proses_gabung_penugasan” *class* “C_penugasan”

Nama class: C_penugasan

Nama operasi: proses_gabung_penugasan(id,index)

Algoritme:

Tabel 5.1 Perancangan Komponen Method Gabung Penugasan

1	Set variabel penugasan = data dari tabel penugasan dimana id_penugasan = id
2	Set variabel array id_ambil_penugasan = unserialize dari kolom id_ambil_penugasan dari variabel penugasan
3	Set variabel id_dosen = id
4	If id_dosen ada pada array id_ambil_penugasan[index]
5	Set variabel key = index dari array id_ambil_penugasan[index] yang bernilai 0
6	If key>=0
7	id_ambil_penugasan[index][i] = id_dosen
8	Set ser_id_ambil = serialize dari array id_ambil_penugasan
9	Deklarasi data_update = kolom id_ambil_penugasan set ser_id_ambil
10	Deklarasi where = kolom id_penugasan find id
11	Variabel res = update data berdasarkan data_update pada where
12	Set pesan sukses
13	Else
14	Set pesan gagal
15	Else
16	Set pesan sudah terdaftar
17	Endif
18	Go to halaman detail penugasan

5.1.2.2 Perancangan Komponen *Method* “kirim_reminder” *class* “C_penugasan”

Nama class: C_penugasan

Nama operasi: kirim_reminder(id)

Algoritme:

Tabel 5.2 Perancangan Komponen Method Kirim Reminder

1	Set variabel where id_penugasan = id Set variabel penugasan = data dari tabel penugasan berdasarkan variabel where
2	Set variabel array dosen = unserialize dari kolom id_ambil_penugasan dari variabel penugasan
3	Set variabel array materi = unserialize dari kolom materi_penugasan dari variabel penugasan
4	Set variabel array soal = unserialize dari kolom tingkat_kesulitan dari variabel penugasan
5	
6	Set variabel array dosen_penugasan = nilai return dari method get_dosen_penugasan(dosen)
7	Set variabel array notifikasi = nilai return dari method get_pesan_notifikasi(penugasan, dosen_penugasan)
8	Set variabel count_email = 0
9	For i=0 sampai i<panjang array dosen_penugasan
10	If notifikasi indeks ke i tersedia
11	Set variabel where id_dosen = dosen_penugasan[\$i]
12	Set data_dosen_penugasan = ambil data dosen berdasarkan variabel where pada database
13	Mengirim email dengan method email_send(email dosen, notifikasi[i])
14	Count_email tambah 1
15	Endif
16	Endfor
17	
18	Set pesan hitung_msg = count_email
19	Go to halaman detail penugasan

5.1.2.3 Perancangan Komponen Method “proses_buat_soal” class “C_penugasan”

Nama class: C_penugasan

Nama operasi: proses_buat_soal()

Algoritme:

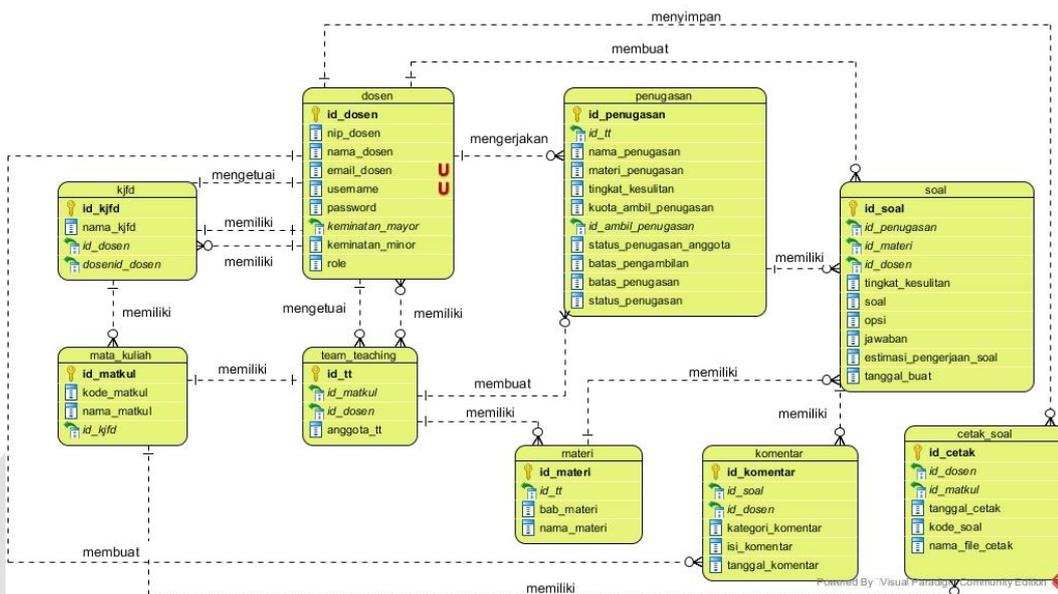
Tabel 5.3 Perancangan Komponen Method Buat Soal

1	get array soal, jawaban benar, opsi, estimasi waktu, tingkat kesulitan, id penugasan, index materi
2	For i=0 sampai i<panjang array soal
3	Set opsi_soal = opsi A,B,C,D indeks ke i
4	Serialize opsi_soal
5	Set data_insert
6	Input data_insert melalui method input_soal
7	Set pesan sukses
8	Endfor
9	Go to halaman detail penugasan



5.1.3 Perancangan Data

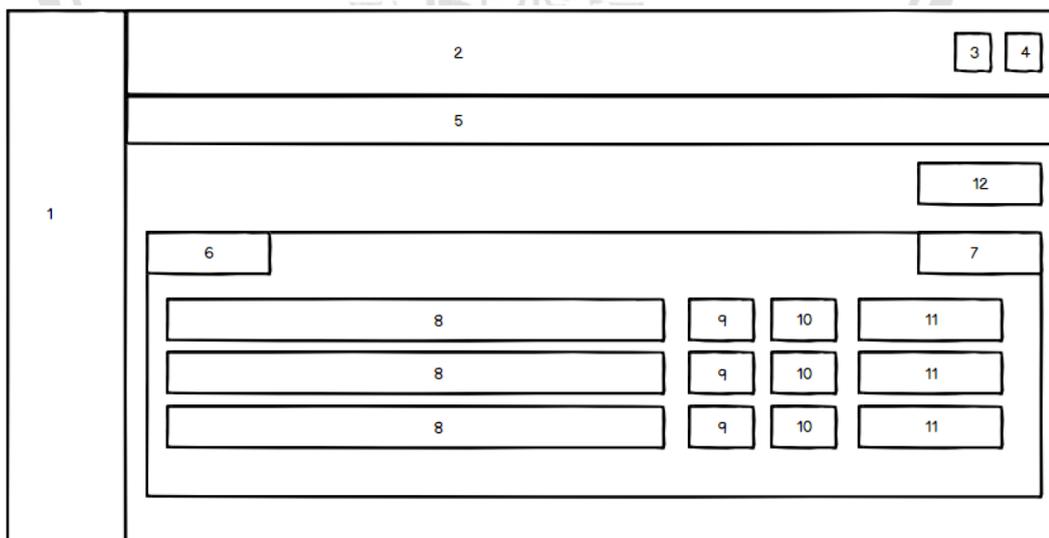
Pada penggunaannya sistem memerlukan basis data untuk mengolah data yang dimasukkan oleh penggunanya. Perancangan data dibutuhkan untuk merancang basis data yang akan dibuat sehingga pada tahap implementasi pengembang memiliki dasar dalam pembuatan basis data. Perancangan data pada sistem pengelolaan soal ini digambarkan pada Gambar 5.5 yang menggambarkan perancangan data menggunakan *Conceptual Data Model*.



Gambar 5.5 *Conceptual Data Model*

5.1.4 Perancangan Antarmuka

5.1.4.1 Perancangan Antarmuka Detail Penugasan



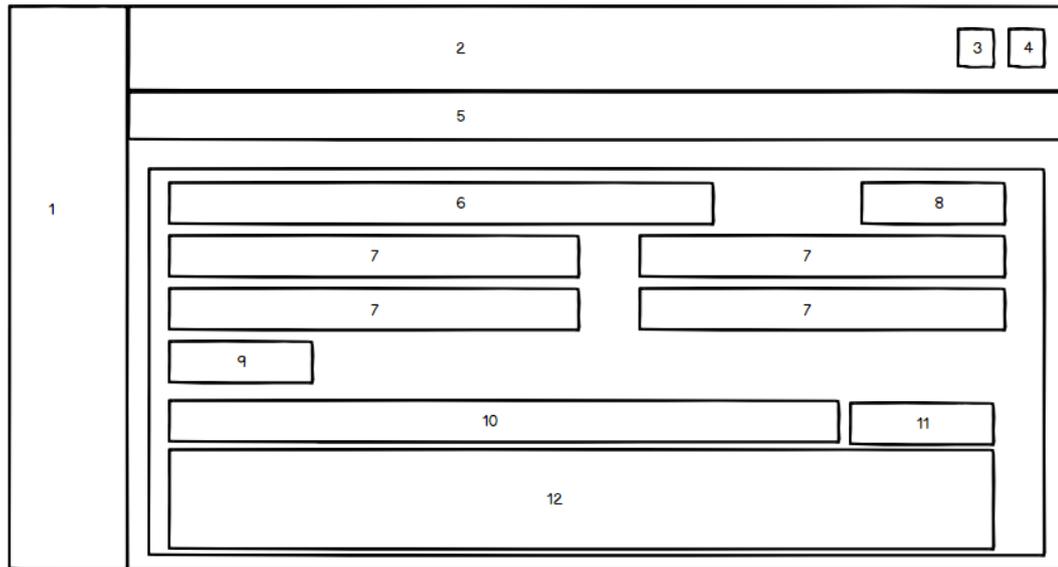
Gambar 5.6 Perancangan antarmuka detail penugasan

Pada Gambar 5.6 merupakan gambaran perancangan antarmuka detail penugasan. Informasi mengenai objek yang terlibat didalam perancangan antarmuka memilih materi terdapat pada Tabel 5.4.

Tabel 5.4 Penjelasan antarmuka detail penugasan

No	Nama Objek	Tipe	Keterangan
1	<i>Sidebar</i>	Navbar	Menampilkan menu yang dapat dipilih oleh pengguna
2	<i>Header</i>	Navbar	Sebagai <i>header</i> untuk informasi pengguna dan notifikasi
3	Notifikasi	<i>dropdown</i>	Untuk menampilkan notifikasi pada sistem
4	Informasi pengguna	<i>Image</i> dan <i>text</i>	Untuk menampilkan menu logout, dan menampilkan nama pengguna
5	Judul halaman	<i>Text</i>	Untuk menampilkan judul halaman yang terbuka
6	Judul penugasan	<i>Text</i>	Untuk menampilkan judul penugasan yang terbuka
7	Kuota memilih materi	<i>Text</i>	Untuk menampilkan kuota pengguna untuk memilih materi
8	Informasi materi penugasan	<i>Text</i>	Untuk menampilkan nama materi dan kebutuhan soal dalam masing-masing tingkat kesulitan
9	Informasi sisa kuota	<i>Text</i>	Untuk menampilkan sisa kuota dalam memilih materi
10	Informasi anggota materi	<i>Image</i>	Untuk menampilkan dosen yang telah memilih materi
11	Tombol memilih, batal memilih, kerjakan, dan <i>edit</i>	<i>Button</i>	Tombol untuk dosen gabung atau keluar dari materi yang dipilih.

5.1.4.2 Perancangan Antarmuka Evaluasi Penugasan



Gambar 5.7 Perancangan antarmuka evaluasi penugasan

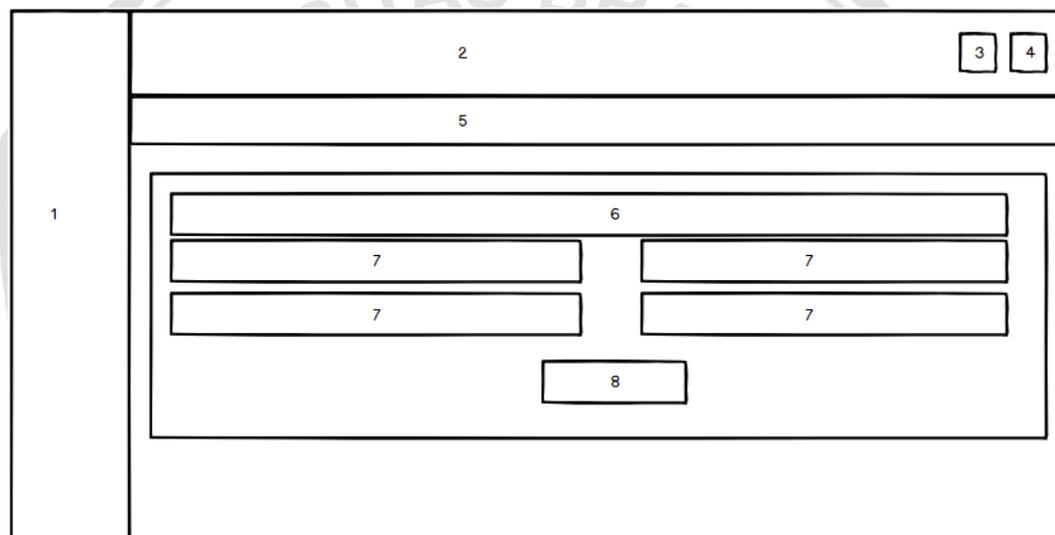
Pada Gambar 5.7 merupakan gambaran perancangan antarmuka evaluasi penugasan. Informasi mengenai objek yang terlibat didalam perancangan antarmuka evaluasi penugasan terdapat pada Tabel 5.5.

Tabel 5.5 Penjelasan antarmuka evaluasi penugasan

No	Nama Objek	Tipe	Keterangan
1	<i>Sidebar</i>	Navbar	Menampilkan menu yang dapat dipilih oleh pengguna
2	<i>Header</i>	Navbar	Sebagai <i>header</i> untuk informasi pengguna dan notifikasi
3	Notifikasi	<i>dropdown</i>	Untuk menampilkan notifikasi pada sistem
4	Informasi pengguna	<i>Image</i> dan <i>text</i>	Untuk menampilkan menu logout, dan menampilkan nama pengguna
5	Judul halaman	<i>Text</i>	Untuk menampilkan judul halaman yang terbuka
6	Soal	<i>Text, image</i>	Untuk menampilkan soal yang akan di evaluasi
7	Opsi jawaban	<i>Text, image</i>	Untuk menampilkan opsi jawaban dari soal

8	Tingkat kesulitan	<i>Text</i>	Untuk menampilkan tingkat kesulitan dari soal
9	Tombol komentar	<i>Button</i>	Untuk memicu munculnya <i>form</i> komentar dan komentar yang telah ada
10	<i>Form input</i> komentar	<i>Dropdown, input</i>	Untuk memberi komentar pada soal
11	Tombol tambah komentar	<i>Button</i>	Tombol untuk menyimpan komentar.
12	Daftar komentar	<i>List</i>	Untuk menampilkan komentar yang telah tersimpan

5.1.4.3 Perancangan Antarmuka Membuat Soal



Gambar 5.8 Perancangan antarmuka membuat soal

Pada Gambar 5.8 merupakan gambaran perancangan antarmuka membuat soal. Informasi mengenai objek yang terlibat didalam perancangan antarmuka membuat soal terdapat pada Tabel 5.6.

Tabel 5.6 Penjelasan antarmuka membuat soal

No	Nama Objek	Tipe	Keterangan
1	<i>Sidebar</i>	Navbar	Menampilkan menu yang dapat dipilih oleh pengguna
2	<i>Header</i>	Navbar	Sebagai <i>header</i> untuk informasi pengguna dan notifikasi

3	Notifikasi	<i>dropdown</i>	Untuk menampilkan notifikasi pada sistem
4	Informasi pengguna	<i>Image dan text</i>	Untuk menampilkan menu logout, dan menampilkan nama pengguna
5	Judul halaman	<i>Text</i>	Untuk menampilkan judul halaman yang terbuka
6	Soal	<i>input</i>	Untuk memasukkan soal
7	Opsi jawaban	<i>input</i>	Untuk memasukkan opsi jawaban dari soal
8	Estimasi waktu	<i>Text</i>	Untuk memasukkan estimasi waktu yang dibutuhkan untuk mengerjakan soal

5.2 Implementasi Sistem

Proses implementasi dilakukan setelah selesai proses perancangan. Implementasi dilakukan berdasarkan hasil yang diperoleh dari tahap analisis kebutuhan dan tahap perancangan. Setiap kebutuhan yang telah digali dan didapatkan akan diimplementasikan pada sistem. Implementasi harus memuat struktur dan kode program berdasarkan kelas-kelas dan algoritme yang sudah dibuat pada bagian perancangan. Pada bagian implementasi ini akan dijelaskan mengenai spesifikasi sistem, implementasi kode program, implementasi data, dan implementasi antarmuka.

5.2.1 Spesifikasi Sistem

Pada bagian ini akan dijelaskan mengenai spesifikasi yang digunakan untuk membangun sistem penentuan prioritas kebutuhan perangkat lunak dengan metode MVGB. Spesifikasi yang akan dijelaskan adalah spesifikasi perangkat keras, perangkat lunak, sistem operasi, dan arsitektur sistem. Pada Tabel 5.7 ditunjukkan spesifikasi perangkat keras yang digunakan oleh pengembang dalam mengembangkan sistem. Pada Tabel 5.8 ditunjukkan spesifikasi perangkat lunak yang digunakan pengembang dalam mengembangkan sistem. Pada Tabel 5.9 ditunjukkan sistem operasi yang digunakan oleh pengembang dalam mengembangkan sistem.

Tabel 5.7 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
Laptop	<i>Processor</i> Intel® Core™ i3-2370M <i>Memory</i> 8 GB <i>Graphic</i> Intel® HD Graphics 3000

Tabel 5.8 Spesifikasi Perangkat Lunak

Nama Komponen	Spesifikasi
Editor Dokumentasi	Microsoft Office Word 2017
Editor Perancangan	Visual Paradigm
Editor Pemrograman	Sublime Text Build 3175
Bahasa Pemrograman	PHP, Javascript, HTML, CSS
Framework	Codeigniter 3.1.8
DBMS	MySQL
Browser	Google Chrome

Tabel 5.9 Sistem Operasi

Nama Komponen	Spesifikasi
Sistem Operasi	Windows 10 Pro 64-bit (10.0, Build 17134)

5.2.2 Implementasi Kode Program

Implementasi kode program didasarkan pada perancangan komponen yang telah dilakukan sebelumnya. Pada bagian ini akan dirubah algoritme-algoritme yang sudah dibuat dalam bentuk pseudocode pada bagian perancangan komponen bentuk bahasa pemrograman. Algoritme yang akan dituliskan implementasinya yaitu memilih materi, mengirim *reminder*, dan membuat soal. Implementasi menggunakan bahasa pemrograman PHP dengan framework codeigneter versi 3.1.8.

5.2.2.1 Implementasi Komponen *Method* “proses_gabung_penugasan” class “C_penugasan”

Nama class: C_penugasan

Nama operasi: proses_gabung_penugasan(id,index)

Algoritme:

Tabel 5.10 Implementasi *Method* Gabung Penugasan

1	<code>\$penugasan = \$this->m_penugasan-</code>
2	<code>>get_penugasan(array("id_penugasan"=>\$id));</code>
3	<code>\$id_ambil_penugasan = unserialize(\$penugasan-</code>
4	<code>>id_ambil_penugasan);</code>
5	<code>\$id_dosen = \$_POST['id_dosen'];</code>
6	<code>if (!in_array(\$id_dosen, \$id_ambil_penugasan[\$index])) {</code>
7	<code> \$key = array_search(0, \$id_ambil_penugasan[\$index]);</code>
8	<code> if (\$key>=0) {</code>
9	<code> \$id_ambil_penugasan[\$index][\$key] = \$id_dosen;</code>
10	<code> \$ser_id_ambil = serialize(\$id_ambil_penugasan);</code>
11	<code> \$data_update = array(</code>

```

12         'id_ambil_penugasan' => $ser_id_ambil,);
13         $where = array('id_penugasan'=>$id);
14         $res = $this->m_penugasan-
15 >update_penugasan($data_update,$where);
16         $this->session->set_flashdata('gabung_msg',
17 'success');
18         }else{
19         $this->session->set_flashdata('gabung_msg', 'failed');
20         }
21     }else{
22     $this->session->set_flashdata('gabung_msg', 'already');
23     }
24     redirect(base_url()."detail-penugasan/".$id);

```

5.2.2.2 Implementasi Komponen Method “*irim_reminder*” class “C_penugasan”

Nama class: C_penugasan

Nama operasi: proses_kirim_reminder(id_penugasan)

Algoritme:

Tabel 5.11 Implementasi Method Kirim Reminder

```

1  $where = array('id_penugasan'=>$id_penugasan);
2  $penugasan = $this->m_penugasan->get_row_penugasan($where);
3  $dosen = unserialize($penugasan->id_ambil_penugasan);
4  $materi = unserialize($penugasan->materi_penugasan);
5  $soal = unserialize($penugasan->tingkat_kesulitan);
6  $dosen_penugasan = $this->get_dosen_penugasan($dosen);
7  $notifikasi = $this-
8  >get_pesan_notifikasi($penugasan,$dosen_penugasan);
9  $count_email = 0;
10 for ($i=0; $i < count($dosen_penugasan); $i++){
11     if (isset($notifikasi[$i])) {
12     $where = array('id_dosen'=>$dosen_penugasan[$i];
13     $data_dosen_penugasan = $this->m_dosen->get_dosen($where);
14     $this->email_send($data_dosen_penugasan-
15 >email_dosen,$notifikasi[$i]);
16     $count_email++;}}
17 $this->session->set_flashdata('kirim_msg', "success");
18 $this->session->set_flashdata('hitung_msg', $count_email);
19 redirect(base_url()."progres-penugasan/".$id_penugasan);

```

5.2.2.3 Implementasi Komponen Method “*proses_buat_soal*” class “C_penugasan”

Nama class: C_penugasan

Nama operasi: proses_buat_soal(id,index)

Algoritme:

Tabel 5.12 Implementasi Method Buat Soal

```

1  $soal = $_POST['soal'];
2  $pgBenar = $_POST['pgBenar'];
3  $pgA = $_POST['pgA'];
4  $pgB = $_POST['pgB'];

```



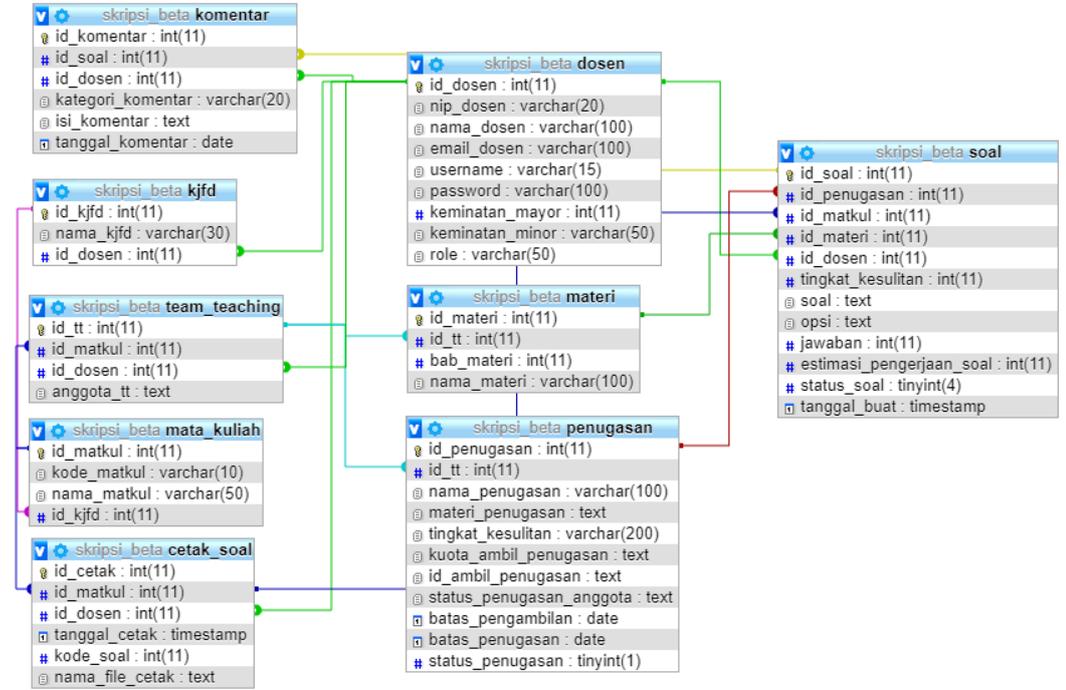
```

5  $pgC = $_POST['pgC'];
6  $pgD = $_POST['pgD'];
7  $estimasi_waktu = $_POST['estimasi_waktu'];
8  $tingkat_kesulitan = $_POST['tingkat_kesulitan'];
9  $id_penugasan = $_POST['id_penugasan'];
10 $index_materi = $_POST['index_materi'];
11 for ($i=0; $i < count($soal) ; $i++) {
12     $opsi = array($pgA[$i], $pgB[$i], $pgC[$i], $pgD[$i]);
13     $ser_opsi = serialize($opsi);
14     $data_insert = array(
15         'id_penugasan' => $id_penugasan,
16         'index_materi' => $index_materi,
17         'id_dosen' => $this->session->userdata('id'),
18         'tingkat_kesulitan' => $tingkat_kesulitan[$i],
19         'soal' => $soal[$i],
20         'opsi' => $ser_opsi,
21         'jawaban' => $pgBenar[$i],
22         'estimasi_pengerjaan_soal' => $estimasi_waktu[$i]);
23     $res = $this->model->insert_data('soal', $data_insert);
24     $this->session->set_flashdata('input_msg', 'success');
25 }
26 redirect(base_url('detail-penugasan/'.$id_penugasan));

```

5.2.3 Implementasi Basis Data

Implementasi data dibuat berdasarkan perancangan data yang telah dibuat *Conceptual Data Model* (CDM). Implementasi basis data disajikan dalam diagram *Physical Data Model* (PDM) pada Gambar 5.9 yang menggambarkan semua tabel dalam basis data dalam sistem berikut dengan kolom dan tipe data untuk masing-masing kolomnya.



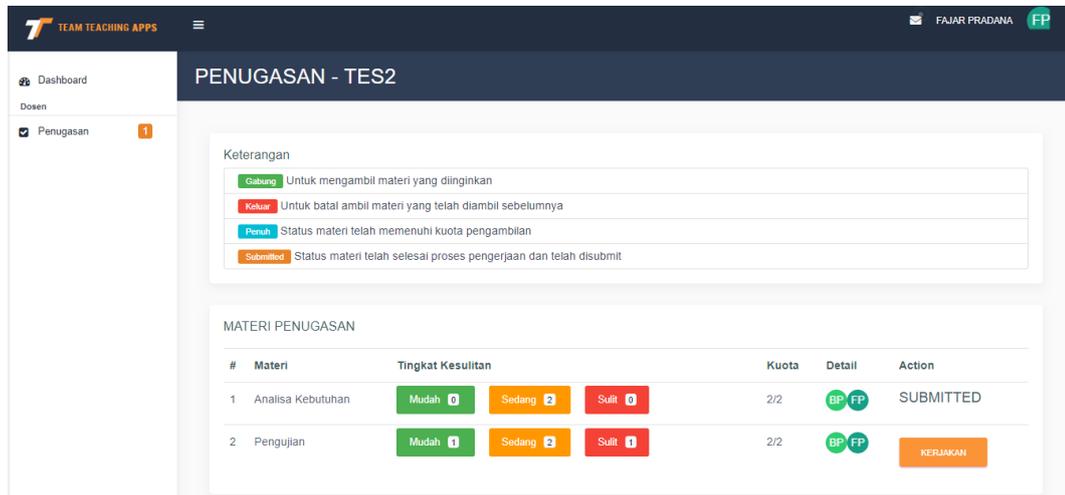
Gambar 5.9 Implementasi basis data



5.2.4 Implementasi Antarmuka

Pada bagian ini akan dipaparkan antarmuka hasil implementasi antarmuka sistem berdasarkan rancangan yang telah dibuat pada bagian perancangan antarmuka.

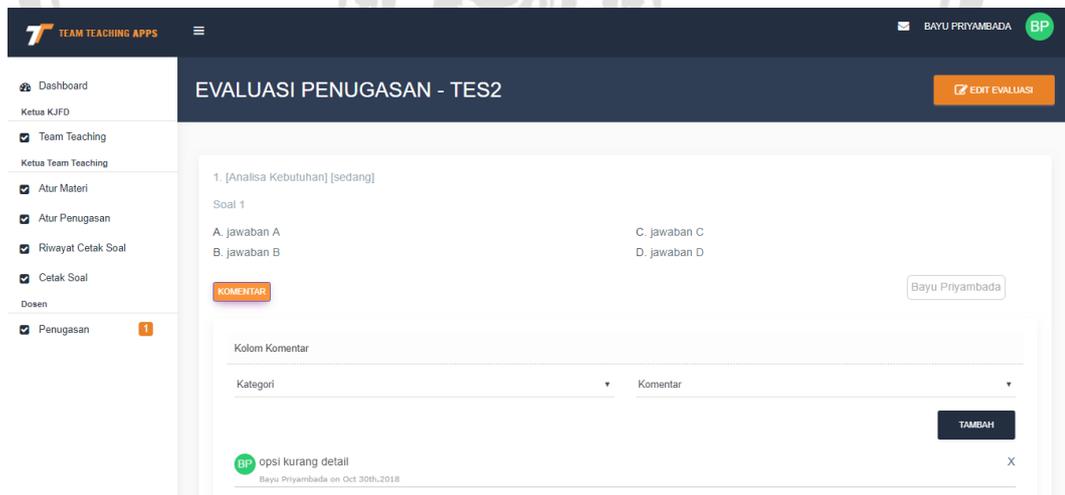
5.2.4.1 Implementasi Antarmuka Detail penugasan



Gambar 5.10 Implementasi antarmuka detail penugasan

Pada implementasi antarmuka detail penugasan Gambar 5.10, pengguna dapat melihat detail penugasan berupa materi, kebutuhan soal beserta tingkat kesulitannya, kuota per materi, dan dosen yang telah bergabung pada materi.

5.2.4.2 Implementasi Antarmuka Evaluasi Penugasan

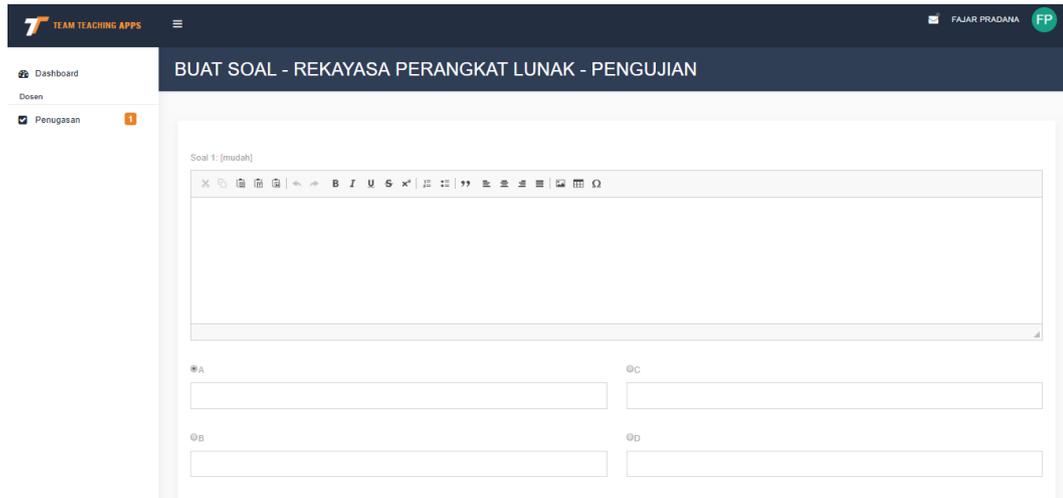


Gambar 5.11 Implementasi antarmuka evaluasi penugasan

Pada implementasi antarmuka evaluasi penugasan Gambar 5.11, pengguna dapat melihat soal-soal yang akan dievaluasi, kolom komentar dan komentar-komentar yang telah diberikan.



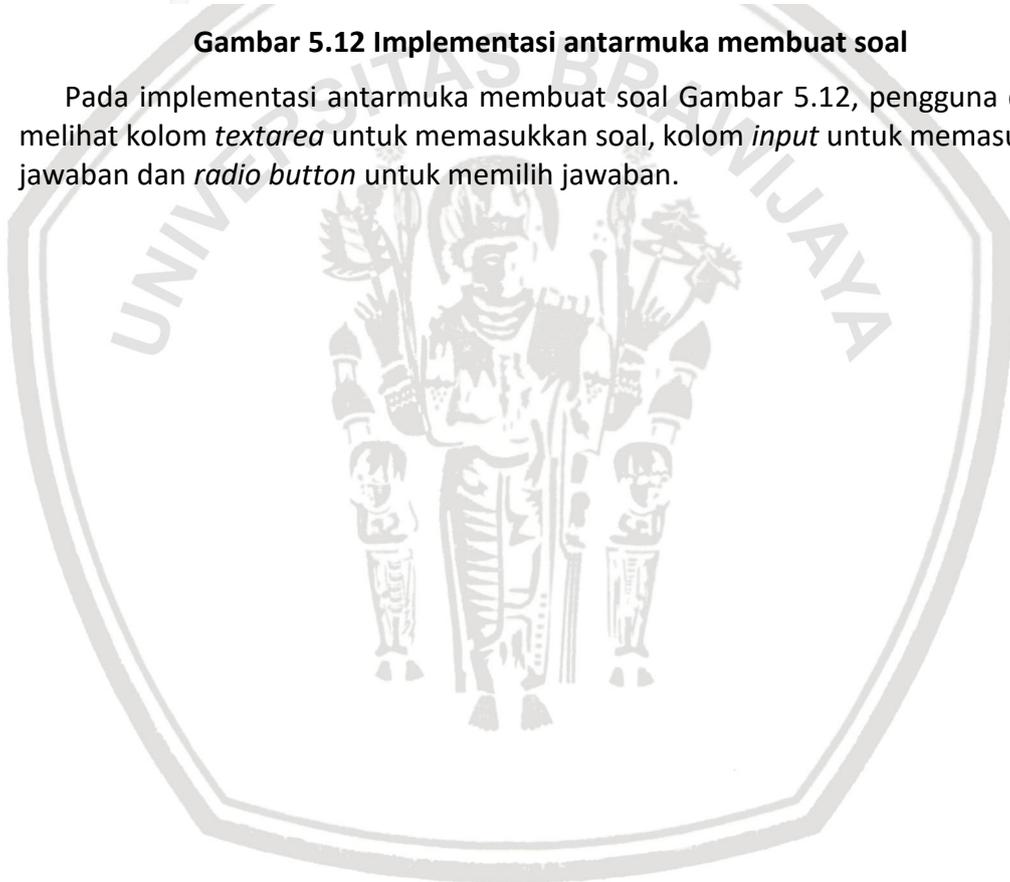
5.2.4.3 Implementasi Antarmuka Membuat Soal



The screenshot displays the 'BUAT SOAL' (Create Question) interface within the 'TEAM TEACHING APPS'. The top navigation bar includes the app logo, a menu icon, and the user name 'FAJAR PRADANA' with a profile icon. The sidebar on the left shows 'Dashboard', 'Dosen', and 'Penugasan' (with a notification badge). The main content area is titled 'BUAT SOAL - REKAYASA PERANGKAT LUNAK - PENGUJIAN'. It features a rich text editor for the question text, followed by four radio button options labeled 'A', 'B', 'C', and 'D', each with an associated input field for the answer.

Gambar 5.12 Implementasi antarmuka membuat soal

Pada implementasi antarmuka membuat soal Gambar 5.12, pengguna dapat melihat kolom *textarea* untuk memasukkan soal, kolom *input* untuk memasukkan jawaban dan *radio button* untuk memilih jawaban.



BAB 6 PENGUJIAN

Tahap pengujian dilakukan setelah melakukan implementasi sistem. Pengujian bertujuan untuk memeriksa apakah implementasi sudah sesuai dengan analisis kebutuhan dan perancangan sistem yang dilakukan. Tahap pengujian yang akan dilakukan adalah pengujian unit, pengujian integrasi, pengujian validasi, dan pengujian *compatibility*.

6.1 Pengujian Unit

Pengujian unit ini berkonsentrasi pada setiap unit (misalnya: komponen, kelas, atau objek) dari perangkat lunak yang diterapkan. Pada pengujian unit ini menggunakan metode *whitebox testing* dengan teknik pengujian *basis path*. *Test case* didapat digunakan untuk mengerjakan basis set yang menjamin pengerjaan setiap perintah minimal satu kali selama uji coba. Langkah-langkah dalam pembuatan *test case* pada *basis path testing* adalah sebagai berikut:

1. Pembuatan diagram alir dari algoritme sistem yang ingin diuji
2. Menentukan *cyclomatic complexity* dari diagram alir
3. Menentukan *independent path* dari diagram alir

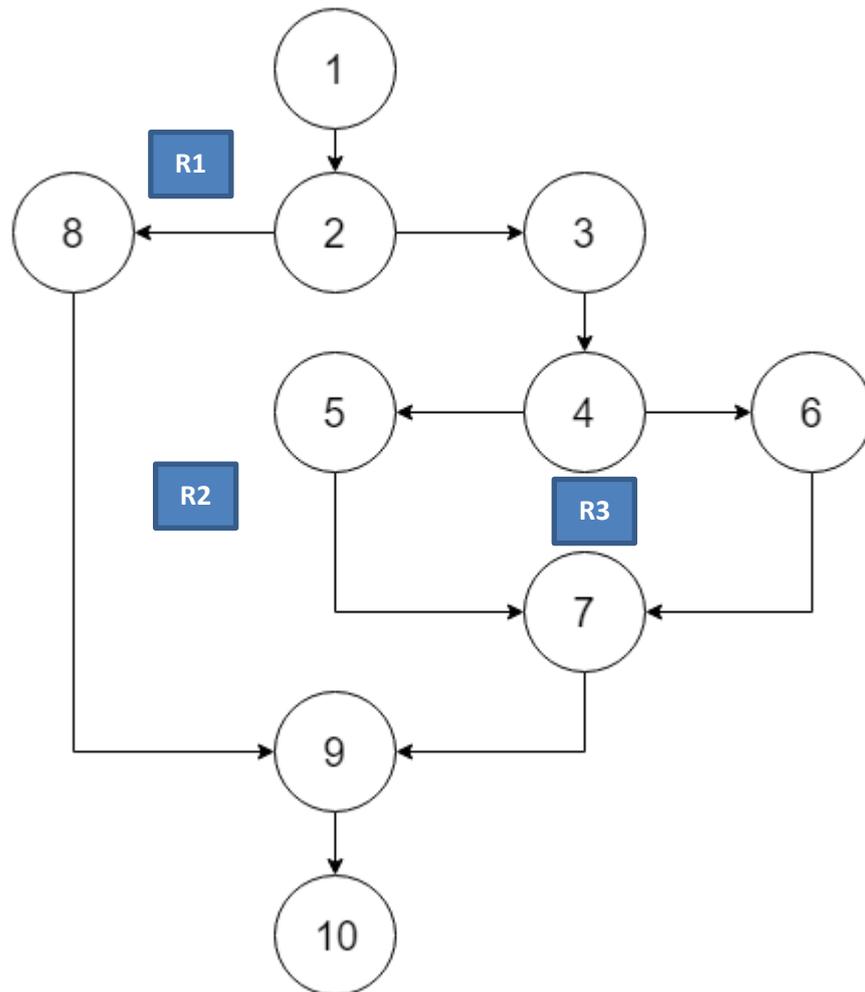
6.1.2 Pengujian Unit *Method* “proses_gabung_penugasan” class “C_penugasan”

1. *Pseudocode*

1	Set variabel penugasan = data dari tabel penugasan dimana id_penugasan = id
1	Set variabel array id_ambil_penugasan = unserialize dari kolom id_ambil_penugasan dari variabel penugasan
1	Set variabel id_dosen = id
2	If id_dosen ada pada array id_ambil_penugasan[index]
3	Set variabel key = index dari array id_ambil_penugasan[index] yang bernilai 0
4	If key >= 0
5	id_ambil_penugasan[index][i] = id_dosen
5	Set ser_id_ambil = serialize dari array id_ambil_penugasan
5	Deklarasi data_update = kolom id_ambil_penugasan set ser_id_ambil
5	Deklarasi where = kolom id_penugasan find id
5	Variabel res= update data berdasarkan data_update dan where
5	Set pesan sukses
6	Else
6	Set pesan gagal
7	Endif
8	Else
8	Set pesan sudah terdaftar
9	Endif
10	Go to halaman detail penugasan

2. Basis Path Testing

2.1 Flow Graph



Gambar 6.1 Flow graph method “proses_gabung_penugasan” class “C_penugasan”

2.2 Cyclomatic Complexity

- $V(G) = \text{Jumlah region} = 3$, ada 5 region R1, R2, R3
- $V(G) = 11 \text{ edges} - 10 \text{ nodes} + 2 = 3$
- $V(G) = 2 \text{ predicate nodes} + 1 = 3$

2.3 Independent Path

- Jalur 1: 1-2-8-9-10
- Jalur 2: 1-2-3-4-5-7-9-10
- Jalur 3: 1-2-3-4-6-7-9-10

Test case dan hasil akan dijelaskan pada Tabel 6.1.

Tabel 6.1 Hasil pengujian unit method “proses_gabung_penugasan” class “C_penugasan”

No	No. Jalur	Prosedur Uji	Expected Result	Result	Status
1	1	Ketika id_dosen ada pada array id_ambil_penugasan	Gagal bergabung, dan menampilkan halaman detail penugasan dengan pesan “sudah tergabung pada materi”	Gagal bergabung, dan menampilkan halaman detail penugasan dengan pesan “sudah tergabung pada materi”	Valid
2.	2	Ketika terdapat nilai 0 pada array id_ambil_penugasan	Berhasil bergabung pada materi dan menampilkan halaman detail penugasan dengan pesan “sukses bergabung”	Berhasil bergabung pada materi dan menampilkan halaman detail penugasan dengan pesan “sukses bergabung”	Valid
3.	3	Ketika tidak terdapat nilai 0 pada array id_ambil_penugasan	Gagal bergabung materi dan menampilkan halaman detail penugasan dengan pesan “gagal bergabung”	Gagal bergabung materi dan menampilkan halaman detail penugasan dengan pesan “gagal bergabung”	Valid

6.1.3 Pengujian Unit Method “kirim_reminder” class “C_penugasan”

1. *Pseudocode*

```

1 Set variabel where id_penugasan = id
1 Set variabel penugasan = data dari tabel penugasan berdasarkan variabel where
1 Set variabel array dosen = unserialize dari kolom id_ambil_penugasan dari variabel penugasan
1 Set variabel array materi = unserialize dari kolom materi_penugasan dari variabel penugasan
1 Set variabel array soal = unserialize dari kolom tingkat_kesulitan dari variabel penugasan

1 Set variabel array dosen_penugasan = nilai return dari method get_dosen_penugasan(dosen)
1 Set variabel array notifikasi = nilai return dari method get_pesan_notifikasi(penugasan, dosen_penugasan)

1 Set variabel count_email = 0

2 For i=0 sampai i<panjang array dosen_penugasan
3 If notifikasi indeks ke I tersedia
4     Set variabel where id_dosen = dosen_penugasan[$i]
4     Set data_dosen_penugasan = ambil data dosen berdasarkan variabel where pada database
    
```



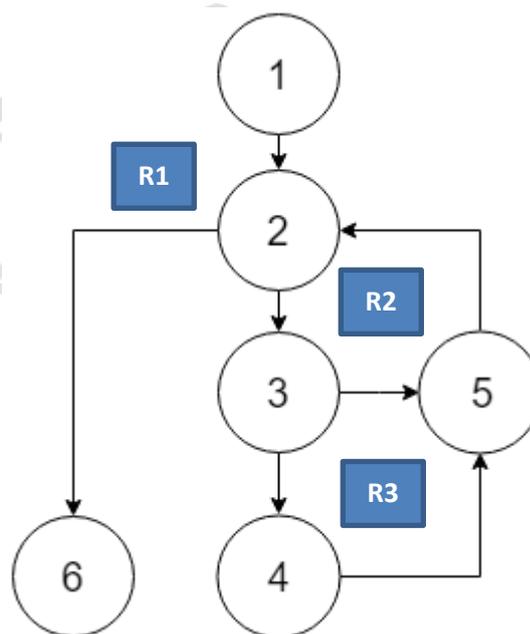
```

4     Mengirim email dengan method email_send(email dosen,
      notifikasi[i]
4     Count_email tambah 1
4     Endif
5 Endfor

6 Set pesan hitung_msg = count_email
6 Go to halaman detail penugasan
  
```

2. Basis Path Testing

2.4 Flow Graph



Gambar 6.2 Flow graph method “*kirim_reminder*” class “*C_penugasan*”

2.5 Cyclomatic Complexity

- $V(G) = \text{Jumlah region} = 3$, ada 3 region R1, R2, R3
- $V(G) = 7 \text{ edges} - 6 \text{ nodes} + 2 = 3$
- $V(G) = 2 \text{ predicate nodes} + 1 = 3$

2.6 Independent Path

- Jalur 1: 1-2-6
- Jalur 2: 1-2-3-5-2-6
- Jalur 3: 1-2-3-4-5-2-6

Test case dan hasil akan dijelaskan pada tabel 6.2 dibawah ini.

Tabel 6.2 Hasil pengujian unit method “ *kirim_reminder* ” class “ *C_penugasan* ”

No	No. Jalur	Prosedur Uji	Expected Result	Result	Status
1.	1	Ketika panjang array dosen_penugasan = 0	Gagal mengirim <i>email</i> , menampilkan halaman progres penugasan dengan pesan “tidak ada email yang dikirim”	Gagal mengirim <i>email</i> , menampilkan halaman progres penugasan dengan pesan “tidak ada email yang dikirim”	Valid
2.	2	Ketika variabel notifikasi pada pada indeks ke i tidak tersedia dengan kondisi panjang array dosen_penugasan > 0	Gagal mengirim <i>email</i> , menampilkan halaman progres penugasan dengan pesan “tidak ada email yang dikirim”	Gagal mengirim <i>email</i> , menampilkan halaman progres penugasan dengan pesan “tidak ada email yang dikirim”	Valid
3.	3	Ketika variabel notifikasi pada pada indeks ke i tersedia dengan kondisi panjang array dosen_penugasan > 0	Berhasil mengirim email, menampilkan halaman progres penugasan dengan pesan “Sukses mengirim email”	Berhasil mengirim email, menampilkan halaman progres penugasan dengan pesan “Sukses mengirim email”	Valid

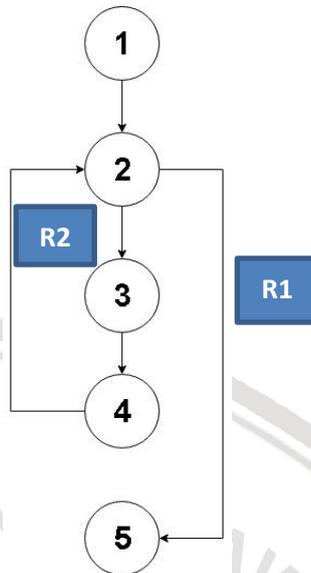
6.1.4 Pengujian Unit Method “ *proses_buat_soal* ” class “ *C_penugasan* ”

1. Pseudocode

1	get array soal, jawaban benar, opsi, estimasi waktu, tingkat kesulitan, id penugasan, index materi
2	For i=0 sampai i<panjang array soal
3	Set opsi_soal = opsi A,B,C,D
3	Serialize opsi soal
3	Set data_insert
3	Input data_insert
3	Set pesan sukses
4	Endfor
5	Go to halaman detail penugasan

2. Basis Path Testing

2.7 Flow Graph



Gambar 6.3 Flow graph method “proses_buat_soal” class “C_penugasan”

2.8 Cyclomatic Complexity

- $V(G) = 2$, ada 2 region R1, R2
- $V(G) = 5 \text{ edges} - 5 \text{ nodes} + 2 = 2$
- $V(G) = 1 \text{ predicate nodes} + 1 = 2$

2.9 Independent Path

- Jalur 1: 1-2-5
- Jalur 2: 1-2-3-4-2-5

Test case dan hasil akan dijelaskan pada tabel 6.3 dibawah ini.

Tabel 6.3 Hasil pengujian unit method “proses_buat_soal” class “C_penugasan”

No	No. Jalur	Prosedur Uji	Expected Result	Result	Status
1	1	Set panjang array soal = 0	Tidak ada soal yang ditambah	Tidak ada soal yang ditambah	valid
2	2	Ketika panjang array soal, jawaban benar, opsi A, opsi B, opsi C, opsi D, tingkat kesulitan, estimasi waktu != 0 dengan id penugasan dan id materi yang telah di set	Berhasil membuat soal dan menampilkan halaman detail penugasan dengan pesan “berhasil membuat soal”	Berhasil membuat soal dan menampilkan halaman detail penugasan dengan pesan “berhasil membuat soal”	valid



6.2 Pengujian Integrasi

Pada pengujian integrasi berfokus dalam menguji integrasi antar unit-unit yang ada, sehingga dapat berjalan dengan sesuai. Pada Tabel 6.4 ditunjukkan identifikasi dan rencana uji yang akan dilakukan. Hasil dari pengujian integrasi ditunjukkan pada Tabel 6.5 sampai Tabel 6.7.

Tabel 6.4 Identifikasi dan Perancangan Pengujian Integrasi

No	Nama Kelas	Nama <i>Method</i>	Tujuan
1	C_penugasan	proses_gabung_penugasan(id_penugasan, index_materi)	Bergabung pada materi penugasan yang ingin dikerjakan
	M_penugasan	get_penugasan(id_penugasan)	
	M_penugasan	update_penugasan(data_update, id_penugasan)	
2	C_penugasan	proses_kirim_reminder(id_penugasan)	Mengirim reminder kepada anggota penugasan
	C_penugasan	get_dosen_penugasan(dosen)	
	C_penugasan	get_pesanan_notifikasi(penugasan, dosen_penugasan)	
	M_dosen	get_dosen(where)	
	C_penugasan	email_send(email_dosen, notifikasi)	
3	C_penugasan	proses_buat_soal()	Membuat soal untuk penugasan
	M_soal	Input_soal(data_insert)	

Tabel 6.5 Hasil Pengujian Integrasi Nomor 1

No Uji	1
Input Pertama	"id_penugasan": "23" "index_materi": "1"
<i>Method</i> dari kelas C_penugasan	proses_gabung_penugasan(id_penugasan, index_materi)
<i>Output</i> Pertama / Input Kedua	"gabung_msg": "success"
<i>Method</i> dari kelas M_penugasan	get_penugasan(id_penugasan)

<i>Output</i> Kedua / Input Ketiga	"id_penugasan": 23 "id_tt": 16 "nama_penugasan": tes "materi_penugasan": a:2:{i:0;s:1:"7";j:1;s:1:"8";} "tingkat_kesulitan": a:2:{i:0;a:3:{i:0;s:1:"1";j:1;s:1:"2";j:2;s:1:"0"};j:1;a:3:{i:0;s:1:"0";j:1;s:1:"2";j:2;s:1:"0"};}} "kuota_ambil_penugasan": a:2:{i:0;s:1:"1";j:1;s:1:"1";} "id_ambil_penugasan": a:2:{i:0;a:1:{i:0;s:1:"9"};j:1;a:1:{i:0;i:0}} "status_penugasan_anggota": a:2:{i:0;a:1:{i:0;i:1};j:1;a:1:{i:0;i:0}} "batas_pengambilan": 2018-11-12 "batas_penugasan": 2018-11-14 "status_penugasan": 0
<i>Method</i> dari kelas M_penugasan	update_penugasan(data_update, id_penugasan)
<i>Expected Result</i>	Memperbarui anggota materi pada <i>database</i>
<i>Result</i>	Memperbarui anggota materi pada <i>database</i>
Status	Valid

Tabel 6.6 Hasil Pengujian Integrasi Nomor 2

No Uji	2
Input Pertama	"id_penugasan": "23"
<i>Method</i> dari kelas C_penugasan	proses_kirim_reminder(id_penugasan)
<i>Output</i> Pertama / Input Kedua	"kirim_msg": "success" "hitung_msg": "1"
<i>Method</i> dari kelas C_penugasan	get_dosen_penugasan(dosen)
<i>Output</i> Kedua / Input Ketiga	"dosen_penugasan": Array ([0] => 9)

<p><i>Method</i> dari kelas C_penugasan</p>	<p>get_pesanan_notifikasi(penugasan, dosen_penugasan)</p>
<p><i>Output</i> Ketiga / <i>Input</i> Keempat</p>	<pre> "notifikasi": array([0] => "<table cellpadding="0" cellspacing="0" style="background-image: url(http://binamulia.org/assets/images/works/banner.jpg);"> <tbody><tr> <td class="pattern" width="450"> <table cellpadding="0" cellspacing="0"> <tbody><tr> <td class="hero"> </td> </tr> <tr> <td align="center" style="font-family: arial,sans-serif; color: #333;"> <h1>Reminder!
</h1> </td> </tr> <tr> <td align="left" style="font-weight: normal; line-height: 20px !important; color: #666; padding-bottom: 20px;text-align: justify;font-family: monospace;color:#000;font-size: 12px;"> <singleline>Halo baru banget, mengingatkan tentang penugasan tes mata kuliah baru pada Team Teaching Apps, anda memiliki tanggungan berupa:

 </singleline> <singleline> baru2: 2 soal
 </singleline> <singleline>Dimohon untuk segera menyelesaikan tanggungan sebelum 2018-11-14.</singleline> </td> </tr> <tr> <td align="left"> </td> </tr> </tbody></table> </td> </tr></tbody></table>") </pre>
<p><i>Method</i> dari kelas M_penugasan</p>	<p>get_dosen(where)</p>
<p><i>Output</i> Kedua / <i>Input</i> Ketiga</p>	<pre> "id_dosen": 9 "nip_dosen": 123 "nama_dosen": baru banget "email_dosen": baru@gmail.com "username": baru "password": 202cb962ac59075b964b07152d234b70 "keminatan_mayor": 1 "keminatan_minor": a:2:{i:0;s:1:"4";i:1;s:1:"5"}; </pre>



	"role": a:3:{i:0;i:4;i:1;i:2;i:2;i:3;}
Method dari kelas C_penugasan	email_send(email_dosen,notifikasi)
Expected Result	Mengirim email kepada anggota penugasan
Result	Mengirim email kepada anggota penugasan
Status	Valid

Tabel 6.7 Hasil Pengujian Integrasi Nomor 3

No Uji	3
Input Pertama	<p>"soal": "Array ([0] => soal1 [1] => Soal2 [2] => Soal3 [3] => Soal4)"</p> <p>"pgBenar": "Array ([0] => 0 [1] => 0 [2] => 4 [3] => 2)"</p> <p>"pgA": "Array ([0] => A1 [1] => A2 [2] => A3 [3] => A4)"</p> <p>"pgB": "Array ([0] => B1 [1] => B2 [2] => B3 [3] => B4)"</p> <p>"pgC": "Array ([0] => C1 [1] => C2 [2] => C3 [3] => C4)"</p> <p>"pgD": "Array ([0] => D1 [1] => D2 [2] => D3 [3] => D4)"</p> <p>"estimasi_waktu": "Array ([0] => 60 [1] => 90 [2] => 60 [3] => 120)"</p> <p>"tingkat_kesulitan": "Array ([0] => 0 [1] => 0 [2] => 1 [3] => 2)"</p> <p>"id_penugasan": "7"</p> <p>"index_materi": "1"</p>
Method dari kelas C_penugasan	proses_buat_soal()
Output Pertama / Input Kedua	"input_msg": "success"
Method dari kelas M_penugasan	Input_soal(data_insert)
Expected Result	Menambah soal penugasan pada <i>database</i>
Result	Menambah soal penugasan pada <i>database</i>
Status	Valid

6.3 Pengujian Validasi

Pengujian validasi dilakukan ketika kebutuhan yang sudah ditetapkan sebagai bagian dari pemodelan kebutuhan divalidasi terhadap perangkat lunak yang telah dibangun. Pengujian validasi dikatakan berhasil apabila semua kebutuhan yang ada telah diimplementasi dengan baik dan sesuai skenario. Pada pengujian validasi ini digunakan metode *blackbox testing*. Dengan *blackbox testing* memungkinkan pembuat perangkat lunak untuk memperoleh kondisi yang terjadi untuk suatu masukan yang akan menjalankan semua kebutuhan. Pengujian validasi akan ditunjukkan pada Tabel 6.8 sampai Tabel 6.82.

Tabel 6.8 Hasil Pengujian Validasi Register

Kode Kebutuhan	SRS-TT-F-1
Nama Kasus Uji	<i>Register</i>
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman login 2. Menekan menu <i>register</i> 3. Mengisi <i>form register</i> dengan nama, <i>e-mail</i> yang belum digunakan, NIP, <i>username</i> yang belum digunakan dan <i>password</i>, kemudian tekan tombol <i>register</i>
Hasil yang diharapkan	Sistem menampilkan halaman login dan pesan sukses mendaftar
Hasil	Sistem menampilkan halaman login dan pesan sukses mendaftar
Status	Valid

Tabel 6.9 Hasil Pengujian Validasi Register Alternatif 1

Kode Kebutuhan	SRS-TT-F-1
Nama Kasus Uji	<i>Register</i>
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman login 2. Menekan menu <i>register</i> 3. Membiarkan terdapat <i>field</i> yang kosong, kemudian tekan tombol <i>register</i>
Hasil yang diharapkan	Sistem menampilkan pesan untuk melengkapi pengisian form.
Hasil	Sistem menampilkan pesan untuk melengkapi pengisian form.
Status	Valid

Tabel 6.10 Hasil Pengujian Validasi *Register* Alternatif 2

Kode Kebutuhan	SRS-TT-F-1
Nama Kasus Uji	<i>Register</i>
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman login 2. Menekan menu <i>register</i> 3. Mengisi <i>form register</i> dengan nama, <i>e-mail</i> yang telah digunakan, NIP, <i>username</i> yang belum digunakan dan <i>password</i>, kemudian tekan tombol <i>register</i>
Hasil yang diharapkan	Sistem menampilkan halaman login dan pesan gagal mendaftar karena <i>e-mail</i> telah digunakan.
Hasil	Sistem menampilkan halaman login dan pesan gagal mendaftar karena <i>e-mail</i> telah digunakan.
Status	Valid

Tabel 6.11 Hasil Pengujian Validasi *Register* Alternatif 3

Kode Kebutuhan	SRS-TT-F-1
Nama Kasus Uji	<i>Register</i>
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman login 2. Menekan menu <i>register</i> 3. Mengisi <i>form register</i> dengan nama, <i>e-mail</i> yang belum digunakan, NIP, <i>username</i> yang telah digunakan dan <i>password</i>, kemudian tekan tombol <i>register</i>
Hasil yang diharapkan	Sistem menampilkan halaman login dan pesan gagal mendaftar karena <i>username</i> telah digunakan.
Hasil	Sistem menampilkan halaman login dan pesan gagal mendaftar karena <i>username</i> telah digunakan.
Status	Valid

Tabel 6.12 Hasil Pengujian Validasi *Login*

Kode Kebutuhan	SRS-TT-F-2
Nama Kasus Uji	<i>Login</i>
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman login

	2. Mengisi <i>form login</i> dengan <i>username</i> dan <i>password</i> yang terdaftar dengan kondisi akun dosen telah diaktifkan
Hasil yang diharapkan	Sistem menampilkan halaman utama.
Hasil	Sistem menampilkan halaman utama.
Status	Valid

Tabel 6.13 Hasil Pengujian Validasi *Login* Alternatif 1

Kode Kebutuhan	SRS-TT-F-2
Nama Kasus Uji	<i>Login</i>
Prosedur	1. Membuka halaman login 2. Membiarkan terdapat field yang kosong, kemudian tekan tombol sign in
Hasil yang diharapkan	Sistem menampilkan pesan untuk melengkapi pengisian form.
Hasil	Sistem menampilkan pesan untuk melengkapi pengisian form.
Status	Valid

Tabel 6.14 Hasil Pengujian Validasi *Login* Alternatif 2

Kode Kebutuhan	SRS-TT-F-2
Nama Kasus Uji	<i>Login</i>
Prosedur	1. Membuka halaman login 2. Mengisi <i>form login</i> dengan <i>username</i> dan <i>password</i> yang terdaftar dengan kondisi akun dosen belum diaktifkan
Hasil yang diharapkan	Sistem menampilkan halaman login dan pesan gagal <i>login</i> karena akun dosen belum diaktifkan.
Hasil	Sistem menampilkan halaman login dan pesan gagal <i>login</i> karena akun dosen belum diaktifkan.
Status	Valid

Tabel 6.15 Hasil Pengujian Validasi *Login* Alternatif 3

Kode Kebutuhan	SRS-TT-F-2
Nama Kasus Uji	<i>Login</i>
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman login 2. Mengisi <i>form login</i> dengan <i>username</i> yang tidak terdaftar atau <i>password</i> yang salah
Hasil yang diharapkan	Sistem menampilkan halaman login dan pesan gagal <i>login</i> karena <i>username</i> atau <i>password</i> salah.
Hasil	Sistem menampilkan halaman login dan pesan gagal <i>login</i> karena <i>username</i> atau <i>password</i> salah.
Status	Valid

Tabel 6.16 Hasil Pengujian Validasi *Logout*

Kode Kebutuhan	SRS-TT-F-3
Nama Kasus Uji	<i>Logout</i>
Prosedur	Memilih menu " <i>logout</i> " pada menu <i>profile</i>
Hasil yang diharapkan	Sistem menampilkan halaman <i>login</i> .
Hasil	Sistem menampilkan halaman <i>login</i> .
Status	Valid

Tabel 6.17 Hasil Pengujian Validasi Melihat Dosen

Kode Kebutuhan	SRS-TT-F-4
Nama Kasus Uji	Melihat Dosen
Prosedur	Membuka halaman "pengaturan dosen"
Hasil yang diharapkan	Sistem menampilkan daftar dosen.
Hasil	Sistem menampilkan daftar dosen.
Status	Valid

Tabel 6.18 Hasil Pengujian Validasi Aktivasi Dosen

Kode Kebutuhan	SRS-TT-F-5
Nama Kasus Uji	Aktivasi dosen
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman "pengaturan dosen"

	<ol style="list-style-type: none"> 2. Menekan tombol “aktifkan” pada akun dosen yang ingin diaktifkan 3. Memilih keminatan mayor dan minor dari dosen yang akan diaktivasi 4. Menekan tombol aktifkan
Hasil yang diharapkan	Sistem menampilkan pesan sukses melakukan aktivasi.
Hasil	Sistem menampilkan pesan sukses melakukan aktivasi.
Status	Valid

Tabel 6.19 Hasil Pengujian Validasi Mengubah Data Dosen

Kode Kebutuhan	SRS-TT-F-6
Nama Kasus Uji	Mengubah data dosen
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “pengaturan dosen” 2. Menekan tombol “<i>edit</i>” pada akun dosen yang ingin diubah datanya 3. Aktor mengubah data dosen dan menekan tombol simpan
Hasil yang diharapkan	Sistem menampilkan pesan “berhasil melakukan perubahan”.
Hasil	Sistem menampilkan pesan “berhasil melakukan perubahan”.
Status	Valid

Tabel 6.20 Hasil Pengujian Validasi Mengubah Data Dosen Alternatif 1

Kode Kebutuhan	SRS-TT-F-6
Nama Kasus Uji	Mengubah data dosen
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “pengaturan dosen” 2. Menekan tombol “<i>edit</i>” pada akun dosen yang ingin diubah datanya 3. Aktor mengubah data dosen dengan membiarkan terdapat <i>field</i> kosong dan menekan tombol simpan

Hasil yang diharapkan	Sistem menampilkan pesan untuk melengkapi pengisian form
Hasil	Sistem menampilkan pesan untuk melengkapi pengisian form
Status	Valid

Tabel 6.21 Hasil Pengujian Validasi Mengubah Data Dosen Alternatif 2

Kode Kebutuhan	SRS-TT-F-6
Nama Kasus Uji	Mengubah data dosen
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “pengaturan dosen” 2. Menekan tombol “<i>edit</i>” pada akun dosen yang ingin diubah datanya 3. Aktor mengubah <i>username</i> dengan <i>username</i> yang telah digunakan dan menekan tombol simpan
Hasil yang diharapkan	Sistem menampilkan pesan gagal karena <i>username</i> telah digunakan
Hasil	Sistem menampilkan pesan gagal karena <i>username</i> telah digunakan
Status	Valid

Tabel 6.22 Hasil Pengujian Validasi Mengubah Data Dosen Alternatif 3

Kode Kebutuhan	SRS-TT-F-6
Nama Kasus Uji	Mengubah data dosen
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “pengaturan dosen” 2. Menekan tombol “<i>edit</i>” pada akun dosen yang ingin diubah datanya 3. Aktor mengubah <i>email</i> dengan <i>email</i> yang telah digunakan dan menekan tombol simpan
Hasil yang diharapkan	Sistem menampilkan pesan gagal karena <i>email</i> telah digunakan
Hasil	Sistem menampilkan pesan gagal karena <i>email</i> telah digunakan
Status	Valid

Tabel 6.23 Hasil Pengujian Validasi Menghapus Dosen

Kode Kebutuhan	SRS-TT-F-7
Nama Kasus Uji	Menghapus dosen
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “pengaturan dosen” 2. Menekan tombol “hapus” dan menekan tombol “ya” untuk konfirmasi menghapus
Hasil yang diharapkan	Sistem menampilkan pesan “berhasil menghapus data”.
Hasil	Sistem menampilkan pesan “berhasil menghapus data”.
Status	Valid

Tabel 6.24 Hasil Pengujian Validasi Menghapus Dosen Alternatif 1

Kode Kebutuhan	SRS-TT-F-7
Nama Kasus Uji	Menghapus dosen
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “pengaturan dosen” 2. Menekan tombol “hapus” dan menekan tombol “tidak” untuk batal menghapus
Hasil yang diharapkan	<i>Pop-up</i> konfirmasi akan tertutup
Hasil	<i>Pop-up</i> konfirmasi akan tertutup
Status	Valid

Tabel 6.25 Hasil Pengujian Validasi Menambah KJFD

Kode Kebutuhan	SRS-TT-F-8
Nama Kasus Uji	Menambah KJFD
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “tambah KJFD” 2. Mengisi <i>form</i> tambah KJFD dan menekan tombol “tambah”
Hasil yang diharapkan	Menampilkan pesan “berhasil menambah KJFD”
Hasil	Menampilkan pesan “berhasil menambah KJFD”
Status	Valid

Tabel 6.26 Hasil Pengujian Validasi Menambah KJFD Alternatif 1

Kode Kebutuhan	SRS-TT-F-8
Nama Kasus Uji	Menambah KJFD
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “tambah KJFD” 2. Mengisi <i>form</i> tambah KJFD dan menekan tombol “tambah” dengan kondisi nama KJFD telah dipakai
Hasil yang diharapkan	Menampilkan pesan “gagal menambah KJFD, nama KJFD telah digunakan”
Hasil	Menampilkan pesan “gagal menambah KJFD, nama KJFD telah digunakan”
Status	Valid

Tabel 6.27 Hasil Pengujian Validasi Melihat KJFD

Kode Kebutuhan	SRS-TT-F-9
Nama Kasus Uji	Melihat KJFD
Prosedur	Membuka halaman “KJFD”
Hasil yang diharapkan	Menampilkan daftar KJFD yang telah ditambahkan
Hasil	Menampilkan daftar KJFD yang telah ditambahkan
Status	Valid

Tabel 6.28 Hasil Pengujian Validasi Melihat KJFD Alternatif 1

Kode Kebutuhan	SRS-TT-F-9
Nama Kasus Uji	Melihat KJFD
Prosedur	Membuka halaman “KJFD” dengan kondisi belum terdapat data KJFD yang ditambahkan
Hasil yang diharapkan	Menampilkan pesan “data belum tersedia”
Hasil	Menampilkan pesan “data belum tersedia”
Status	Valid

Tabel 6.29 Hasil Pengujian Validasi Mengubah KJFD

Kode Kebutuhan	SRS-TT-F-10
Nama Kasus Uji	Mengubah KJFD

Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “KJFD” 2. Menekan tombol “<i>edit</i>” pada KJFD yang ingin diubah datanya 3. Aktor mengubah data KJFD dan menekan tombol simpan
Hasil yang diharapkan	Sistem menampilkan pesan “berhasil melakukan perubahan”.
Hasil	Sistem menampilkan pesan “berhasil melakukan perubahan”.
Status	Valid

Tabel 6.30 Hasil Pengujian Validasi Mengubah KJFD Alternatif 1

Kode Kebutuhan	SRS-TT-F-10
Nama Kasus Uji	Mengubah KJFD
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “KJFD” 2. Menekan tombol “<i>edit</i>” pada KJFD yang ingin diubah datanya 3. Aktor mengubah data KJFD dan menekan tombol simpan, dengan kondisi nama KJFD telah digunakan
Hasil yang diharapkan	Sistem menampilkan pesan “Gagal melakukan perubahan, nama KJFD telah digunakan”.
Hasil	Sistem menampilkan pesan “Gagal melakukan perubahan, nama KJFD telah digunakan”.
Status	Valid

Tabel 6.31 Hasil Pengujian Validasi Menghapus KJFD

Kode Kebutuhan	SRS-TT-F-11
Nama Kasus Uji	Menghapus KJFD
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “KJFD” 2. Menekan tombol “hapus” dan menekan tombol “ya” untuk menghapus
Hasil yang diharapkan	Sistem menampilkan pesan “berhasil menghapus data”.
Hasil	Sistem menampilkan pesan “berhasil menghapus data”.
Status	Valid

Tabel 6.32 Hasil Pengujian Validasi Menghapus KJFD Alternatif 1

Kode Kebutuhan	SRS-TT-F-11
Nama Kasus Uji	Menghapus KJFD
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “KJFD” 2. Menekan tombol “hapus” dan menekan tombol “tidak” untuk batal menghapus
Hasil yang diharapkan	<i>Pop-up</i> konfirmasi akan tertutup
Hasil	<i>Pop-up</i> konfirmasi akan tertutup
Status	Valid

Tabel 6.33 Hasil Pengujian Validasi Menambah Mata Kuliah

Kode Kebutuhan	SRS-TT-F-12
Nama Kasus Uji	Menambah Mata Kuliah
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “tambah Mata Kuliah” 2. Mengisi <i>form</i> tambah Mata Kuliah dan menekan tombol “tambah”
Hasil yang diharapkan	Menampilkan pesan “berhasil menambah Mata Kuliah”
Hasil	Menampilkan pesan “berhasil menambah Mata Kuliah”
Status	Valid

Tabel 6.34 Hasil Pengujian Validasi Menambah Mata Kuliah Alternatif 1

Kode Kebutuhan	SRS-TT-F-12
Nama Kasus Uji	Menambah Mata Kuliah
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “tambah Mata Kuliah” 2. Mengisi <i>form</i> tambah Mata Kuliah dan menekan tombol “tambah” dengan kondisi kode mata kuliah telah dipakai
Hasil yang diharapkan	Menampilkan pesan “gagal menambah mata kuliah, kode Mata Kuliah telah digunakan”

Hasil	Menampilkan pesan “gagal menambah mata kuliah, kode Mata Kuliah telah digunakan”
Status	Valid

Tabel 6.35 Hasil Pengujian Validasi Melihat Mata Kuliah

Kode Kebutuhan	SRS-TT-F-13
Nama Kasus Uji	Melihat Mata Kuliah
Prosedur	Membuka halaman “Mata Kuliah”
Hasil yang diharapkan	Menampilkan daftar Mata Kuliah yang telah ditambahkan
Hasil	Menampilkan daftar Mata Kuliah yang telah ditambahkan
Status	Valid

Tabel 6.36 Hasil Pengujian Validasi Melihat Mata Kuliah Alternatif 1

Kode Kebutuhan	SRS-TT-F-13
Nama Kasus Uji	Melihat Mata Kuliah
Prosedur	Membuka halaman “Mata Kuliah” dengan kondisi belum terdapat data Mata Kuliah yang ditambahkan
Hasil yang diharapkan	Menampilkan pesan “data belum tersedia”
Hasil	Menampilkan pesan “data belum tersedia”
Status	Valid

Tabel 6.37 Hasil Pengujian Validasi Mengubah Mata Kuliah

Kode Kebutuhan	SRS-TT-F-14
Nama Kasus Uji	Mengubah Mata Kuliah
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “Mata Kuliah” 2. Menekan tombol “<i>edit</i>” pada Mata Kuliah yang ingin diubah datanya 3. Aktor mengubah data Mata Kuliah dan menekan tombol simpan
Hasil yang diharapkan	Sistem menampilkan pesan “berhasil melakukan perubahan”.

Hasil	Sistem menampilkan pesan “berhasil melakukan perubahan”.
Status	Valid

Tabel 6.38 Hasil Pengujian Validasi Mengubah Mata Kuliah Alternatif 1

Kode Kebutuhan	SRS-TT-F-14
Nama Kasus Uji	Mengubah Mata Kuliah
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “Mata Kuliah” 2. Menekan tombol “<i>edit</i>” pada Mata Kuliah yang ingin diubah datanya 3. Aktor mengubah data Mata Kuliah dan menekan tombol simpan, dengan kondisi kode Mata Kuliah telah digunakan
Hasil yang diharapkan	Sistem menampilkan pesan “Gagal melakukan perubahan, kode Mata Kuliah telah digunakan”.
Hasil	Sistem menampilkan pesan “Gagal melakukan perubahan, kode Mata Kuliah telah digunakan”.
Status	Valid

Tabel 6.39 Hasil Pengujian Validasi Menghapus Mata Kuliah

Kode Kebutuhan	SRS-TT-F-15
Nama Kasus Uji	Menghapus Mata Kuliah
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “Mata Kuliah” 2. Menekan tombol “hapus” dan menekan tombol “ya” untuk menghapus
Hasil yang diharapkan	Sistem menampilkan pesan “berhasil menghapus data”.
Hasil	Sistem menampilkan pesan “berhasil menghapus data”.
Status	Valid

Tabel 6.40 Hasil Pengujian Validasi Menghapus Mata Kuliah Alternatif 1

Kode Kebutuhan	SRS-TT-F-15
Nama Kasus Uji	Menghapus Mata Kuliah
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “Mata Kuliah”

	2. Menekan tombol “hapus” dan menekan tombol “tidak” untuk batal menghapus
Hasil yang diharapkan	Pop-up konfirmasi akan tertutup
Hasil	Pop-up konfirmasi akan tertutup
Status	Valid

Tabel 6.41 Hasil Pengujian Validasi Melihat *Team Teaching*

Kode Kebutuhan	SRS-TT-F-16
Nama Kasus Uji	Melihat <i>Team Teaching</i>
Prosedur	Membuka halaman “ <i>team teaching</i> ”
Hasil yang diharapkan	Menampilkan daftar <i>Team Teaching</i> berdasarkan mata kuliah yang ada
Hasil	Menampilkan daftar <i>Team Teaching</i> berdasarkan mata kuliah yang ada
Status	Valid

Tabel 6.42 Hasil Pengujian Validasi Melihat *Team Teaching* Alternatif 1

Kode Kebutuhan	SRS-TT-F-16
Nama Kasus Uji	Melihat <i>Team Teaching</i>
Prosedur	Membuka halaman “ <i>team teaching</i> ” dengan kondisi belum terdapat data Mata Kuliah yang ditambahkan
Hasil yang diharapkan	Menampilkan pesan “data belum tersedia”
Hasil	Menampilkan pesan “data belum tersedia”
Status	Valid

Tabel 6.43 Hasil Pengujian Validasi Mengubah *Team Teaching*

Kode Kebutuhan	SRS-TT-F-17
Nama Kasus Uji	Mengubah <i>Team Teaching</i>
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “<i>team teaching</i>” 2. Menekan tombol “<i>edit</i>” pada <i>team teaching</i> yang ingin diubah datanya 3. Aktor mengubah data <i>team teaching</i> dan menekan tombol simpan

Hasil yang diharapkan	Sistem menampilkan pesan “berhasil melakukan perubahan”.
Hasil	Sistem menampilkan pesan “berhasil melakukan perubahan”.
Status	Valid

Tabel 6.44 Hasil Pengujian Validasi Menambah Materi

Kode Kebutuhan	SRS-TT-F-18
Nama Kasus Uji	Menambah Materi
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “tambah materi” 2. Mengisi <i>form</i> tambah Mata Kuliah dan menekan tombol “tambah”
Hasil yang diharapkan	Menampilkan pesan “berhasil menambah materi”
Hasil	Menampilkan pesan “berhasil menambah materi”
Status	Valid

Tabel 6.45 Hasil Pengujian Validasi Melihat Materi

Kode Kebutuhan	SRS-TT-F-19
Nama Kasus Uji	Melihat Materi
Prosedur	Membuka halaman “atur materi”
Hasil yang diharapkan	Menampilkan daftar materi yang telah ditambahkan
Hasil	Menampilkan daftar materi yang telah ditambahkan
Status	Valid

Tabel 6.46 Hasil Pengujian Validasi Melihat Materi Alternatif 1

Kode Kebutuhan	SRS-TT-F-19
Nama Kasus Uji	Melihat Materi
Prosedur	Membuka halaman “atur materi” dengan kondisi belum terdapat data materi yang ditambahkan
Hasil yang diharapkan	Menampilkan pesan “data belum tersedia”
Hasil	Menampilkan pesan “data belum tersedia”
Status	Valid

Tabel 6.47 Hasil Pengujian Validasi Mengubah Materi

Kode Kebutuhan	SRS-TT-F-20
Nama Kasus Uji	Mengubah Materi
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “Mata Kuliah” 2. Menekan tombol “<i>edit</i>” pada Mata Kuliah yang ingin diubah datanya 3. Aktor mengubah data Mata Kuliah dan menekan tombol simpan
Hasil yang diharapkan	Sistem menampilkan pesan “berhasil melakukan perubahan”.
Hasil	Sistem menampilkan pesan “berhasil melakukan perubahan”.
Status	Valid

Tabel 6.48 Hasil Pengujian Validasi Menghapus Materi

Kode Kebutuhan	SRS-TT-F-21
Nama Kasus Uji	Menghapus Materi
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “atur materi” 2. Menekan tombol “hapus” dan menekan tombol “ya” untuk menghapus
Hasil yang diharapkan	Sistem menampilkan pesan “berhasil menghapus data”.
Hasil	Sistem menampilkan pesan “berhasil menghapus data”.
Status	Valid

Tabel 6.49 Hasil Pengujian Validasi Menghapus Materi Alternatif 1

Kode Kebutuhan	SRS-TT-F-21
Nama Kasus Uji	Menghapus Materi
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “KJFD” 2. Menekan tombol “hapus” dan menekan tombol “tidak” untuk batal menghapus
Hasil yang diharapkan	<i>Pop-up</i> konfirmasi akan tertutup
Hasil	<i>Pop-up</i> konfirmasi akan tertutup

Status	Valid
--------	-------

Tabel 6.50 Hasil Pengujian Validasi Menambah Penugasan

Kode Kebutuhan	SRS-TT-F-22
Nama Kasus Uji	Menambah Penugasan
Prosedur	1. Membuka halaman “tambah penugasan” 2. Mengisi <i>form</i> tambah penugasan dan menekan tombol “tambah”
Hasil yang diharapkan	Menampilkan pesan “berhasil menambah penugasan”
Hasil	Menampilkan pesan “berhasil menambah penugasan”
Status	Valid

Tabel 6.51 Hasil Pengujian Validasi Melihat Penugasan

Kode Kebutuhan	SRS-TT-F-23
Nama Kasus Uji	Melihat Penugasan
Prosedur	Membuka halaman “atur penugasan”
Hasil yang diharapkan	Menampilkan daftar penugasan yang telah ditambahkan
Hasil	Menampilkan daftar penugasan yang telah ditambahkan
Status	Valid

Tabel 6.52 Hasil Pengujian Validasi Melihat Penugasan Alternatif 1

Kode Kebutuhan	SRS-TT-F-23
Nama Kasus Uji	Melihat Penugasan
Prosedur	Membuka halaman “atur penugasan” dengan kondisi belum terdapat data penugasan yang ditambahkan
Hasil yang diharapkan	Menampilkan pesan “data belum tersedia”
Hasil	Menampilkan pesan “data belum tersedia”
Status	Valid

Tabel 6.53 Hasil Pengujian Validasi Melihat Progres Penugasan

Kode Kebutuhan	SRS-TT-F-24
Nama Kasus Uji	Melihat Penugasan
Prosedur	Membuka halaman “lihat progres”
Hasil yang diharapkan	Menampilkan progres penugasan yang telah dibuat
Hasil	Menampilkan progres penugasan yang telah dibuat
Status	Valid

Tabel 6.54 Hasil Pengujian Validasi Mengubah Penugasan

Kode Kebutuhan	SRS-TT-F-25
Nama Kasus Uji	Mengubah Penugasan
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “atur penugasan” 2. Menekan tombol “<i>edit</i>” pada penugasan yang ingin diubah datanya 3. Aktor mengubah data penugasan dan menekan tombol simpan
Hasil yang diharapkan	Sistem menampilkan pesan “berhasil melakukan perubahan”.
Hasil	Sistem menampilkan pesan “berhasil melakukan perubahan”.
Status	Valid

Tabel 6.55 Hasil Pengujian Validasi Menghapus Penugasan

Kode Kebutuhan	SRS-TT-F-26
Nama Kasus Uji	Menghapus Penugasan
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “atur penugasan” 2. Menekan tombol “hapus” dan menekan tombol “ya” untuk menghapus
Hasil yang diharapkan	Sistem menampilkan pesan “berhasil menghapus data”.
Hasil	Sistem menampilkan pesan “berhasil menghapus data”.
Status	Valid

Tabel 6.56 Hasil Pengujian Validasi Menghapus Penugasan Alternatif 1

Kode Kebutuhan	SRS-TT-F-26
Nama Kasus Uji	Menghapus Penugasan
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “atur penugasan” 2. Menekan tombol “hapus” dan menekan tombol “tidak” untuk batal menghapus
Hasil yang diharapkan	<i>Pop-up</i> konfirmasi akan tertutup
Hasil	<i>Pop-up</i> konfirmasi akan tertutup
Status	Valid

Tabel 6.57 Hasil Pengujian Validasi Mengatur Anggota Materi Penugasan

Kode Kebutuhan	SRS-TT-F-27
Nama Kasus Uji	Mengatur Anggota Materi Penugasan
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “detail penugasan” 2. Memilih anggota <i>team teaching</i> dan memilih untuk memasukkan atau mengeluarkan anggota dengan menekan tombol “gabung” atau “keluar” 3. Menekan tombol “ya” dari <i>pop-up</i> konfirmasi untuk bergabung atau keluar
Hasil yang diharapkan	Sistem menampilkan pesan sukses.
Hasil	Sistem menampilkan pesan sukses.
Status	Valid

Tabel 6.58 Hasil Pengujian Validasi Mengatur Anggota Materi Penugasan Alternatif 1

Kode Kebutuhan	SRS-TT-F-27
Nama Kasus Uji	Mengatur Anggota Materi Penugasan
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “detail penugasan” 2. Memilih anggota <i>team teaching</i> dan memilih untuk memasukkan atau mengeluarkan anggota dengan menekan tombol “gabung” atau “keluar”

	3. Menekan tombol “ya” dari <i>pop-up</i> konfirmasi untuk gabung atau keluar
Hasil yang diharapkan	<i>Pop-up</i> konfirmasi akan tertutup
Hasil	<i>Pop-up</i> konfirmasi akan tertutup
Status	Valid

Tabel 6.59 Hasil Pengujian Validasi Gabung Materi Penugasan

Kode Kebutuhan	SRS-TT-F-28
Nama Kasus Uji	Gabung Materi Penugasan
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “detail penugasan” 2. Menekan tombol “gabung” pada materi yang dipilih 3. Menekan tombol “ya” dari <i>pop-up</i> konfirmasi untuk gabung pada materi
Hasil yang diharapkan	Sistem menampilkan pesan sukses.
Hasil	Sistem menampilkan pesan sukses.
Status	Valid

Tabel 6.60 Hasil Pengujian Validasi Gabung Materi Penugasan Alternatif 1

Kode Kebutuhan	SRS-TT-F-28
Nama Kasus Uji	Gabung Materi Penugasan
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “detail penugasan” 2. Menekan tombol “gabung” pada materi yang dipilih 3. Menekan tombol “tidak” dari <i>pop-up</i> konfirmasi untuk gabung pada materi
Hasil yang diharapkan	<i>Pop-up</i> konfirmasi akan tertutup
Hasil	<i>Pop-up</i> konfirmasi akan tertutup
Status	Valid

Tabel 6.61 Hasil Pengujian Validasi Keluar Materi Penugasan

Kode Kebutuhan	SRS-TT-F-29
Nama Kasus Uji	Keluar Materi Penugasan
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “detail penugasan”

	<ol style="list-style-type: none"> 2. Menekan tombol “keluar” pada materi yang dipilih 3. Menekan tombol “ya” dari <i>pop-up</i> konfirmasi untuk keluar pada materi
Hasil yang diharapkan	Sistem menampilkan pesan sukses.
Hasil	Sistem menampilkan pesan sukses.
Status	Valid

Tabel 6.62 Hasil Pengujian Validasi Keluar Materi Penugasan Alternatif 1

Kode Kebutuhan	SRS-TT-F-29
Nama Kasus Uji	Keluar Materi Penugasan
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “detail penugasan” 2. Menekan tombol “keluar” pada materi yang dipilih 3. Menekan tombol “tidak” dari <i>pop-up</i> konfirmasi untuk keluar pada materi
Hasil yang diharapkan	<i>Pop-up</i> konfirmasi akan tertutup
Hasil	<i>Pop-up</i> konfirmasi akan tertutup
Status	Valid

Tabel 6.63 Hasil Pengujian Validasi Membuat Soal

Kode Kebutuhan	SRS-TT-F-30
Nama Kasus Uji	Membuat soal
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “detail penugasan” 2. Menekan tombol “kerjakan” pada materi yang ingin dibuat soalnya 3. Mengisi <i>form</i> buat soal dan menekan tombol “tambah”
Hasil yang diharapkan	Menampilkan pesan “berhasil membuat soal”
Hasil	Menampilkan pesan “berhasil membuat soal”
Status	Valid

Tabel 6.64 Hasil Pengujian Validasi Mengubah Soal

Kode Kebutuhan	SRS-TT-F-31
Nama Kasus Uji	Mengubah soal
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “detail penugasan” 2. Menekan tombol “edit” pada materi yang ingin diubah soalnya 3. Mengubah isi <i>form</i> ubah soal dan menekan tombol “simpan”
Hasil yang diharapkan	Menampilkan pesan “berhasil mengubah soal”
Hasil	Menampilkan pesan “berhasil mengubah soal”
Status	Valid

Tabel 6.65 Hasil Pengujian Validasi Submit Soal

Kode Kebutuhan	SRS-TT-F-32
Nama Kasus Uji	Submit soal
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “detail penugasan” 2. Menekan tombol “submit” pada materi yang soalnya akan disubmit 3. Menekan tombol “ya” dari <i>pop-up</i> konfirmasi untuk submit soal
Hasil yang diharapkan	Sistem menampilkan pesan sukses.
Hasil	Sistem menampilkan pesan sukses.
Status	Valid

Tabel 6.66 Hasil Pengujian Validasi Submit Soal Alternatif 1

Kode Kebutuhan	SRS-TT-F-32
Nama Kasus Uji	Submit Soal
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “detail penugasan” 2. Menekan tombol “submit” pada materi yang soalnya akan disubmit 3. Menekan tombol “tidak” dari <i>pop-up</i> konfirmasi untuk submit soal
Hasil yang diharapkan	<i>Pop-up</i> konfirmasi akan tertutup

Hasil	<i>Pop-up</i> konfirmasi akan tertutup
Status	Valid

Tabel 6.67 Hasil Pengujian Validasi Mempercepat Evaluasi

Kode Kebutuhan	SRS-TT-F-33
Nama Kasus Uji	Mempercepat Evaluasi
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “Lihat progres” 2. Menekan tombol “evaluasi sekarang” 3. Menekan tombol “ya” dari <i>pop-up</i> konfirmasi untuk mempercepat evaluasi
Hasil yang diharapkan	Sistem menampilkan pesan sukses.
Hasil	Sistem menampilkan pesan sukses.
Status	Valid

Tabel 6.68 Hasil Pengujian Validasi Mempercepat Evaluasi Alternatif 1

Kode Kebutuhan	SRS-TT-F-33
Nama Kasus Uji	Mempercepat Evaluasi
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “Lihat progres” 2. Menekan tombol “evaluasi sekarang” 3. Menekan tombol “ya” dari <i>pop-up</i> konfirmasi untuk mempercepat evaluasi
Hasil yang diharapkan	<i>Pop-up</i> konfirmasi akan tertutup
Hasil	<i>Pop-up</i> konfirmasi akan tertutup
Status	Valid

Tabel 6.69 Hasil Pengujian Validasi Mengirim Reminder

Kode Kebutuhan	SRS-TT-F-34
Nama Kasus Uji	Mengirim Reminder
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “Lihat progres” 2. Menekan tombol “kirim reminder” 3. Menekan tombol “ya” dari <i>pop-up</i> konfirmasi untuk mengirim reminder
Hasil yang diharapkan	Sistem menampilkan pesan sukses.
Hasil	Sistem menampilkan pesan sukses.

Status	Valid
--------	-------

Tabel 6.70 Hasil Pengujian Validasi Mengirim Reminder Alternatif 1

Kode Kebutuhan	SRS-TT-F-34
Nama Kasus Uji	Mengirim Reminder
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “Lihat progres” 2. Menekan tombol “kirim reminder” 3. Menekan tombol “tidak” dari <i>pop-up</i> konfirmasi untuk mengirim reminder
Hasil yang diharapkan	<i>Pop-up</i> konfirmasi akan tertutup
Hasil	<i>Pop-up</i> konfirmasi akan tertutup
Status	Valid

Tabel 6.71 Hasil Pengujian Validasi Memberi Komentar

Kode Kebutuhan	SRS-TT-F-35
Nama Kasus Uji	Memberi Komentar
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “Evaluasi penugasan” 2. Menekan tombol “komentar” 3. Mengisi <i>form</i> komentar dan menekan tombol “tambah”
Hasil yang diharapkan	Sistem menampilkan pesan sukses.
Hasil	Sistem menampilkan pesan sukses.
Status	Valid

Tabel 6.72 Hasil Pengujian Validasi Menghapus Komentar

Kode Kebutuhan	SRS-TT-F-36
Nama Kasus Uji	Menghapus Komentar
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “Evaluasi penugasan” 2. Menekan tombol “komentar” 3. Menekan tombol “X” pada komentar yang telah dibuat 4. Menekan tombol “ya” pada <i>pop-up</i> konfirmasi untuk menghapus komentar
Hasil yang diharapkan	Sistem menampilkan pesan “berhasil menghapus data”.

Hasil	Sistem menampilkan pesan “berhasil menghapus data”.
Status	Valid

Tabel 6.73 Hasil Pengujian Validasi Menghapus Komentar Alternatif 1

Kode Kebutuhan	SRS-TT-F-36
Nama Kasus Uji	Menghapus Komentar
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “Evaluasi penugasan” 2. Menekan tombol “komentar” 3. Menekan tombol “X” pada komentar yang telah dibuat 4. Menekan tombol “tidak” pada <i>pop-up</i> konfirmasi untuk menghapus komentar
Hasil yang diharapkan	<i>Pop-up</i> konfirmasi akan tertutup
Hasil	<i>Pop-up</i> konfirmasi akan tertutup
Status	Valid

Tabel 6.74 Hasil Pengujian Validasi Mengevaluasi Soal

Kode Kebutuhan	SRS-TT-F-37
Nama Kasus Uji	Mengevaluasi Soal
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “Edit Evaluasi” 2. Mengubah soal yang ingin dievaluasi 3. Menekan tombol “simpan”
Hasil yang diharapkan	Sistem menampilkan pesan sukses.
Hasil	Sistem menampilkan pesan sukses.
Status	Valid

Tabel 6.75 Hasil Pengujian Validasi Menutup Evaluasi

Kode Kebutuhan	SRS-TT-F-38
Nama Kasus Uji	Menutup Evaluasi
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “Atur Penugasan” 2. Menekan tombol “tutup evaluasi” 3. Menekan tombol “ya” pada <i>pop-up</i> konfirmasi untuk menutup evaluasi

Hasil yang diharapkan	Sistem menampilkan pesan sukses.
Hasil	Sistem menampilkan pesan sukses.
Status	Valid

Tabel 6.76 Hasil Pengujian Validasi Menutup Evaluasi Alternatif 1

Kode Kebutuhan	SRS-TT-F-38
Nama Kasus Uji	Menutup Evaluasi
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman "Atur Penugasan" 2. Menekan tombol "tutup evaluasi" 3. Menekan tombol "tidak" pada <i>pop-up</i> konfirmasi untuk batal menutup evaluasi
Hasil yang diharapkan	<i>Pop-up</i> konfirmasi akan tertutup
Hasil	<i>Pop-up</i> konfirmasi akan tertutup
Status	Valid

Tabel 6.77 Hasil Pengujian Validasi Review Soal

Kode Kebutuhan	SRS-TT-F-39
Nama Kasus Uji	Review Soal
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman "Cetak soal pilih" 2. Memilih soal yang ingin dicetak 3. Menekan tombol "review"
Hasil yang diharapkan	Sistem menampilkan review soal yang telah dipilih
Hasil	Sistem menampilkan review soal yang telah dipilih
Status	Valid

Tabel 6.78 Hasil Pengujian Validasi Cetak Soal Acak

Kode Kebutuhan	SRS-TT-F-40
Nama Kasus Uji	Cetak Soal Acak
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman "Cetak soal" 2. Mengisi <i>form</i> cetak soal acak 3. Menekan tombol "review" 4. Menekan tombol "cetak" pada halaman review soal
Hasil yang diharapkan	Sistem menampilkan mencetak soal secara acak

Hasil	Sistem menampilkan mencetak soal secara acak
Status	Valid

Tabel 6.79 Hasil Pengujian Validasi Cetak Soal Pilih

Kode Kebutuhan	SRS-TT-F-41
Nama Kasus Uji	Cetak Soal Pilih
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “Cetak soal” 2. Menekan tombol “pilih soal” 3. Memilih soal pada halaman “cetak soal pilih” 4. Menekan tombol “review” 5. Menekan tombol “cetak” pada halaman review soal
Hasil yang diharapkan	Sistem menampilkan mencetak soal yang dipilih
Hasil	Sistem menampilkan mencetak soal yang dipilih
Status	Valid

Tabel 6.80 Hasil Pengujian Melihat Riwayat Cetak Soal

Kode Kebutuhan	SRS-TT-F-42
Nama Kasus Uji	Melihat Riwayat Cetak Soal
Prosedur	Membuka halaman “Riwayat Cetak Soal”
Hasil yang diharapkan	Sistem menampilkan riwayat cetak soal
Hasil	Sistem menampilkan riwayat cetak soal
Status	Valid

Tabel 6.81 Hasil Pengujian Validasi Melihat Riwayat Cetak Soal Alternatif 1

Kode Kebutuhan	SRS-TT-F-42
Nama Kasus Uji	Melihat Riwayat Cetak Soal
Prosedur	Membuka halaman “Riwayat Cetak Soal” dengan kondisi belum terdapat data riwayat cetak soal yang dilakukan
Hasil yang diharapkan	Menampilkan pesan “data belum tersedia”
Hasil	Menampilkan pesan “data belum tersedia”
Status	Valid

Tabel 6.82 Hasil Pengujian Validasi Evaluasi Bank Soal

Kode Kebutuhan	SRS-TT-F-43
Nama Kasus Uji	Evaluasi Bank Soal
Prosedur	<ol style="list-style-type: none"> 1. Membuka halaman “Dashboard” 2. Menekan tombol evaluasi pada grafik bank soal 3. Mengubah soal sesuai kebutuhan 4. Menekan tombol simpan
Hasil yang diharapkan	Sistem akan menampilkan pesan sukses
Hasil	Sistem akan menampilkan pesan sukses
Status	Valid

6.4 Pengujian *Compatibility*

Pengujian *compatibility* dapat dilihat pada Tabel 6.82 dan hasil pengujian *compatibility* dapat dilihat pada Gambar 6.4, pengujian menggunakan aplikasi SortSite 5.

Tabel 6.83 Pengujian *Compatibility*

Kode Kebutuhan	SRS-TT-NF-1
Prosedur	Menjalankan sistem pada browser yang berbeda
Hasil yang diharapkan	Fungsi pada sistem dapat berjalan dengan baik pada 8 browser yang berbeda
Hasil	Fungsi pada sistem dapat berjalan dengan baik pada 8 browser yang berbeda
Status	Valid

The screenshot shows the 'Compatibility' tab in SortSite 5. It displays a table of browser compatibility results for various browsers and their versions. The table includes columns for Browser, IE, Edge, Firefox, Safari, Opera, Chrome, iOS, and Android. The results show that the system is compatible with all tested browsers and versions, with no issues reported.

Browser	IE	Edge	Firefox	Safari	Opera	Chrome	iOS	Android				
Version	11	17	61	≤ 10	11	54	68	≤ 9	10	11	≤ 3	4*
Critical Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Major Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Minor Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Key:
 ● Missing content or functionality
 ● Major layout or performance problems
 ● Minor layout or performance problems

* Most Android devices from 4.1 onwards use Chrome as the default browser, older versions use the original Android browser

Priority	Description and URL	Guideline and Line#	Count
>	Expand all 0 issues		

Gambar 6.4 Hasil Pengujian *Compatibility*

Pada Gambar 6.4 ditunjukkan hasil pengujian *compatibility* dimana terdapat tabel yang menunjukkan tingkat kecocokan dari sistem yang berjalan terhadap browser yang digunakan. Pada pengujian ini sistem diuji dari 3 kategori yakni *critical issues*, *major issues*, dan *minor issues*. Pada *critical issues* jika terjadi ketidakcocokan antara sistem dengan browser maka terdapat konten yang hilang atau fungsional yang tidak dapat digunakan. Pada *major issues* dan *minor issues* pengaruh yang dihasilkan jika terjadi ketidakcocokan adalah pada segi tampilan.



BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan hasil analisis kebutuhan, perancangan, implementasi, dan pengujian yang telah dilakukan, maka dapat disimpulkan sebagai berikut:

1. Berdasarkan analisis kebutuhan yang telah dilakukan, sistem pengelolaan soal memiliki 43 kebutuhan fungsional dan 1 kebutuhan nonfungsional. Didapatkan juga 5 aktor yang terhubung dengan sistem, yakni pengguna, ketua jurusan, koordinator KJFD, ketua *team teaching*, dan dosen. Kemudian dari analisis yang telah dilakukan dilakukan pemodelan berupa *usecase diagram* untuk menjelaskan perilaku sistem tampak luar, dan *usecase scenario* untuk menjelaskan *usecase diagram*.
2. Berdasarkan perancangan yang telah dilakukan, hasil yang didapatkan berupa perancangan arsitektur yang menghasilkan *sequence diagram* dan *class diagram*. Terdapat perancangan komponen dimana menghasilkan rancangan algoritme dari 3 fungsi utama yang ada pada sistem. Pada Perancangan basis data, basis data dimodelkan menggunakan *Conceptual Data Model (CDM)* dan yang terakhir adalah perancangan antarmuka yang menghasilkan *layout* sebagai rancangan awal tampilan sistem, *wireframe* digunakan sebagai media pembuatan perancangan antarmuka. Berdasarkan implementasi yang telah dilakukan dihasilkan spesifikasi sistem dimana menjelaskan tentang spesifikasi perangkat keras, spesifikasi perangkat lunak, dan sistem operasi yang digunakan dalam mengembangkan sistem. Terdapat juga implementasi kode program yang menghasilkan kode program yang dibuat berdasarkan perancangan komponen yang telah dibuat. Implementasi basis data dilakukan dan menghasilkan *Physical Data Model (PDM)* dan implementasi antarmuka menghasilkan tampilan jadi dari perancangan antarmuka yang telah dibuat.
3. Berdasarkan pengujian yang telah dilakukan, didapatkan hasil 100% valid pada semua pengujian. Pengujian yang dilakukan pada sistem berupa pengujian unit, pengujian integrasi, pengujian validasi, dan pengujian *compatibility*. Pada pengujian unit digunakan metode *whitebox testing* dengan membuat *basis path testing* dan menguji 3 *method* utama dari sistem. Pada pengujian integrasi dilakukan integrasi antar *method* untuk melihat apakah dalam integrasinya *method* dapat menjalankan fungsinya. Pada pengujian validasi dilakukan untuk menguji skenario yang dibuat dan juga alternatif yang ada, didapatkan 75 kasus uji. Pada pengujian *compatibility* digunakan aplikasi SortSite 5 yang dapat menguji *compatibility* dari *website*, hasil yang diberikan adalah sistem dapat berjalan dengan baik pada 8 *browsers* berbeda.

7.2 Saran

Saran yang diberikan untuk pengembangan sistem pengelolaan soal sebagai aplikasi pengelolaan soal ini selanjutnya antara lain:

1. Dapat mengenali kemiripan soal antar dosen yang mengambil materi yang sama atau dengan soal yang sudah dibuat sebelumnya.
2. Dapat mendeteksi adanya kesalahan tulis, tanda baca dan kebakuan bahasa pada soal dan opsi yang dibuat.



DAFTAR PUSTAKA

- Nasution, H.M. Farid. 2016. Hubungan Metode Mengajar Dosen, Keterampilan Belajar, Sarana Belajar dan Lingkungan Belajar dengan Prestasi Belajar Mahasiswa.
- Astrini, Dhien dan Elvira, Melly. 2016. Model Bank Soal Berbasis WEB Untuk Meningkatkan Kualitas Pendidikan di Era Masyarakat Ekonomi ASEAN.
- Raintung, Firanda. 2016. Pentingnya Efisiensi Kerja Dalam Meningkatkan Kinerja Karyawan Pada PT. Kimia Farma Trading and Distribution.
- Putra, Eko R. 2013. Pengaruh Disiplin Kerja dan Lingkungan Kerja Terhadap Kinerja Karyawan di Workshop PT Dunia Marine Internusa Pekanbaru.
- Dunggio, Mardjan. 2013. Semangat dan Disiplin Kerja Terhadap Produktivitas Kerja Karyawan Pada PT. Jasa Raharja (PERSERO) Cabang Sulawesi Utara.
- Retnawati, Heri dan Hadi, Samsul. 2014. Sistem Bank Soal Daerah Terkalibrasi Untuk Menyongsong Era Desentralisasi.
- Mutmainah, Siti. 2017. Studi Kelayakan Pengembangan Aplikasi Bank Soal Dalam Rumah Belajar.
- A.S Rosa dan Salahuddin M. 2011. Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek). Bandung: Modula.
- Pressman, Roger S., 2010. Software Engineering A Practitioner's Approach. 7th Ed. New York: McGraw-Hill.
- Sommerville, Ian., 2011. Software engineering. 9th ed. London: Addison-Wesley.
- Singh, D., Thakur, A., dan Chaudhary, A. 2015. A Comparative Study between Waterfall and Incremental Software Development Life Cycle Model.
- Kumar, N., Zadgaonkar, A.S., dan Shukla, A. 2013. Evolving a New Software Development Life Cycle Model SDLC-2013 with Client Satisfaction.
- BPMN 2018. Tentang BPMN. Tersedia di: < <http://www.bpmn.org/> > [Diakses 9 November 2018]
- Bothell, Timothy W. 2001. 14 Rules for Writing Multiple-Choice Questions.
- Parkes, Jay dan Zimmaro, Dawn. 2016. Learning and Assessing with Multiple-Choice Questions in College Classrooms. Routledge, Taylor & Francis Group, 2016.
- Kementerian Pendidikan dan Kebudayaan. 2017. Panduan Penulisan Soal SMA/MA-SMK Tahun 2017.
- Mar, J., Masters, G.N., dan Keeves, J.P., 1999. Advances in measurement in educational research and assessment
- Thorndike, R.L. 1982. Applied Psychometrics. Boston Houghton Mifflin.

Codeigniter 2018. Welcome to CodeIgniter. Tersedia di: <
https://www.codeigniter.com/user_guide/general/welcome.html> [Diakses
9 November 2018]

PHPWord 2018. Introduction. Tersedia di: <
<https://phpword.readthedocs.io/en/latest/intro.html>> [Diakses 9 November
2018]

KBBI 2018. KBBI Daring - Soal. Tersedia di: <
<https://kbbi.kemdikbud.go.id/entri/soal>> [Diakses 26 November 2018]

