

RANCANG BANGUN APLIKASI BANK SOAL UNIVERSITAS SYIAH KUALA BERBASIS WEB

TUGAS AKHIR

Diajukan untuk melengkapi tugas-tugas dan
memenuhi syarat-syarat guna memperoleh gelar Sarjana Komputer

Oleh:

MR. HADAFEE MUDO
2008107010101



**PROGRAM STUDI INFORMATIKA JURUSAN INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN
ALAM UNIVERSITAS SYIAH KUALA
DARUSSALAM, BANDA ACEH
APRIL, 2024**

KATA PENGANTAR

Segala puji dan syukur kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya kepada kita semua, sehingga penulis dapat menyelesaikan penulisan Tugas Akhir yang berjudul “**RANCANG BANGUN APLIKASI BANK SOAL UNIVERSITAS SYIAH KUALA BERBASIS WEB**” yang telah dapat diselesaikan sesuai rencana. Penulis banyak mendapatkan berbagai pengarahan, bimbingan, dan bantuan dari berbagai pihak. Oleh karena itu, melalui tulisan ini penulis mengucapkan rasa terima kasih kepada:

1. Ayah dan Ibu sebagai kedua orang tua penulis yang senantiasa selalu mendukung aktivitas dan kegiatan yang penulis lakukan baik secara moral maupun material serta menjadi motivasi terbesar bagi penulis untuk menyelesaikan Tugas Akhir ini.
2. Ibu Rini Deviani, S.T., M.Eng selaku Dosen pembimbing I yang telah banyak memberikan bimbingan dan arahan kepada penulis, sehingga penulis dapat menyelesaikan Tugas Akhir ini.
3. Bapak Husaini, S.ST., M.Sc. selaku Dosen selaku Dosen Pembimbing II yang telah banyak memberikan bimbingan dan arahan kepada penulis, sehingga penulis dapat menyelesaikan Tugas Akhir ini.
4. Bapak Razief Perucha Fauzie Afidh, S.Si, M.Sc. selaku Dosen Wali.
5. Bapak Alim Misbullah, S.Si., M.S., selaku Ketua Program Studi Informatika
6. Seluruh dosen di Jurusan Informatika Fakultas Matematika dan Ilmu Pengetahuan Alam yang tidak bisa disebutkan satu persatu yang telah memberikan ilmu dan bimbingannya kepada penulis selama penulis menuntut ilmu di Jurusan Informatika.
7. Teman seperjuangan Informatika angkatan 2020, serta seluruh pihak yang telah terlibat membantu dalam menyelesaikan Tugas Akhir ini.

Penulis juga menyadari segala ketidaksempurnaan yang terdapat didalamnya baik dari segi materi, cara, ataupun bahasa yang disajikan. Seiring dengan ini penulis mengharapkan kritik dan saran dari pembaca yang sifatnya dapat berguna untuk

kesempurnaan Tugas Akhir ini. Harapan penulis semoga tulisan ini dapat bermanfaat bagi banyak pihak dan untuk perkembangan ilmu pengetahuan

Banda Aceh, 18 April 2024

Mr.Hadafee Mudo

NPM. 2008107010101

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI	iii
DAFTAR TABEL	v
DAFTAR GAMBAR.....	vi
BAB I PENDAHULUAN.....	1
1.1 LATAR BELAKANG	1
1.2 RUMUS MASALAH.....	3
1.3 TUJUAN PENELITIAN	3
1.4 MANFAAT PENELITIAN.....	3
BAB II TINJAUAN PUSTAKA	5
2.1 BANK SOAL	5
2.2 WEBSITE	7
2.3 LARAVEL	7
2.4 PHP	7
2.5 MYSQL.....	8
2.6 UNIFIED MODELING LANGUAGE	9
2.6.1 USE CASE DIAGRAM.....	9
2.6.2 ACTIVITY DIAGRAM	9
2.7 ENTITIY RELATIONSHIP DIAGRAM	11
2.8 WATERFALL MODEL	11
2.9 BLACK BOX TESTING	14
2.10 USABILITY TESTING	15
2.11 PERFORMANCE TESTING.....	17
2.12 PENELITIAN TERKAIT	19
BAB III METODE PENELITIAN.....	21
3.1 TEMPAT DAN WAKTU PENELITAIN	21
3.2 ALAT DAN BAHAN	22
3.2.1 Perangkat Keras	22
3.2.2 Perangkat Lunak	22
3.3 METODE PENELITIAN	22
3.3.1 Identifikasi Masalah.....	23
3.3.2 Studi Literatur	23

3.3.3	Analisa Kebutuhan Sistem.....	24
3.3.3.1	Kebutuhan Fungsional	24
3.3.3.2	Kebutuhan Non-Fungsional	24
3.3.4	Perancangan Sistem	25
3.3.4.1	Identifikasi Pengguna.....	25
3.3.4.2	<i>Use Case Diagram</i>	25
3.3.4.3	<i>Design Prototype</i> Aplikasi	27
3.3.5	Implementasi Sistem.....	32
3.3.6	Pengujian Sistem.....	32
3.3.6.1	Pengujian <i>Fungsional</i>	32
3.3.6.2	Pengujian Non-Fungsional.....	37
3.3.6.3	Pengujian <i>Usability</i>	41
3.3.7	Analisa Hasil.....	42
3.3.8	Penulisan Laporan.....	42
DAFTAR PUSTAKA.....		43

DAFTAR TABEL

Tabel 2.1 Entity Relationship Diagram (ERD)	11
Tabel 2.2 Daftar Pertanyaan Metode UMUX (Finstad, 2010).....	15
Tabel 2.3 Skor SUS (Borsci dkk., 2015).....	16
Tabel 3.1 Jadwal pelaksanaan penelitian bulan Maret-Mei 2024	21
Tabel 3.2 Jadwal pelaksanaan penelitian bulan Juni-Juli 2024.....	21
Tabel 3.3 Skenario Pengujian <i>Manual Black Box</i>	33
Tabel 3.4 <i>Detail Thread Group</i> untuk <i>Load Test</i>	37
Tabel 3.5 <i>Detail Thread Group</i> untuk <i>Stress Test</i>	39

DAFTAR GAMBAR

Gambar 2.1 Simbol-simbol pada <i>Activity Diagram</i>	10
Gambar 2.2 Model Pengembangan <i>Waterfall</i> (Pressman 2002)	12
Gambar 2.3 Siklus Evaluasi Desain <i>Interface</i> (Pressman 2002).....	13
Gambar 2.4 Perbandingan skala dan skor pada SUS (Bangor dkk., 2009).....	17
Gambar 3.1 Skema Proses Kerja.....	23
Gambar 3.2 <i>Use Case Diagram</i>	26
Gambar 3.3 <i>Design Prototype</i> Halaman Beranda	27
Gambar 3.4 <i>Design Prototype</i> Halaman Login	28
Gambar 3.5 <i>Design Prototype</i> Halaman Cari Soal	28
Gambar 3.6 <i>Design Prototype</i> Halaman Daftar Soal	29
Gambar 3.7 <i>Design Prototype</i> Halaman Tambah Soal	30
Gambar 3.8 <i>Design Prototype</i> Halaman Mengelola Matakuliah.	31
Gambar 3.9 <i>Design Prototype</i> Halaman Tambah Matakuliah.	31
Gambar 3.10 <i>Design Prototype</i> Halaman Profil	32
Gambar 3.11 Aktivitas Pengguna Mahasiswa untuk <i>Load Test</i>	38
Gambar 3.12 Aktivitas Pengguna Dosen untuk <i>Load Test</i>	39
Gambar 3.13 Aktivitas Pengguna Mahasiswa untuk <i>Stress Test</i>	40
Gambar 3.14 Aktivitas Pengguna Dosen untuk <i>Stress Test</i>	41

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Pendidikan merupakan salah satu upaya untuk meningkatkan kualitas hidup manusia. Kegiatan belajar mengajar dalam kaitannya dengan pendidikan sebenarnya telah dijalani seorang manusia sejak lahir, baik dalam lingkungan informal (keluarga dan masyarakat) maupun lingkungan formal (sekolah, perguruan tinggi, dan lembaga pendidikan formal lainnya). Kualitas pendidikan dipengaruhi oleh tiga hal yaitu tujuan, proses dan hasil pembelajaran. Tujuan pembelajaran mengarahkan proses pembelajaran yang akan dilakukan oleh pendidik. Proses pembelajaran dilakukan berdasarkan pada tujuan yang telah ditetapkan. Proses pembelajaran terjadi jika terdapat interaksi antara pendidik dan peserta didik. Hal ini erat kaitannya dengan sumber belajar yang diperlukan untuk melaksanakan proses pembelajaran. Sumber belajar berupa guru maupun dosen, buku, dan lingkungan sehingga tingkat kompetensi yang dicapai peserta didik menjadi tujuan akhir dalam melaksanakan proses pembelajaran. (Astrini dan Melly, 2016)

Sebagaimana proses belajar-mengajar yang umumnya berlaku, setiap mahasiswa yang mengikuti perkuliahan dalam satu semester harus melalui dua tahapan evaluasi proses pembelajaran, yaitu ujian tengah semester dan akhir semester. Dalam kegiatan evaluasi proses belajar-mengajar tersebut, tidak hanya mencakup pelaksanaan pengujian saja, namun juga ada aktivitas-aktivitas lain yang mengikutinya, diantaranya seperti pembuatan soal, pengawasan, pengoreksian, dan penilaian. (Bernadette Tynan, 2004)

Pelatihan mengerjakan soal ujian memiliki peran penting dalam persiapan ujian karena tidak hanya memperkuat pemahaman materi tetapi juga meningkatkan keterampilan dalam menjawab soal. Berlatih mengerjakan soal secara rutin, siswa dapat mengidentifikasi kelemahan mereka dan meningkatkan peluang mendapatkan nilai tinggi. Selain itu, pelatihan ini membantu siswa mengelola waktu selama ujian, sehingga mereka dapat menjawab soal dengan efisien. Mengetahui pola soal dan cara menjawabnya, siswa dapat lebih percaya diri menghadapi ujian sebenarnya. Pelatihan

ini juga membantu dalam memahami teknik-teknik pengerjaan soal dan memperoleh pengalaman yang diperlukan untuk menghadapi situasi ujian.(Plook TCAS, 2020)

Bank soal adalah kumpulan soal ujian yang dapat dengan mudah diakses untuk digunakan dalam mempersiapkan ujian. Penggunaan bank soal dan perangkat lunak bank soal yang baik adalah solusi untuk mempersiapkan soal baru dan kemudahan dalam pengelolaan administrasinya, baik soal yang digunakan oleh seorang guru dari kelas kecil, dosen universitas dengan berbagai bidang keilmuan, atau direktur program multifaset untuk sertifikasi, lisensi, akuntabilitas, atau survey penilaian acuan norma (Ward, 2005).

Berdasarkan observasi terhadap seluruh situs web Universitas Syiah Kuala, ditemukan bahwa tidak ada informasi atau tempat penyimpanan soal ujian lama yang dapat diakses oleh mahasiswa. Setiap mahasiswa memerlukan informasi terkait soal ujian lama untuk mempersiapkan diri menghadapi ujian, dan ketidak beradaan sumber daya ini membuat mahasiswa kesulitan dalam mencari informasi yang sesuai. Ketidak mampuan untuk mengakses bank soal juga membuat mahasiswa sulit untuk mengulang atau mempelajari materi sesuai kebutuhan ujian. Selain itu, ketiadaan akses ke soal ujian lama juga mengakibatkan mahasiswa kehilangan peluang untuk menjadi akrab dengan format soal dan jenis pertanyaan.

Cara mengatasi masalah ini adalah dengan membuat Situs web Bank Soal yang dapat diakses oleh mahasiswa. Bank soal ini harus mencakup soal ujian lama, soal kuis atau soal latihan dari berbagai mata kuliah agar mahasiswa dapat mempelajari dan mengulang materi kapan saja. Bank soal harus memiliki sistem manajemen yang memungkinkan mahasiswa untuk mengakses dan mencari informasi dengan cepat dan efisien.

Selain itu, bank soal sebaiknya mendukung kegiatan diskusi, komentar, atau *Feedback* terkait soal-soal yang ada di dalamnya, sehingga mahasiswa dapat mendapatkan dukungan tambahan dalam proses pembelajaran. Kemampuan untuk berbagi pengalaman dan pengetahuan dari mengerjakan ujian lama dengan orang lain dalam komunitas pendidikan akan menciptakan lingkungan belajar yang penuh kolaborasi dan saling mendukung.

1.2 RUMUS MASALAH

Rumusan masalah yang akan diselesaikan dalam Penelitian ini adalah sebagai berikut :

1. Bagaimana merancang aplikasi untuk mempermudah dalam mengakses dan mencari soal?
2. Bagaimana menciptakan mekanisme pengelolaan soal agar mudah diakses dan disimpan?
3. Bagaimana cara mendiskusikan soal yang ada di dalam Aplikasi?
4. Bagaimana menguji fungsionalitas menggunakan *Black Box*, serta melakukan *Usability testing* dengan metode *Usability Metric For User Experience* (UMUX) dan *Performance Test*.

1.3 TUJUAN PENELITIAN

Tujuan yang ingin dicapai pada penelitian ini yaitu sebagai berikut :

1. Mengembangkan aplikasi dengan desain inovatif untuk mempermudah akses terhadap soal dan meningkatkan efisiensi belajar.
2. Menerapkan metode penyimpanan yang lebih efisien untuk memudahkan manajemen soal.
3. Membangun sistem komentar untuk bisa mendiskusikan soal.
4. Menguji fungsionalitas menggunakan *Black Box*, , serta melakukan *Usability testing* dengan metode *Usability Metric For User Experience* (UMUX) dan *Performance Testing*.

1.4 MANFAAT PENELITIAN

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Memudahkan mahasiswa dalam mengakses dan mencari berbagai soal, meningkatkan efisiensi belajar, dan persiapan ujian.
2. Penerapan metode penyimpanan yang lebih efisien akan secara langsung meningkatkan optimasi manajemen soal, memudahkan pengelolaan, pencarian, dan aksesibilitas soal.
3. Mendorong interaksi antara pengguna melalui tempat komentar, membangun komunitas belajar yang aktif dan berkolaborasi.

4. Menyediakan sistem aplikasi web Bank Soal yang sesuai kebutuhan serta nyaman dan efektif untuk gunakan.

BAB II

TINJAUAN PUSTAKA

2.1 BANK SOAL

Selama ini pemahaman sebagian orang sering terkecoh antara bank soal dan kumpulan soal. Permasalahan pengertian bank soal berkaitan dengan asumsi bahwa bank soal merupakan kumpulan soal-soal semata. Untuk memahami pengertian bank soal, sebaiknya dipahami terlebih dahulu latar belakang munculnya bank soal. Setiap guru memiliki tugas dalam proses pembelajaran untuk melakukan penilaian terhadap hasil belajar siswa. Untuk melakukan penilaian diperlukan pengukuran, salah satunya dengan menggunakan tes.

Biasanya ketika seorang guru ingin melakukan penilaian terhadap siswa mengenai suatu materi tertentu, maka guru tersebut membuat dan/atau memilih beberapa soal yang sesuai dengan kompetensi yang akan dicapai siswa. Guru tersebut dikatakan telah membuat sebuah instrumen tes. Namun hal ini tentu saja mengandung banyak kelemahan.

Paling tidak ada dua kelemahan mendasar (Sumardyono, 2011). Perangkat tes yang dibuat setiap kali akan melakukan penilaian tidak teruji secara statistik karena tidak melalui ujicoba untuk menguji validitas dan reliabilitasnya, termasuk pengujian daya beda dan efektivitas pengecoh (distractor) pada tipe soal pilihan ganda. Belum lagi validitas konstruksinya dapat pula dipertanyakan karena penyusunannya yang seadanya dan kurang terencana. Jika pun ingin diperoleh tes yang benar-benar berkualitas, tentu memerlukan waktu yang tidak sedikit setiap kali akan melakukan penilaian. Keberadaan bank soal bermanfaat untuk mengatasi masalah guru ketika ingin melakukan penilaian. Setiap kali akan mengkonstruksi tes untuk penilaian, para guru tinggal mengambil butir-butir soal yang telah ada di bank soal. Selain mempermudah dalam penyusunan instrumen tes, juga menjamin kualitas instrumen yang akan dipakai.

Berikut ini akan dipaparkan pengertian bank soal yang dikemukakan oleh beberapa ahli. Gronlund dalam (Rudner, 1999) menyatakan *"Item banks are files of various suitable test items that are "coded by subject area, instructional level, instructional objective measured, and various pertinent item characteristics (e.g., item difficulty*

and discriminating power).” Bank soal adalah sekumpulan soal yang layak yang dikodekan berdasarkan subjek materi, tingkat pembelajaran, tujuan pembelajaran yang diukur (baca: SK atau KD), dan bermacam karakteristik lain butir soal yang berguna.

Bank soal tidak hanya digunakan dalam bidang akademik tetapi dapat juga digunakan dalam disiplin ilmu lain. Bank soal juga dapat dimanfaatkan untuk memenuhi kebutuhan tertentu dari bidang tertentu. Secara umum, bank soal memberikan keuntungan fasilitasi mudah, hasilnya cepat dan pelaksanaannya relatif murah (biaya dapat ditekan) karena tidak memerlukan kertas kerja dan pemeriksaan hasil dapat dilakukan saat itu juga (Widana, 2014).

Menurut Wright & Bell, (1984) menyatakan bahwa bank soal merupakan komposisi dari pertanyaan-pertanyaan yang terkoordinasi, dikembangkan, didefinisikan, dan dikuantifikasikan sehingga memberikan definisi yang operasional. Dengan demikian, bank soal bukan hanya kumpulan sejumlah soal saja.

Menurut Masters, G.N. dan Keeves, J.P. (Ed) dalam (Dhien & Kumaidi, 2021) mendefinisikan bank soal sebagai kumpulan yang relatif besar, yang mempermudah dalam memperoleh pertanyaan-pertanyaan penyusun tes. “Mudah” memiliki pengertian bahwa soal-soal tersebut diberi indeks, terstruktur, dan diberi keterangan sehingga mudah dalam pemilihannya untuk disusun sebagai perangkat tes pada suatu ujian.

Senada dengan pengertian-pengertian di atas, Choppin dalam (Latuconsina & Yunanto, 2017) memberikan definisi bahwa bank soal merupakan sekumpulan dari butir-butir tes yang diorganisasikan dan dikatalogkan untuk mencapai jumlah tertentu berdasarkan isi dan juga karakteristik butir. Karakteristik butir ini meliputi tingkat kesulitan, reliabilitas, validitas dan lain-lain.

Dalam pengembangan bank soal kecil, memang mungkin dilakukan tanpa bantuan computer. Tetapi dalam pengembangan bank soal yang besar, tidak mungkin mengembangkan bank soal tanpa bantuan computer. Hal ini disebabkan karena dalam pengembangan bank soal yang besar, ada beberapa tahapan yang tidak mungkin dilakukan tanpa bantuan computer (Retnawati, 2014).

2.2 WEBSITE

Website merupakan suatu layanan yang dapat digunakan oleh pemakai komputer di mana komputer yang digunakan harus terhubung ke internet (Kurtiyahningsih, 2011)

Website atau situs *web* adalah setiap computer atau tempat (*space*) dalam sebuah *computer* yang terhubung dengan internet dan menjalankan fungsi dan proses sebagai *server web* yang berisi dokumen-dokumen dalam *format* HTML (Husda & Wangdra, 2016). Selain itu ada juga yang berpendapat bahwa, *website* merupakan metode untuk menampilkan informasi di internet, baik itu berupa teks, gambar video dan suara maupun interaktif yang memiliki keuntungan yang menghubungkan (*link*) dari dokumen dengan dokumen lainnya (*hypertext*) yang dapat di akses melalui *browser* (Yuhefizar, 2012).

2.3 LARAVEL

Framework Laravel merupakan perangkat lunak untuk membuat sebuah web yang memudahkan programmer yang didalamnya ada berbagai fungsi. Ada plugin dan konsep yang berguna untuk membuat suatu sistem tertentu agar terstruktur. Laravel merupakan *framework* aplikasi web dengan sintaks yang elegan dan ekspresif yang dibangun dengan menyenangkan dan dari pengalaman pembuat yang kreatif agar dapat memuaskan pengguna, sampai saat ini laravel pernah menjadi yang banyak digunakan ditahun 2015 (SitePoint, 2015).

2.4 PHP

Hypertext Preprocessor atau sering disebut PHP merupakan sebuah bahasa *scripting* tingkat tinggi yang dipasang pada dokumen HTML. Sebagian besar sintaks dalam PHP mirip dengan bahasa C, Java dan Perl. Namun, pada PHP ada beberapa fungsi yang lebih spesifik. Tujuan utama dari penggunaan bahasa ini adalah untuk memungkinkan perancang *web* yang dinamis dan dapat bekerja secara otomatis (Setiawan, 2017). PHP adalah bahasa pemrograman *script server-side* yang didesain untuk pengembangan *web*. Selain itu, PHP juga bisa digunakan sebagai bahasa pemrograman umum (wikipedia). PHP dikembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan sekarang dikelola oleh The PHP Group. Situs resmi PHP beralamat di

<http://www.php.net>. PHP disebut bahasa pemrograman *server side* karena PHP diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman *client-side* seperti JavaScript yang diproses pada *web browser (client)* (Sabanayo, 2014).

2.5 MYSQL

MySQL termasuk jenis RDBMS(*Relation Database Management System*), pada MySQL sebuah database mengandung satu beberapa tabel, table terdiri dari sejumlah kolom dan baris. Dalam bahasa SQL, pada umumnya informasi tersimpan dalam tabel-tabel yang secara logik merupakan struktur dua dimensi yang terdiri atas baris-baris data yang berada dalam satu atau lebih kolom.

Menurut Sidik (2014:333)”MySQL merupakan *software database* yang termasuk paling populer dilingkungan linux, kepopuleran ini ditunjang karena Performansi *query* dari databasenya yang saat itu biasa dikatakan paling cepat dan jarang bermasalah”.

Menurut Setiawan dalam jurnal Khairil dkk (2012 : 60) MySQL adalah *database* yang menghubungkan *script* PHP menggunakan perintah *query* dan *escape character* yang sama dengan PHP.

Secara umum bahasa SQL dibagi menjadi tiga bagian (Setiawan dalam khairil, dkk 2012:60) yaitu:

1. DDL (*Data Definition Language*) Digunakan untuk membangun objek-objek dalam sebuah database seperti tabel. Dengan DDL kita dapat menentukan tata letak baris, definisi kolom, kolom-kolom kunci, lokasi file dan strategi penyimpanan.
2. DML (*Data Manipulasi Language*) Digunakan dalam manipulasi suatu tabel didalam *database* (menambah, mengedit, mencari dan menghapus). Perintah-perintah DML adalah SELECT, INSERT, UPDATE dan DELETE.
3. DCL (*Data Control Language*) Digunakan untuk menangani masalah keamanan dalam *database server*, dan hak-hak pada objek-objek *database*. Perintah-perintah yang dipakai Berdasarkan teori tersebut dapat disimpulkan bahwa MySQL adalah perangkat lunak jenis *database* yang digunakan untuk membangun aplikasi *web* dengan menggunakan perintah *query* dan *escape character* yang sama dengan PHP

dan paling terkenal dilingkungan linux karena performansi *query* dari *database* jarang bermasalah.

2.6 UNIFIED MODELING LANGUAGE

Unified Modeling Language (UML) merupakan suatu bahasa dan akan digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. Kebutuhan dan penggunaan akan UML dari hari ke hari dalam industri terus meningkat. Sehingga dapat dikatakan standar dalam menjadikannya bahasa pemodelan yang umum dalam pengembangan suatu sistem. (Rama dan Jones, 2008).

UML memiliki berbagai macam pemodelan dalam diagram untuk memodelkan aplikasi yang berorientasi pada objek. Dari berbagai macam pemodelan diagram tersebut yang paling sering digunakan dalam pembangunan aplikasi yang berorientasi pada objek yaitu *Use Case Diagram*, *Sequence Diagram*, *Collaboration Diagram*, dan *Class Diagram* (Aziz, 2005). Pada penelitian ini, penulis akan menggunakan *Use Case Diagram*, dan *ActivityDiagram*.

2.6.1 USE CASE DIAGRAM





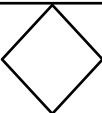
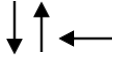
Use case Diagram adalah model diagram yang akan digunakan ketika menjelaskan secara visual terkait konteks dari interaksi yang akan berlangsung antara aktor dengan sistem yang dirancang. (Kurniawan, 2017)

Use case Diagram dapat digambarkan melalui elips yang berisi kegiatan dari sistem dan akan dilakukan oleh aktor atau yang bersangkutan. Elips yang saling berkaitan akan mewakili satu tujuan dari sistem. Sehingga hal tersebut dapat dengan mudah menjelaskan urutan langkah yang harus diambil untuk mencapai tujuan tertentu. (Bentley dan Witten, 2007).

2.6.2 ACTIVITY DIAGRAM

Activity Diagram adalah model diagram yang menggambarkan aliran aktivitas yang terjadi pada suatu sistem dalam bentuk kumpulan aksi, dimulai dari bagaimana aksi tersebut dimulai, hasil dari keputusan yang didapatkan hingga proses akhir dari rangkaian aksi. (Suendri, 2018).

Symbol-simbol yang digunakan pada *activity Diagram* dapat dilihat pada Gambar 2.1 berikut.


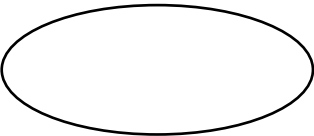
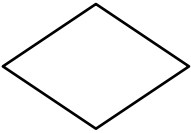

No	Gambar	Nama	Keterangan
1		Action	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
2		Activity	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
3		Initial Node	Bagaimana objek dibentuk atau diawali.
4		Activity final Node	Bagaimana objek dibentuk dan diakhiri.
5		Decision	Digunakan Untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu.
6		Line Connection	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya.

Gambar 2.1 Simbol-simbol pada *Activity Diagram*.

2.7 ENTITIY RELATIONSHIP DIAGRAM

Entity Relationship Diagram (ERD) merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh *System Analys* dalam tahap analisis persyaratan proyek pengembangan sistem. Sementara seolah-olah teknik diagram atau alat peraga memberikan dasar untuk desain database relasional yang mendasari sistem informasi yang dikembangkan. ERD bersama-sama dengan detail pendukung merupakan model data yang pada gilirannya digunakan sebagai spesifikasi untuk database. (Brady dan Loonam : 2010)

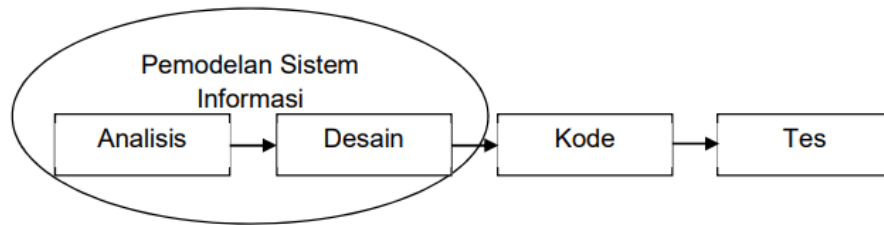
Tabel 2.1 *Entity Relationship Diagram* (ERD)

No.	Simbol	Keterangan
1		Entitas
2		Atribut
3		Hubungan
4		Garis

2.8 WATERFALL MODEL

Model proses perangkat lunak merupakan gambaran dari proses pengembangan perangkat lunak. Setiap model proses perangkat lunak dapat menjelaskan proses dari sudut pandang tertentu, sehingga dapat memberikan informasi mengenai proses yang dilakukan. Model proses pengembangan dari sistem informasi yang akan dibangun menggunakan Model *Waterfall*. Menurut Pressman (2012) Model Waterfall (model air terjun) merupakan suatu model pengembangan secara sekuensial. Model *Waterfall*

bersifat sistematis dan berurutan dalam membangun sebuah perangkat lunak. Proses pembuatannya mengikuti alur dari mulai analisis, desain, kode, pengujian dan pemeliharaan. Model pengembangan *Waterfall* memiliki beberapa kelebihan, antara lain: dapat mudah dipahami dan dapat diterapkan dalam proses pengembangan perangkat lunak. Gambar 2.2 merupakan model pengembangan *Waterfall*



Gambar 2.2 Model Pengembangan *Waterfall* (Pressman 2002)

Tahap-tahap dari model pengembangan Waterfall ini, yaitu:

1. Analisis kebutuhan perangkat lunak.

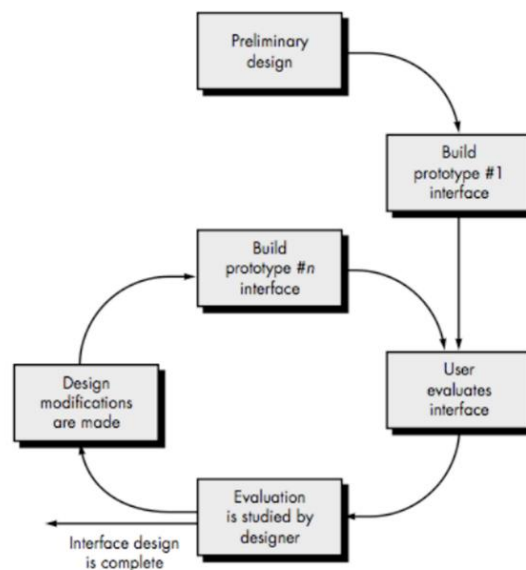
Analisis kebutuhan perangkat lunak merupakan tahap pengumpulan kebutuhan yang diintensifkan dan difokuskan. Untuk mengetahui kebutuhan perangkat lunak, seperti apa yang dibutuhkan oleh pengguna. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan. Tujuan dari analisis kebutuhan yaitu merangkum hal-hal apa saja yang diinginkan pengguna dan mencari kebutuhan apa saja yang diperlukan dalam mengembangkan perangkat lunak. Kebutuhan untuk sistem maupun perangkat lunak didokumentasikan dan dilihat kembali oleh pengguna apakah sudah sesuai dengan yang diinginkan (Pressman 2002, 219).

2. Desain

Desain merupakan suatu tahapan yang berfokus pada desain untuk membuat perangkat lunak seperti: struktur data, arsitektur perangkat lunak, *user interface* (antarmuka), dan prosedur pengkodean. Tahap desain dilakukan dengan menerjemahkan kebutuhan perangkat lunak berdasarkan dari hasil analisis kebutuhan ke dalam bentuk desain, sehingga dapat diimplementasikan menjadi program pada tahap implementasi. Desain didokumentasikan dan menjadi bagian dari konfigurasi perangkat lunak (Pressman 2002). Ada beberapa jenis pemodelan

perangkat lunak, salah satu pemodelan perangkat lunak yang digunakan dalam tahapan ini yaitu *Unified Modeling Language* (UML), yang merupakan gambaran mengenai perangkat lunak yang akan dibuat. UML dibuat untuk mempermudah pengembang dalam membuat suatu perangkat lunak (Pressman 2002, 987).

Dalam tahap desain dilakukan evaluasi agar memenuhi kebutuhan pengguna dengan membuat sebuah *prototype*, kemudian diberikan kepada pengguna untuk dievaluasi, apakah sudah sesuai apa belum. Apabila belum sesuai kemudian diperbaiki dan membuat *prototype* yang baru, kemudian diberikan kepada pengguna lagi sampai memenuhi kebutuhan pengguna. Untuk lebih jelasnya dapat dilihat pada lingkaran evaluasi pada Gambar 2.3.



Gambar 2.3 Siklus Evaluasi Desain *Interface* (Pressman 2002).

3. Implementasi (Pembuatan Kode)

Desain merupakan tahap menerjemahkan desain sistem kedalam perangkat lunak berdasarkan desain yang telah dibuat. Penerjemahan desain menggunakan kode bahasa program sehingga dapat berjalan dengan baik. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis. Hasil akhir dari tahap ini adalah menghasilkan sistem informasi yang sesuai dengan desain yang sudah dibuat (Pressman 2002).

Dalam tahap implementasi dilakukan evaluasi untuk mengurangi kesalahan dengan menjalankan kode yang dibuat, apakah sudah sesuai dengan spesifikasi

yang ditentukan dan tidak terjadi kesalahan. Evaluasi berguna untuk mengurangi kesalahan yang terjadi sebelum semua bagian digabungkan menjadi satu kesatuan perangkat lunak.

4. Pengujian

Proses pengujian berfokus untuk mengurangi kesalahan yang terjadi ketika sistem informasi dijalankan dan menguji kualitas dari sistem informasi. Pengujian sistem terdiri dari pengujian fungsi dan kualitas sistem informasi. Pengujian fungsi digunakan untuk mengecek apakah fungsi yang dilakukan berjalan dengan baik. Pengujian dilakukan dengan menggunakan White-box testing dan Black-box testing. White-box testing digunakan untuk melakukan pengecekan tentang logika internal dan struktur kode, apakah sudah benar (Khan 2012). Black-box testing digunakan untuk menguji sistem informasi apakah masukan yang diterima dan keluaran sudah beroperasi dengan benar. Pengujian kualitas sistem digunakan untuk mengecek apakah sistem yang dibuat sudah layak untuk digunakan (Pressman 2002).

2.9 BLACK BOX TESTING

Black Box Testing adalah pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak (Watkins dan Mills, 2010). *Black Box Testing* bekerja dengan cara mengabaikan mekanisme internal atau struktur sistem dan berfokus pada output yang dihasilkan sebagai respons dari *input* yang dipilih dan dieksekusi. Pengujian *Black Box* dilakukan untuk mengevaluasi pemenuhan dari sistem dengan persyaratan fungsional yang ditentukan dan hasil prediksi yang sesuai (Khan dkk., 2011). Fokus utama dari *Black Box Testing* adalah spesifikasi fungsionalitas perangkat lunak. Menurut Mustaqbal dkk. (2015), dalam *black box testing* biasanya menemukan hal-hal berikut:

1. Kesalahan dari user interface.
2. Kesalahan struktur data dan akses database.
3. Kesalahan performa.
4. Kesalahan inisiasi dan terminasi.
5. Fungsi yang tidak benar atau tidak ada.

2.10 USABILITY TESTING

Usability testing merupakan kegiatan pengujian pengumpulan data mengenai suatu produk dalam tahap pengembangan. Tujuan utama *usability testing* antara lain untuk mengumpulkan data kuantitatif dengan mengukur waktu untuk mempelajarinya, performa kecepatan dan tingkat *human error* (Wahl, 2000). *Usability testing* dilakukan dengan melibatkan pengguna dalam proses pengujian usabilitynya. Pengujian ini diharapkan akan mendapatkan kelebihan dan kekurangan dari setiap aspek yang ada pada aplikasi. Oleh karena itu, diperlukan dokumentasi pengalaman aktual seluruh calon pengguna aplikasi atau produk saat dievaluasi (Wesfix, 2017).

2.9.1 Usability Metric for User Experience (UMUX)

Usability Metric for User Experience (UMUX) merupakan metode pengujian dengan *usability testing* yang akan digunakan pada penelitian ini. UMUX merupakan skala *Likert* dengan menggunakan empat item untuk melakukan suatu penilaian subjektif pada manfaat aplikasi. UMUX dibuat dengan hasil yang diperoleh dengan skala *system usability* yang memiliki 10 item sebanding. UMUX sangat mudah digunakan sebagai modul kegunaan dalam matriks user experience. UMUX bertujuan untuk menyederhanakan kinerja dari *system usability scale* (SUS) dengan penyesuaian langkah yang penting. UMUX dapat dilakukan secara online sebagai survei dalam pengujian kegunaan aplikasi (Finstad, 2010). UMUX terdiri dari 4 pertanyaan dengan skala *Likert* 1-7. Berikut merupakan daftar pertanyaan metode UMUX yang dapat dilihat pada Tabel 2.1.

Tabel 2.2 Daftar Pertanyaan Metode UMUX (Finstad, 2010)

No.	Pertanyaan	Skor
1	Aplikasi ini sesuai dengan kebutuhan saya	1-7
2	Saya memiliki pengalaman buruk dalam menggunakan aplikasi ini	1-7
3	Aplikasi ini mudah digunakan	1-7
4	Saya harus menghabiskan banyak waktu untuk menggunakan aplikasi ini dengan benar	1-7

Rumus perhitungan dapat dilihat pada persamaan 2.1.

$$UMUX = \frac{[P1 - 1] + [7 - P2] + [P3 - 1] + [7 - P4]}{24} \times 100$$

Keterangan:

P1 = Pertanyaan satu.

P2 = Pertanyaan dua.

P3 = Pertanyaan tiga.

P4 = Pertanyaan empat.

Studi terbaru menunjukkan bahwa skor SUS lebih mendekati skor UMUX-lite. Sedangkan skor dari UMUX menghasilkan nilai yang lebih tinggi. jika hanya bergantung kepada skor UMUX, praktisi mungkin tidak perlu melakukan analisis usability lebih lanjut disebabkan nilai dari UMUX boleh jadi akan bernilai sangat memuaskan. Namun dengan skor SUS dan UMUX-lite, memiliki kemungkinan lain sehingga dapat melakukan perbaikan pada keseluruhan pengalaman interaksi pengguna. Hasil dari skor UMUX kadang terlalu positif sehingga perhitungan skor UMUX lebih efektif menggunakan persamaan UMUX-LITE. UMUX-LITE sendiri memiliki korelasi dalam menghitung skor sehingga untuk menghitung skor UMUX-LITE menjadi SUS dapat menggunakan persamaan 2.2 (Borsci dkk., 2015).

$$UMUX - LITE = .65([Item1score] + [Item2score] - 2)100/12) + 22.9$$

Keterangan:

Item 1 score = Pertanyaan satu.

Item 2 score = Pertanyaan tiga.

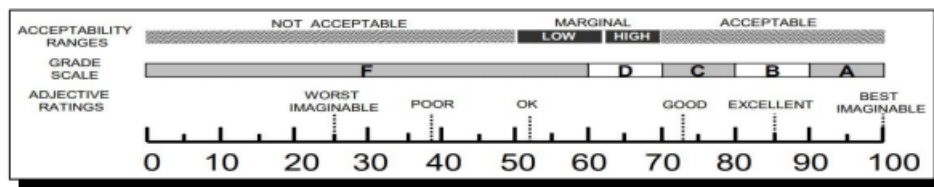
Berikut merupakan pedoman skor SUS yang dapat dilihat pada Tabel 2.2

Tabel 2.3 Skor SUS (Borsci dkk., 2015)

Grade	Range
Grade A+	(84.1-100)
Grade A	(80.8-84.0)
Grade A-	(78.9-80.7)
Grade B+	(77.2-78.8)
Grade B	(74.1-77.1)

Grade B-	(72.6-74.0)
Grade C+	(71.1-72.5)
Grade C	(65.0-71.0)
Grade C-	(62.7-64.9)
Grade D	(51.8-62.6)
Grade F	(0-51.7)

Hasil pengujian dapat diterima atau tidak dapat dilihat pada gambar 2.4.



Gambar 2.4 Perbandingan skala dan skor pada SUS (Bangor dkk., 2009)

2.11 PERFORMANCE TESTING

Performance testing, sesuai dengan namanya, adalah pengujian *non-functional* yang dilakukan untuk mengetahui *behavior* atau performa dari sebuah sistem saat dihadapkan pada situasi yang bervariasi sehingga dapat menghindari kegagalan sistem seperti contohnya system down. (Hariaty Lumbantobing, 2021)

Dalam *performance testing* ada beberapa faktor yang menjadi fokus pada pengujian ini, diantaranya :

- **Speed** : melihat kecepatan sistem dalam memberikan respon dari setiap *request*
- **Scalability** : menentukan berapa jumlah *load/user/thread* maksimum yang dapat ditangani system.
- **Stability** : menganalisis kondisi sistem saat diberikan beban yang bervariasi, apakah stabil atau tidak.

Secara umum, ada 4 jenis *performance testing* dengan tujuan pengujian yang berbeda-beda. Saya akan menjabarkannya secara detail dengan contoh kasus yang umum terjadi.

1. Load Testing

Merupakan bagian dari teknik pengujian *performance* dimana tujuan pengujian ini adalah untuk menentukan *target load* dari sebuah sistem.

Pengujian ini diperlukan saat adanya sebuah *event* musiman yang menyebabkan lalu lintas (*traffic*) sistem bertambah seiring berjalannya waktu.

Contohnya, pada sebuah aplikasi penjualan tiket yang *traffic*-nya perlahan akan bertambah saat sebuah *event* mulai mendekati. Misalnya acara tahun baru yang terjadi di tanggal 1 Januari, mungkin jauh-jauh hari seiring mendekatnya tahun baru, *traffic* dari aplikasi penjualan tiket juga akan bertambah sehingga pengujian ini diperlukan untuk mengetahui ketahanan sistem dengan menaikkan *load* perlahan-lahan pada setiap skenario-nya, kemudian menganalisis penggunaan *resource*-nya apakah ada perbedaan yang signifikan atau tidak.

2. Stress Testing

Merupakan bagian dari teknik pengujian *performance* yang dijalankan untuk mengetahui limit atau batas kemampuan dari sebuah sistem dengan cara membanjiri lalu lintas (*traffic*) sistem sampai sistem tersebut *crash*. Salah satu alasan utama dilakukannya pengujian ini adalah untuk menghindari terjadinya kegagalan sistem (*system down*).

Contoh kasus yang paling umum adalah saat sebuah aplikasi *e-commerce* mengadakan promo besar-besaran, sehingga banyak *user* yang mengakses aplikasi tersebut di jam tertentu (misalnya ada waktu tertentu untuk menggunakan promo) pada waktu yang bersamaan. Setelah dilakukannya pengujian ini, diharapkan tim pengembang dapat membangun sebuah sistem yang lebih kebal terhadap kasus *traffic* yang dibanjiri oleh banyak *user*.

3. Endurance Testing

Merupakan bagian dari teknik pengujian *performance* yang dijalankan untuk menganalisis *behavior* atau performa sistem dengan *load* yang normal namun dalam jangka waktu yang panjang. Pengujian ini dilakukan untuk menghindari terjadinya kebocoran memori (*memory leak*). Di beberapa instansi, *endurance testing* tidak menjadi bagian dari tanggung jawab seorang *Software Tester/QA*. Pengujian ini biasanya dilakukan oleh seorang *performance engineer*, atau *role* yang lebih ahli di bidang ini.

4. Spike Testing

Merupakan bagian dari teknik pengujian *performance* yang dijalankan untuk mengetahui jumlah *load system* apakah sesuai dengan ekspektasi. *Spike* dan *load testing* memiliki tujuan yang mirip. Bedanya dengan *load testing* adalah pengujian ini dilakukan dengan mengirim *load* yang banyak dalam waktu bersamaan. Contohnya adalah saat mengakses *website* hasil ujian SBMPTN. Dengan pengujian ini, kita dapat mengetahui *user* atau *load* maksimum yang dapat mengakses *website* tersebut pada waktu yang bersamaan.

2.12 PENELITIAN TERKAIT

Penelitian terkait yang dilakukan oleh (Musriatun Napiah, 2020) dalam merancang sistem penyajian Bank Soal untuk jenjang sekolah menengah atas berbasis web. Penelitian ini bertujuan untuk merancang aplikasi yang mampu mendokumentasikan seluruh soal dan kegiatan selama pelaksanaan ujian dan pemberian tugas. Analisis dan desain menggunakan metode *waterfall*. Penelitian diawali dengan proses melakukan studi literature, pengumpulan data, analisis sistem dan kebutuhan, membuat pemodelan sistem menggunakan UML, merancang implementasi aplikasi, melakukan pengujian dengan *Black Box Testing* dan *maintance* atau pemeliharaan data sistem. Aplikasi Bank Soal berbasis web dirancang memiliki fitur untuk pendataan soal dan penjadwalannya, monitoring pelaksanaan ujian dan *monitoring upload* tugas siswa. Hasil penelitian ini dapat digunakan sebagai dasar pengembangan aplikasi Bank Soal berbasis web dalam rangka pengelolaan evaluasi belajar yang lebih interkatif di studi kasus untuk jenjang Sekolah Menengah Atas, sehingga tidak hanya terfokus di pembelajaran di kelas, tetapi siswa juga bisa mengaksesnya via internet di mana saja.

Penelitian terkait yang dilakukan oleh (Mustamiin, 2020) dengan judul penelitian Rancang Bangun Sistem Manajemen Soal Dan Ujian Berbasis *Website* Menggunakan *Framework* Laravel. Penelitian ini bertujuan untuk mempermudah dosen dalam memberikan ujian dan proses pemberian nilai ujian yang efektif dan efisien. Hasil dari penelitian ini, dapat disimpulkan bahwa dengan adanya sistem

manajemen bank soal dan ujian berbasis *website*, diharapkan dapat membentuk kinerja dosen dalam mengelola soal dan ujian.

Penelitian terkait yang dilakukan oleh (Natayasa, 2020) dengan judul Sistem Informasi Silabus, RPP, dan Bank Soal Berbasis Web. Penelitian ini menjelaskan bahwa dibutuhkan sistem informasi untuk memastikan kesesuaian antara soal dengan materi yang diajarkan. Model pengembangan yang digunakan untuk membangun sistem informasi ini adalah model *Waterfall*. Hasil akhir dari sistem informasi ini adalah dapat digunakan oleh guru untuk menyusun Silabus, Rencana Pelaksanaan Pembelajaran, dan Soal yang sesuai dengan materi yang diajarkan.

BAB III METODE PENELITIAN

3.1 TEMPAT DAN WAKTU PENELITIAN

Penelitian dilakukan di fakultas MIPA Universitas Syiah Kuala Banda Aceh. Waktu yang diperlukan untuk penelitian ini selama kurang lebih 5 bulan dari bulan Maret hingga bulan juli 2024.

Tabel 3.1 Jadwal pelaksanaan penelitian bulan Maret-Mei 2024

No	Kegiatan	Maret				April				Mei			
		Minggu Ke				Minggu Ke				Minggu Ke			
		1	2	3	4	1	2	3	4	1	2	3	4
1	Pengajuan Judul												
2	Studi Literatur												
3	Penyusunan BAB I												
4	Penyusunan BAB II												
5	Penyusunan BAB III												
6	Bimbingan dan Revisi Proposal												
7	Seminar Proposal												
8	Revisi Seminar Proposal												
9	Pelaksanaan Penelitian												

Tabel 3.2 Jadwal pelaksanaan penelitian bulan Juni-Juli 2024

No	Kegiatan	Juni				Juli			
		Minggu Ke				Minggu Ke			
		1	2	1	1	1	1	3	4
1	Pelaksanaan Penelitian								
2	Penyusunan BAB IV								
3	Penyusunan BAB V								
4	Seminar Hasil								
5	Revisi Seminar Hasil								
6	Sidang Skripsi								
7	Revisi Hasil Sidang Skripsi								

3.2 ALAT DAN BAHAN

Berikut ini adalah alat dan bahan yang digunakan dalam mengerjakan Tugas Akhir ini.

3.2.1 Perangkat Keras

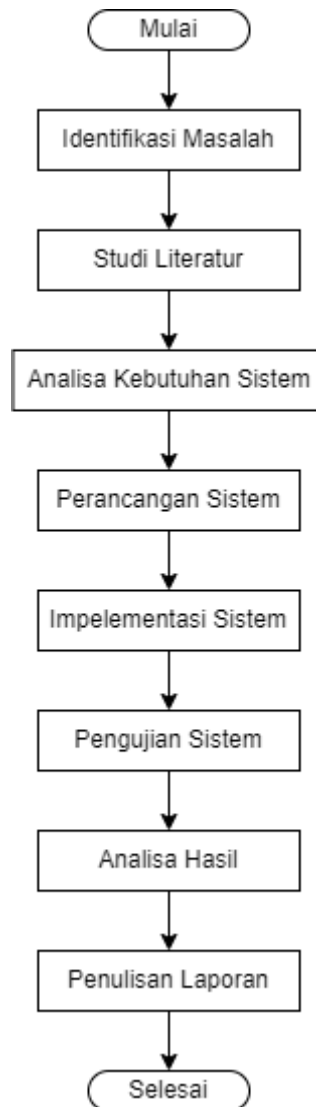
- Laptop Acer Nitro 5 dengan Prosesor Intel® Core™ I5 Gen 7, RAM 16 GB dan Penyimpanan dengan HDD 1TB + SSD 256GB

3.2.2 Perangkat Lunak

- Sistem Operasi Windows 11 Pro
- PHP versi 8.2.12
- *Framework* Laravel versi 10.43.0
- *Java Development Kit* versi 21.0.2
- Apache JMeter versi 5.6.3
- XAMPP versi 3.3.0
- Aplikasi *Design Prototype* Figma
- *Browser*: Google Chrome
- *Visual Studio Code*

3.3 METODE PENELITIAN

Metode penelitian yang dilakukan pada penelitian ini menggunakan metode *waterfall* yang terdiri dari beberapa tahapan, tahapan-tahapan tersebut dapat dilihat pada gambar 3.1 di bawah ini.



Gambar 3.1 Skema Proses Kerja

3.3.1 Identifikasi Masalah

Tahapan ini merupakan tahap yang dilakukan untuk merumuskan masalah, menentukan batasan masalah, serta tujuan dan manfaat dari aplikasi yang akan dibuat.

3.3.2 Studi Literatur

Studi literatur merupakan tahap yang dilakukan untuk mengumpulkan, mempelajari dan mencari bahan sebagai referensi atau landasan untuk dapat menyelesaikan setiap rumusan masalah selama proses penelitian berlangsung. Studi literatur dapat dilakukan dengan cara mencari situs *website*, artikel-artikel terpercaya dan jurnal-jurnal yang terkait dengan penelitian.

3.3.3 Analisa Kebutuhan Sistem

Tahapan analisa kebutuhan sistem ini yaitu menganalisa kebutuhan sistem dengan merancang hal-hal dasar dalam perancangan sistem. Pada tahap analisa kebutuhan sistem dan tahapan-tahapan selanjutnya, penulis akan menerapkan model *Unified Modeling Language* (UML) yang merupakan sebuah "bahasa" yang telah umum digunakan untuk visualisasi, merancang, dan mendokumentasikan sistem perangkat lunak. Adapun konsep analisisnya yaitu dengan mengidentifikasi pengguna, membuat *Use Case Diagram* dan *Activity Diagram*.

3.3.3.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang didefinisikan berdasarkan fungsionalitas sistem. Berikut adalah kebutuhan fungsionalnya:

1. Pengguna dapat melakukan *login* ke dalam sistem menggunakan akun mahasiswa dan dosen USK.
2. Pengguna dapat melihat dan mengunduh soal secara langsung.
3. Soal-soal harus dapat dikelompokkan berdasarkan mata kuliah.
4. Pengguna dapat berinteraksi dalam diskusi tentang soal dengan komentar dari pengguna lain.
5. Sistem harus memungkinkan pengguna akun dosen untuk menambah, mengedit, dan menghapus soal.
6. Pengguna dapat mengedit profil mereka.

3.3.3.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional merupakan kebutuhan perangkat keras dan perangkat lunak yang menjadi batasan kebutuhan apabila tidak dapat dikerjakan oleh sistem itu sendiri. Berikut adalah kebutuhan non fungsionalnya:

1. Sistem dapat dijalankan oleh beberapa *software web browser*.
2. Sistem memiliki *database* yang memadai
3. Sistem memiliki desain antar muka yang dapat dengan mudah dipahami dan digunakan
4. Sistem harus minim kesalahan dan masalah

3.3.4 Perancangan Sistem

Setelah melakukan tahap analisa kebutuhan sistem, maka selanjutnya dilakukanlah adalah tahap perancangan. Pada tahap ini, dilakukan perancangan *use case diagram*, *activity diagram* dan desain *interface* sistem informasi yang akan dibangun.

3.3.4.1 Identifikasi Pengguna

Berikut merupakan daftar pengguna yang telah penulis tetapkan :

1. Mahasiswa

Mahasiswa adalah pengguna yang menggunakan aplikasi untuk mengakses, mencari, dan mengerjakan soal-soal ujian sebagai bagian dari persiapan mereka. Mereka juga dapat berinteraksi dalam diskusi tentang soal dengan pengguna lainnya.

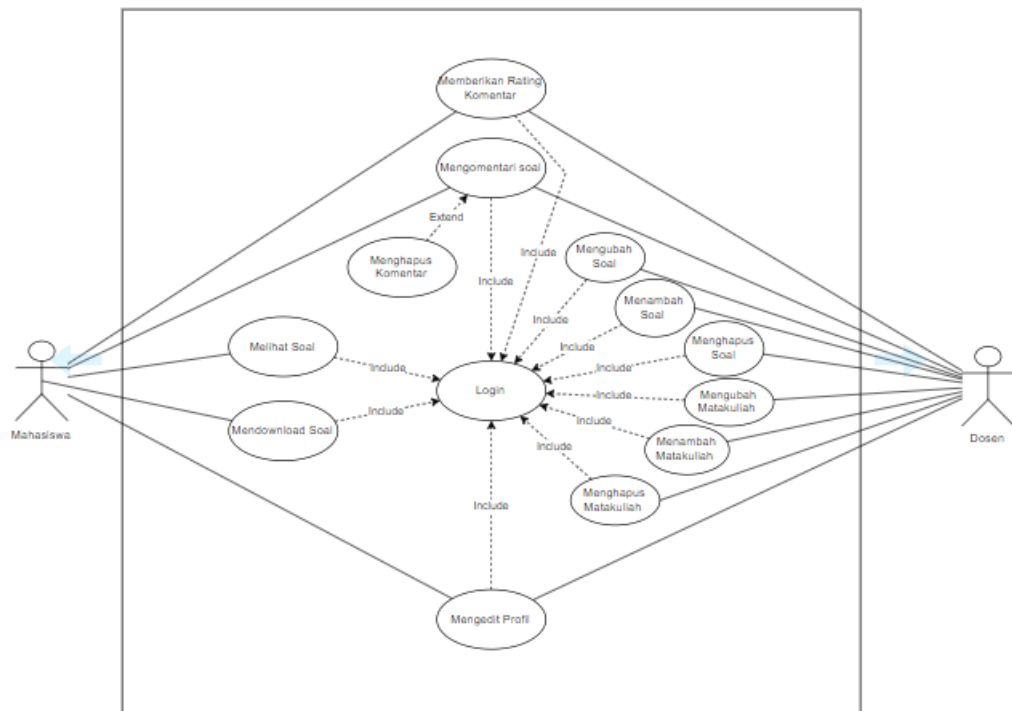
2. Dosen

Dosen adalah pengguna yang memiliki peran tambahan dalam sistem. Mereka dapat menambah, mengedit, dan menghapus soal-soal dalam bank soal. Selain itu, mereka juga dapat berpartisipasi dalam diskusi dengan pengguna lainnya.

3.3.4.2 Use Case Diagram

Use case diagram adalah diagram yang menjelaskan interaksi antara komponen atau pihak yang terkait (aktor) dan sistem. selain itu *Use case diagram* dapat mendeskripsikan kebutuhan fungsional dari aplikasi sehingga dapat mengetahui apa saja yang dapat dilakukan oleh sistem. setiap *case* yang terdapat di dalam *Use case* akan menjadi fitur aplikasi pada tahap pembuatan aplikasinya. *Use case* memiliki beberapa hubungan seperti *include* dan *extend*. *Use case* yang memiliki hubungan *include* berarti tidak dapat dilakukan sebelum *Use case* yang dituju dilakukan. Sedangkan *extend* berarti *Use case* tersebut merupakan aktivitas hasil perluasan dari *Use case* yang dituju. *Use case* yang dituju dari hubungan *extend* masih dapat dilakukan walaupun *Use case* perluasannya tidak dilakukan.

Gambar 3.2 berikut ini merupakan *Use Case Diagram* yang telah dirancang:



Gambar 3.2 merupakan *Use case* yang menjelaskan interaksi antara aktor dan sistem. Aktor terbagi menjadi dua peran yaitu Mahasiswa dan Dosen. Adapun penjelasan terkait *use case* kedua aktor tersebut adalah sebagai berikut:

1. Mahasiswa

Mahasiswa harus *login* terlebih dahulu menggunakan akun mahasiswa USK mereka sebelum dapat melihat berbagai soal yang tersedia dalam bank soal. Setelah berhasil *login*, mahasiswa memiliki akses penuh untuk melihat dan mengunduh soal-soal secara langsung dari bank soal tersebut. Mereka juga diberikan kemampuan untuk memberikan komentar atau membagi jawaban terhadap soal-soal yang mereka lihat, sehingga memungkinkan adanya interaksi antara mahasiswa dalam memahami dan memecahkan soal-soal tersebut. Selain itu, mahasiswa juga diberikan opsi untuk memberikan nilai atau *rating* terhadap komentar yang diberikan oleh pengguna lain terhadap soal tertentu. Hal ini memberikan gambaran kepada mahasiswa lainnya tentang kualitas dan kebergunaan komentar tersebut. Terakhir,

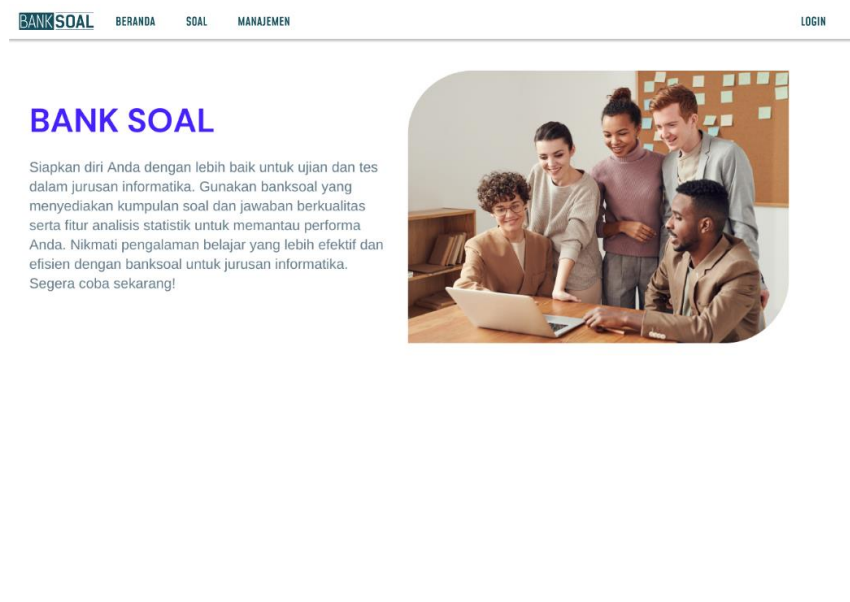
mahasiswa juga memiliki kemampuan untuk mengedit profil mereka agar informasi profil tetap terkini.

2. Dosen

Dosen dapat *login* ke dalam sistem menggunakan akun dosen USK mereka. Setelah berhasil *login*, dosen memiliki akses penuh untuk melakukan sejumlah tindakan terkait soal dalam bank soal. Mereka dapat menambahkan soal baru, mengubah konten soal yang sudah ada, atau bahkan menghapus soal yang tidak relevan atau sudah tidak digunakan. Selain itu, dosen juga memiliki fasilitas untuk memberikan komentar atau saran tambahan terhadap setiap soal yang ada di bank soal. Jika ada komentar yang dianggap tidak pantas atau tidak sesuai, dosen juga memiliki kewenangan untuk menghapusnya. Selain aktivitas terkait dengan soal, dosen juga diberikan kemampuan untuk mengedit profil mereka, sehingga mereka dapat memperbarui informasi pribadi.

3.3.4.3 Design Prototype Aplikasi

1. *Design Prototype* Halaman Beranda



Gambar 3.3 *Design Prototype* Halaman Beranda

2. Design Prototype Halaman Login

Gambar 3.4 Design Prototype Halaman Login

3. Design Prototype Halaman Cari Soal

Semester 1				
No	Kode	Nama Matakuliah	SKS	
1	MPA104	PENGANTAR FISIKA	3	Lihat
2	MPA105	PENGANTAR BIOLOGI	3	Lihat
3	MPA106	PENDIDIKAN PANCASILA DAN KEWARGANEGARAAN	2	Lihat
4	MPA107	PENGANTAR KIMIA	3	Lihat
5	MPA108	PENGANTAR KALKULUS	3	Lihat
6	MPA109	PENGANTAR TIK	2	Lihat
7	MPA110	PEMBINAAN KARAKTER I	0	Lihat
8	MPA111	PENGETAHUAN KEBENCANAAN DAN LINGKUNGAN	2	Lihat

Gambar 3.5 Design Prototype Halaman Cari Soal

4. *Design Prototype* Halaman Daftar Soal

BANK SOAL BERANDA SOAL MANAJEMEN

Daftar Soal

Mata Kuliah PEGANTAR BIOLOGI

No	Nama Soal	Dibuat oleh	Update	Download	
1	Latihan 1	NATHAN DRAKE	13-05-23	↓	Lihat
2	Latihan 2	NATHAN DRAKE	13-05-23	↓	Lihat
3	Latihan 3	NATHAN DRAKE	13-05-23	↓	Lihat
4	UTS 2022	NATHAN DRAKE	13-05-23	↓	Lihat
5	UTS 2023	NATHAN DRAKE	13-05-23	↓	Lihat
6	UAS 2020	NATHAN DRAKE	13-05-23	↓	Lihat
7	UTS 2021	NATHAN DRAKE	13-05-23	↓	Lihat
8	UAS 2022	NATHAN DRAKE	13-05-23	↓	Lihat

Sebelum **1** 2 Lanjut

Gambar 3.6 *Design Prototype* Halaman Daftar Soal

5. *Design Prototype* Halaman Soal

BANK SOAL BERANDA SOAL MANAJEMEN

Menampilkan Soal format file PDF

[Download](#)

Komentar

Isi Komentar ...

SUTHICHAI Mahasiswa 4.0 (40)

GOOD!!!

[Balas](#)

NATHAN DRAKE Dosen 4.8 (6)

@GREENEE E-JAM Thank!

[Balas](#)

Isi Komentar ...

SUTHICHAI Mahasiswa 4.0 (40)

ini benar?

[Balas](#)

Gambar 3.18 *Design Prototype* Halaman Soal

6. *Design Prototype* Halaman Mengelola Soal

BANK SOAL BERANDA SOAL MANAJEMEN

Mengelola Soal
Mengelola Matakuliah

Jenjang: SI Fakultas: MIPA Prodi: Informatika Semester: Semester 1 Matakuliah: Biologi

Daftar Soal

Mata Kuliah Biologi

No	Nama Soal	Dibuat Oleh	Update
1	latihan 1	dafee	2024-02-13 08:22:18
2	UAS 2023	dafee	2024-02-13 08:22:41
3	UTS2023	dafee	2024-02-13 08:23:07

Tambah Soal

Gambar 3.19 *Design Prototype* Halaman Mengelola Soal

7. *Design Prototype* Halaman Tambah Soal

BANK SOAL BERANDA SOAL MANAJEMEN

Mengelola Soal
Mengelola Matakuliah

Jenjang: SI Fakultas: MIPA Prodi: Informatika Semester: Semester 1 Matakuliah: Biologi

Tambah Soal

Nama Soal:
Isi nama soal

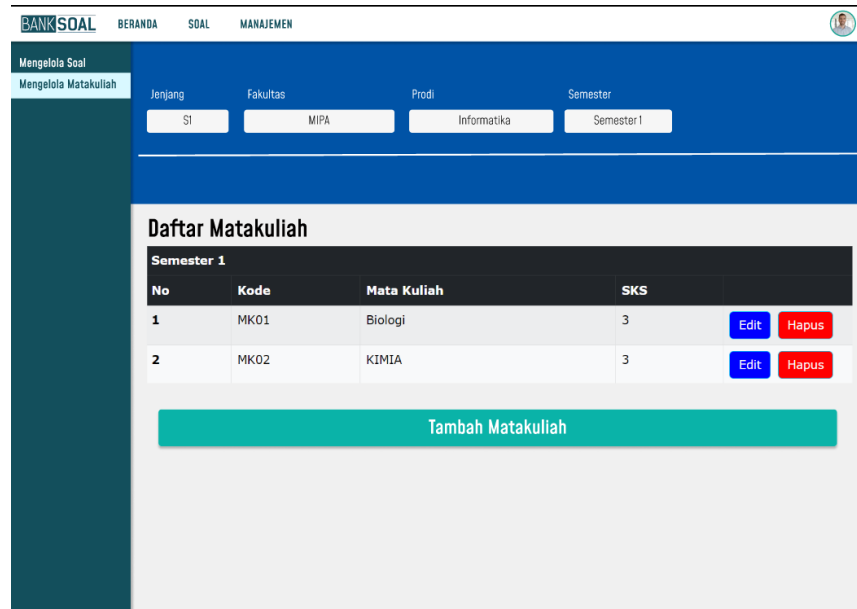
Pilih File:
Pilih file

Kode Matakuliah:
Isi kode Matakuliah

Submit
Batal

Gambar 3.7 *Design Prototype* Halaman Tambah Soal

8. *Design Prototype* Halaman Mengelola Matakuliah

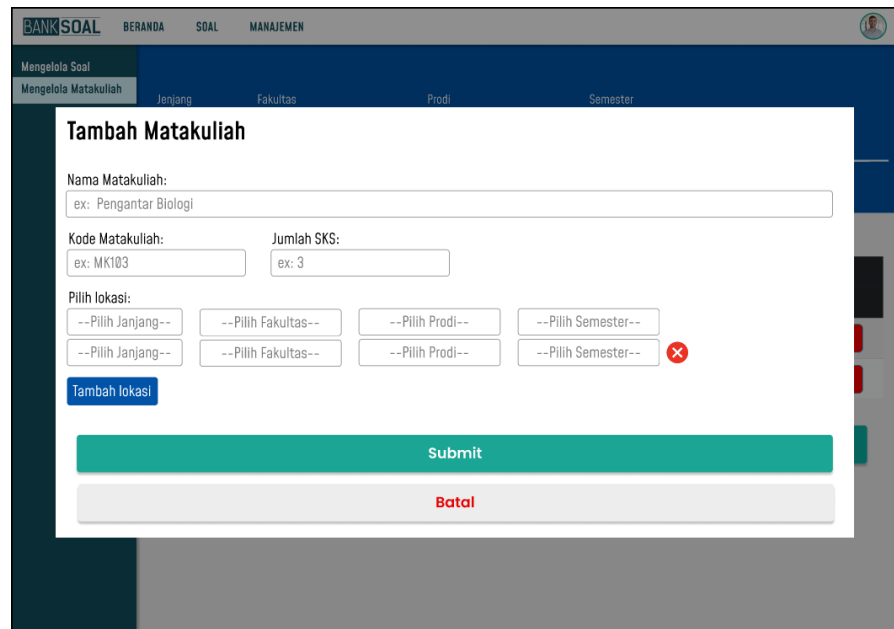


The image shows a web application interface for managing courses. At the top, there is a navigation bar with the logo 'BANK SOAL' and links for 'BERANDA', 'SOAL', and 'MANAJEMEN'. Below this, a sidebar on the left contains the text 'Mengelola Soal' and 'Mengelola Matakuliah'. The main content area has a header with filters for 'Jenjang' (S1), 'Fakultas' (MIPA), 'Prodi' (Informatika), and 'Semester' (Semester 1). Below the filters, the title 'Daftar Matakuliah' is displayed. A table lists courses for 'Semester 1' with columns for 'No', 'Kode', 'Mata Kuliah', and 'SKS'. The table contains two rows: one for 'Biologi' (MK01) and one for 'KIMIA' (MK02), both with 3 SKS. Each row has 'Edit' and 'Hapus' buttons. Below the table is a green button labeled 'Tambah Matakuliah'.

No	Kode	Mata Kuliah	SKS
1	MK01	Biologi	3
2	MK02	KIMIA	3

Gambar 3.8 *Design Prototype* Halaman Mengelola Matakuliah.

9. *Design Prototype* Halaman Tambah Matakuliah



The image shows a web application interface for adding a new course. At the top, there is a navigation bar with the logo 'BANK SOAL' and links for 'BERANDA', 'SOAL', and 'MANAJEMEN'. Below this, a sidebar on the left contains the text 'Mengelola Soal' and 'Mengelola Matakuliah'. The main content area has a header with filters for 'Jenjang', 'Fakultas', 'Prodi', and 'Semester'. Below the filters, the title 'Tambah Matakuliah' is displayed. The form contains several input fields: 'Nama Matakuliah' (with an example 'Pengantar Biologi'), 'Kode Matakuliah' (with an example 'MK103'), and 'Jumlah SKS' (with an example '3'). There are also four dropdown menus for 'Pilih lokasi' (Jenjang, Fakultas, Prodi, Semester). A blue button labeled 'Tambah lokasi' is positioned below the dropdowns. At the bottom of the form, there are two large buttons: a green 'Submit' button and a grey 'Batal' button.

Gambar 3.9 *Design Prototype* Halaman Tambah Matakuliah.

10. *Design Prototype* Halaman Profil

The image shows a web browser window with a blue header bar. The header contains the text 'BANK SOAL' in a bold, sans-serif font, followed by navigation links: 'BERANDA', 'SOAL', and 'MANAJEMEN'. On the right side of the header is a small circular profile picture icon. The main content area is titled 'Profil' in a bold, sans-serif font. Below the title is a profile picture of a man with short brown hair and a beard, wearing a light blue shirt. To the right of the profile picture are four input fields: 'Nama Lengkap:' with the value 'Suthichat Suwanbandit', 'Nomor Telepon:' with the value '000000000000', 'Alamat E-mail:' with the value 'suthichai@gmail.com', and 'Deskripsi:' with a placeholder '.....'. Below these fields is a blue button labeled 'Edit Profil'.

Gambar 3.10 *Design Prototype* Halaman Profil

3.3.5 Implementasi Sistem

Setelah melakukan perancangan dengan *Use Case Diagram*, dan *Activity Diagram* maka langkah selanjutnya dilanjutkan dengan implementasi sistem. Pada tahap ini dilakukan penulisan *code* program. Untuk pembangunan sistem Bank Soal ini menggunakan aplikasi berbasis *web*. Untuk pembangunan aplikasi *web* digunakan Bahasa pemrograman PHP dengan menggunakan *framework* Laravel, *Database Management System* (DBMS) yang digunakan adalah *MySQL*.

3.3.6 Pengujian Sistem

Salah satu prinsip utama yang dijadikan sebagai tolak ukur keberhasilan dari pengembangan perangkat lunak adalah nilai pengujian itu sendiri. Pengujian dilakukan untuk melihat apakah produk yang dibangun atau dihasilkan sudah sesuai dengan kebutuhan (*requirement*) atau belum. Pengujian yang akan dilakukan pada sistem ini yaitu pengujian fungsional, pengujian non-fungsional dan pengujian *Usability*.

3.3.6.1 Pengujian Fungsional

Pengujian fungsionalitas pada penelitian ini menggunakan metode *Manual Black Box* yang berfokus pada pengujian fungsionalitas dari sistem. Pengujian dianggap berhasil apabila seluruh skenario pengujian yang telah

ditentukan dapat berjalan sesuai dengan yang ditentukan dan dianggap gagal jika terjadi sebaliknya. Hasil pengujian disajikan dalam bentuk tabel dengan luaran berhasil tidaknya pengujian yang dilakukan pada Pengguna. Untuk detail skenario dapat dilihat pada tabel 3.3 berikut :

Tabel 3.3 Skenario Pengujian *Manual Black Box*

No	Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian
A. Mahasiswa				
1	<i>Login</i> akun	Mengisi NPM dan <i>Password</i> , Tekan tombol <i>Login</i>	<i>Login</i> berhasil dan akan arah ke halaman beranda.	
2	Mencari Matakuliah	Mencari soal dengan memenuhi syarat <i>dropdown search</i>	Memuncul Daftar Matakuliah	
3	Melihat daftar Soal	Memilih Matakuliah dan menekan tombol Lihat	Menampilkan Halaman Daftar Soal	
4	Melihat Soal	Memilih Soal dan menekan tombol Lihat	Menampilkan Halaman Soal	
5	Mendownload Soal	Menekan tombol <i>Download</i>	<i>Download</i> berhasil	
6	Mengomentari Soal	Mengisi komentar dan menekan tombol kirim	Memuncul komentar di <i>Layout</i> komentar	
7	Memberikan <i>Rating</i> Komentar	Memberikan rating dengan memilih ikon jumlah bintang dan mengklik.	Mengubah warna ikon bintang dan mengupdate nilai rata-rata	

8	Membalas Komentar	Menekan tombol balas, mengisi komentar dan tekan tombol kirim	Komentar tersebut akan muncul di bawah komentar yang dibalas	
9	Menghapus Komentar mahasiswa sendiri	Menekan tombol Hapus dan menekan tombol Konfirmasi	Menghilang komentar tersebut didalam <i>layout</i> komentar	
10	Melihat Profil	Menekan ikon Profil	Menampilkan Halaman Profil	
11	Mengedit profil	Menekan tombol <i>Edit</i> , mengedit profil dan menekan tombol <i>Submit</i>	Mengupdate data profil di halaman profil	
12	<i>Logout</i> Akun	Menekan tombol <i>Logout</i>	<i>Logout</i> berhasil dan Menampilkan halaman logon	
B. Dosen				
1	<i>Login</i> Akun	Mengisi NIP dan <i>Password</i> , Tekan tombol <i>Login</i>	<i>Login</i> berhasil dan akan arah ke halaman beranda.	
2	Menambah Soal	Menekan tombol Tambah Soal, mengisi <i>formular</i> dan menekan tombol <i>Submit</i>	Memuncul <i>Pop-up</i> berhasil	
3	Mengedit Soal	Menekan tombol Edit, mengisi	Memuncul <i>Pop-up</i> berhasil	

		<i>formular</i> dan menekan tombol <i>Submit</i>		
4	Menghapus Soal	Menekan tombol Hapus dan menekan tombol konfirmasi	Memuncul <i>Pop-up</i> berhasil	
5	Menambah Matakuliah	Menekan tombol Tambah Matakuliah, mengisi <i>formular</i> dan menekan tombol <i>Submit</i>	Memuncul <i>Pop-up</i> berhasil	
6	Mengubah Matakuliah	Menekan tombol Edit, mengisi <i>formular</i> dan menekan tombol <i>Submit</i>	Memuncul <i>Pop-up</i> berhasil	
7	Menghapus Matakuliah	Menekan tombol Hapus dan menekan tombol konfirmasi	Memuncul <i>Pop-up</i> berhasil	
8	Mencari Matakuliah	Mencari soal dengan memenuhi syarat <i>dropdown search</i>	Memuncul Daftar Matakuliah	
9	Melihat daftar Soal	Memilih Matakuliah dan menekan tombol Lihat	Menampilkan Halaman Daftar Soal	
10	Melihat Soal	Memilih Soal dan menekan tombol Lihat	Menampilkan Halaman Soal	
11	Mendownload Soal	Menekan tombol <i>Download</i>	<i>Download</i> berhasil	

12	Mengomentari Soal	Mengisi komentar dan menekan tombol kirim	Memuncul komentar di <i>Layout</i> komentar	
13	Memberi Rating Komentar	Memberikan rating dengan memilih ikon jumlah bintang dan mengklik.	Mengubah warna ikon bintang dan mengupdate nilai rata-rata	
14	Membalas Komentar	Menekan tombol balas, mengisi komentar dan tekan tombol kirim	Komentar tersebut akan muncul di bawah komentar yang dibalas	
15	Menghapus Komentar	Menekan tombol Hapus dan menekan tombol Konfirmasi	Menghilang komentar tersebut didalam <i>layout</i> komentar	
16	Melihat Profil	Menekan ikon Profil	Menampilkan Halaman Profil	
17	Mengubah profil	Menekan tombol <i>Edit</i> , mengedit profil dan menekan tombol <i>Submit</i>	Mengupdate data profil di halaman profil	
18	<i>Logout</i> Akun	Menekan tombol <i>Logout</i>	<i>Logout</i> berhasil dan Menampilkan halaman logon	

3.3.6.2 Pengujian Non-Fungsional

Pengujian fungsional pada penelitian ini menggunakan metode *performance test* untuk memastikan kinerja aplikasi sesuai dengan harapan. Dalam *performance test*, terdapat dua jenis pengujian yang akan dilakukan, yaitu *load test* dan *stress test*.

1. Load Test

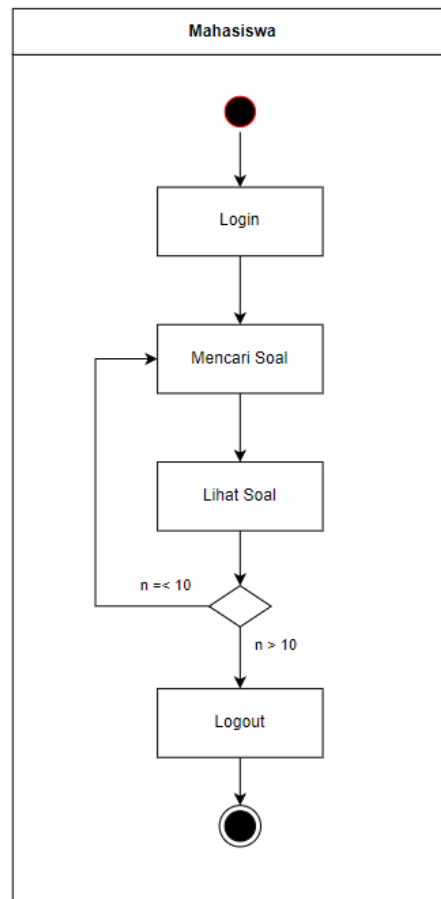
Load test pada penelitian ini bertujuan untuk mengidentifikasi batasan kapasitas sistem dan respon aplikasi terhadap lonjakan pengguna. Hasil dari pengujian ini akan membantu memastikan bahwa aplikasi dapat beroperasi dengan baik dan memberikan pengalaman pengguna yang memuaskan dalam berbagai situasi penggunaan yang mungkin terjadi. Berikut merupakan skenario untuk *Load Test*.

Tabel 3.4 Detail *Thread Group* untuk *Load Test*

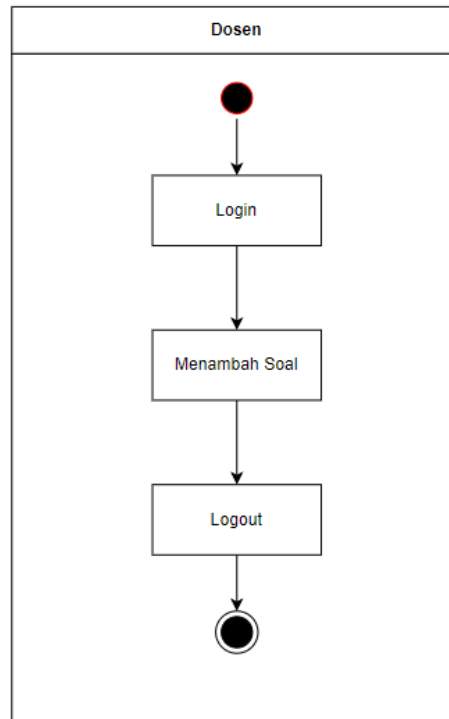
No	Total Pengguna	Pengguna Mahasiswa	Pengguna Dosen	Ramp-up Period (Detik)
1	25	22	3	50
2	50	45	5	100
3	75	67	8	150
4	100	90	10	200

Pengujian ini melakukan *Load test* sebanyak empat kali dengan jumlah pengguna yang berbeda pada setiap pengujian. *Load test* dilakukan dengan bertambahnya jumlah pengguna dari 25 hingga 100, dengan penambahan bertahap pada setiap sesi pengujian. Dari total pengguna yang ditambahkan, 90% dianggap sebagai pengguna mahasiswa dan 10% sebagai pengguna dosen, mencerminkan proporsi pengguna di lingkungan aplikasi yang sebenarnya. Periode ramp-up dalam load test diatur agar setara dengan dua kali jumlah pengguna, memungkinkan simulasi bertahap dari beban pengguna selama pengujian.

Untuk skenario *load test*, terdapat dua aktivitas yang berbeda untuk pengguna mahasiswa dan dosen:



Gambar 3.11 Aktivitas Pengguna Mahasiswa untuk *Load Test*.



Gambar 3.12 Aktivitas Pengguna Dosen untuk *Load Test*.

2. *Stress Test*

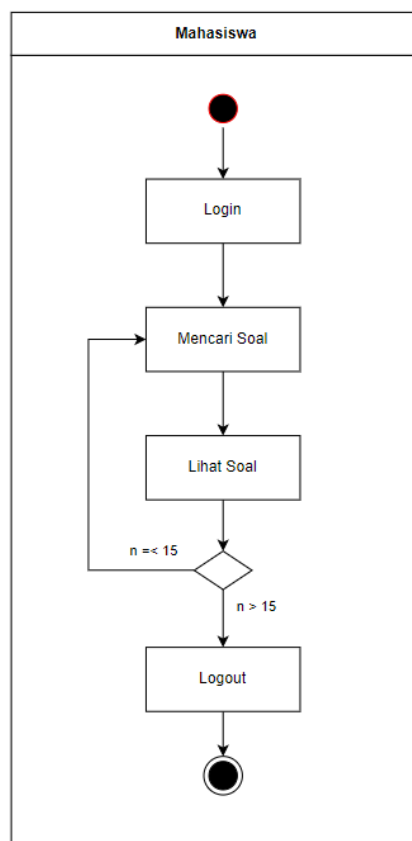
Stress test pada penelitian ini bertujuan untuk mengevaluasi kinerja sistem saat dihadapkan pada situasi beban yang ekstrem, sehingga dapat mengidentifikasi batasan kapasitas dan respons aplikasi terhadap lonjakan pengguna. Dengan melakukan *stress test* ini, diharapkan aplikasi dapat teruji secara menyeluruh dan dapat beroperasi secara optimal bahkan dalam situasi beban yang paling tinggi, sehingga memberikan pengalaman pengguna yang memuaskan dalam berbagai situasi penggunaan yang mungkin terjadi. Berikut merupakan skenario untuk *Stress Test*.

Tabel 3.5 Detail *Thread Group* untuk *Stress Test*.

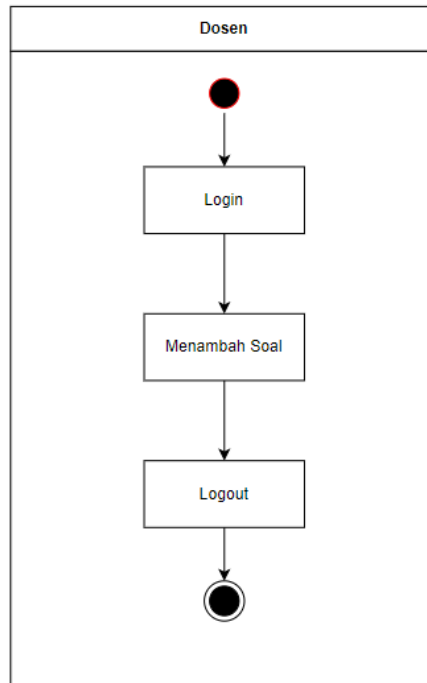
No	Total Pengguna	Pengguna Mahasiswa	Pengguna Dosen	Ramp-up Period (Detik)
1	75	67	8	75
2	100	90	10	100
3	125	112	13	125
4	150	135	15	150

Pengujian ini melanjutkan dengan melakukan *stress test* setelah *load test*, dengan menguji aplikasi dalam kondisi beban yang semakin tinggi. *Stress test* dilakukan dengan peningkatan jumlah pengguna dari 75 hingga 150, dengan penambahan bertahap pada setiap sesi pengujian. Dari jumlah total pengguna, 90% dianggap sebagai pengguna mahasiswa dan 10% sebagai pengguna dosen. Periode ramp-up dalam *stress test* diatur agar sesuai dengan jumlah pengguna, memungkinkan simulasi bertahap dari beban pengguna selama pengujian.

Untuk skenario *Stress test*, terdapat dua aktivitas yang berbeda untuk pengguna mahasiswa dan dosen:



Gambar 3.13 Aktivitas Pengguna Mahasiswa untuk *Stress Test*.



Gambar 3.14 Aktivitas Pengguna Dosen untuk *Stress Test*.

3.3.6.3 Pengujian Usability

Pengujian *Usability* dilakukan untuk mengevaluasi pengalaman pengguna terhadap aplikasi yang akan dibangun. Pengujian ini menggunakan metode *Usability Metric for User Experience* (UMUX). UMUX adalah metode penilaian kualitatif singkat yang mengukur kegunaan *usability* umum dari suatu produk. Pengujian *usability* dilakukan dengan cara melakukan survei kepada calon pengguna. Calon pengguna akan diminta untuk mengisi kuesioner yang pada tiap pertanyaan memiliki rentang skala 1-7. Metode UMUX menargetkan kegunaan dengan menilai efektivitas, efisiensi, dan kepuasan. Pengujian yang menggunakan metode ini dilakukan dengan mengumpulkan data dari responden yang melakukan pengujian. Pengujian dapat dilakukan baik secara luring maupun daring dengan menargetkan total responden sebanyak 15 orang.

3.3.7 Analisa Hasil

Tahap Analisa hasil merupakan tahap dalam mengumpulkan hasil pengujian yang telah dilakukan dan selanjutnya akan dianalisa untuk mendapatkan kesimpulan dari pengujian tersebut. Hasil dari tiga pengujian akan dianalisis, yaitu pengujian fungsionalitas, pengujian *Usability*, dan pengujian kinerja (*Performance Test*). Hasil pengujian fungsionalitas, yang menggunakan metode *Black Box Testing*, dianalisis untuk mengevaluasi kesesuaian antara implementasi sistem dengan desain yang telah direncanakan. Sementara itu, hasil pengujian *Usability*, yang menggunakan metode UMUX (*Usability Metric for User Experience*), dianalisis untuk menilai tingkat kelayakan sistem informasi dalam penggunaannya. Hasil dari pengujian kinerja akan dievaluasi untuk menentukan sejauh mana sistem mampu menangani beban dan performa dalam situasi penggunaan yang sesungguhnya.

3.3.8 Penulisan Laporan

Tahap penulisan laporan merupakan tahap paling akhir di mana hasil analisa yang sudah didapatkan dijabarkan dalam laporan tugas akhir. Penulisan laporan ini dijabarkan secara terstruktur dan sesuai dengan penelitian yang telah dilakukan. Dengan penulisan laporan ini maka akan diketahui hasil penelitian secara detail.

DAFTAR PUSTAKA

- Adelheid; Andre; dan Khairil. 2012. “Buku Pintar Mengusai PHP dan MySQL”. Mediakita.
- Astrini, D., & Melly, A. (2016). Model Bank Soal Berbasis Web Untuk Meningkatkan Kualitas Pendidikan di Era Masyarakat Ekonomi Asean. *Kimia Dan Pendidikan Kimia (JKPK)*, 1(3), 194–202.
- Aziz, M. F. (2005). *Object Oriented Programming dengan PHP 5*. Komputindo., Jakarta.
- Bernadette Tynan.(2004). *Melatih Anak berpikir seperti Jenius*. Penerbit:PT Gramedia Pustaka Utama.29-31.
- Betha Sidik. (2014). Pemrograman WEB PHP. Bandung. Informatika Bandung.
- Brady. M dan Loonam. J. (2010). *Exploring the use of entity-relationship diagramming as a technique to support grounded theory inquiry*. Bradford: Emerald Group Publishing.
- Borsci, S., Federici, S., Bacci, S., Gnaldi, M., dan Bartolucci, F. (2015). Assessing user satisfaction in the era of user experience: Comparison of the sus, umux, and umux-lite as a function of product experience. *International journal of human-computer interaction*, 31(8):484–495.
- Dhien, A., & Kumaidi, K. (2021). Developing An Achievement Test in Learning of Physics for Assessment. *Jurnal Pendidikan Fisika Indonesia*, 17(1), 60–71.
- Finstad, K. (2010). The usability metric for user experience. *Interacting with Computers*, 22:323–327.
- Hariaty Lumbantobing, (2021). *Performance Testing : Pengertian, Fokus, Jenis, dan Contoh*.
- Husda, N. E., & Wangdra, Y. (2016). *Pengantar Teknologi. Informasi, Ed.Revisi*. Jakarta: Baduose Media.

- Khan, M. dkk. (2011). Different approaches to black box testing technique for finding errors. *International Journal of Software Engineering & Applications (IJSEA)*, 2(4).
- Kurtiyaningsih, Pemograman Basis Data Berbasis WEB Menggunakan PHP dan MySQL, Yogyakarta: *Graha Ilmu*, 2011.
- Latuconsina, N. M., & Yunanto, P. W. (2017). Pembuatan Bank Soal Dan Analisis Butir Soal Mata Kuliah Kriptografi Untuk Mahasiswa Program Studi Pendidikan Teknik Informatika Dan Komputer Universitas Negeri Jakarta. *PINTER : Jurnal Pendidikan Teknik Informatika Dan Komputer*, 1(2), 142– 145. <https://doi.org/10.21009/pinter.1.2.7>.
- Musriatun Napiah, M. R. (2020). Rancang Sistem Penyajian Bank Soal Untuk Jenjang. *Jurna Infortech, Volume 2 No.2*.
- Mustamiin, M. (2020). Rancang Bangun Sistem Manajemen Soal Dan Ujian Berbasis Website Menggunakan Framework Laravel. *Jurnal IKRA-ITH Informatika*, 58-63.
- Mustaqbal, M. S., Firdaus, R. F., dan Rahmadi, H. (2015). Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN). *Jurnal Ilmiah Teknologi Informasi Terapan*, 1:31–36.
- Natayasa, I. K. (2020). Sistem Informasi Silabus, RPP, dan Bank Soal Berbasis Web. *Jurnal Ilmiah Teknologi dan Komputer*.
- Plook TCAS. (2020). *Documentation*. <https://www.trueplookpanya.com/tcas/article/detail/84501> (Diakses Tanggal 01 Februari 2024).
- Rama, D. V. dan Jones, F. L. (2008). *Sistem Informasi Akuntansi*. Salemba Empat, Jakarta.
- Retnawati, Heri, (2014). *Pengembangan Bank Soal Matematika*. Yogyakarta: Jurdik Matematika FMIPA UNY.

- Roger S. Pressman, (2002). *Rekayasa Perangkat Lunak Pendekatan Praktisi (Buku Satu)*, ANDI Yogyakarta.
- Rudner, L. (1999). *Item banking. Practical Assessment, Research and Evaluation*, 6(4), 1998–1999. <https://doi.org/10.1017/cbo9780511732980.019>
- Sabanayo, D. I. (2014). Sistem Pendukung Keputusan Pemilihan Karyawan Terbaik Menggunakan Metode SAW Pada PT. Berkah Cahaya Muria Kudus.
- Setiawan, D. (2017). *Buku Sakti Pemrograman Web*. Bantul: Start Up.
- SitePoint, 2015. The Most PHP Framework for 2015 : SitePoint Survey Result. <https://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/>.
- Suendri (2018). Implementasi Diagram UML (*Unified Modelling Language*) Pada Perancangan Sistem Informasi Remunerasi Dosen dengan *Database Oracle* (Studi Kasus: UIN Sumatera Utara Medan). *Jurnal Ilmu Komputer dan Informatika*, 3(1).
- Sumardiyono, (2011). *Pengembangan dan Pengelolaan Bank Soal Matematika di KKG/MGMP*. 8–10.
- Wahl, N. J. (2000). Student-Run Usability Testing. In *Thirteenth Conference on Software Engineering Education and Training*, pages 123–131.
- Ward, Annie W., (2005). *Guidelines for the Development of Item Banks*. Educational Measurement: Issues and Practice. California : ITEMS. 13(1). 34-39
- Watkins, J. dan Mills, S. (2010). *Testing IT: an off-the-shelf software testing process*. Cambridge University Press.
- Wesfix, T. (2017). *Branding Itu Dipraktekin*. Grasindo, Jakarta.
- Widana, I. W. (2014). Pengembangan Bank Soal. *Jurnal EMASAINS*, III(2), 186– 197.
- WRIGHT, B. D., & BELL, S. R. (1984). ITEM BANKS: WHAT, WHY, HOW. *Journal of Educational Measurement*, 21(4), 331–345. <https://doi.org/https://doi.org/10.1111/j.1745-3984.1984.tb01038.x>

Yuhefizar. (2012). *Cara Mudah Membangun Website Interaktif. Menggunakan CMS Joomla Edisi Revisi*. Jakarta: PT. Elex Media. Komputindo.