

**RANCANG BANGUN SISTEM INFORMASI *TRY OUT* UNTUK SOAL
CPNS BERBASIS *WEB***

PROPOSAL SKRIPSI

Dibuat sebagai syarat pelaksanaan penelitian mahasiswa program sarjana

Oleh :

Rifan Alamsyah

NIM : 1606020



**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TINGGI TEKNOLOGI GARUT**

2020

LEMBAR PENGESAHAN

**RANCANG BANGUN SISTEM INFORMASI *TRY OUT* UNTUK SOAL
CPNS BERBASIS *WEB***

PROPOSAL SKRIPSI

Oleh:

Rifan Alamsyah

NIM: 1606020

Proposal Skripsi ini telah diperiksa dan disetujui oleh:
Pembimbing,

Asep Deddy S., M.Kom
NIDN: 0417026202

Mengetahui,
Ketua Program Studi Teknik Informatika,

Dede Kurniadi, S.Kom., M.Kom
NIDN: 0402098301

RINGKASAN

Pegawai Negeri Sipil(PNS) adalah warga negara Indonesia yang memenuhi syarat tertentu, diangkat sebagai pegawai ASN secara tetap oleh pejabat pembina kepegawaian untuk menduduki jabatan. Sistem informasi *try out* dibutuhkan untuk memberikan sarana berlatih untuk test Calon PNS (CPNS) dan juga untuk meningkatkan kualitas CPNS itu sendiri. Dengan menerapkan sistem *Computer Assessted Test* (CAT) akan memberikan gambaran bagaimana *test* CPNS yang sesungguhnya. Sistem informasi ini juga akan memiliki fitur *review* soal yang akan mengkoreksi jawaban yang salah sehingga akan memberikan pengetahuan lebih kepada peserta CPNS. Tujuan penelitian ini adalah merancang bangun sistem informasi *try out* untuk soal CPNS berbasis *Web* yang dapat digunakan untuk waktu yang tidak terbatas sehingga dapat digunakan dalam jangka waktu yang panjang. Metodologi yang digunakan adalah *Rational Unified Process* (RUP) serta menggunakan pemodelan *Unified Modeling Language* (UML). Hasil dari penelitian ini berupa Rancang Bangun Sistem Informasi *Try Out* Untuk Soal CPNS Berbasis *Web*.

KATA PENGANTAR

Assalammu'alaikum Warahmatullahi Wabarakatu..

Puji dan syukur penyusun panjatkan kehadirat Allah SWT yang telah memberikan berkat, rahmat, dan karunia-Nya sehingga penyusun dapat menyelesaikan laporan Proposal Tugas Akhir tepat pada waktunya. Shalawat serta salam semoga tercurah limpahkan kepada junjungan Nabi kita semua yakni Nabi Muhammad SAW, kepada keluarganya, para sahabatnya dan semoga sampai kepada kita selaku umatnya.

Dalam penyusunan laporan Proposal Tugas Akhir ini penyusun mendapat banyak bantuan dan dukungan dari beberapa pihak, oleh karena itu penyusun mengucapkan rasa terimakasih yang sebesar-besarnya, kepada ayah dan ibunda tercinta yang tiada henti-hentinya memberikan do'a dan dorongan sampai dengan saat ini dan juga rekan-rekan informatika 2016 yang telah membantu. Tidak lupa juga penyusun ucapkan banyak terima kasih yang sebesar-besarnya kepada yang terhormat :

1. Bapak Dr. H. Hilmi Aulawi, ST., MT., Selaku Ketua Sekolah Tinggi Teknologi Garut (STTG);
2. Bapak Dede Kurniadi, S.kom., M.kom selaku Ketua Jurusan Teknik Informatika yang telah memberikan dukungan;
3. Bapak Ridwan Setiawan, M.Kom., Selaku Sekertaris Program Studi Teknik Informatika;
4. Bapak Asep Deddy S., M.Kom selaku dosen pembimbing dalam dukungan dan bimbingan yang penuh dengan kesabaran dan perhatian dalam penyusunan Laporan Tugas Akhir ini,

Tidak lupa penulis ucapkan terimakasih kepada Bapak dan Ibu tercinta, adik, saudara-saudara dan seluruh keluarga serta orang-orang terdekat yang sudah memberikan bantuan, dorongan dan motivasi bagi penulis selama melakukan penyusunan Proposal Skripsi. Dan semua pihak yang tidak bisa disebutkan satu persatu yang telah memberikan bantuan dan inspirasi dalam penyusunan Proposal Skripsi.

Penulis berharap semoga penyusunan Proposal Skripsi ini dapat bermanfaat khususnya bagi penulis pribadi, umumnya bagi semua yang membaca Proposal Skripsi ini.

Wassalammu'alaikum Warahmatullahi Wabarakatu.

Garut, 16 Maret 2018

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN

| | |
|--|----|
| RINGKASAN | i |
| KATA PENGANTAR | ii |
| DAFTAR ISI..... | iv |
| DAFTAR GAMBAR | v |
| DAFTAR TABEL..... | vi |
| 1. PENDAHULUAN | 1 |
| 1.1. Latar Belakang | 1 |
| 1.2. Masalah Penelitian | 3 |
| 1.3. Tujuan Penelitian..... | 3 |
| 1.4. Pertanyaan Penelitian | 3 |
| 1.5. Cakupan Penelitian..... | 4 |
| 1.6. Manfaat Penelitian..... | 4 |
| 2. TINJAUAN PUSTAKA..... | 6 |
| 2.1. Penelitian Rujukan | 6 |
| 2.2. Kesenjangan Penelitian | 7 |
| 2.3. Sistem Informasi..... | 9 |
| 2.4. Konsep Dasar Perancangan | 9 |
| 2.5. Rekayasa Perangkat Lunak..... | 9 |
| 2.6. <i>Website</i> | 10 |
| 2.7. <i>Aplikasi Web</i> | 10 |
| 2.8. <i>Rational Unified Process</i> (RUP) | 10 |
| 2.9. <i>Unified Modeling Language</i> (UML)..... | 12 |
| 2.10. <i>Software</i> Pendukung..... | 18 |
| 3. METODOLOGI PENELITIAN..... | 23 |
| 3.1. Kerangka Pemikiran | 23 |
| 3.2 <i>Work Breakdown Structure</i> (WBS)..... | 25 |
| 4. JADWAL RENCANA KEGIATAN PENELITIAN | 33 |
| DAFTAR PUSTAKA | 68 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 2.1. Tahapan Rational Unified Process | 10 |
| Gambar 2.2. Diagram <i>Unified Modeling Language</i> | 12 |
| Gambar 2.3. Tampilan <i>Home Visual Studio Code</i> | 18 |
| Gambar 2.4. Tampilan <i>Workspace Microsoft Visio</i> | 19 |
| Gambar 2.5. Tampilan awal <i>Laravel</i> | 20 |
| Gambar 2.6. Tampilan XAMPP..... | 21 |
| Gambar 2.7. Tampilan awal <i>GoogleChrome</i> | 22 |
| Gambar 3.1. Kerangka Pemikiran | 24 |
| Gambar 3.2. <i>Work Breakdown Structure (WBS)</i> | 25 |
| Gambar 3.3. Diagram Alur Perancangan Sistem | 26 |

DAFTAR TABEL

| | |
|--|-------------------------------------|
| Tabel 2.1. Kesenjangan Penelitian | 7 |
| Tabel 2.2. Simbol-Simbol Diagram <i>Use Case</i> | 13 |
| Tabel 2.3. Notasi <i>Activity</i> Diagram | 14 |
| Tabel 2.4. Notasi <i>Sequence</i> Diagram | 15 |
| Tabel 2.5. Simbol-Simbol Pada <i>Class Diagram</i> | 17 |
| Tabel 3.1. Detail Activity | 29 |
| Tabel 3.2. Sumber Daya Penelitian | 31 |
| Tabel 4.1. Jadwal Rencana Kegiatan Penelitian | Error! Bookmark not defined. |

1. PENDAHULUAN

1.1. Latar Belakang

Pegawai Negeri Sipil (PNS) menurut peraturan pemerintah nomor 11 tahun 2017 tentang manajemen pegawai negeri sipil adalah warga negara Indonesia yang memenuhi syarat tertentu, diangkat sebagai Pegawai ASN secara tetap oleh pejabat pembina kepegawaian untuk menduduki jabatan pemerintahan. Menurut data Badan Kepegawaian dan Diklat (BKD) Garut pada tahun 2019 terdapat 21.703 orang mendaftarkan diri sebagai Calon PNS (CPNS) akan tetapi formasi kebutuhan CPNS di lingkungan pemerintahan kabupaten Garut tahun anggaran 2019 sebanyak 838 formasi, dikarenakan kuota penerimaan formasi yang terbatas membuat seleksi CPNS menjadi hal yang sulit. Faktor jenis soal pada *test* CPNS juga membuat CPNS harus belajar materi yang beragam, menurut Peraturan Menteri Pendayagunaan Aparatur Negara dan Reformasi Republik Indonesia, *test* CPNS memiliki 3 jenis soal yakni meliputi Test Wawasan Kebangsaan (TWK) sebanyak 35 soal, Test Intelegensi Umum (TIU) sebanyak 30 soal dan Tes Karakteristik Pribadi (TKP) sebanyak 35 soal, oleh karena banyaknya soal yang harus dipelajari peserta calon PNS (CPNS) dituntut untuk memperbanyak latihan soal CPNS, baik dengan mengerjakan soal pada buku CPNS yang tersedia di toko-toko ataupun dengan mengerjakan soal dari aplikasi yang tersedia di paket pembelian buku yang memiliki sistem *Try Out* berbasis *Computer Asseted Test* (CAT) sehingga CPNS dapat belajar sesuai dengan *test* CPNS yang sesungguhnya. *Try out* adalah suatu mekanisme yang digunakan sebagai sebuah latihan bagi siswa sebelum melaksanakan ujian yang sesungguhnya (Aisah & Haryati, 2019). Dengan sistem informasi *Try Out* ini diharapkan dapat membantu CPNS untuk berlatih mengerjakan soal serupa yang di prediksi akan ada pada *test* CPNS sesungguhnya, sehingga nantinya memberikan kesempatan lebih besar untuk lulus.

Pada penelitian sebelumnya yang berjudul “Rancang Bangun Aplikasi *Bank Soal* Program Studi Teknik Informatika Sekolah Tinggi Teknologi Garut” menjelaskan bahwa hasil dari penelitian tersebut adalah aplikasi *Bank Soal* untuk menyimpan soal-soal hasil evaluasi (Azizah & Fitriani, 2016). Penelitian kedua yang berjudul “Perancangan Aplikasi Survei Kepuasan Mahasiswa Berbasis

Kuesioner *Online*” memiliki hasil Aplikasi Survei Kepuasan Mahasiswa berbasis kuesioner *online* untuk memfasilitasi mahasiswa dalam melakukan proses pengisian survei sehingga mahasiswa bisa mengisi survei dimana saja dan kapan saja tanpa ada batasan ruang dan waktu (Kurniadi & Islami, 2018). Pada penelitian ketiga yang berjudul “Aplikasi Tryout Ujian *Online* untuk SMA/SMK Sederajat (*Online Examination Tryout Application for Senior and Vocational High School Students*)”, membahas tentang program tryout yang digunakan sebagai sarana latihan siswa dalam menghadapi ujian nasional (Kumalasari, Erika, & Mustikasari, 2017). Pada penelitian keempat yang berjudul “Pembuatan Aplikasi Sistem Ujian *Online* Pada SMK Garuda Nusantara Bekasi”, pada penelitian ini membuat aplikasi ujian *online* sehingga memberi kemudahan bagi siswa, guru dan pengawas ujian dalam pelaksanaan dan pengolahan data dan hasil ujian (Riyadi, Hermaliani, & Utami, 2019). Pada penelitian terakhir yang berjudul “Aplikasi Learning Manajemen Sistem Dan Ulangan *Online* Berbasis Web” menjelaskan bahwa hasil penelitian menyediakan fitur-fitur untuk membantu guru dalam mengelola pengumuman, materi, tugas dan ulangan. Aplikasi ini juga membantu siswa dalam mengunduh materi, mengunggah pengumpulan tugas, melaksanakan ulangan secara *online* dan melihat nilai-nilai tugas dan ulangan (Ekaputri, Suryatiningsih, & Siswanto, 2016). Berdasarkan penelitian-penelitian yang dirujuk terlihat bahwa terdapat konsep seperti *bank* soal dan kuisisioner yang memiliki sistem tanya jawab yang akan berguna dalam pembuatan sistem penyimpanan dan struktur soal dan jawaban, serta pada penelitian yang dirujuk juga memiliki konsep *try out* dan ulangan berbasis *online* yang akan memudahkan pengguna dalam melakukan proses *try out* atau ulangan tanpa harus mengerjakan dalam kertas, dan soalpun dapat dinilai secara otomatis sehingga dapat memudahkan penilai atau guru. Pada penelitian yang dirujuk juga memiliki persamaan pembuatan basis aplikasi yakni berbasis *web*, oleh karena itu penulis akan membangun aplikasi berbasis *web* dengan alasan aplikasi akan lebih fleksibel untuk diakses dimana saja dan pada perangkat apa saja tanpa melakukan instalasi terlebih dahulu, serta aplikasi akan memiliki tampilan yang dapat menyesuaikan ukuran layar secara responsif sehingga pengguna dapat menggunakan aplikasi secara maksimal di perangkat masing-masing. Selain itu aplikasi *try out* yang akan dirancang di desain

menyerupai sistem CAT dengan fitur batas waktu pengerjaan, sehingga dapat melatih para peserta CPNS untuk mengelola waktu pengerjaan soal. Merujuk pada kelima penelitian sebelumnya, penulis mengambil judul penelitian Rancang Bangun Sistem Informasi *Try Out* Untuk Soal CPNS Berbasis *Web*.

1.2. Masalah Penelitian

Berdasarkan latar belakang serta jurnal rujukan yang diambil, terdapat beberapa masalah yaitu sebagai berikut:

1. Pada jurnal yang dirujuk, aplikasi belum ada fitur pembahasan setelah mengerjakan *try out* atau ulangan *online*.
2. Aplikasi belum memiliki informasi lanjutan dari nilai hasil *try out* atau ulangan.
3. Waktu pengerjaan *try out* atau ulangan pada aplikasi tidak fleksibel.

1.3. Tujuan Penelitian

Adapun tujuan dari penelitian ini dapat dirumuskan diantaranya adalah untuk:

1. Untuk membuat rancang bangun sistem informasi *try out* untuk soal CPNS berbasis *web*.
2. Untuk memudahkan peserta CPNS mendapatkan latihan soal sesuai dengan *test* CPNS yang sesungguhnya.

1.4. Pertanyaan Penelitian

Berdasarkan Masalah Penelitian yang ada diatas terdapat masalah yaitu, diantaranya :

1. Bagaimana membuat rancang bangun sistem informasi *try out* untuk soal CPNS?
2. Bagaimana membuat sistem informasi dengan latihan soal yang sesuai dengan *test* CPNS?

1.5. Cakupan Penelitian

Agar dalam pembahasan ini dapat mengarah pada tujuan yang telah direncanakan, untuk menghindari pembahasan yang bisa melebar, maka penulis membatasi laporan Skripsi ini kedalam beberapa cakupan penelitian ini, adalah sebagai berikut:

1. Aplikasi *Try Out* soal CPNS;
2. Metodologi yang digunakan pada perancangan sistem informasi *try out* untuk soal CPNS berbasis *web* ini menggunakan *Rational Unified Process* (RUP);
3. Dalam aplikasi menerapkan sistem *try out* yang memiliki batas waktu pengerjaan soal;
4. Memiliki bank soal yang disertai dengan penjelasan tiap soal;
5. Memiliki fitur review jawaban ketika selesai mengerjakan *try out*;
6. Pengguna aplikasi ini adalah masyarakat yang ingin belajar soal CPNS khususnya akan mengikuti ujian CPNS; dan
7. Bahasa pemrograman yang digunakan adalah PHP dan javascript serta MySQL sebagai penyimpanan *database*.

1.6. Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Bagi pengguna (masyarakat) selaku pengguna akhir dari aplikasi ini mendapatkan informasi seputar *try out* atau ulangan online soal CPNS yang dapat diakses dimana saja dan kapan saja; dan
2. Dengan adanya aplikasi ini diharapkan dapat membantu meningkatkan kualitas CPNS yang akan menjadi Aparatur Sipil Negara yang telah terlatih dan memiliki wawasan yang luas.

1.7. Sistematika

Proposal skripsi ini disusun dengan sistematika sebagai berikut:

1. **PENDAHULUAN**, berisi latar belakang, masalah, tujuan, pertanyaan, cakupan, dan manfaat penelitian, serta sistematika proposal;

2. **TINJAUAN PUSTAKA**, berisi analisis rinci terhadap penelitian sebelumnya dan hubungannya dengan penelitian yang akan dilaksanakan;
3. **METODOLOGI PENELITIAN**, berisi kerangka pemikiran, beserta penjelasan tentang tahapan penelitian berikut aktivitas dan tekniknya, serta waktu, tempat, dan sumber daya yang digunakan.
4. **JADWAL RENCANA KEGIATAN PENELITIAN**, berisi rencana jadwal penelitian secara detail disajikan dalam bentuk tabel.

2. TINJAUAN PUSTAKA

2.1. Penelitian Rujukan

Dalam penyusunan penelitian ini merujuk dari jurnal atau hasil penelitian sebelumnya, dimana terdapat beberapa keterkaitan dalam penelitian yang dilakukan baik secara langsung ataupun secara tidak langsung. Hasil penelitian tersebut sebagai berikut :

1. Penelitian rujukan pertama berjudul “Rancang Bangun Aplikasi *Bank Soal* Program Studi Teknik Informatika Sekolah Tinggi Teknologi Garut”. Pada penelitian ini menggunakan metodologi berorientasi objek dengan pendekatan *Unified Software Development Process* dengan pemodelan *Unified Modeling Language*. Untuk bahasa pemrograman yang digunakan adalah PHP dan untuk *database* menggunakan MySQL. Adapun hasil dari penelitian ini menghasilkan aplikasi bank soal di program studi teknik informatika mampu memberikan solusi dalam mengatur penyimpanan soal-soal dengan baik dan dosen dapat menggunakan soal-soal sebelumnya yang pernah dibuat untuk digunakan pada waktu berikutnya serta sejarah pembuatan soal-soal dapat dicari dengan mudah;
2. Penelitian rujukan kedua berjudul “Perancangan Aplikasi Survei Kepuasan Mahasiswa Berbasis Kuesioner Online”. Pada penelitian ini menggunakan metodologi Unified Approach yang meliputi tahapan Object Oriented Analysis, tahapan Object Oriented Design serta menggunakan Component Based Development pada tahapan Implementasi, dan untuk pemodelannya menggunakan Unified Modelling Language. Untuk bahasa pemrograman yang digunakan adalah PHP dan untuk *database* menggunakan MySQL. Adapun hasil dari penelitian ini berupa Aplikasi Survei Kepuasan Mahasiswa berbasis kuesioner online untuk memfasilitasi mahasiswa dalam melakukan proses pengisian survei sehingga mahasiswa bisa mengisi survei dimana saja dan kapan saja tanpa ada batasan ruang dan waktu;
3. Penelitian rujukan ketiga berjudul “Aplikasi Tryout Ujian *Online* untuk SMA/SMK Sederajat (*Online Examination Tryout Application for Senior and Vocational High School Students*)”. Metodologi yang digunakan

adalah *Waterfall*. Untuk bahasa pemrograman yang digunakan adalah PHP dan MySQL sebagai penyimpanan *database*. Hasil dari penelitian ini berupa Aplikasi Tryout ujian online dan Manual book untuk ditujukan untuk semua pengguna aplikasi, baik guru, siswa, maupun admin aplikasi;

4. Penelitian keempat berjudul “Pembuatan Aplikasi Sistem Ujian *Online* Pada SMK Garuda Nusantara Bekasi”. Metodologi yang digunakan adalah *Waterfall*. Untuk bahasa pemrograman yang digunakan adalah PHP dan penyimpanan *database* yang digunakan adalah MySQL. Hasil dari penelitian ini adalah aplikasi ujian online sehingga memberi kemudahan bagi siswa, guru dan pengawas ujian dalam pelaksanaan dan pengolahan data dan hasil ujian pada SMK Garuda Nusantara Bekasi.
5. Penelitian terakhir berjudul “Aplikasi Learning Manajemen Sistem Dan Ulangan Online Berbasis Web”. Metodologi yang digunakan pada penelitian ini adalah Prototipe (Prototyping Model). Bahasa pemrograman yang digunakan adalah PHP dengan menggunakan *framework CodeIgniter* dan untuk penyimpanan *database* menggunakan MySQL. Hasil dari penelitian ini berupa aplikasi Learning Manajemen Sistem dan Ulangan Online, untuk membantu pihak sekolah dan siswa dalam penyampaian pengumuman, pembagian materi yang diajarkan, pemberian dan pengumpulan tugas siswa, pelaksanaan dan pemeriksaan ulangan, serta merekap nilai ulangan siswa.

2.2. Kesenjangan Penelitian

Berdasarkan penelitian yang telah dirujuk sebelumnya dan sudah dijelaskan, maka terdapat kesenjangan dan permasalahan diantaranya :

Tabel 2.1. Kesenjangan Penelitian

| No | Judul Penelitian | Cakupan Penelitian | Kesenjangan Penelitian |
|----|--|--|--|
| 1 | Rancang Bangun Aplikasi <i>Bank Soal</i> Program Studi Teknik Informatika Sekolah Tinggi Teknologi Garut | Perancangan sistem informasi dari aplikasi pada penelitian berfokus pada penyimpanan soal. | Jika pada penelitian sebelumnya memberikan kemudahan dalam penyimpanan soal, maka pada penelitian ini konsep dari soal tersebut, dapat |

| No | Judul Penelitian | Cakupan Penelitian | Kesenjangan Penelitian |
|----|---|---|---|
| | (Azizah & Fitriani, 2016) | | diimplementasikan sehingga dapat digunakan dalam <i>try out</i> |
| 2 | Perancangan Aplikasi Survei Kepuasan Mahasiswa Berbasis Kuesioner Online (Kurniadi & Islami, 2018) | Aplikasi yang dirancang memiliki sistem informasi yang hanya mencakup pertanyaan dan jawaban, serta jawaban hanya berbentuk angka dari 1 sampai 6 | Jika perekapan di penelitian sebelumnya perekapan nilai didapatkan dari bobot jawaban yang dipilih, maka pada penelitian ini perekapan didapatkan dari kebenaran dari jawaban yang dipilih serta jawaban berbentuk <i>text</i> ataupun gambar |
| 3 | Aplikasi Tryout Ujian Online untuk SMA/SMK Sederajat (<i>Online Examination Tryout Application for Senior and Vocational High School Students</i>) (Kumalasari, Erika, & Mustikasari, 2017) | Pada sistem informasi <i>try out</i> yang dirancang sebelumnya, peserta hanya dapat mengerjakan <i>try out</i> apabila telah dimulai oleh admin sehingga waktu pengerjaan tidak fleksibel | Jika waktu pengerjaan <i>try out</i> pada penelitian sebelumnya memiliki waktu pengerjaan yang terbatas dan tergantung admin, maka pada penelitian ini waktu pengerjaan dapat dilakukan kapan saja serta dapat diulangi sebanyak apapun peserta mau |
| 4 | Pembuatan Aplikasi Sistem Ujian Online Pada SMK Garuda Nusantara Bekasi (Riyadi, Hermaliani, & Utami, 2019) | Pada penelitian sebelumnya sistem informasi <i>try out</i> , jawaban haruslah berbentuk <i>text</i> | Jika pada penelitian sebelumnya jenis jawaban haruslah berbentuk <i>text</i> , maka pada penelitian ini jawaban dapat berbentuk <i>text</i> atau gambar sesuai dengan soal CPNS yang ditentukan oleh pemerintah |
| 5 | Aplikasi Learning Manajemen Sistem Dan Ulangan | Pada perancangan sistem informasi <i>try out</i> sebelumnya, | Jika pada penelitian sebelumnya tidak adanya fitur <i>review</i> jawaban |

| No | Judul Penelitian | Cakupan Penelitian | Kesenjangan Penelitian |
|----|--|---|---|
| | Online Berbasis Web (Ekaputri, Suryatiningsih, & Siswanto, 2016) | sistem <i>try out</i> tidak memiliki fitur <i>review</i> jawaban setelah menyelesaikan <i>try out</i> | setelah menyelesaikan <i>try out</i> , maka pada penelitian ini akan diterapkan fitur <i>review</i> jawaban sehingga peserta <i>try out</i> dapat mengetahui kekurangan dari hasil <i>try out</i> |

Berdasarkan kesenjangan penelitian sebelumnya yang telah dijelaskan, maka penelitian yang selanjutnya yaitu Rancang Bangun Sistem Informasi *Try Out* Untuk Soal CPNS Berbasis *Web*, sehingga aplikasi akan memiliki fitur *try out* meliputi pengerjaan *try out*, *review* jawaban dan unduh materi CPNS. Aplikasi ini akan membantu masyarakat yang berniat mengikuti *test* CPNS untuk belajar dan mengerjakan soal *try out* berbasis *Computer Asseted Test* (CAT), dimana sistem CAT ini digunakan pada *test* CPNS yang sesungguhnya.

2.3. Sistem Informasi

Sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk menyediakan kepada pihak luar tertentu dengan informasi yang diperlukan untuk pengambilan keputusan (Antonio & Safriadi, 2012)

2.4. Konsep Dasar Perancangan

Perancangan atau rancang merupakan serangkaian prosedur untuk menerjemahkan hasil analisa dari sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem diimplementasikan (Buchari, Sentinuwo, & Lantang, 2015).

2.5. Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah aplikasi dari sebuah pendekatan kuantifiabel, disiplin dan sistematis kepada pengebangan, operasi, dan

pemeliharaan perangkat lunak yaitu aplikasi dan rekayasa perangkat lunak (Pressman R. S., 2002).

2.6. Website

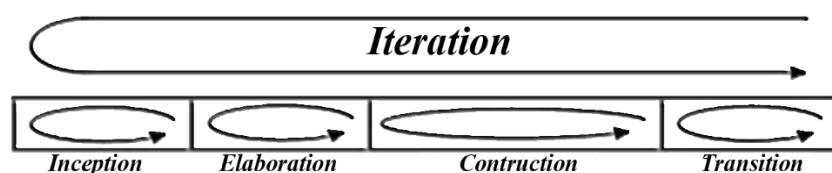
Website adalah keseluruhan halaman-halaman web yang terdapat dalam sebuah domain yang mengandung informasi (Yuhefizar, Mooduto, & Hidayat, 2009).

2.7. Aplikasi Web

Aplikasi merupakan penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi output (Irawan & Setiyorini, 2017). Sedangkan aplikasi *web* merupakan sekumpulan file *hypertext* yang saling terhubung untuk menunjukkan informasi-informasi tertentu dengan menggunakan teks-teks. Aplikasi web saat ini sedang berevolusi menjadi lingkungan komputasi yang canggih tidak hanya menyajikan fitur-fitur mandiri, fungsi-fungsi komputasi, dan isi untuk pengguna akhir, namun juga terintegrasi dengan sistem basis data yang dimiliki oleh perusahaan dan juga terintegrasi dengan aplikasi-aplikasi bisnis lainnya (Pressman R. , 2012).

2.8. Rational Unified Process (RUP)

Rational Unified Process (RUP) adalah pendekatan pengembangan perangkat lunak yang dilakukan secara berulang-ulang, fokus pada arsitektur, lebih diarahkan berdasarkan penggunaan kasus. *RUP* juga merupakan sebuah produk proses perangkat lunak yang dikembangkan oleh *Rational Software* yang diakuisisi oleh IBM pada bulan Februari 2003 (Sukamto & Shalahuddin, 2018). *RUP* memiliki empat buah tahapan yang dapat dilakukan secara iteratif.



Gambar 2.1. Tahapan Rational Unified Process

(Sukamto & Shalahuddin, 2018)

Berikut merupakan penjelasan untuk setiap tahapan pada *RUP* (Sukamto & Shalahuddin, 2018) :

1. Tahap pertama adalah *Inception* (permulaan) lebih pada pemodelan proses bisnis yang dibutuhkan (*business modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*). Berikut adalah tahapan yang dibutuhkan pada tahap ini:
 - a. Memahami ruang lingkup dari proyek (termasuk pada biaya, waktu, kebutuhan, resiko dan lain sebagainya)
 - b. Membangun kasus bisnis yang dibutuhkan.

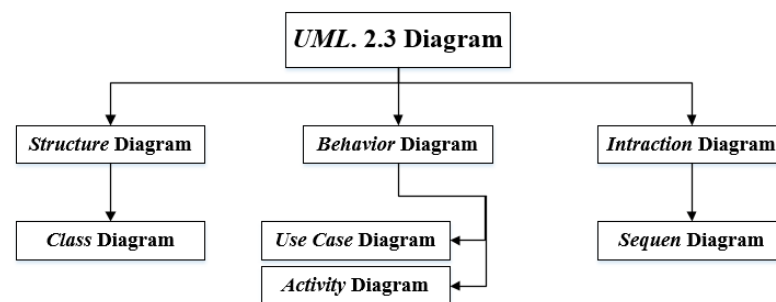
Ditahap ini memetakan kebutuhan sistem, dengan menggunakan *Work Breakdown Structure (WBS)*.

2. Tahap kedua adalah *Elaboration* (perluasan/perencanaan), tahap ini lebih difokuskan kepada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan bisa dibuat atau tidak. Mendeteksi resiko yang mungkin terjadi dari arsitektur yang dibuat. Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada purwarupa sistem (*prototype*). Pada tahap ini menggunakan pemodelan dan *UML*.
3. Tahap ketiga adalah *Construction* (konstruksi), tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. Tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program. Tahap ini menghasilkan produk perangkat lunak. Pada tahap ini, mengimplementasikan kode program dengan XAMPP yang didukung dengan perangkat lunak lainnya.
4. Tahap keempat adalah *Transition* (transisi), tahap ini lebih pada instalasi sistem agar dapat dimengerti oleh *user*. Tahap ini menghasilkan produk perangkat lunak. Aktivitas pada tahap ini termasuk pada pelatihan *user*, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan *user*. Tahapan uji coba ini menggunakan pengujian *black-box*, yang bertujuan untuk menguji sistem dengan analisis atau tahapan yang ada pada metodologi *RUP*.

Akhir dari keempat fase ini adalah produk perangkat lunak yang sudah lengkap. Keempat fase pada *RUP* dijalankan secara berurutan dan iteratif dimana sistem interaksi dapat digunakan untuk memperbaiki interaksi berikutnya.

2.9. Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah salah satu standar bahasa yang paling banyak digunakan didunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.



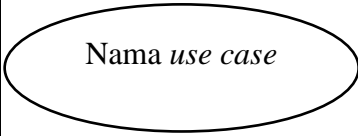
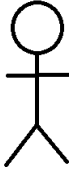


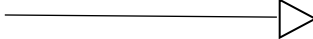
Gambar 2.2. Diagram *Unified Modeling Language*

(Sukamto & Shalahuddin, 2018)

Adapun *UML* yang akan digunakan untuk penelitian ini meliputi diantaranya (Sukamto & Shalahuddin, 2018) :

1. *Use Case* atau diagram *use case*, merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan *login* antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi yang ada. Berikut adalah simbol-simbol yang ada didalam diagram *use case* :

Tabel 2.2. Simbol-Simbol Diagram *Use Case*


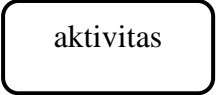
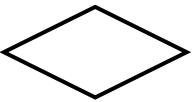

| Simbol | Deskripsi |
|---|---|
| <p><i>Use case</i></p>  | <p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>.</p> |
| <p>Aktor/<i>actor</i></p>  | <p>Orang, proses, atau sistem lain yang berintraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p> |
| <p>Asosiasi / <i>association</i></p>  | <p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> yang memiliki <i>login</i> dengan aktor.</p> |
| <p>Ekstensi/ <i>extend</i></p> <p><< <i>extend</i> >></p>  | <p>Relasi <i>use case</i> tambahan menuju sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.</p> |
| <p>Generalisasi/ <i>generalization</i></p>  | <p>Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana salah satu fungsi adalah fungsi yang lebih umum dari lainnya.</p> |

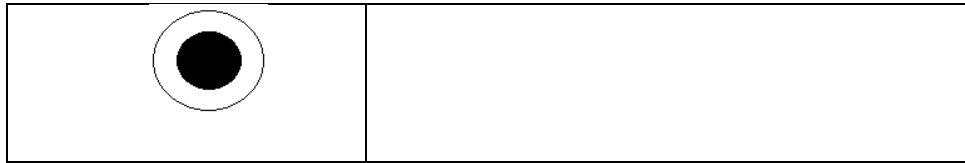
| | |
|--|--|
| <p>Menggunakan / <i>include</i> / <i>uses</i></p> <p><i><<include>></i> →</p> <p><i><<uses>></i> →</p> | <p>Relasi tambahan ke sebuah <i>use case</i>, dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.</p> |
|--|--|

Sumber : Sukamto & Shalahuddin (2018)

2. *Activity Diagram*, menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu di perhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan oleh aktor, jadi aktivitas apa yang dilakukan oleh sistem.

Tabel 2.3. Notasi *Activity Diagram*



| Simbol | Deskripsi |
|--|--|
| <p>status awal</p>  | <p>Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.</p> |
| <p>aktivitas</p>  | <p>Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.</p> |
| <p>percabangan / <i>decision</i></p>  | <p>Asosiasi percabangan, dimana jika ada pilihan lebih dari satu aktivitas.</p> |
| <p>penggabungan / <i>join</i></p>  | <p>Asosiasi penggabungan dimana lebih dari satu aktivitas, digabungkan menjadi satu aktivitas.</p> |
| <p>status akhir</p> | <p>Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir dari aktivitas.</p> |


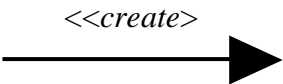
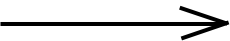


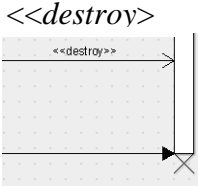


Sumber : Sukanto & Shalahuddin (2018)

3. *Sequence* Diagram, menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan kemudian diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek apa saja yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang dibuat menjadi objek itu.

Tabel 2.4. Notasi *Sequence* Diagram

| Simbol | Deskripsi |
|--|---|
| aktor  Nama_aktor atau <div style="border: 1px solid black; padding: 2px; display: inline-block;">Nama_aktor</div> tanpa waktu aktif | Orang, proses, atau sistem lain yang <i>berlogin</i> dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari <i>actor</i> adalah gambar orang, tapi <i>actor</i> belum tentu merupakan orang. |
| garis hidup / <i>lifeline</i>  | Menyatakan kehidupan suatu objek. |
| objek <div style="border: 1px solid black; padding: 2px; display: inline-block;">Nama_objek : nama kelas</div> | Menyatakan objek yang <i>berlogin</i> pesan. |




| | |
|---|--|
| <p>Waktu aktif</p>  | <p>Menyatakan objek dalam keadaan aktif dan <i>berlogin</i>, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.</p> |
| <p>Pesan tipe <i>create</i></p>  | <p>Menyatakan suatu objek akan membuat objek yang lain, arah panah mengarah pada objek yang akan dibuat.</p> |
| <p>Pesan tipe <i>call</i></p> <p>1 :</p>  | <p>Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.</p> |
| <p>Pesan tipe <i>send</i></p> <p>1 : masukan</p>  | <p>Menyatakan bahwa suatu objek mengirimkan data/masukan informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p> |
| <p>Pesan tipe <i>return</i></p> <p>1 : keluaran</p>  | <p>Menyatakan suatu objek yang telah menjalankan suatu operasi atau metode yang menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang akan menerima kembalian itu.</p> |
| <p>Pesan tipe <i>destroy</i></p>  | <p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang akan diakhiri, sebaliknya jika ada <i>create</i> maka akan ada <i>destroy</i>.</p> |


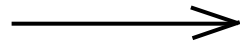
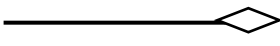
Sumber : Sukamto & Shalahuddin (2018)

4. *Class Diagram*, menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sebuah sistem. Diagram

kelas dibuat agar pembuat program membuat kelas-kelas sesuai dengan rancangan di dalam diagram kelas, agar antara dokumentasi perancangan dan perangkat lunak dapat sinkron. Kelas-kelas yang ada pada struktur sistem harus bisa melakukan fungsi-fungsi sesuai dengan kebutuhan sistem, sehingga pembuat perangkat lunak atau *programmer* dapat membuat kelas-kelas di dalam program perangkat lunak sesuai dengan perancangan diagram kelas yang telah dibuat.

Tabel 2.5. Simbol-Simbol Pada *Class Diagram*

| Simbol | | | Deskripsi |
|---|------------|--|--|
| | Nama_Kelas | | Kelas pada struktur sistem. |
| | +Atribut | | |
| | +operasi() | | |
| kelas Antarmuka /  Nama_interface | | | Sama dengan konsep dalam <i>interface</i> dalam pemrograman berorientasi objek. |
| Asosiasi /  | | | Relasi antar kelas dengan makna umum, asosiasi biasanya disertai dengan <i>multiplicity</i> . |
| Asosiasi berarah <i>directed association/associatio</i>  | | | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> . |

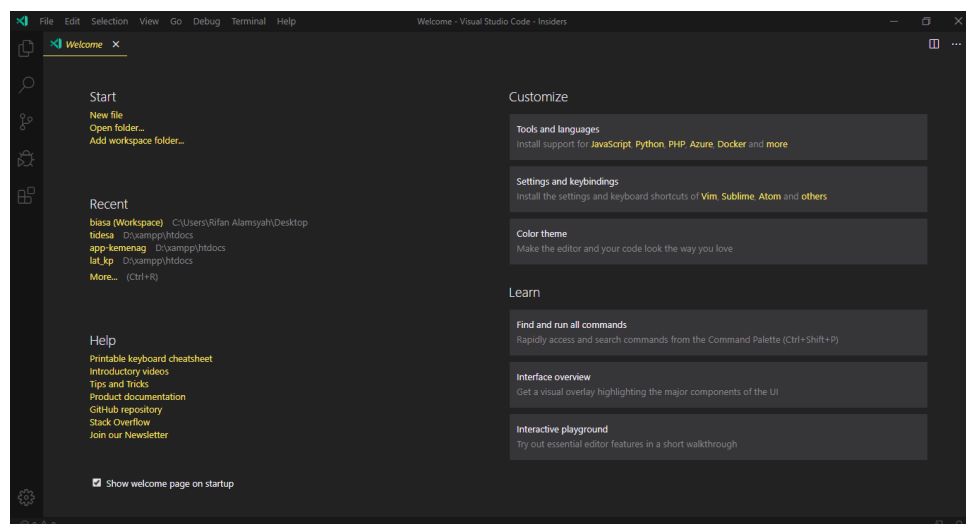
| | |
|--|--|
| <p>generalisasi</p>  | <p>Relasi antar kelas dengan makna / arti generalisasi-spesialisasi (umum - khusus).</p> |
| <p>Kebergantungan <i>/dependency association</i></p>  | <p>Relasi antar kelas dengan makna kebergantungan antar sebuah kelas.</p> |
| <p>agregasi <i>/aggregation</i></p>  | <p>Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>).</p> |

Sumber : Sukamto & Shalahuddin (2018)

2.10. Software Pendukung

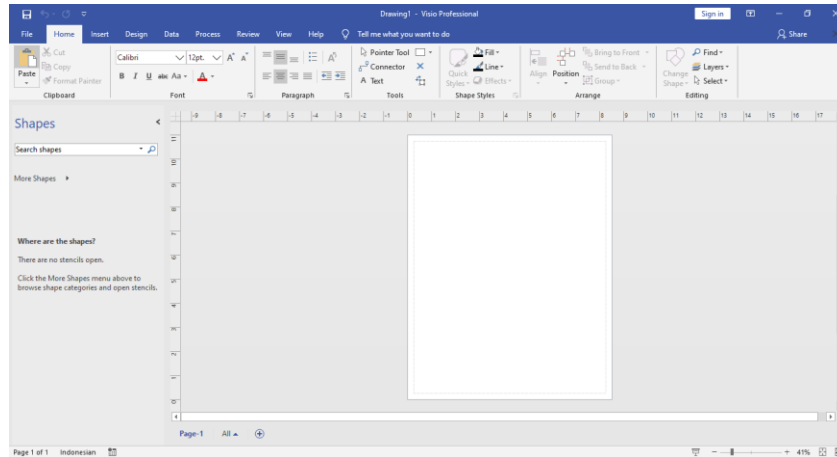
Adapun *tools* / alat bantu dalam pengimplementasian sistem informasi *try out* untuk soal cpns berbasis *web* yaitu sebagai berikut :

1. *Visual Studio Code* adalah editor kode yang ringan namun kuat yang berjalan di desktop dan tersedia untuk Windows, macOS dan Linux. Muncul dengan dukungan bawaan untuk JavaScript, TypeScript dan Node.js dan memiliki ekosistem ekstensi yang kaya untuk bahasa lain (seperti C ++, C #, Java, Python, PHP, Go) dan runtimes (seperti .NET dan Unity). (VSCode, 2020);



Gambar 2.3. Tampilan Home Visual Studio Code

2. *Microsoft visio* adalah sebuah program aplikasi komputer yang dirilis oleh *Microsoft Corporation*. Aplikasi ini menggunakan grafik vektor untuk membuat diagram (Ratna, 2014);



Gambar 2.4. Tampilan *Workspace* Microsoft Visio

3. Bahasa pemrograman, yang digunakan adalah sebagai berikut.
- HTML (Hyper Text Markup Language)*, yaitu bahasa pemformatan teks untuk dokumen-dokumen pada jaringan komputer yang sering disebut sebagai *world wide web*, yang kemudian dapat diakses untuk menampilkan berbagai informasi di dalam sebuah penjelajahan *web* internet / *browser* (Rerung, 2018);
 - CSS (Cascading Style Sheet)*, adalah bahasa yang dapat digunakan untuk mendefinisikan bagaimana suatu bahasa *markup* ditampilkan pada suatu media dimana bahasa *markup* ini salah satunya adalah *HTML* (Rerung, 2018);
 - PHP*, merupakan singkatan dari *Hypertext Preprocessor* adalah bahasa pemrograman *script server-side* yang didesain untuk pengembangan *web*. *PHP* disebut sebagai 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)*. *PHP* dapat digunakan dengan gratis (*free*) dan bersifat *open source* (Lalompoh, 2018);
 - JavaScript*, adalah bahasa *scripting* yang populer diinternet dan dapat bekerja di sebagian besar *browser* populer seperti *Internet Explorer*

(IE), *Mozilla FireFox*, *Netscape* dan *Opera*. Kode *Javascript* dapat disisipkan dalam halaman *web* menggunakan *script* (Sunyoto, 2017).

4. *Framework*, adalah sebuah kerangka kerja. *Framework* juga dapat diartikan juga sebagai kumpulan dari *script* (terutama *class* dan *function*) yang dapat membantu *developer / programmer* dalam menangani berbagai masalah-masalah dalam pemrograman seperti koneksi ke *database*, pemanggilan *variabel*, *file*, dll. sehingga *developer* lebih cepat membangun aplikasi (Warsito, Yusup, & Yulianto, 2014). Pada pembuatan aplikasi, menggunakan *framework Codeigniter* dan *Bootstrap*.
 - a. *Laravel*, adalah sebuah *framework PHP* yang dirilis dibawah lisensi MIT, dengan kode sumber yang sudah disediakan oleh Github, sama halnya seperti *framework-framework* yang lain, *Laravel* dibangun dengan konsep MVC (Model-Controller-View), kemudian *Laravel* dilengkapi juga command line tool yang bernama *Artisan* yang bisa digunakan untuk packaging bundle dan instalasi bundle melalui command prompt (Aminudin, 2015);

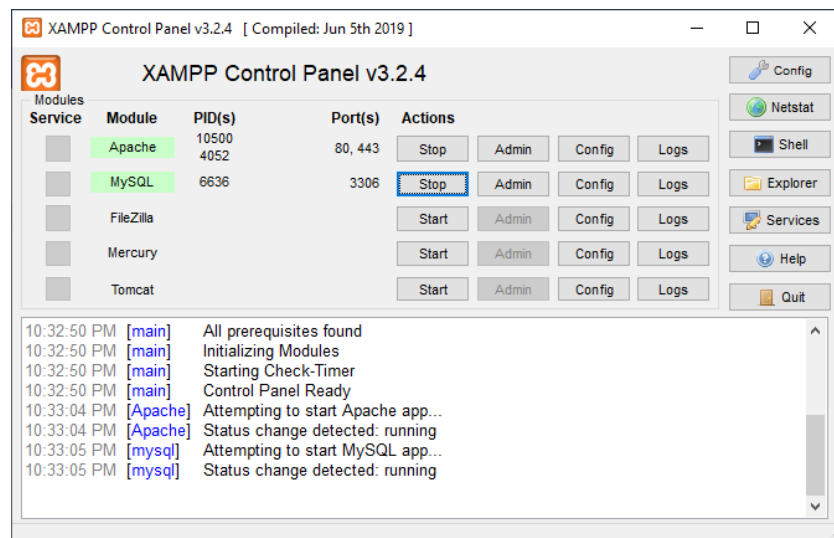


Gambar 2.5. Tampilan awal *Laravel*

- b. *Bootstrap*, adalah *framework front-end* yang *powerfull* untuk pengembangan sebuah aplikasi *web* agar lebih cepat dan mudah. *bootstrap* menggunakan *HTML*, *CSS*, dan *Javascript* (Enterprise, 2016).
5. *Command Prompt (CMD)*, untuk memakai fungsi dari *framework Laravel*, maka dibutuhkan *CMD*. Pengertian *Command Prompt* atau yang sering kita

sebut dengan istilah *CMD* adalah suatu perintah *DOS* berbasis teks pada sistem operasi *Windows* (mstekno, 2017);

6. XAMPP, adalah perangkat lunak bantu yang dapat digunakan untuk melakukan pemrograman berbasis web tanpa harus melakukan instalasi paket perangkat lunak yang terpisah-pisah. XAMPP berfungsi sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL* database dan penerjemah bahasa yang ditulis kedalam bahasa pemrograman *PHP* dan *Perl* (Lalompoh, 2018).

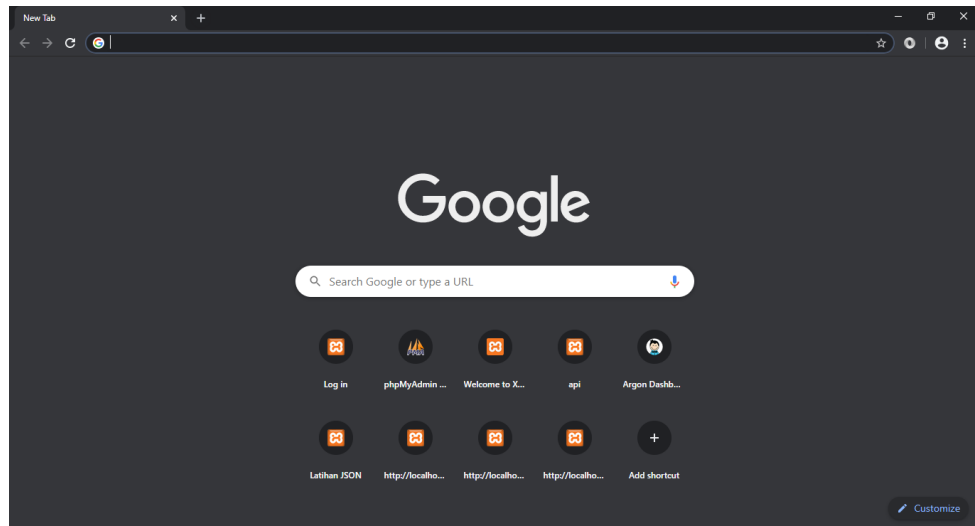


Gambar 2.6. Tampilan XAMPP

- a. *Apache*, sebuah nama *web server* yang bertanggung jawab pada *request-response* HTTP dan *logging* informasi secara detail. Selain itu, *Apache* juga disebut sebagai suatu *web server* yang mengikuti standar protokol HTTP, dan tentu saja sangat digemari (Irza, Zulhendra, & Efrizon, 2017);
- b. *MySQL*, adalah sebuah *server database open source* yang terkenal yang digunakan berbagai aplikasi terutama untuk *server* atau membuat *web*. *MySQL* berfungsi sebagai *SQL(Structured Query Language)* yang dimiliki sendiri dan sudah diperluas, umumnya bersamaan dengan *PHP* untuk membuat aplikasi *server* yang dinamis dan lebih *powerfull* (Lalompoh, 2018);
- c. *PHPMyadmin*, adalah sebuah aplikasi *open source* / gratis yang berfungsi untuk memudahkan manajemen *MySQL*. *PhpMyAdmin*

dapat membuat *database*, membuat tabel, memasukan, menghapus dan mengubah data dan terasa lebih mudah, tanpa perlu mengetikkan perintah *SQL* secara manual (Hidayat, Marlina, & Utami, 2017).

7. *GoogleChrome*, *google* meluncurkan web browser dengan meluncurkan *Chrome*, sebuah *web browser* yang memiliki kecepatan, kemudahan penggunaan yang baik. (Faiz, Umar, & Yudhana, 2017);



Gambar 2.7. Tampilan awal *GoogleChrome*

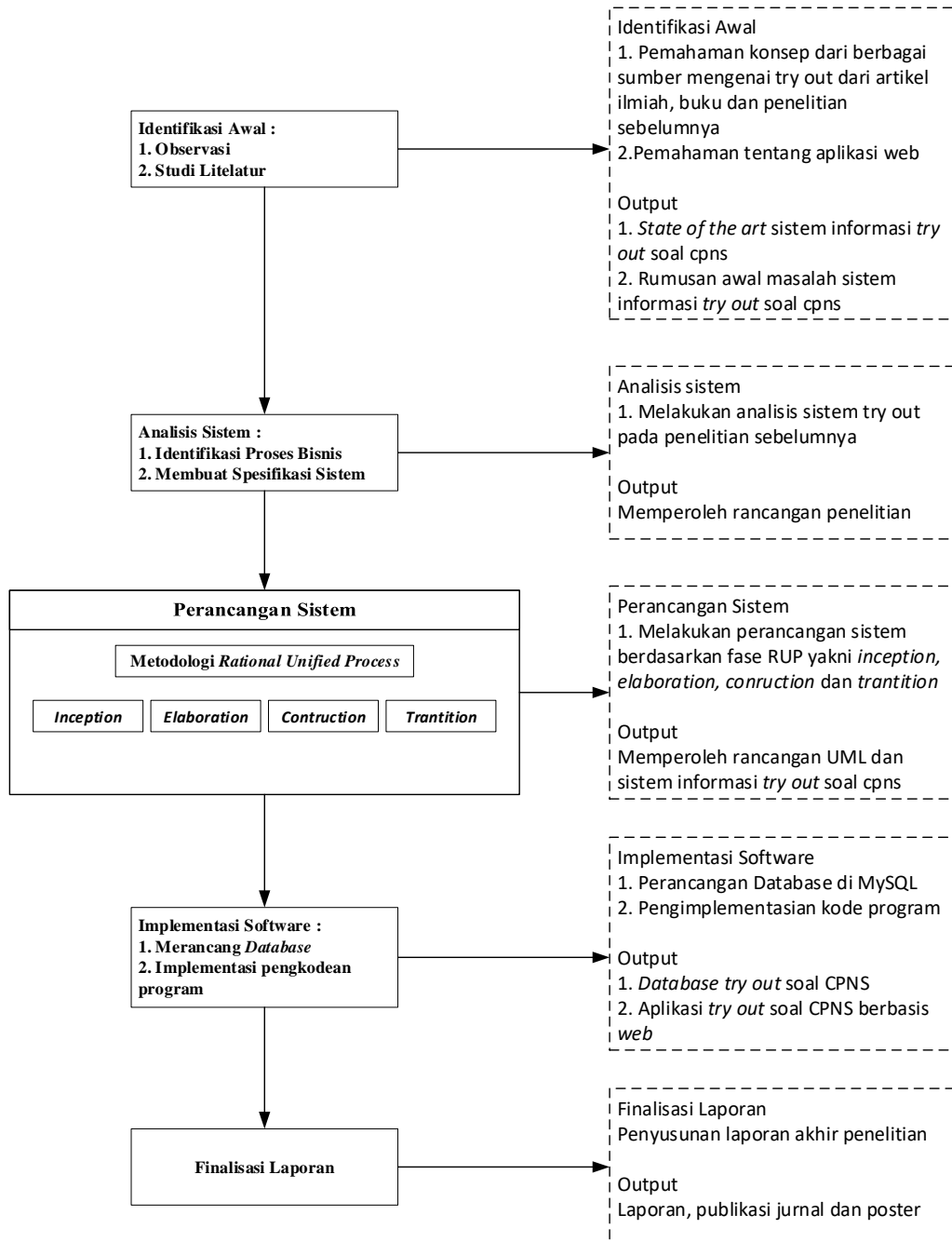
3. METODOLOGI PENELITIAN

Pada penelitian kali ini, metodologi yang digunakan adalah metodologi Rational Unified Process (RUP), RUP merupakan sebuah framework dengan struktur yang longgar dalam proses pengembangan software. Metodologi RUP digambarkan dengan bahasa pemodelan Unified Modeling Language (UML) yang merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, Object Modeling Technique (OMT) dan Object Oriented Software Engineering (OOSE) (Munawar, 2018). Tahapan yang akan digambarkan pada pengembangan aplikasi ini mencakup 4 tahapan, yaitu tahap pembuatan Use Case Diagram, Class Diagram, Activity Diagram dan Sequence Diagram. Dengan menggunakan metodologi RUP, proses pengembangannya dilakukan secara iteratif atau berulang, melalui 4 fase, yaitu Inception, Elaboration, Construction dan Transition.

3.1. Kerangka Pemikiran

Kerangka pemikiran merupakan suatu diagram yang digunakan untuk menjelaskan secara garis besar alur logika berjalannya sebuah penelitian. Kerangka pemikiran pula berfungsi sebagai explanation yaitu untuk menjelaskan, prediction untuk memprediksi atau meramalkan dan control yaitu sebagai pengendalian suatu gejala pada penelitian. Pada penelitian kali ini yang dilakukan pertama kali yaitu melakukan identifikasi awal penelitian, dengan melakukan observasi dan studi literature. Observasi dilakukan untuk mencari informasi mengenai penelitian yang akan dilakukan, sedangkan studi literatur dilakukan untuk mendapatkan rumusan masalah serta kesenjangan penelitian. Kemudian langkah selanjutnya yang dilakukan adalah identifikasi proses bisnis dan pembuatan spesifikasi sistem. Dilakukan untuk menentukan teknik pengolahan data dan fungsi penelitian yang dilakukan. Lalu pada langkah perancangan sistem dan aplikasi, dilakukan perancangan sistem sesuai dengan metodologi yang digunakan yaitu RUP dengan fase-fase yang harus dilakukan. Pada perancangan aplikasi, dilakukan perancangan struktur menu dan interface, agar aplikasi yang akan dibuat sudah tergambar dengan jelas. Terdapat pula langkah implementasi aplikasi, yang berfungsi untuk pembuatan database dan pengimplementasian kode program agar menjadi suatu aplikasi yang utuh. Kemudian langkah pengujian aplikasi, langkah ini dilakukan

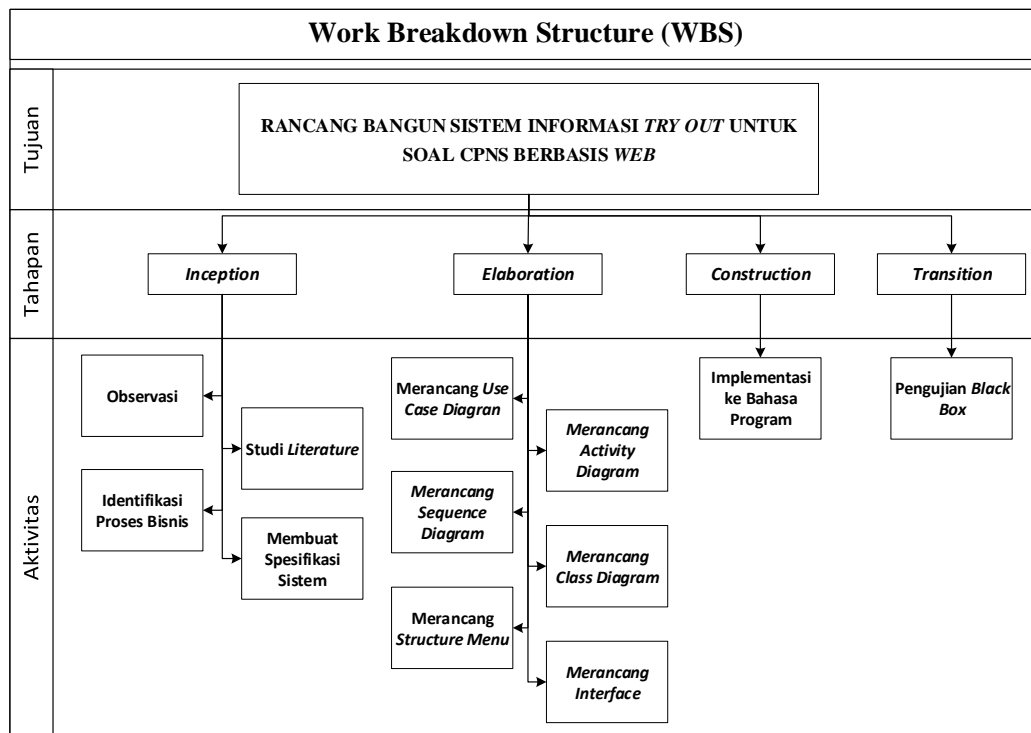
untuk menguji apakah aplikasi yang dibuat sudah berfungsi dengan baik atau tidak. Yang terakhir langkah finalisasi laporan, berisi laporan penelitian yang telah tersusun.



Gambar 3.1. Kerangka Pemikiran

3.2 Work Breakdown Structure (WBS)

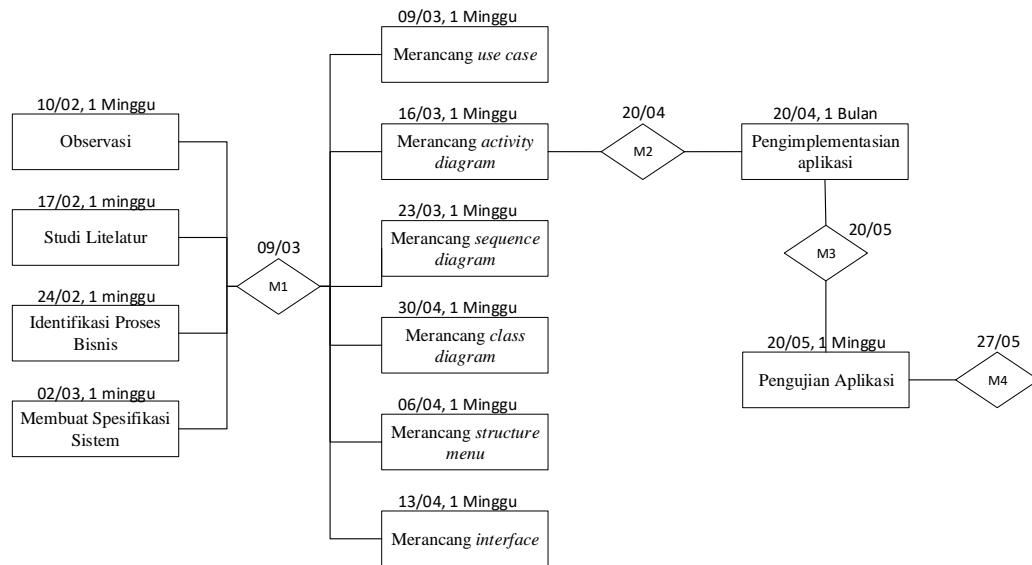
Work Breakdown Structure (WBS) merupakan sebuah cara yang digunakan untuk mendefinisikan dan mengelompokkan tugas-tugas dari sebuah proyek menjadi bagian-bagian kecil sehingga lebih mudah di atur (Andi Maddeppungeng, 2015). Urutan aktivitas yang dilakukan mulai dari perencanaan sampai dengan pengujian digambarkan menggunakan *Work Breakdown Structure (WBS)*. Berikut merupakan gambaran *WBS* menurut tujuan dan metodologi yang digunakan, seperti pada gambar 3.1.:



Gambar 3.2. Work Breakdown Structure (WBS)

Tahapan yang pertama dalam *WBS* adalah *inception*, dimana pada tahapan ini memiliki aktivitas pemodelan proses bisnis dan aktivitas pemodelan kebutuhan sistem, yang mana diperoleh dari hasil observasi dan studi *literature*. Tahapan kedua yaitu *elaboration*, pada tahapan ini berfokus pada arsitektur sistem, tahapan ini lebih pada analisis dan desain sistem serta implementasi purwarupa sistem (*prototype*), dimana tahapan ini di modelkan menggunakan *UML*, yang terdiri dari *use case diagram*, *activity diagram*, *sequence diagram*, *class diagram*, serta merancang struktur menu dan merancang *interface*. Tahapan ketiga yaitu *construction*, tahapan ini lebih pada implementasi sistem yang berfokus pada kode

program. Tahapan yang keempat yaitu *transition*, pada tahap ini juga dilakukan pengujian sistem apakah sudah memenuhi harapan dari *user*. Keempat tahapan ini merupakan aktivitas serta tahapan sebagai proses perancangan dan pembangunan dari aplikasi.



Gambar 3.3. Diagram Alur Perancangan Sistem

Penjelasan:

1. M1 = *Milestone* tahap 1 (*Inception*)
2. M2 = *Milestone* tahap 2 (*Elaboration*)
3. M3 = *Milestone* tahap 3 (*Construction*)
4. M4 = *Milestone* tahap 4 (*Transition*)

Adapun penjelasan dari *diagram* alur aktivitas pada gambar 3.3. yaitu sebagai berikut:

1. *Inception*, tahapan ini merupakan persiapan dari penelitian. Dalam tahapan *inception* dilakukan beberapa aktivitas yaitu:
 - a. Observasi, pada tahap ini dilakukan pengamatan dan analisa, pengumpulan data seperti jenis soal CPNS dan bagaimana sistem CAT berjalan. Dari hasil observasi ini didapatkan informasi mengenai sistem informasi *try out* CPNS.

b. Studi literatur, pada tahap ini dilakukan pencarian terhadap penelitian sebelumnya serta referensi teori yang relevan. Referensi tersebut berisi tentang:

- 1) Sistem *try out* CPNS;
- 2) Metodologi Penelitian;
- 3) Aplikasi Berbasis *Web*;
- 4) *Software* pendukung.

Referensi tersebut didapatkan dari buku, jurnal, maupun artikel laporan.

c. Identifikasi proses bisnis yang berjalan, pada tahap ini berupa penentuan target *actor* sebagai pengguna dari sistem dan menggambarkan hubungan antara pengguna dengan system. Pada hasil identifikasi proses bisnis didapatkan *actor* yang akan menggunakan aplikasi adalah masyarakat dan untuk khususnya adalah peserta CPNS.

d. Pembuatan spesifikasi sistem, tahap ini menggambarkan fungsi dari sistem dan kinerja sistem tersebut pada pengaplikasiannya sebagai aplikasi berbasis *web*. Pada hasil pembuatan spesifikasi sistem didapatkan informasi spesifikasi minimum sistem yang dapat menjalankan aplikasi *try out* CPNS yakni sebagai berikut:

- a. Windows dan Mac OS, spesifikasi *prosessor core 2 duo*, ram 1 gb dan *disk space* 1 gb;
- b. Android dan IOS, spesifikasi *prosessor single core 1.1 Ghz*, ram 512 mb dan *disk space* 100 mb;

2. *Elaboration*, pada tahap ini dilakukan aktivitas yaitu perancangan *use case*, *activity diagram*, *class diagram* dan *sequence diagram*, serta melakukan perancangan untuk struktur menu dan *interface* dari aplikasi yang akan dibuat. Pada tahapan *elaboration* didapatkan hasil rancangan sebagai berikut :

- a. *use case* aplikasi *try out* meliputi: *use case login* aplikasi yang menggambarkan hak akses *admin* dan *user*; *use case* registrasi menggambarkan proses pendaftaran oleh *user*; *use case try out* yang

menggambarkan proses *try out* dari pengerjaan soal sampai pembahasan soal; *use case* pengelolaan *user* yang menggambarkan pengelolaan *user* oleh *admin*; *use case* pengelolaan soal yang menggambarkan pengelolaan soal untuk implementasi *try out*; *use case* pengelolaan materi yang menggambarkan pengelolaan materi oleh *admin* yang dapat diakses oleh *user*; dan *use case* pengumuman yang menggambarkan bagaimana *admin* memasang pengumuman untuk *user*;

- b. *activity diagram* aplikasi *try out* meliputi: *activity diagram* login yaitu aktivitas awal untuk masuk kedalam aplikasi sesuai hak akses; *activity diagram* registrasi yaitu aktivitas user mendaftarkan diri untuk mendapatkan hak akses; *activity diagram try out* yakni aktivitas user mengerjakan *try out*; *activity diagram* pengelolaan soal yaitu aktivitas mengelola soal oleh *admin*; *activity diagram* pengelolaan *user* yaitu aktivitas mengelola *user* aplikasi; *activity diagram* pengelolaan materi yaitu aktivitas pengelolaan materi oleh *admin*; *activity diagram* pengumuman;
 - c. *sequence diagram* aplikasi *try out* meliputi : *sequence diagram* login; *sequence diagram* registrasi; *sequence diagram try out*; *sequence diagram* pengelolaan soal; *sequence diagram* pengelolaan *user*; *sequence diagram* pengelolaan materi; *sequence diagram* pengumuman;
 - d. *class diagram* aplikasi *try out* yang menggambarkan hubungan *class* login, *class* registrasi, *class try_out*, *class* soal, *class* user, *class* materi dan *class* pengumuman;
3. *Construction*, tahap ini melakukan implementasi kode program menggunakan bahasa pemrograman yang digunakan dan sesuai dengan pemodelan pada tahap *elaboration* sehingga akan aplikasi *try out* untuk soal CPNS yang siap diuji.
 4. *Transition*, pada tahap ini dilakukan pengujian aplikasi yang telah dibuat pada tahap *construction*, pengujian ini dilakukan untuk mengetahui kinerja dan fungsi dari aplikasi yang dibuat.

Adapun tabel dari aktivitas diatas yang bisa dilihat pada tabel 3.1. *detail activity* berikut:

Tabel 3.1. Detail Activity

| No. | Aktivitas | Masukan | Proses | Keluaran |
|-----|---|---|--|--|
| 1. | Observasi | Masalah Penelitian | Pengamatan dan analisa | Hasil Observasi |
| 2. | Studi Literatur | Buku, Jurnal dan Artikel Penelitian | Kesenjangan penelitian dan referensi teori | Informasi yang berkaitan dengan penelitian |
| 3. | Identifikasi Proses Bisnis | Proses bisnis yang berjalan pada try out CPNS | Penentuan target pengguna (<i>actor</i>) | Aktivitas dari pengguna sistem dengan sistem yang dibuat |
| 4. | Membuat Spesifikasi Sistem | Kebutuhan aplikasi yang akan dibuat | Penggambaran fungsi dari sistem serta kinerjanya | Rincian mengenai sistem yang akan dibuat |
| 5. | Merancang <i>use case</i> dan skenario <i>diagram</i> | Proses bisnis yang sedang berjalan | Penggambaran <i>use case diagram</i> | Gambar rancangan skenario <i>use case diagram</i> |

| No. | Aktivitas | Masukan | Proses | Keluaran |
|-----|--|--|---|--|
| 6. | Merancang <i>class diagram</i> | Berdasarkan <i>use case diagram</i> | Penggambaran <i>class diagram</i> | Gambar rancangan <i>class diagram</i> |
| 7. | Merancang <i>activity diagram</i> | Berdasarkan <i>use case diagram</i> | Penggambaran <i>activity diagram</i> | Gambar rancangan <i>activity diagram</i> |
| 8. | Merancang <i>sequence diagram</i> | Berdasarkan <i>use case diagram</i> | Penggambaran <i>sequence diagram</i> | Gambar rancangan <i>sequence diagram</i> |
| 9. | Merancang struktur menu dan <i>interface</i> | Berdasarkan <i>use case, class diagram, activity diagram</i> dan <i>sequence diagram</i> | Membuat gambaran menu dari aplikasi yang akan dibuat dan tampilan aplikasi tersebut | Gambar struktur menu dan <i>interface</i> aplikasi |
| 10. | Implementasi aplikasi | Berdasarkan tahapan dari beberapa diagram dan gambar yang dibuat pada tahap sebelumnya | Penyatuan semua tahap perancangan dengan implementasi kode program | Sistem Informasi <i>try out</i> untuk soal CPNS |

| No. | Aktivitas | Masukan | Proses | Keluaran |
|-----|--------------------------------------|----------------------------|--|--------------------------|
| 11. | Pengujian <i>BlackBox Testing</i> | Aplikasi yang telah dibuat | Pengujian aplikasi pada pengguna (<i>user</i>) | Hasil pengujian aplikasi |

Adapun sumber daya yang terlibat dalam penelitian, dapat dilihat pada tabel sumber daya penelitian berikut ini:

Tabel 3.2. Sumber Daya Penelitian

| No. | Aktivitas | Manusia | Perangkat |
|-----|---|----------------|---|
| 1. | Observasi | Rifan Alamsyah | Laptop |
| 2. | Studi Literatur | Rifan Alamsyah | Buku, Jurnal dan Artikel Penelitian |
| 3. | Identifikasi Proses Bisnis | Rifan Alamsyah | Laptop, Kertas A4 |
| 4. | Membuat Spesifikasi Sistem | Rifan Alamsyah | Laptop, Kertas A4 |
| 5. | Merancang <i>use case</i> dan skenario <i>diagram</i> | Rifan Alamsyah | Laptop dan <i>Microsoft Visio</i> |
| 6. | Merancang <i>class diagram</i> | Rifan Alamsyah | Laptop dan <i>Microsoft Visio</i> |
| 7. | Merancang <i>activity diagram</i> | Rifan Alamsyah | Laptop dan <i>Microsoft Visio</i> |
| 8. | Merancang <i>sequence diagram</i> | Rifan Alamsyah | Laptop dan <i>Microsoft Visio</i> |
| 9. | Merancang struktur menu dan <i>interface</i> | Rifan Alamsyah | Laptop, <i>Microsoft Visio</i> , <i>Sublime Text</i> dan <i>Google Chrome</i> |

| No. | Aktivitas | Manusia | Perangkat |
|-----|-----------------------|----------------|--|
| 10. | Implementasi Aplikasi | Rifan Alamsyah | <i>XAMPP, Sublime Text, Laptop dan Google Chrome</i> |
| 11. | Pengujian Aplikasi | Rifan Alamsyah | <i>XAMPP, Laptop dan Google Chrome</i> |

4. HASIL DAN PEMBAHASAN

4.1 Hasil Penelitian

Perancangan sekaligus pembuatan Sistem Informasi Try Out Untuk soal CPNS Berbasis *Web* menggunakan metodologi *Rational Unified Procces (RUP)*. Berikut adalah hasil dari pembahasan penelitian aktivitas yang terdapat pada metodologi *RUP*.

4.1.1 Inception

Pada tahap pertama ini dilakukan identifikasi proses bisnis yang didapat dari referensi dokumen dan jurnal terkait penelitian yang ada sebelumnya. Pada tahap ini juga akan digambarkan diagram *use case* dan diagram aktivitas.

1. Observasi, pada kegiatan ini dilakukan pencarian data yang berhubungan dengan *try out* dan soal yang ada dalam *test* CPNS dengan membaca dan menelaah soal CPNS yang ada di buku dan mengetahui proses bisnis dari sistem yang sudah ada pada *test* sesungguhnya. Hasil observasi menunjukkan bahwa soal CPNS memiliki tingkat kesulitan tersendiri, karena soal CPNS memiliki 3 jenis soal yakni Tes Intelegensi Umum (TIU), Test Wawasan Kebangsaan (TWK) dan Tes Karakteristik Pribadi (TKP). Hasil observasi menunjukkan juga bahwa soal CPNS memiliki jumlah 100 soal dengan komposisi 35 soal TKP, 35 soal TIU dan 30 soal TWK serta dikerjakan dalam waktu 90 menit. Oleh karena itu peserta CPNS haruslah terus berlatih atau *tryout*, baik itu latihan mandiri dengan mengerjakan di buku ataupun mengikuti kegiatan *tryout* yang biasa diselenggarakan oleh organisasi tertentu, dan oleh sebab itu dirancanglah Sistem Informasi CPNS yang memiliki latihan *tryout* sesuai dengan *test* CPNS yang sesungguhnya.
2. Studi litelatur, pada penelitian ini didapatkan kesenjangan penelitian yang dibandingkan dengan penelitian sebelumnya, untuk jurnal yang pertama hanya berfokus pada penyimpanan soal saja dan tidak diimplementasikan pada fitur apapun. Jurnal kedua memiliki kekurangan pada jenis penjawaban soal dan perlu pengembangan untuk variasi jawaban. Jurnal ketiga memiliki kekurangan pada waktu pengerjaan *try out* yang terbatas

dan hanya bisa dimulai oleh admin. Jurnal Keempat, jenis jawaban yang digunakan hanyalah berbentuk teks. Jurnal kelima, *try out* tidak memiliki fitur review soal jadi peserta *try out* tidak bisa mengetahui jawaban yang benar dan pembahasanya. Sedangkan penelitian sekarang penulis membuat Sistem Informasi *Try Out* Untuk Soal CPNS Berbasis *Web*, dengan dilengkapi fitur review soal, jawaban yang beragam sesuai dengan standar jenis soal CPNS dan dapat diakses kapanpun dimanapun dengan perangkat yang memiliki akses ke internet. Dengan adanya pengumpulan data ini, sehingga menjadi salah satu penunjang dalam menyelesaikan sistem informasi *try out* untuk soal CPNS berbasis *web*.

3. Identifikasi proses bisnis, secara singkat proses bisnis dari sistem informasi *try out* soal CPNS berbasis *web* yakni terdapat 2 *role* pengguna yang dapat mengakses aplikasi yakni *user* dan *admin* dimana setiap *role* memiliki tugas yang berbeda, *user* dapat mengerjakan tryout, melihat jawaban hasil tryout dan melihat materi serta informasi seputar CPNS, sedangkan *admin* memiliki tugas untuk mengelola soal yang akan dikerjakan oleh *user* dan mengelola materi serta informasi yang akan di sajikan kepada *user*.
4. Membuat spesifikasi sistem didefinisikan untuk menentukan kebutuhan apa saja yang menunjang dalam membangun aplikasi. Rincian persyaratan meliputi sebagai berikut :
 - a. Persyaratan tampilan, aplikasi ini haruslah memiliki tampilan web yang responsif baik dibuka melalui *desktop* ataupun *smartphone*, karena tujuan aplikasi dapat dibuka dimana dan kapan saja jadi aplikasi haruslah memiliki tampilan yang baik untuk setiap *device* yang digunakan.
 - b. Persyaratan sistem digunakan untuk tahap awal kebutuhan spesifikasi aplikasi yakni:
 - 1) Spesifikasi fungsional sistem, yakni aplikasi dapat diakses secara *online* dan memiliki perbedaan hak akses.
 - 2) Spesifikasi nonfungsional sistem, adapun persyaratan minimum untuk sistem informasi yang akan dibangun adalah sebagai berikut :

- a) Sisi Pengembang, diambil dari website pemrograman terkemuka yaitu dicoding dengan alamat <https://www.dicoding.com/academies/123> diakses pada 29 mei 2029 bahwa persyaratan minimum untuk pengembangan *web* adalah sebagai berikut :

Processor : Intel Celeron atau lebih baik

Ram : 1Gb atau lebih besar

Resolusi Layar : 1366x768px atau lebih baik

- b) Sisi Pengguna, untuk sisi pengguna yang mengakses aplikasi lebih fleksibel karena platform yang digunakan adalah *web*, maka dari itu perangkat yang digunakan adalah perangkat yang dapat mengakses internet melalui *browser* yang ada.

c. Persyaratan pengembangan

- 1) Perancangan sistem menggunakan pemodelan *Unified Modeling Language* dengan *use case* diagram sebagai gambaran antara pengguna dengan sistem;
- 2) Perancangan *database* dengan menggunakan fitur artisan pada *framework laravel* yang diimplementasikan pada XAMPP; dan
- 3) Implementasi program ke bahasa pemrograman PHP dengan menggunakan *framework laravel*.

- d. Batasan Sistem, adapun batasan sistem yang dibuat untuk user hanya dapat mengakses informasi dan materi serta mengerjakan latihan soal/*tryout* yang terdapat pada *web* dan disediakan oleh admin.

4.1.2 Elaboration

Dimana pada tahapan ini menentukan penggambaran awal suatu sistem dan pembuatan rancangan, atau disebut juga sebagai arsitektur sistem. Penggambaran awalnya diimplementasikan kedalam *use case diagram*, *activity diagram*, *sequence diagram*, *class diagram*, struktur menu dan *interface*.

4.1.2.1 Perancangan Use Case Diagram

Perancangan *use case diagram* dimulai dengan indentifikasi aktor dan indentifikasi *use case diagram*.

1. Identifikasi aktor, adalah proses menentukan seseorang yang *login* dengan sistem. Dari pengumpulan data yang tersedia, didapatkan aktor dari sistem informasi *tryout* untuk soal CPNS diantaranya sebagai berikut :

Tabel 4.1 Identifikasi Aktor

| No | Aktor | Aktifitas |
|----|-------|--|
| 1 | Admin | Aktor yang mengelola informasi dan soal yang ada pada aplikasi |
| 2 | User | Aktor yang menggunakan aplikasi dan melihat informasi seputar CPNS serta mengerjakan <i>tryout</i> yang disediakan |

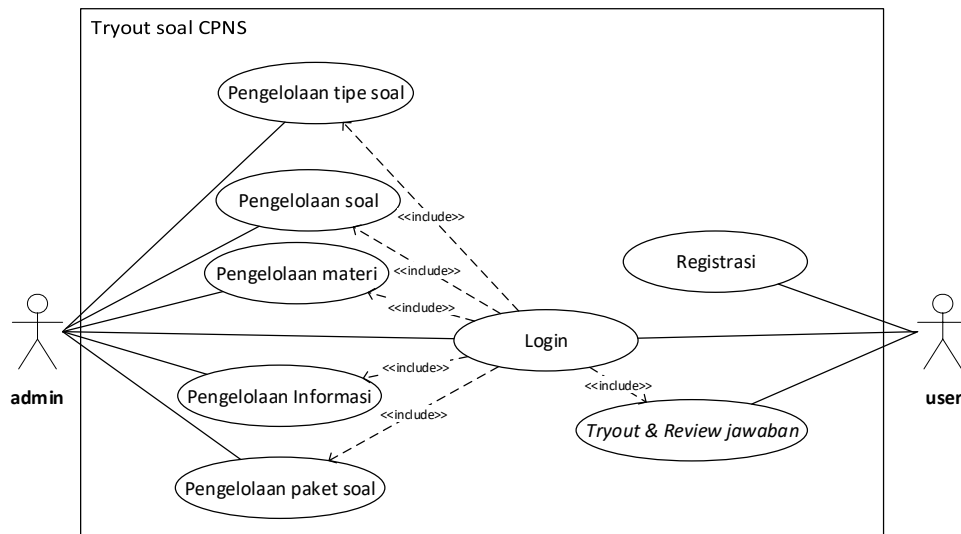
2. Identifikasi *use case*, berikut ini adalah hasil indentifikasi *use case* pada *tryout* soal CPNS dapat dilihat pada tabel 4.2

Tabel 4.2 Identifikasi Use Case

| No | Use Case | Deskripsi |
|----|--------------|---|
| 1 | <i>Login</i> | <i>Login</i> digunakan untuk memberikan akses kepada user atau admin guna mengakses aplikasi, dilengkapi juga dengan fitur lupa <i>password</i> |

| No | Use Case | Deskripsi |
|----|----------------------------|--|
| 2 | Registrasi | Registrasi digunakan untuk pendaftaran user yang ingin mendapatkan akses aplikasi |
| 3 | Pengelolaan paket soal | Mengelola paket soal yang dapat dipilih oleh user |
| 4 | Pengelolaan tipe soal | Tipe soal berguna untuk membedakan soal yang disajikan pada penginputan soal |
| 5 | Pengelolaan soal | Pengelolaan soal berfungsi untuk memasukan soal berdasarkan paket yang dipilih |
| 6 | Pengelolaan materi | Pengelolaan materi dimasukan oleh admin dan bisa diakses oleh user |
| 7 | Pengelolaan informasi CPNS | Pengelolaan informasi berisi tips dan jadwal-jadwal penting seputar CPNS |
| 8 | Tryout & Review jawaban | Use case tryout ini berfungsi untuk mengumpulkan jawaban yang diberi oleh user dengan sistem tryout dan jawaban akan diperiksa sistem yang diberikan oleh user guna belajar dari kesalahan |

3. *Use case*, digunakan untuk mengetahui fungsi apa saja yang berada pada sebuah sistem dan siapa saja yang berhak menggunakan fungsi yang ada.



Gambar 4.1 Use Case Diagram

4. Skenario *use case*, adapun skenario yang menjelaskan *use case* diagram yang menjelaskan sistem informasi *tryout* soal CPNS sebagai berikut:
- Skenario *use case login*, alur *login* yang digunakan untuk mengakses aplikasi dari sisi aktor dan reaksi sistem pada tabel 4.3

Tabel 4.3 Skenario Use Case Login

| Aksi Aktor | Reaksi sistem |
|---|---|
| 1. Mengakses halaman <i>login</i> | |
| | 2. Menampilkan <i>form login</i> |
| 3. Memasukan <i>email</i> dan <i>password</i> | |
| | 4. Melakukan validasi terhadap <i>email</i> dan <i>password</i> |
| | 5. Masuk ke halaman utama sesuai <i>role</i> |

- b. Skenario *use case* registrasi, alur registrasi yang dilakukan oleh user guna mendapatkan akses aplikasi dijelaskan pada tabel 4.4

Tabel 4.4 Skenario *Use Case* Registrasi

| Aksi Aktor | Reaksi Sistem |
|-----------------------------------|---|
| 1. Mengakses halaman registrasi | |
| | 2. Menampilkan halaman registrasi |
| 3. Memasukan data registrasi user | |
| | 4. Melakukan validasi dan memasukan data user ke database |
| | 5. Masuk ke halaman utama user |

- c. Skenario *use case* pengelolaan paket soal, paket soal dikelola oleh admin dan dijelaskana pada tabel 4.6

Tabel 4.5 Skenario *Use Case* Pengelolaan Paket Soal

| Aksi Aktor | Reaksi Sistem |
|---|-----------------------------------|
| 1. Mengakses halaman pengelolaan paket soal | |
| | 2. Menampilkan halaman paket soal |

| Aksi Aktor | Reaksi Sistem |
|---|--|
| 3. Memilih tombol tambah paket dan memasukan data paket | |
| | 4. Melakukan validasi dan melakukan tambah data paket ke database |
| | 5. Kembali ke halaman paket soal dengan notifikasi berhasil ditambahkan |
| 6. Memilih edit tambah paket dan mengedit data yang telah ada | |
| | 7. Melakukan validasi dan melakukan edit data paket di database sesuai id yang dipilih |
| | 8. Kembali ke halaman paket soal dengan notifikasi berhasil diedit |
| 9. Memilih tombol hapus paket | |
| | 10. Mekonfirmasi aksi dan menghapus paket sesuai id yang dipilih dari database |
| | 11. Kembali ke halaman paket soal dengan notifikasi berhasil dihapus |

- d. Skenario *use case* pengelolaan tipe soal, tipe soal berguna untuk membedakan tipe dari setiap soal yang ditampilkan seperti yang dijelaskan pada tabel 4.6

Tabel 4.6 Skenario *Use Case* Pengelolaan Tipe Soal

| Aksi Aktor | Reaksi Sistem |
|---|--|
| 1. Mengakses halaman pengelolaan tipe soal | |
| | 2. Menampilkan halaman tipe soal |
| 3. Memilih tombol tambah tipe soal dan memasukan data tipe soal | |
| | 4. Melakukan validasi dan melakukan tambah data tipe soal ke database |
| | 5. Kembali ke halaman tipe soal dengan notifikasi berhasil ditambahkan |
| 6. Memilih edit tambah tipe soal dan mengedit data yang telah ada | |
| | 7. Melakukan validasi dan melakukan edit data tipe soal di database sesuai id yang dipilih |

| Aksi Aktor | Reaksi Sistem |
|-----------------------------------|--|
| | 8. Kembali ke halaman tipe soal dengan notifikasi berhasil diedit |
| 9. Memilih tombol hapus tipe soal | |
| | 10. Mekonfirmasi aksi dan menghapus tipe soal sesuai id yang dipilih dari database |
| | 11. Kembali ke halaman tipe soal dengan notifikasi berhasil dihapus |

- e. Skenario *use case* soal, pengelolaan soal dilakukan oleh admin dan digunakan untuk mengelola soal yang akan digunakan dlama *tryout* oleh user seperti yang dijelaskan pada tabel 4.7

Tabel 4.7 Skenario *Use Case* Pengelolaan Soal

| Aksi Aktor | Reaksi Sistem |
|--|---|
| 1. Mengakses halaman pengelolaan soal sesuai paket yang dipilih | |
| | 2. Menampilkan halaman pengelolaan soal |
| 3. Memilih tombol tambah soal dan memasukan data soal sesuai tipe soal | |

| Aksi Aktor | Reaksi Sistem |
|--|---|
| | 4. Melakukan validasi dan melakukan tambah data soal ke database |
| | 5. Kembali ke halaman pengelolaan soal dengan notifikasi berhasil ditambahkan |
| 6. Memilih edit tambah soal dan mengedit data yang telah ada | |
| | 7. Melakukan validasi dan melakukan edit data soal di database sesuai id yang dipilih |
| | 8. Kembali ke halaman pengelolaan soal dengan notifikasi berhasil diedit |
| 9. Memilih tombol hapus soal | |
| | 10. Mekonfirmasi aksi dan menghapus soal sesuai id yang dipilih dari database |
| | 11. Kembali ke halaman pengelolaan soal dengan notifikasi berhasil dihapus |

- f. Skenario *use case* pengelolaan informasi, berguna untuk menampilkan informasi seputar CPNS kepada user dan dilakukan oleh admin seperti dijelaskan pada tabel 4.8

Tabel 4.8 Skenario *Use Case* Pengelolaan Informasi

| Aksi Aktor | Reaksi Sistem |
|---|--|
| 1. Mengakses halaman pengelolaan informasi | |
| | 2. Menampilkan halaman informasi |
| 3. Memilih tombol informasi dan memasukan data informasi CPNS | |
| | 4. Melakukan validasi dan melakukan tambah data informasi ke database |
| | 5. Kembali ke halaman informasi dengan notifikasi berhasil ditambahkan |
| 6. Memilih edit tambah informasi dan mengedit data yang telah ada | |
| | 7. Melakukan validasi dan melakukan edit data informasi di database sesuai id yang dipilih |

| Aksi Aktor | Reaksi Sistem |
|-----------------------------------|--|
| | 8. Kembali ke halaman informasi dengan notifikasi berhasil diedit |
| 9. Memilih tombol hapus informasi | |
| | 10. Mekonfirmasi aksi dan menghapus informasi sesuai id yang dipilih dari database |
| | 11. Kembali ke halaman informasi dengan notifikasi berhasil dihapus |

- g. Skenario *use case* pengelolaan materi, pengelolaan materi digunakan untuk memeberikan wawasan mengenai materi CPNS seperti dijelaskan pada tabel 4.9

Tabel 4.9 Skenario *Use Case* Pengelolaan Materi

| Aksi Aktor | Reaksi Sistem |
|--|-------------------------------|
| 1. Mengakses halaman pengelolaan materi | |
| | 2. Menampilkan halaman materi |
| 3. Memilih tombol materi dan memasukan data materi | |

| Aksi Aktor | Reaksi Sistem |
|--|---|
| | 4. Melakukan validasi dan melakukan tambah data materi ke database |
| | 5. Kembali ke halaman materi dengan notifikasi berhasil ditambahkan |
| 6. Memilih edit tambah materi dan mengedit data yang telah ada | |
| | 7. Melakukan validasi dan melakukan edit data materi di database sesuai id yang dipilih |
| | 8. Kembali ke halaman materi dengan notifikasi berhasil diedit |
| 9. Memilih tombol hapus materi | |
| | 10. Mekonfirmasi aksi dan menghapus materi sesuai id yang dipilih dari database |
| | 11. Kembali ke halaman informasi dengan notifikasi berhasil dihapus |

- h. Skenario *use case tryout*, user mengerjakan soal sesuai paket yang dipilih user sesuai tabel 4.10

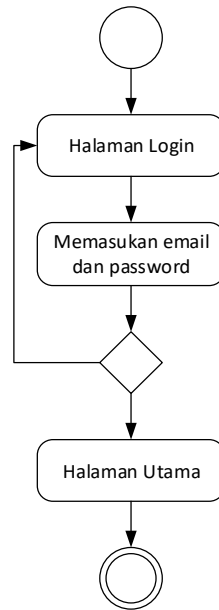
Tabel 4.10 Skenario *Use Case Tryout & Review Jawaban*

| Aksi Aktor | Reaksi Sistem |
|---|--|
| 1. Mengakses halaman <i>tryout</i> dan memilih paket soal | |
| | 2. Menampilkan halaman <i>tryout</i> dan memulai <i>tryout</i> |
| 3. Mengerjakan soal dan menekan tombol selesai | |
| | 4. Memasukan data jawaban user ke database |
| | 5. Masuk ke halaman <i>review</i> soal |
| 6. Menampilkan <i>review</i> pengerjaan user yang sudah diperiksa | |

4.1.2.2 Perancangan *Activity Diagram*

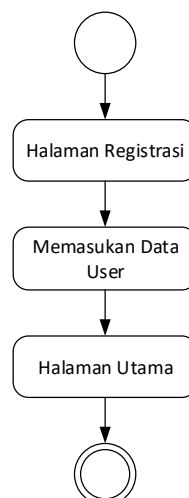
Digunakan sebagai pemodelan sebuah sistem dari *use case* yang telah dibuat. Adapun *Activity diagram* yang telah dirancang adalah sebagai berikut:

1. *Activity diagram login*, adapun aliran kerja dari aktivitas *login* dapat dilihat pada gambar 4.2:



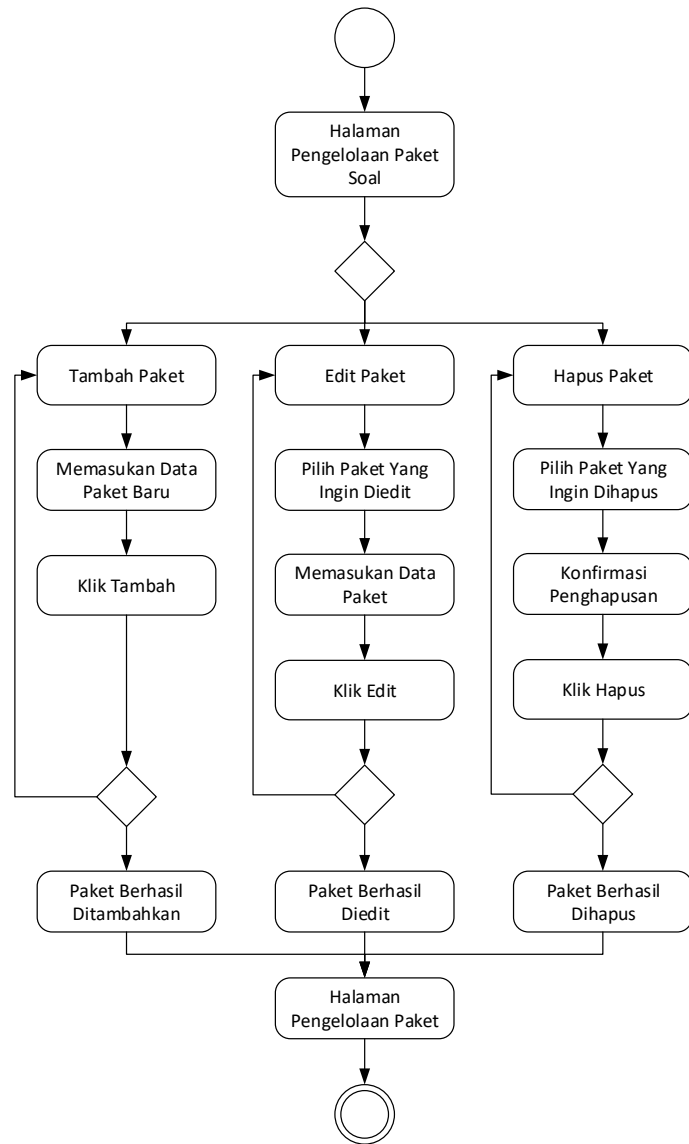
Gambar 4.2 *Activity Diagram Login*

2. *Activity diagram registrasi*, adapun aliran kerja dari aktivitas registrasi dapat dilihat pada Gambar 4.3.



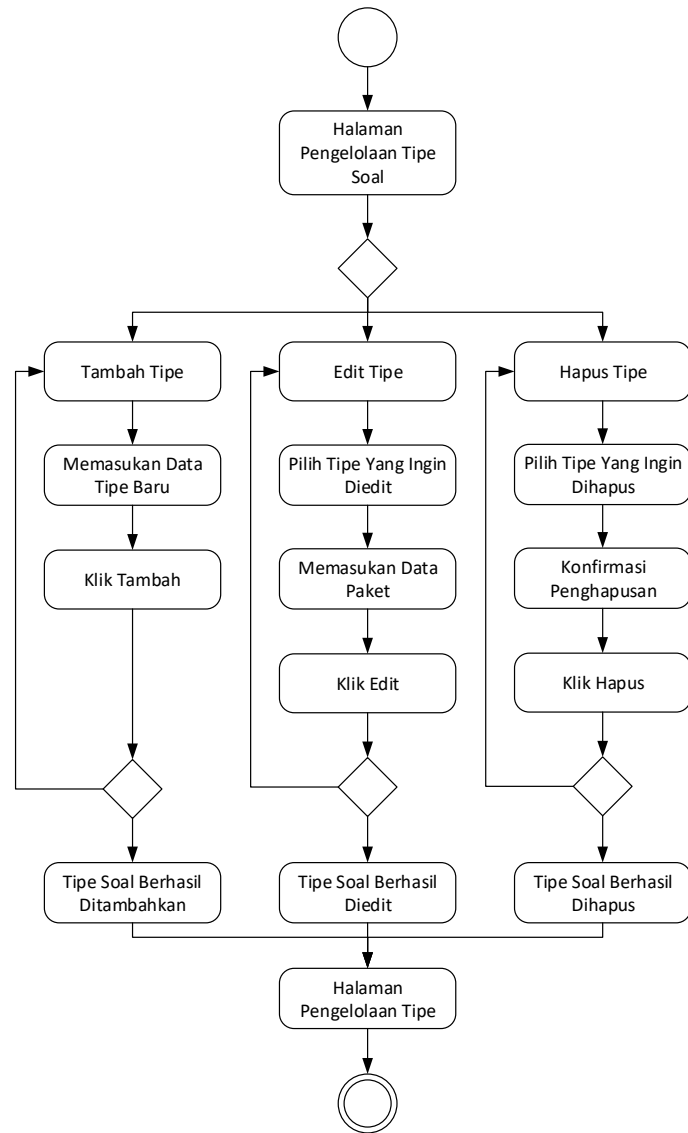
Gambar 4.3 *Activity Diagram Registrasi*

3. *Activity* diagram pengelolaan paket, adapun aliran kerja dari aktivitas pengelolaan paket soal dapat dilihat pada gambar 4.4



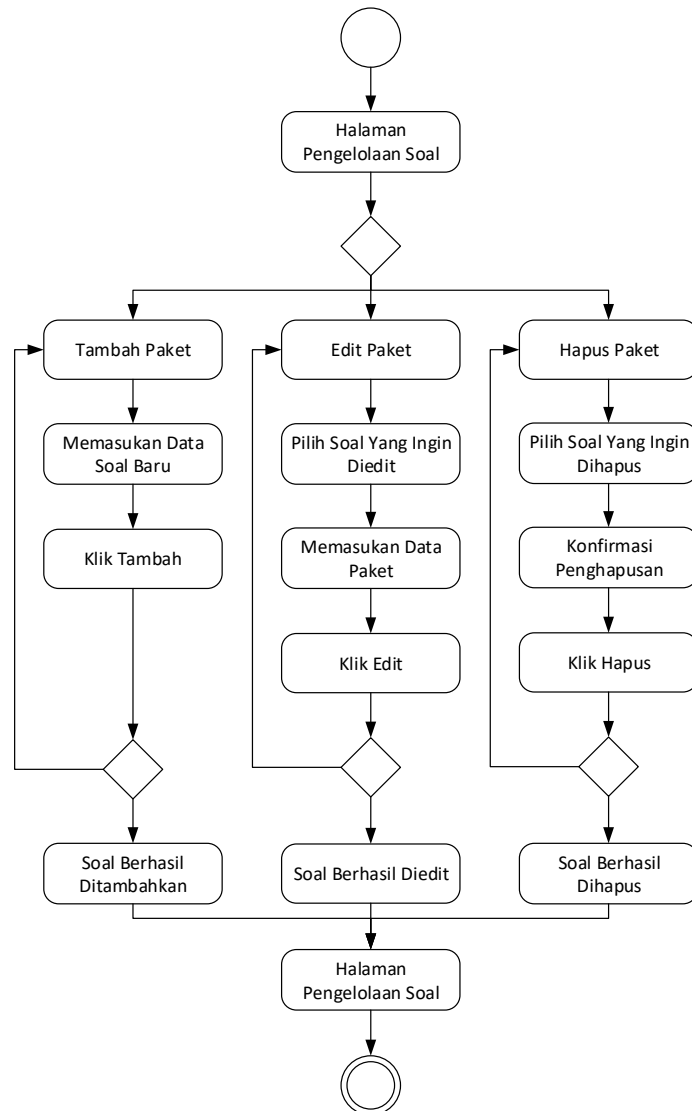
Gambar 4.4 Activity Diagram Pengelolaan Paket

4. *Activity* diagram pengelolaan tipe soal, adapun aliran kerja dari aktivitas pengelolaan tipe soal dapat dilihat pada gambar 4.5



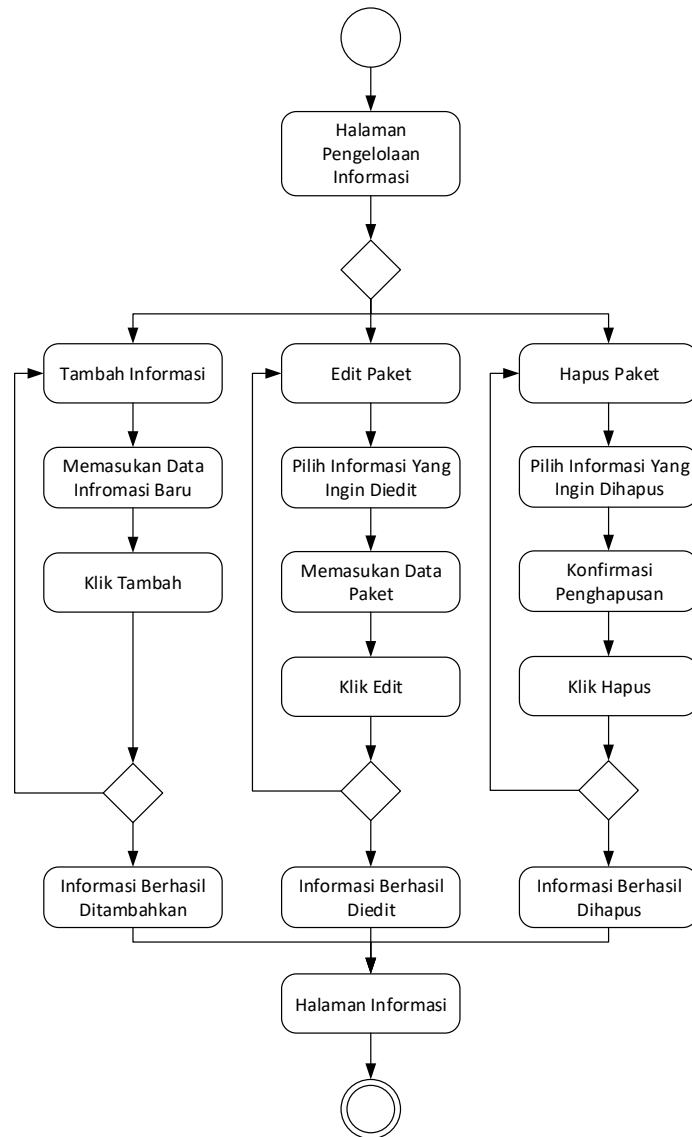
Gambar 4.5 Activity Diagram Pengelolaan Paket Tipe Soal

5. *Activity* diagram pengelolaan soal, adapun aliran kerja dari aktivitas pengelolaan soal dapat dilihat pada gambar 4.6



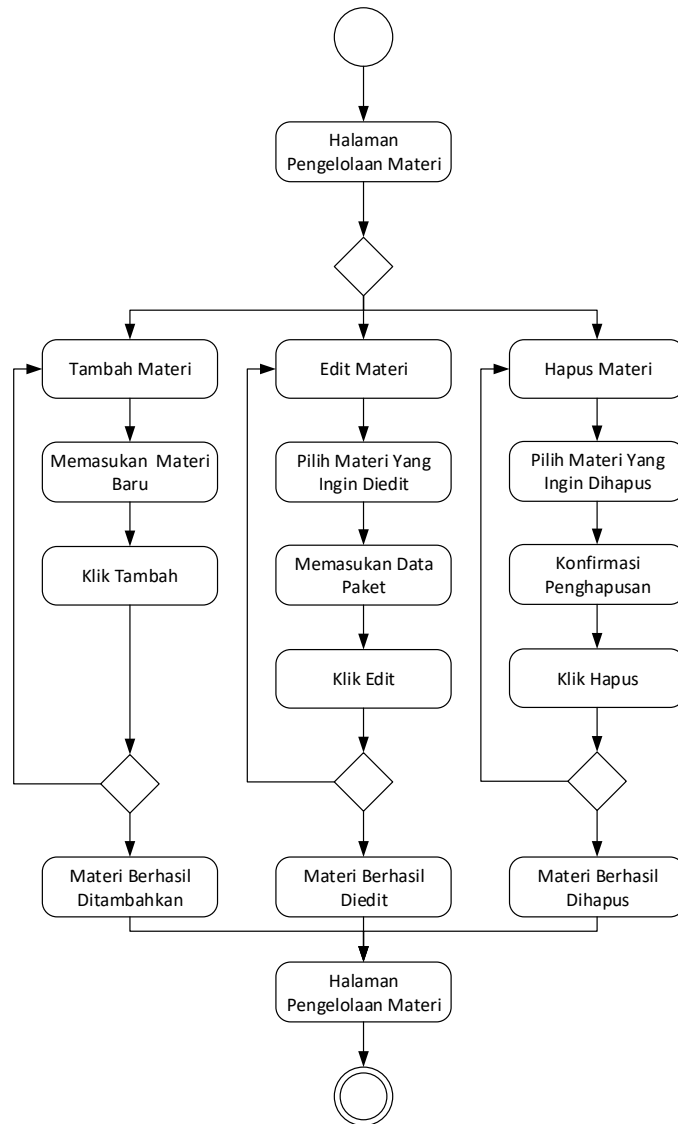
Gambar 4.6 Activity Diagram Pengelolaan Soal

6. *Activity* diagram pengelolaan soal, adapun aliran kerja dari aktivitas pengelolaan informasi dapat dilihat pada gambar 4.7



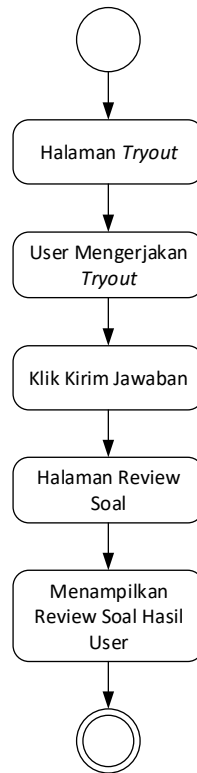
Gambar 4.7 Activity Diagram Pengelolaan Informasi

7. *Activity* diagram pengelolaan materi, adapun aliran kerja dari aktivitas pengelolaan Materi dapat dilihat pada gambar 4.8



Gambar 4.8 Activity Diagram Pengelolaan Materi

8. *Activity* diagram pengelolaan *tryout* dan *review* jawaban, adapun aliran kerja dari aktivitas *tryout* dan *review* jawaban dapat dilihat pada gambar 4.9

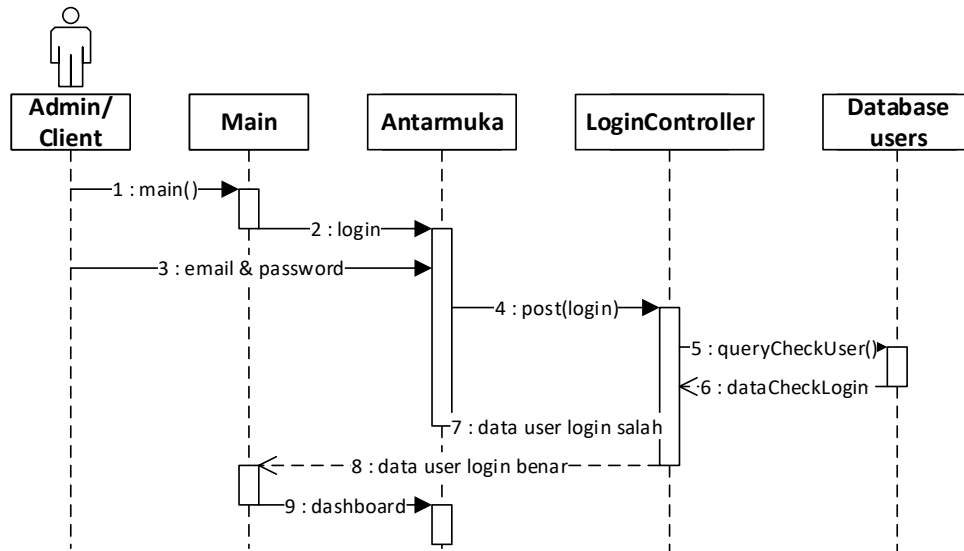


Gambar 4.9 Activity Diagram Tryout & Review Jawaban

4.1.2.3 Perancangan *Sequence Diagram*

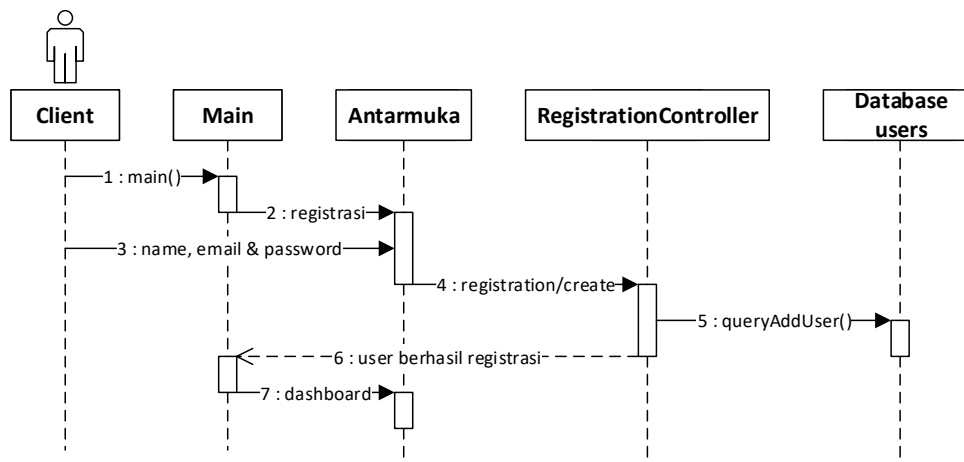
Sequence diagram menggambarkan *login* aktor dengan sistem serta menjelaskan sistem itu sendiri berjalan. Adapun *sequence diagram* yang telah dirancang adalah sebagai berikut :

1. *Sequence diagram login*, adapun *login* objek dari aktivitas *login* dapat dilihat pada gambar 4.10



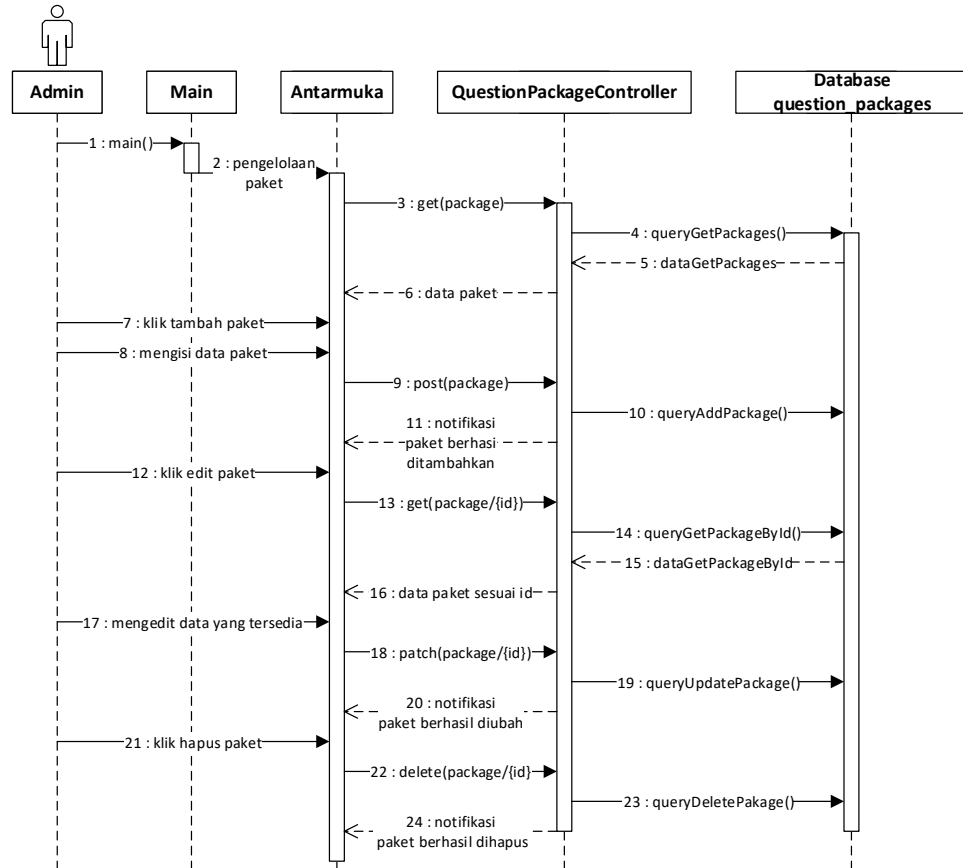
Gambar 4.10 *Sequence diagram login*

2. *Sequence diagram registrasi client*, adapun *login* objek dari aktivitas registrasi yang dilakukan oleh *client* guna mendapatkan hak akses aplikasi dapat dilihat pada gambar 4.11



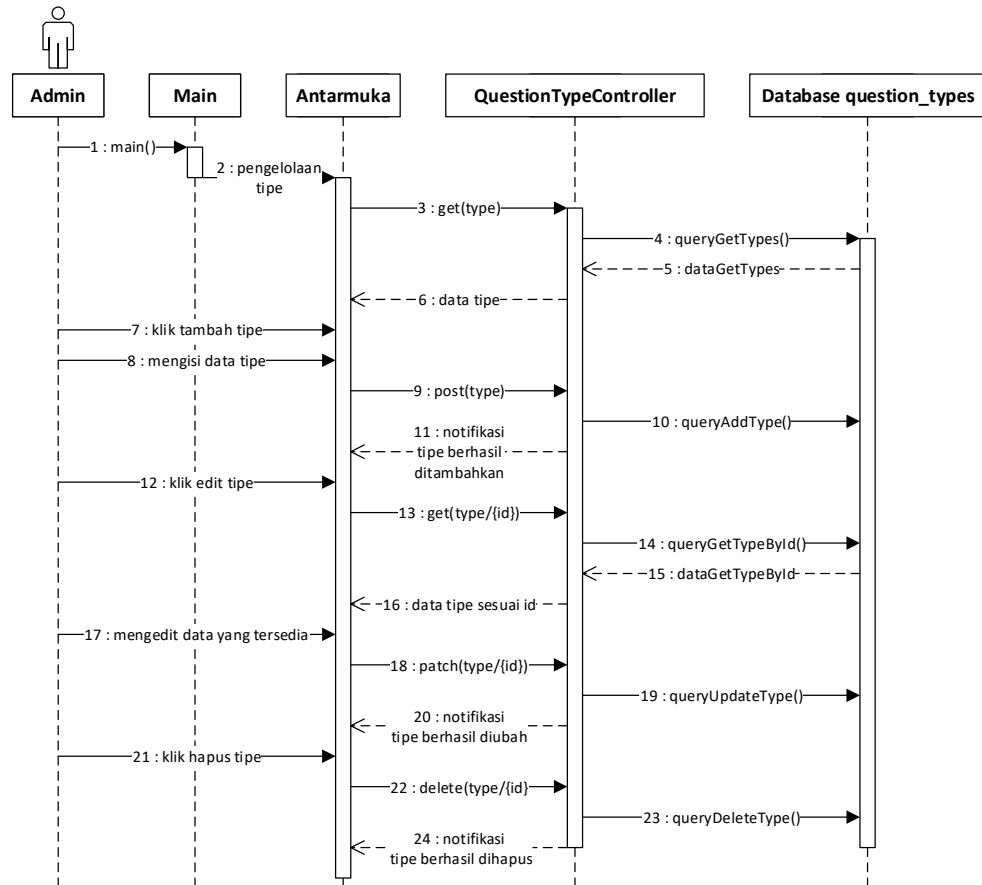
Gambar 4.11 *Sequence diagram registrasi*

3. *Sequence* diagram pengelolaan paket, menjelaskan *login* objek dari aktivitas pengelolaan paket yang berfungsi untuk mengelola paket soal yang dapat dilihat pada gambar 4.12



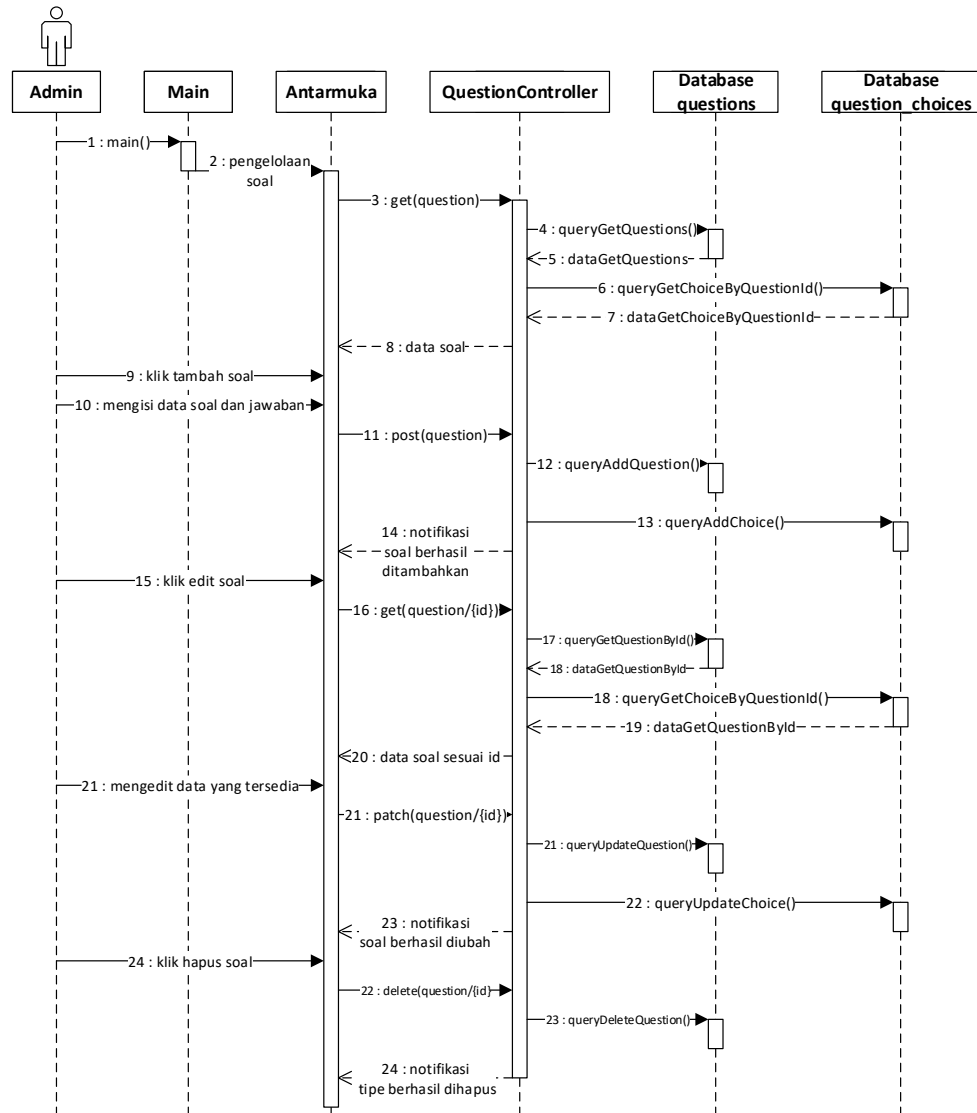
Gambar 4.12 Sequence diagram pengelolaan tipe

4. Sequence diagram pengelolaan tipe menjelaskan bagaimana *login* dari objek yang berguna untuk mengelola tipe soal, detail sequence diagram pengelolaan tipe dijelaskan pada gambar 4.13



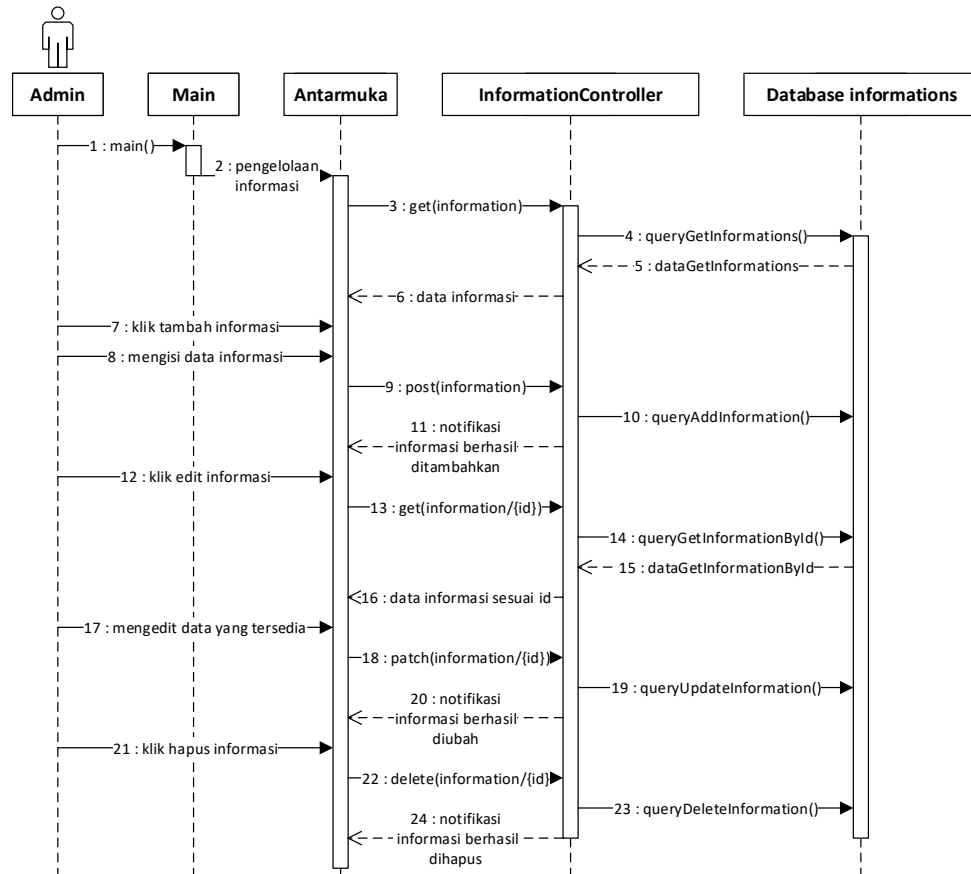
Gambar 4.13 Sequence Diagram Pengelolaan Paket

5. *Sequence* diagram pengelolaan soal menjelaskan *login* dari objek guna mengelola soal, detail *sequence* diagram pnglolaan soal dapat dilihat pada gambar 4.14



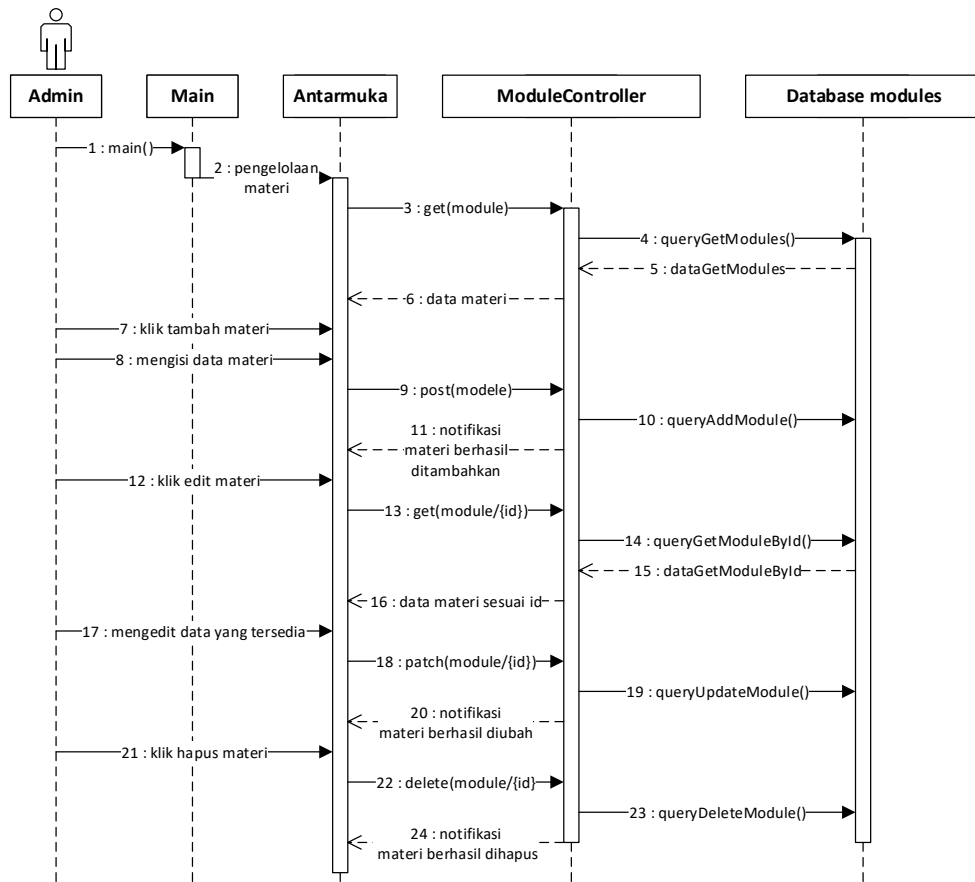
Gambar 4.14 *Sequence* Diagram Pengelolaan Soal

6. *Sequence* diagram pengelolaan informasi menjelaskan *login* antara objek yang mengelola informasi yang dapat diakses oleh user. Detail *sequence* diagram pengelolaan informasi dapat dilihat pada gambar 4.15



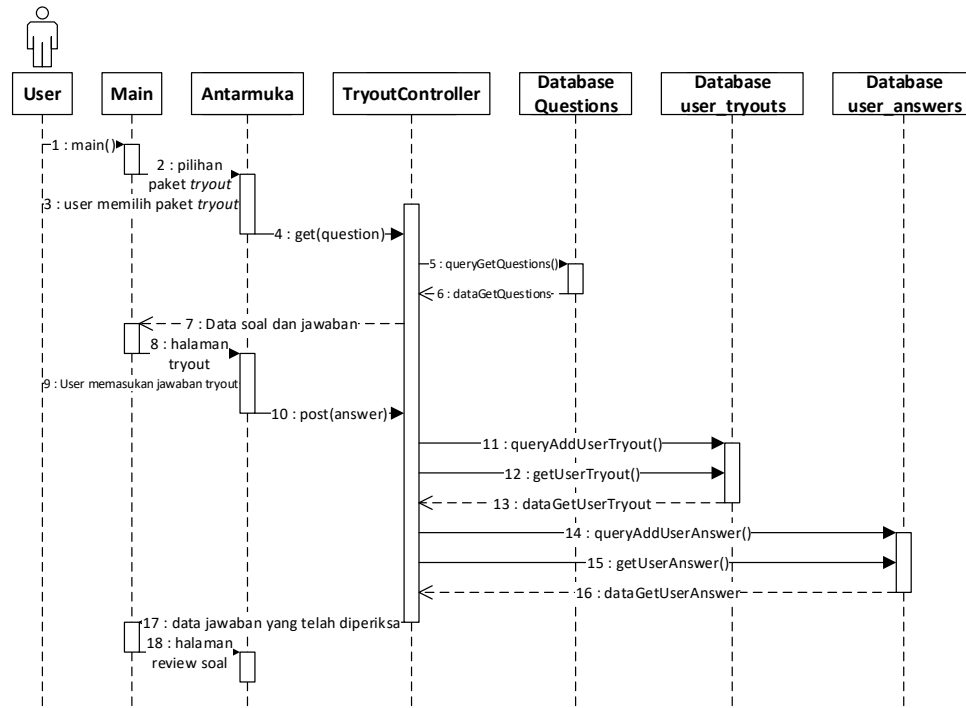
Gambar 4.15 Sequence Diagram Pengelolaan Informasi

7. *Sequence* diagram pengelolaan materi merupakan representasi dari *login* antara objek yang mengelola materi yang dapat dilihat pada gambar 4.16



Gambar 4.16 *Sequence* Diagram Pengelolaan Materi

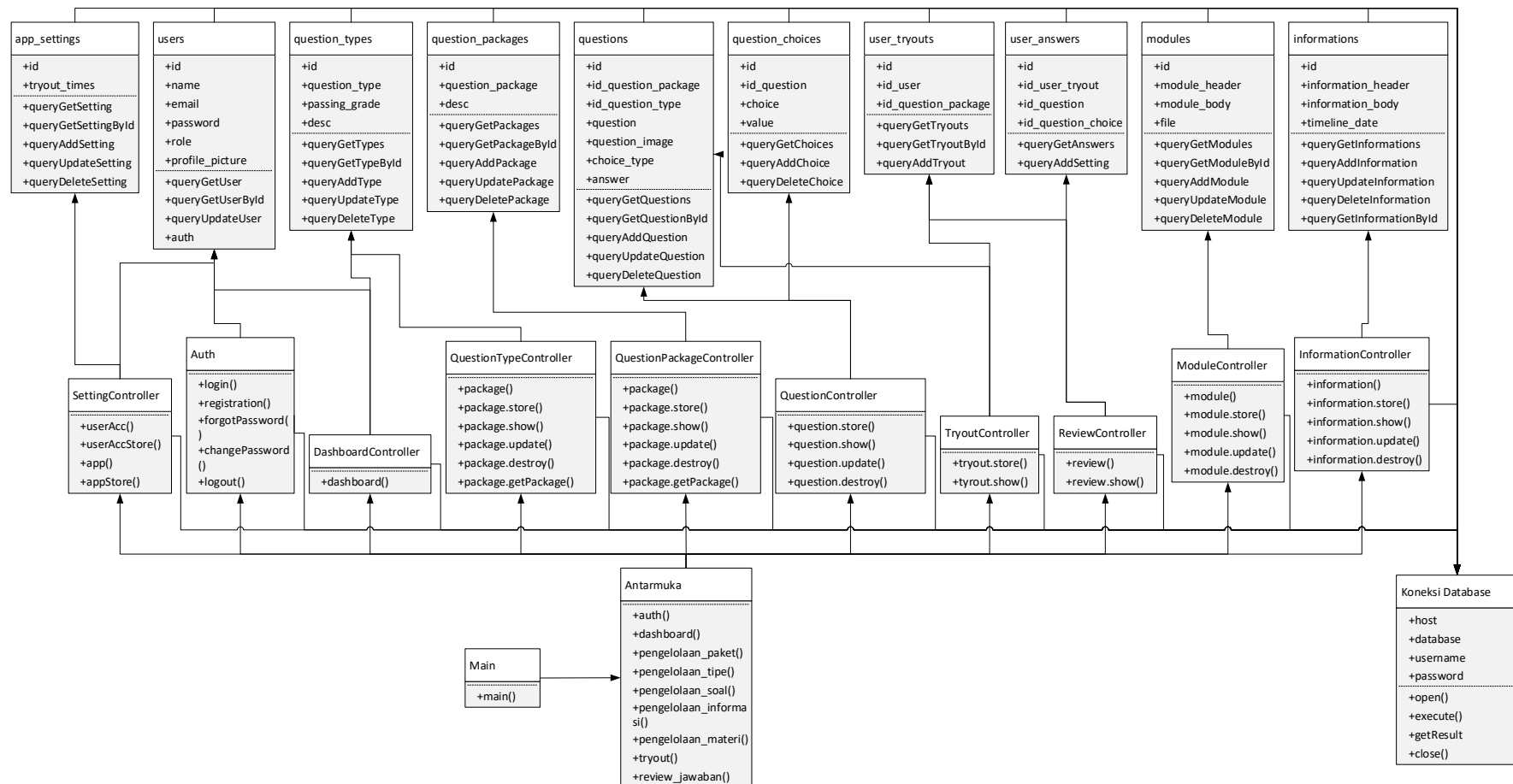
8. *Sequence diagram tryout dan review jawaban* menjelaskan *login* antara objek yang mengatur tryout dan review jawaban dan dapat dilihat lebih detail pada gambar 4.17



Gambar 4.17 Sequence Diagram Tryout dan Review Jawaban

4.1.2.4 Perancangan Class Diagram

Class diagram sering disebut juga dengan desain pemrograman karena merepresentasikan struktur dari program itu sendiri. Berikut ini merupakan *class* diagram dari sistem informasi *tryout* untuk soal CPNS pada gambar 4.18

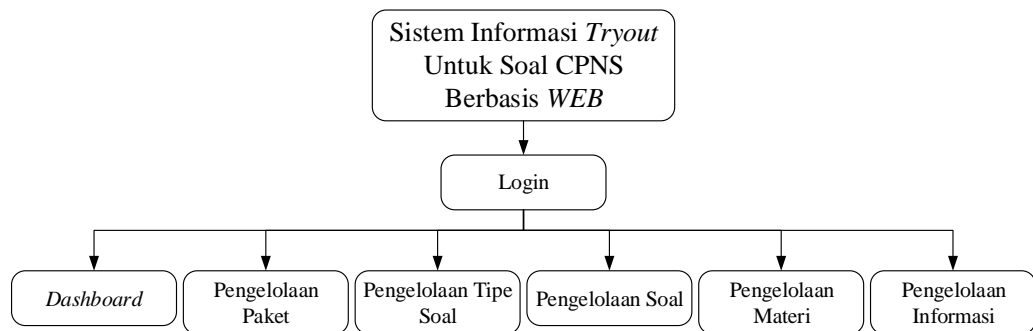


Gambar 4.18 Class Diagram

4.1.2.5 Perancangan Tampilan Struktur Menu

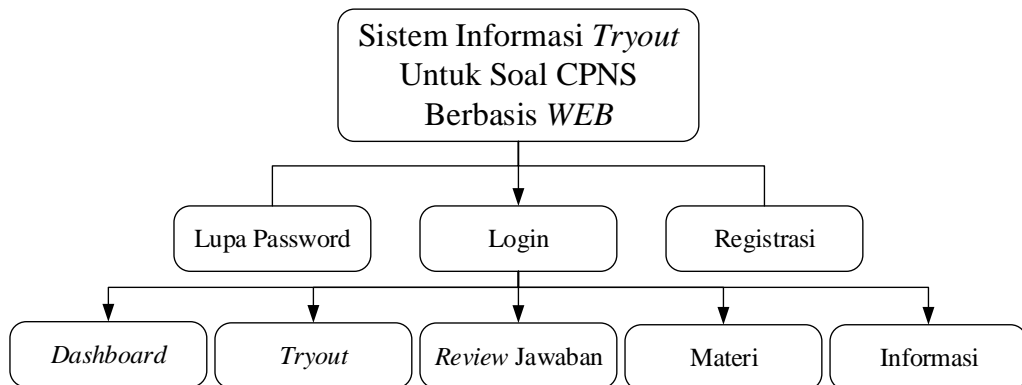
Struktur menu berfungsi untuk menggambarkan sistem dapat diakses berdasarkan tingkatan menu serta *role* yang tersedia. Adapun tampilan struktur menu yang dirancang sebagai berikut :

1. Struktur menu admin, merupakan rancangan dari tampilan untuk admin. Adapun tampilan struktur menu untuk admin dapat dilihat pada gambar 4.19



Gambar 4.19 Struktur Menu Admin

2. Struktur menu user, merupakan rancangan dari tampilan untuk user. Adapun tampilan struktur menu untuk user dapat dilihat pada gambar 4.20



Gambar 4.20 Struktur Menu User

4.1.2.6 Perancangan *Interface*

Perancangan *interface* adalah perancangan gambaran dari tampilan secara garis besar yang akan dibuat. Adapun tampilan *interface* yang telah dirancang sebagai berikut :

1. *Interface login* merupakan perancangan dari tampilan *login* yang terdapat form dan fitur yang akan dibuat. Berikut rancangan *interface login* yang dapat dilihat pada Gambar 4.21

| Login | |
|--------------------------------------|--------------------------|
| Email | <input type="text"/> |
| Password | <input type="password"/> |
| <input type="checkbox"/> Remember me | Lupa Password? |
| <input type="button" value="Login"/> | |

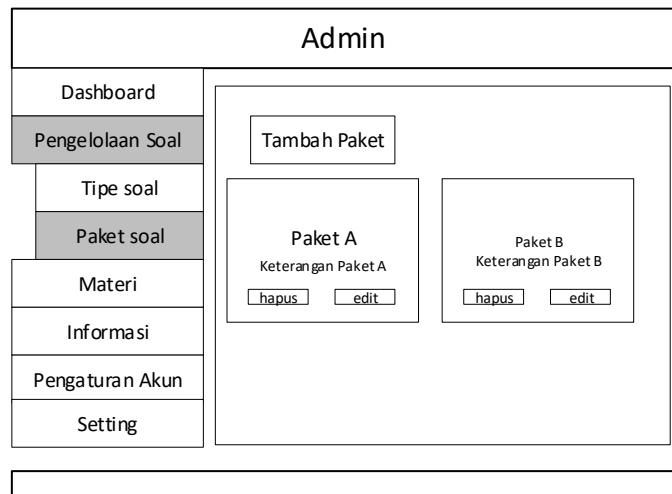
Gambar 4.21 *Interface Login*

2. *Interface registrasi* merupakan rancangan dari tampilan registrasi yang dapat dilihat pada gambar 4.22

| Registrasi | |
|---|--------------------------|
| Nama | <input type="text"/> |
| Email | <input type="text"/> |
| Password | <input type="password"/> |
| <input type="button" value="Registrasi"/> | |

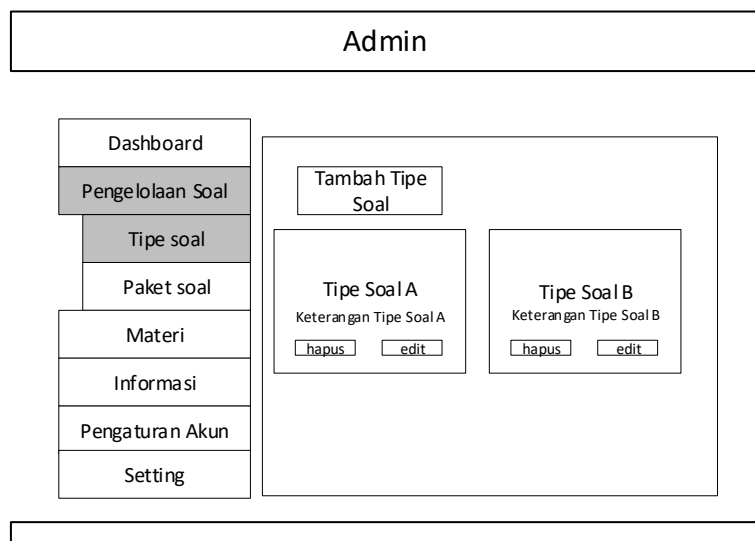
Gambar 4.22 *Interface Registrasi*

3. *Interface* pengelolaan paket merupakan perancangan tampilan dari pengelolaan paket soal seperti pada gambar 4.23



Gambar 4.23 Interface Pengelolaan Paket Soal

4. *Interface* pengelolaan tipe soal merupakan gambaran dari pengelolaan tipe soal yang membedakan soal satu dengan yang lainnya, seperti yang digambarkan 4.24



Gambar 4.24 Interface Pengelolaan Tipe

5. *Interface* pengelolaan informasi merupakan gambaran tampilan yang akan dibuat untuk mengelola informasi seperti pada gambar 4.25

| | | | | | | | | | | | | | | | | | | | | | |
|---|--|------------------|-----------|------------|--------|-----------|-----------------|---------|---|------------------|--|-------------|--|-------------|--|-------------|--|-------------|--|-------------|--|
| Admin | | | | | | | | | | | | | | | | | | | | | |
| <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Dashboard</td></tr> <tr><td>Pengelolaan Soal</td></tr> <tr><td> Tipe soal</td></tr> <tr><td> Paket soal</td></tr> <tr><td> Materi</td></tr> <tr style="background-color: #e0e0e0;"><td>Informasi</td></tr> <tr><td> Pengaturan Akun</td></tr> <tr><td> Setting</td></tr> </table> | Dashboard | Pengelolaan Soal | Tipe soal | Paket soal | Materi | Informasi | Pengaturan Akun | Setting | <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: center; padding: 5px;">Tambah Informasi</td> </tr> <tr> <td style="padding: 5px;">Informasi A</td> <td style="text-align: right; padding: 5px;"> <input type="button" value="hapus"/> <input type="button" value="edit"/> </td> </tr> <tr> <td style="padding: 5px;">Informasi B</td> <td style="text-align: right; padding: 5px;"> <input type="button" value="hapus"/> <input type="button" value="edit"/> </td> </tr> <tr> <td style="padding: 5px;">Informasi C</td> <td style="text-align: right; padding: 5px;"> <input type="button" value="hapus"/> <input type="button" value="edit"/> </td> </tr> <tr> <td style="padding: 5px;">Informasi D</td> <td style="text-align: right; padding: 5px;"> <input type="button" value="hapus"/> <input type="button" value="edit"/> </td> </tr> <tr> <td style="padding: 5px;">Informasi E</td> <td style="text-align: right; padding: 5px;"> <input type="button" value="hapus"/> <input type="button" value="edit"/> </td> </tr> </table> | Tambah Informasi | | Informasi A | <input type="button" value="hapus"/> <input type="button" value="edit"/> | Informasi B | <input type="button" value="hapus"/> <input type="button" value="edit"/> | Informasi C | <input type="button" value="hapus"/> <input type="button" value="edit"/> | Informasi D | <input type="button" value="hapus"/> <input type="button" value="edit"/> | Informasi E | <input type="button" value="hapus"/> <input type="button" value="edit"/> |
| Dashboard | | | | | | | | | | | | | | | | | | | | | |
| Pengelolaan Soal | | | | | | | | | | | | | | | | | | | | | |
| Tipe soal | | | | | | | | | | | | | | | | | | | | | |
| Paket soal | | | | | | | | | | | | | | | | | | | | | |
| Materi | | | | | | | | | | | | | | | | | | | | | |
| Informasi | | | | | | | | | | | | | | | | | | | | | |
| Pengaturan Akun | | | | | | | | | | | | | | | | | | | | | |
| Setting | | | | | | | | | | | | | | | | | | | | | |
| Tambah Informasi | | | | | | | | | | | | | | | | | | | | | |
| Informasi A | <input type="button" value="hapus"/> <input type="button" value="edit"/> | | | | | | | | | | | | | | | | | | | | |
| Informasi B | <input type="button" value="hapus"/> <input type="button" value="edit"/> | | | | | | | | | | | | | | | | | | | | |
| Informasi C | <input type="button" value="hapus"/> <input type="button" value="edit"/> | | | | | | | | | | | | | | | | | | | | |
| Informasi D | <input type="button" value="hapus"/> <input type="button" value="edit"/> | | | | | | | | | | | | | | | | | | | | |
| Informasi E | <input type="button" value="hapus"/> <input type="button" value="edit"/> | | | | | | | | | | | | | | | | | | | | |

Gambar 4.25 *Interface* Pengelolaan Informasi

6. *Interface* pengelolaan materi adalah perancangan dari tampilan pengelolaan materi, seperti yang telah dirancang pada gambar 2.6

| | | | | | | | | | | | | | | | | | | | | | |
|---|--|------------------|-----------|------------|--------|-----------|-----------------|---------|---|---------------|--|----------|--|----------|--|----------|--|----------|--|----------|--|
| Admin | | | | | | | | | | | | | | | | | | | | | |
| <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Dashboard</td></tr> <tr><td>Pengelolaan Soal</td></tr> <tr><td> Tipe soal</td></tr> <tr><td> Paket soal</td></tr> <tr style="background-color: #e0e0e0;"><td>Materi</td></tr> <tr><td> Informasi</td></tr> <tr><td> Pengaturan Akun</td></tr> <tr><td> Setting</td></tr> </table> | Dashboard | Pengelolaan Soal | Tipe soal | Paket soal | Materi | Informasi | Pengaturan Akun | Setting | <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: center; padding: 5px;">Tambah Materi</td> </tr> <tr> <td style="padding: 5px;">Materi A</td> <td style="text-align: right; padding: 5px;"> <input type="button" value="hapus"/> <input type="button" value="edit"/> </td> </tr> <tr> <td style="padding: 5px;">Materi B</td> <td style="text-align: right; padding: 5px;"> <input type="button" value="hapus"/> <input type="button" value="edit"/> </td> </tr> <tr> <td style="padding: 5px;">Materi C</td> <td style="text-align: right; padding: 5px;"> <input type="button" value="hapus"/> <input type="button" value="edit"/> </td> </tr> <tr> <td style="padding: 5px;">Materi D</td> <td style="text-align: right; padding: 5px;"> <input type="button" value="hapus"/> <input type="button" value="edit"/> </td> </tr> <tr> <td style="padding: 5px;">Materi E</td> <td style="text-align: right; padding: 5px;"> <input type="button" value="hapus"/> <input type="button" value="edit"/> </td> </tr> </table> | Tambah Materi | | Materi A | <input type="button" value="hapus"/> <input type="button" value="edit"/> | Materi B | <input type="button" value="hapus"/> <input type="button" value="edit"/> | Materi C | <input type="button" value="hapus"/> <input type="button" value="edit"/> | Materi D | <input type="button" value="hapus"/> <input type="button" value="edit"/> | Materi E | <input type="button" value="hapus"/> <input type="button" value="edit"/> |
| Dashboard | | | | | | | | | | | | | | | | | | | | | |
| Pengelolaan Soal | | | | | | | | | | | | | | | | | | | | | |
| Tipe soal | | | | | | | | | | | | | | | | | | | | | |
| Paket soal | | | | | | | | | | | | | | | | | | | | | |
| Materi | | | | | | | | | | | | | | | | | | | | | |
| Informasi | | | | | | | | | | | | | | | | | | | | | |
| Pengaturan Akun | | | | | | | | | | | | | | | | | | | | | |
| Setting | | | | | | | | | | | | | | | | | | | | | |
| Tambah Materi | | | | | | | | | | | | | | | | | | | | | |
| Materi A | <input type="button" value="hapus"/> <input type="button" value="edit"/> | | | | | | | | | | | | | | | | | | | | |
| Materi B | <input type="button" value="hapus"/> <input type="button" value="edit"/> | | | | | | | | | | | | | | | | | | | | |
| Materi C | <input type="button" value="hapus"/> <input type="button" value="edit"/> | | | | | | | | | | | | | | | | | | | | |
| Materi D | <input type="button" value="hapus"/> <input type="button" value="edit"/> | | | | | | | | | | | | | | | | | | | | |
| Materi E | <input type="button" value="hapus"/> <input type="button" value="edit"/> | | | | | | | | | | | | | | | | | | | | |

Gambar 4.26 *Interface* Pengelolaan Materi

7. *Interface tryout* adalah gambar dari tampilan *tryout* yang akan dikerjakan oleh user, seperti yang telah dirancang pada gambar 4.27

User

Dashboard

Tryout

Review Jawaban

Materi

Informasi

Pengaturan Akun

Pertanyaan 1+1 =

| | | |
|-----|--|-----|
| a 1 | | c 3 |
| b 2 | | d 4 |
| e 5 | | |

Planet terbesar

| | | |
|-------------|--|--------|
| a merkurius | | c bumi |
| b venus | | d mars |
| e jupiter | | |

Gambar 4.27 *Interface Tryout*

DAFTAR PUSTAKA

- Aisah, N., & Haryati, T. (2019). SISTEM TRY OUT ONLINE DAN MEDIA PEMBELAJARAN PERSIAPAN UJIAN NASIONAL STUDI KASUS SMKN 2 BANDUNG. *jurnal komputa*.
- Aminudin. (2015). *Cara Efektif Belajar Framework Laravel*. (L. Hakim, Penyunt.) Yogyakarta: Lokomedia.
- Andi Maddeppungeng, I. S. (2015). Analisis Pengendalian Penjadwalan Pembangunan Gedung Administrasi Universitas Pendidikan Indonesia (UPI) Kampus Serang Menggunakan Meotde Work Breakdown Structure (WBS) dan Kurva-S. *Jurnal Fondasi*.
- Antonio, H., & Safriadi, N. (2012). Rancang Bangun Sistem Informasi Administrasi Informatika. *ELKHA*.
- Azizah, S. H., & Fitriani, L. (2016). RANCANG BANGUN APLIKASI BANK SOAL PROGRAM STUDI TEKNIK INFORMATIKA SEKOLAH TINGGI TEKNOLOGI GARUT. *jurnal algoritma*.
- Buchari, M. Z., Sentinuwo, S. R., & Lantang, O. A. (2015). Rancang Bangun Video Animasi 3 Dimensi Untuk Mekanisme Pengujian Kendaraan Bermotor di Dinas Perhubungan, Kebudayaan, Pariwisata, Komunikasi dan Informasi. *E-journal Teknik Informatika*, 1.
- CodeIgniter*. (2020). Diambil kembali dari CodeIgniter: <http://www.codeigniter.com>
- Dewi, R., Emida, & Rasyid, Y. (2018). Pengaruh Discovery Learning Model Berbantuan Media Objek Langsung Terhadap Keterampilan Menulis Teks Laporan Observasi. *Jurnal Pendidikan Bahasa dan Sastra Indonesia*, 169.
- Ekaputri, M. K., Suryatiningsih, & Siswanto, B. (2016). APLIKASI LEARNING MANAJEMEN SISTEM DAN ULANGAN ONLINE. *ilmu terapan*.
- Enterprise, J. (2016). *Pemrograman Bootstrap Untuk Pemula*. Jakarta: PT Elex Media Komputindo.
- Faiz, M. N., Umar, R., & Yudhana, A. (2017). Implementasi Live Forensics untuk Perbandingan Browser. *JISKA*, 109.
- Hidayat, R., Marlina, S., & Utami, L. D. (2017). Perancangan Sistem Informasi Penjualan Barang Handmade Berbasis Website Dengan Metode Waterfall. *Simposium Nasional Ilmu Pengetahuan dan Teknologi*, 177.

- Irawan, A., & Setiyorini, N. K. (2017). Rancang Bangun Aplikasi helpdesk Dengan Menggunakan Pendekatan Knowledge Management System Pada Seksi Teknisi. *Jurnal ProTekInfo*, 5.
- Irza, I. F., Zulhendra, & Efrizon. (2017). Analisis Perbandingan Kinerja Web Server Apache dan Nginx Menggunakan Httpperf. *Jurnal Vokasional Teknik Elektronika & Informatika*, 76.
- Kharistiani, E., & Aribowo, E. (2013). Sistem Informasi Geografis Pemetaan Potensi SMA/SMK Berbasis Web. *Jurnal Sarjana Teknik Informatika*, 716.
- Kumalasari, E., Erika, & Mustikasari, D. (2017). Aplikasi Tryout Ujian Online untuk SMA/SMK Sederajat (Online Examination Tryout Application for Senior and Vocational High. *jurnal edukasi*.
- Kurniadi, D., & Islami, A. F. (2018). Aplikasi Survei Kepuasan Mahasiswa berbasis kuesioner online untuk. *Perancangan Aplikasi Survei Kepuasan Mahasiswa Berbasis Kuesioner Online*.
- Lalompoh, H. (2018). Pengembangan Aplikasi Pelaporan Kuliah Kerja Terpadu Universitas Sam Ratulangi. *Jurnal Teknik Informatika*, 2.
- mastekno. (2017, desember 7). *mastekno*. Dipetik november 2, 2018, dari mastekno.com: <https://www.mastekno.com/id/pengertian-fungsi-command-prompt-cmd-lengkap/>
- Pressman, R. (2012). *Rekayasa Perangkat Lunak Pendekatan Praktisi Edisi 7*. Yogyakarta: ANDI.
- Pressman, R. S. (2002). *Rekayasa Perangkat Lunak (buku 1)*. yogyakarta: Andi.
- Putratama, S. V. (2016). *Pemrograman Web dengan Menggunakan PHP dan Framework Codeigniter*. Yogyakarta: deepublish.
- Rerung, R. R. (2018). *Pemrograman Web Dasar*. Yogyakarta: CV Budi Utama.
- Riyadi, A., Hermaliani, E. H., & Utami, D. Y. (2019). PEMBUATAN APLIKASI SISTEM UJIAN ONLINE PADA SMK GARUDA NUSANTARA BEKASI. *junal ilmiah sinus*.
- Sukamto, R. A., & Shalahuddin, M. (2018). *Rekayasa Perangkat Lunak Terstruktur Dan Berorientasi Objek*. Bandung: Informatika Bandung.
- Sunyoto, A. (2017). *AJAX Membangun Web dengan Teknologi Asynchrone JavaScript dan XML*. Yogyakarta: Andi.

- Tigris. (2019). *Tigris*. Dipetik januari 23, 2019, dari Tigris.org:
<http://argouml.tigris.org/>
- VSCode. (2020). *Visual Studio Code Docs*. Diambil kembali dari Visual Studio Code: <http://code.visualstudio.com>
- Warsito, A. B., Yusup, M., & Yulianto. (2014). Kajian Yii Framework Dalam Pengembangan. *JournalL CCIT*, 439.
- Yuhefizar, Mooduto, H., & Hidayat, R. (2009). *Cara mudah membangun website interaktif menggunakan content management system joomla (CMS)*. Jakarta: Elex Media Komputindo.