

# ANALISIS DAN PERANCANGAN SISTEM EVALUASI HASIL BELAJAR UNTUK PERGURUAN TINGGI

**Berliyanto**

*Program Studi Teknik Informatika, FTI, Institut Teknologi Budi Utomo Jakarta  
berli@itbu.ac.id*

## **Abstract**

*Learning evaluation or examination is an important activity in higher education. Since the Covid-19 pandemic, all universities in Indonesia have been forced to implement online learning (e-Learning) including the examination activity. In information system context, online exams require more resources than routine teaching and learning activities. It could be a problem especially if the available resources are limited. The proposed solution for this problem is to separate the system for online exams from the learning management system (LMS). However, the process of acquiring a new system, either self-developed or purchasing an existing system, requires a requirements analysis process. Without it, the system tends to fail to meet the institutions needs. The focus of this research is to analyze the requirements for an online learning evaluation system in higher education. There are three research questions, namely: (1) What are the requirements for system? (2) What is the requirement model?, and (3) To what extent can the requirement model be developed into learning evaluation system design? The methodology used follows the stages of the system development life cycle (SDLC), with an emphasis on the analysis and design stages. The research steps include identifying system requirements, modeling the requirements obtained, and designing the system based on those requirements models. The results of this study are the identification of 3 business requirements, 13 user requirements, and 20 system requirements. Behavioral and structural models are then made using swimlane diagrams and entity relationship diagrams. At the end of the research, interface and system architecture designs are made in accordance with these requirement models. A high fidelity prototype of the system is also developed.*

**Keywords:** *system analysis, system design, online evaluation system, requirement analysis*

## **1. PENDAHULUAN**

Pandemi covid-19 memiliki dampak yang besar bagi dunia pendidikan. Di Indonesia, setiap perguruan tinggi diwajibkan untuk melakukan aktivitas pembelajaran secara daring (*e-learning*) (Mendikbud, 2020). Tujuannya adalah untuk mengusahakan proses belajar-mengajar tidak terhenti selama pandemi.

Proses pembelajaran yang dilakukan setiap semester memiliki serangkaian aktivitas, salah satunya adalah evaluasi hasil belajar. Umumnya, evaluasi hasil belajar dilakukan di tengah dan akhir semester. Dua bentuk evaluasi hasil belajar ada umum dilakukan adalah soal pilihan ganda dan soal esai. Dua bentuk soal ujian seperti itu masih umum ditemukan pada perguruan tinggi yang ada di Indonesia.

Dalam konteks e-learning, kegiatan evaluasi hasil belajar membutuhkan sumber daya yang lebih besar dibandingkan dengan proses belajar sehari-hari. Evaluasi hasil belajar secara daring mengharuskan mahasiswa untuk mengakses *e-learning platform* secara bersamaan. Hal tersebut berbeda dengan pertemuan harian yang waktunya bersifat fleksibel. Saat evaluasi hasil belajar dilakukan, *server* yang menjalankan aplikasi harus menangani *request* dari *concurrent user* dengan jumlah yang lebih banyak. Di samping itu, isu waktu juga menjadi hal yang perlu diperhatikan. Ujian biasanya memiliki batas waktu tertentu. *Traffic* yang padat dari *client* menuju *server* saat ujian sangat berpotensi untuk menjadi penghambat pelaksanaan ujian.

Satu solusi yang diusulkan untuk mengatasi masalah yang terjadi saat evaluasi hasil belajar dilakukan secara daring adalah mengembangkan sistem sendiri khusus untuk ujian. Dengan kata lain, perguruan tinggi perlu memisahkan sistem untuk ujian daring dari *learning management system* (LMS). Pemisahan sistem untuk ujian dari LMS meringankan beban server karena tidak semua sumber dayanya bisa dibuat terpisah dengan proses belajar mengajar harian. Disamping itu, mengembangkan sistem sendiri lebih menjamin terpenuhinya kebutuhan bisnis perguruan tinggi (Dennis, Wixom, & Roth, 2014).

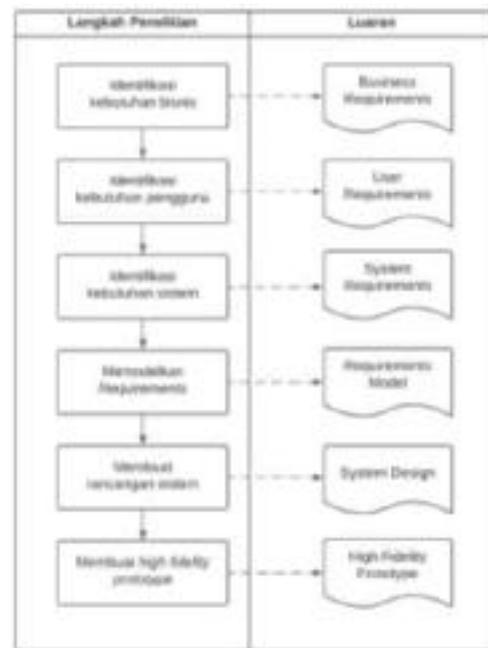
Untuk mencapai solusi tersebut, perguruan tinggi perlu melakukan akuisisi sistem baru. Dalam hal ini adalah sistem evaluasi hasil belajar. Ada tiga cara akuisisi yang dapat dilakukan. Pertama adalah mengembangkan sistem sendiri secara *in house*. Kedua adalah dengan membeli paket sistem yang sudah jadi. Dan ketiga adalah mengembangkan sistem secara *oursource* ke pihak ketiga.

Proses akuisisi sistem baru, baik yang dikembangkan sendiri ataupun membeli sistem yang sudah ada, membutuhkan proses analisis kebutuhan (*requirement analysis*). Tanpa proses tersebut, sistem yang dihasilkan cenderung gagal untuk memenuhi kebutuhan perguruan tinggi (Tilley & Rosenblatt, 2016). Hal tersebut kemudian melatarbelakangi penelitian ini.

Fokus penelitian ini adalah untuk menganalisis kebutuhan sistem evaluasi hasil belajar daring dalam konteks pendidikan tinggi. Tujuannya adalah mendapatkan *system requirements* yang valid dan sesuai dengan kondisi perguruan tinggi secara umum. Untuk mencapai tujuan tersebut, ada tiga pertanyaan penelitian yang perlu dijawab. Pertama adalah apa saja kebutuhan sistem yang ada? Lalu kedua adalah bagaimana model kebutuhannya? Dan ketiga, sejauh mana model kebutuhan tersebut dapat dikembangkan menjadi rancangan sistem ujian daring? Tiga pertanyaan tersebut terjawab melalui penelitian ini.

## 2. METODOLOGI

Penelitian yang dilakukan ini menggunakan metode kualitatif yang mengikuti tahapan *system development life cycle* (SDLC) sebagai kerangka kerja. Gambar 1. menunjukkan langkah-langkah penelitian yang dilakukan beserta dengan luaran dari tiap langkah tersebut. Pada dasarnya langkah penelitian dilakukan untuk menjawab tiga pertanyaan penelitian yang dijelaskan pada bagian sebelumnya.



Gambar 1. Langkah dan Luaran Penelitian

Seperti yang terlihat pada Gambar 1., ada enam langkah yang dilakukan dalam penelitian ini. Mulai dari identifikasi kebutuhan bisnis pada langkah pertama, hingga membuat *high fidelity prototype* di langkah terakhir. Langkah pertama hingga ketiga dilakukan untuk menjawab pertanyaan penelitian yang pertama. Sementara itu langkah keempat dilakukan untuk menjawab pertanyaan penelitian yang kedua. Dan pertanyaan penelitian yang ketiga dijawab setelah melakukan langkah kelima dan keenam. Semua pertanyaan penelitian selesai dijawab setelah semua langkah dilakukan.

### 2.1 Identifikasi *Business Requirements*

Langkah pertama yang dilakukan dalam penelitian ini adalah

mengidentifikasi kebutuhan bisnis atau organisasi. Dalam penelitian ini, konteks organisasinya adalah perguruan tinggi. Identifikasi kebutuhan bisnis dilakukan dengan melakukan studi dokumen milik tiga perguruan tinggi di Indonesia. Perguruan tinggi yang dipelajari masing-masing berbentuk universitas, institut, dan sekolah tinggi. Visi, misi, dan tujuan setiap perguruan tinggi dipelajari untuk mendapatkan kesamaan pola kebutuhan bisnis.

## 2.2 Identifikasi User Requirements

Kebutuhan pengguna diidentifikasi setelah kebutuhan bisnis perguruan tinggi berhasil didapatkan. Pada langkah ini dilakukan observasi pada *learning management system* masing-masing perguruan tinggi. Aktivitas yang diamati terbatas pada pelaksanaan evaluasi hasil belajar termasuk *quiz*, ujian tengah semester, dan ujian akhir semester. Lingkup observasi adalah pada program studi teknik informatika yang ada di perguruan tinggi masing-masing. Selain observasi dilakukan juga wawancara terhadap tiga orang dosen yang mewakili perguruan tinggi masing-masing. Wawancara dilakukan secara daring dengan format semi terstruktur. Hasil dari tahap ini adalah daftar kebutuhan pengguna terhadap sistem. Semua teknik pengumpulan data yang dilakukan untuk identifikasi requirement sejalan dengan apa yang direkomendasikan oleh Chakraborty, Sarker, dan Sarker (2010).

## 2.3 Identifikasi System Requirements

Jenis *requirement* ketiga yang diidentifikasi adalah *system requirement* atau kebutuhan bisnis. Requirements ini didapatkan dengan menganalisis *business* dan *user requirement* yang sudah diidentifikasi pada langkah sebelumnya. Observasi terhadap sistem sejenis yang sudah ada juga dilakukan untuk mendapatkan pola kesamaan fitur dan fungsi sistem. Luaran yang dihasilkan dari langkah ini adalah daftar *system requirement* yang terdiri atas *functional non-functional requirement*. Dua jenis *requirement* tersebut disarankan oleh

Pressman & Maxim (2020) untuk diidentifikasi ketika menentukan *system requirement*.

## 2.4 Memodelkan Requirements

Langkah keempat yang dilakukan dalam penelitian ini adalah membuat model *requirement*. Ada tiga model yang dibuat yaitu *functional model*, *process model*, dan *data model*. *Functional model* dibuat dengan menggunakan *use case analysis*. Sementara itu *process model* digambarkan dalam bentuk *workflow* dengan *swimlane diagram*. Dan *data model* dibuat dalam bentuk *entity relationship diagram* (ERD). Pertanyaan penelitian yang kedua terjawab setelah langkah keempat ini dilakukan.

## 2.5 Membuat Rancangan Sistem

Langkah kelima dalam penelitian ini adalah membuat rancangan (*design*) berdasarkan model-model kebutuhan sistem. Tahap ini dilakukan untuk menguji apakah *system requirement* dan *requirement model* pada langkah-langkah sebelumnya dapat dikembangkan menjadi *system design*. Ada tiga rancangan sistem yang dikembangkan yaitu rancangan antarmuka, rancangan penyimpanan data, dan rancangan arsitektur sistem.

## 2.6 Membuat Purwarupa Sistem

Langkah terakhir yang dilakukan dalam penelitian ini adalah membuat purwarupa sistem. Purwarupa dibuat dalam bentuk *high fidelity prototype* yang menurut Preece, Rogers, dan Sharp (2019) adalah jenis purwarupa yang paling mendekati rancangan akhir suatu sistem. Purwarupa dibangun untuk memvalidasi rancangan sistem yang dibuat. Baik purwarupa dan juga rancangan sistem yang dihasilkan pada langkah sebelumnya dilakukan untuk menjawab pertanyaan penelitian yang ketiga. Kesimpulan dalam penelitian ini disusun setelah keenam langkah penelitian selesai dilakukan.

## 3. ANALISIS DAN PEMBAHASAN

Setiap langkah penelitian yang sudah dilakukan menghasilkan luaran yang kemudian dijadikan masukan untuk

langkah selanjutnya. Hasil dari setiap langkah tersebut didapatkan melalui proses analisis secara kualitatif. Hasil dan analisis untuk masing-masing langkah penelitian tersebut dijelaskan pada bagian ketiga ini.

### 3.1 Hasil Identifikasi *Requirements*

Tiga jenis *requirements* yang dihasilkan dalam penelitian ini adalah *business requirement*, *user requirement*, dan *system requirement*. Bagaimana masing-masing *requirement* tersebut didapatkan dijelaskan pada bagian metodologi, yaitu pada langkah penelitian pertama, kedua, dan ketiga.

Ada tiga *business requirement* yang berhasil diidentifikasi, yaitu:

- [BR01] Perguruan Tinggi diwajibkan untuk melakukan evaluasi hasil belajar terhadap peserta didiknya.
- [BR02] Perguruan Tinggi harus menyelenggarakan semua kegiatan belajar mengajar secara daring, termasuk kegiatan evaluasi hasil belajar.
- [BR03] Perguruan Tinggi perlu membuat laporan hasil evaluasi belajar peserta didiknya setiap semester.

Tiga kebutuhan bisnis tersebut didapatkan setelah mempelajari dokumen dari tiga perguruan tinggi dan peraturan pemerintah yang berlaku saat penelitian ini dilakukan.

Tabel 1. Contoh User Requirements

Kode	User Requirement
[UR01]	Pengguna bisa mengakses sistem secara daring melalui web browser.
[UR02]	Pengajar bisa membuat sesi ujian daring untuk mahasiswa.
[UR03]	Pengajar bisa menentukan batas waktu pelaksanaan ujian daring.
...	...
[UR13]	Bagian akademik bisa mengunduh laporan rekapitulasi nilai dalam bentuk <i>spreadsheet</i> .

Analisis terhadap *business requirements* kemudian dilakukan pada tahap selanjutnya untuk mendapatkan

*user requirement*. Proses analisis tersebut dilengkapi dengan observasi dan wawancara seperti yang sudah dijelaskan prosedurnya di bagian metodologi. Hasil yang didapatkan adalah sebanyak 13 kebutuhan. Tabel 1. menampilkan 4 contoh *user requirements* di antaranya.

Tahap terakhir dari proses identifikasi *requirements* adalah mengidentifikasi *system requirements*. Analisis secara kualitatif kembali dilakukan pada tahap ini. Sebuah pertanyaan dijadikan panduan untuk mengidentifikasi kebutuhan sistem, yaitu: "apa yang bisa dilakukan sistem untuk memenuhi semua kebutuhan pengguna dan kebutuhan bisnis yang ada?". Dari tahap ini dihasilkan sebanyak 20 kebutuhan sistem. Tabel 2. memuat semua kebutuhan sistem yang berhasil diidentifikasi oleh penelitian ini.

Tabel 2. *System Requirements*

Kode	Software Requirements
[SR01]	Sistem berbentuk aplikasi berbasis web.
[SR02]	Sistem bisa dijalankan dengan baik di berbagai <i>browser</i> populer.
[SR03]	Sistem menyediakan <i>user role</i> pengajar, mahasiswa, dan bagian akademik
[SR04]	Sistem memungkinkan pengajar untuk membuat sesi ujian.
[SR05]	Sistem memungkinkan pengajar untuk menambahkan soal.
[SR06]	Sistem memungkinkan pengajar untuk menentukan waktu ujian.
[SR07]	Sistem memungkinkan pengajar untuk melihat nilai peserta ujian.
[SR08]	Sistem memungkinkan pengguna untuk mengunduh rekapitulasi nilai dalam bentuk <i>spreadsheet</i> .
[SR09]	Sistem memungkinkan pengajar untuk memberikan nilai secara manual.

Tabel 2. *System Requirements* (Lanjutan)

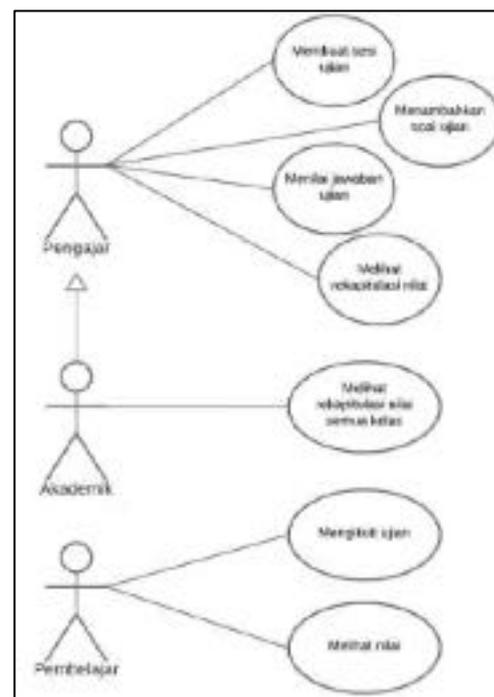
Kode	Software Requirements
[SR10]	Sistem memungkinkan penilaian jawaban ujian mahasiswa secara otomatis.
[SR11]	Sistem memungkinkan mahasiswa mengikuti ujian.
[SR12]	Sistem menampilkan hitung mundur waktu ujian yang tersisa.
[SR13]	Sistem menampilkan soal ujian secara responsif dengan perangkat pengguna.
[SR14]	Sistem menampilkan informasi apakah mahasiswa sudah mengikuti ujian atau belum.
[SR15]	Sistem memungkinkan mahasiswa untuk melihat nilainya sendiri.
[SR16]	Sistem memungkinkan mahasiswa untuk menyimpan jawaban sementara.
[SR17]	Sistem menyimpan jawaban yang sudah terisi oleh mahasiswa secara otomatis setelah waktu ujian berakhir.
[SR18]	Sistem memungkinkan bagian akademik untuk melakukan semua hal yang bisa dilakukan pengajar.
[SR19]	Sistem memungkinkan bagian akademik untuk mengunduh rekapitulasi nilai evaluasi hasil belajar dalam bentuk spreadsheet.
[SR20]	Sistem memungkinkan bagian akademik melihat laporan statistik pelaksanaan evaluasi hasil belajar.

Setiap kebutuhan sistem yang diidentifikasi pada Tabel 2., setidaknya memenuhi satu kebutuhan pengguna atau kebutuhan bisnis. Misalnya, kebutuhan bisnis [BR03] dipenuhi oleh kebutuhan sistem [SR19]. Sementara itu, kebutuhan sistem [SR19] juga memenuhi kebutuhan pengguna [UR13]. Kondisi yang sama juga berlaku untuk kebutuhan sistem lainnya. Pada dasarnya pertanyaan panduan yang sudah disebutkan juga berfungsi untuk memastikan setiap kebutuhan sistem relevan dengan kebutuhan bisnis dan juga kebutuhan pengguna. Dua puluh kebutuhan sistem ini merupakan jawaban untuk pertanyaan penelitian yang pertama. Selanjutnya,

dari kebutuhan sistem tersebut dibuatkan modelnya.

### 3.2 Model-model Requirements

Dari kebutuhan sistem yang berhasil diidentifikasi, selanjutnya dibuatkan model-model kebutuhannya. Ada tiga model kebutuhan yang dibuat dalam penelitian ini, yaitu *functional model*, *process model*, dan *data model*. Masing-masing model saling terkait satu sama lain dan semuanya dihasilkan melalui analisis terhadap kebutuhan sistem.



Gambar 2. Use Case Diagram

Model fungsional didapatkan setelah melakukan *use case analysis*. Dari analisis didapatkan tiga orang aktor yang ada di dalam sistem ini. Ketiga aktor tersebut adalah pengajar, pembelajar, dan akademik. Selain aktor, pada tahap ini juga diidentifikasi sebanyak tujuh buah *use case* utama. Hubungan antara aktor dengan *use case* yang bisa dilakukannya dinyatakan dalam bentuk *use case diagram* seperti yang dapat dilihat pada Gambar 2.

Seperti yang terlihat pada Gambar 2, aktor pengajar memiliki 4 buah *use case*. Pengajar bisa membuat sesi ujian, menambahkan soal ujian, menilai

jawaban ujian, dan melihat rekapitulasi nilai. Semua yang bisa dilakukan oleh pengajar tersebut bisa dilakukan juga oleh bagian akademik. Oleh sebab itu, notasi generalisasi ditambahkan pada *use case diagram* (Gambar 2) dari aktor akademik menuju aktor pengajar. Sebagai tambahan, bagian akademik ini bisa melihat rekapitulasi nilai untuk semua kelas. Sementara itu, yang bisa dilakukan oleh pembelajar adalah mengikuti ujian dan melihat nilai yang diperoleh.

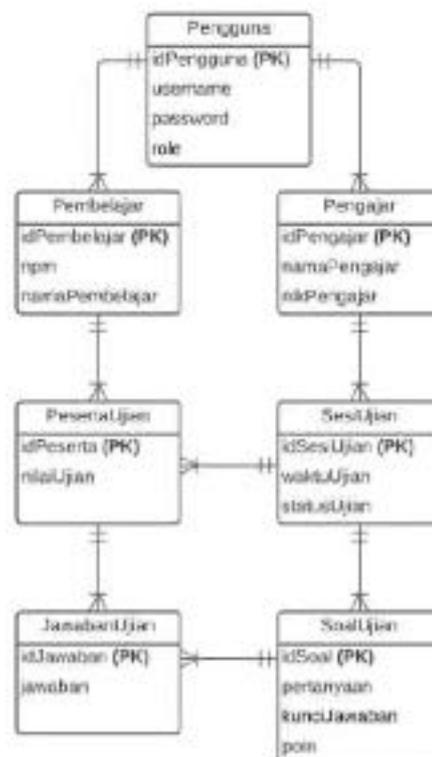


Gambar 3. Workflow Sistem

Model proses dari sistem dinyatakan dalam bentuk alur kerja (*workflow*). Alur kerja sistem ini digambarkan dengan *swimlane diagram* yang hasilnya dapat dilihat pada Gambar 3. Berbeda dengan *use case diagram* yang menampilkan 3 aktor, *workflow* yang dibuat hanya mencakup 2 aktor saja yaitu pengajar dan pembelajar. Hal tersebut dilakukan tanpa mengesampingkan peran aktor akademik di dalam sistem. Pada dasarnya semua yang bisa dilakukan pengajar bisa juga dilakukan oleh akademik. Oleh sebab itu,

kolom pengajar pada Gambar 3 berlaku juga untuk akademik. Pada praktiknya, satu sesi ujian bisa dibuat baik oleh pengajar ataupun bagian akademik. Hal tersebut didasari pada kebutuhan bisnis bahwa pengajar bisa mendelegasikan tugas pengaturan ujian kepada bagian akademik.

Seperti yang terlihat pada Gambar 3, setiap pelaksanaan ujian melewati sembilan proses. Proses tersebut diawali dengan membuat sesi ujian baru yang dilakukan oleh pengajar. Dilanjutkan dengan membuat soal dan mengaktifkan sesi ujian. Ketika sesi ujiannya sudah aktif, pembelajar bisa memulai ujian dan menjawab pertanyaan. Pembelajar dianggap selesai melakukan ujian apabila telah berhasil mengirimkan jawaban. Pengajar kemudian dapat memeriksa jawaban yang terkirim tersebut. Di akhir pelaksanaan ujian, pengajar dapat melihat laporan pelaksanaan ujian. Dan pengajar bisa melihat nilai yang dipublikasikan. Seperti itu cara membaca *workflow* sistem pada Gambar 3.



Gambar 4. Entity Relationship Diagram

Model kebutuhan ketiga yang dihasilkan dari penelitian ini adalah model data. Secara teori, sistem informasi berbasis web dapat menggunakan *relational database* untuk mengelola data yang ada di dalamnya. Penelitian ini menerapkan teori tersebut dan memodelkan data untuk sistem dalam bentuk *relational database*. Model data yang dihasilkan dinotasikan dalam *entity relationship (ER) diagram*. Gambar 4 menunjukkan *ER diagram* yang menjadi luaran dari penelitian ini.

Model data untuk sistem ujian daring terdiri atas 7 entitas yang saling terkait. Tujuh entitas tersebut dan juga keterkaitan antar entitasnya disajikan pada Gambar 4. Entitas pertama adalah pengguna yang menyimpan atribut *username*, *password*, dan *role*. Atribut *role* pada entitas pengguna menentukan apakah seorang pengguna merupakan pengajar atau mahasiswa. Di sisi lain, aktor bagian akademik juga disimpan sebagai pengajar karena atribut yang diperlukan sama dengan pengajar. Setiap pengajar berelasi langsung dengan entitas 'SesiUjian'. Terdapat kebutuhan yang menyatakan bahwa satu pengajar bisa membuat banyak sesi ujian. Dengan demikian relasi yang dipilih untuk memodelkannya adalah relasi one-to-many seperti yang terlihat pada Gambar 4. Seperti itu seterusnya cara membaca model data yang menjadi luaran penelitian ini.

Model kebutuhan fungsional, model proses, dan model data merupakan luaran yang dihasilkan pada tahap analisis kebutuhan sistem. Ketiga model tersebut sekaligus menjadi jawaban untuk pertanyaan penelitian yang kedua. Dari model-model *requirements* ini kemudian dibuatkan rancangan sistemnya.

### 3.3 Rancangan Sistem

Penelitian ini menghasilkan dua rancangan sistem dari model kebutuhan yang telah diidentifikasi pada tahap sebelumnya. Pertama adalah rancangan antarmuka yang menggambarkan tempat pengguna berinteraksi dengan sistem.

Dan kedua adalah rancangan arsitektur sistem yang menjelaskan spesifikasi hardware dan software yang diperlukan untuk menjalankan sistem secara optimal. Di dalam rancangan arsitektur tersebut juga dijelaskan seperti apa rancangan programnya.



Gambar 5. Rancangan Antarmuka Halaman Membuat Soal

Rancangan antarmuka sistem dibuat dalam bentuk *wireframe*. Rancangan antarmuka yang dibuat mengikuti setiap proses pada *workflow* (Gambar 3). Total rancangan antarmuka yang dibuat adalah 10 buah, dengan komposisi 9 rancangan untuk setiap proses pada *workflow* dan 1 rancangan untuk *template* utama. Contoh rancangan antarmuka yang dihasilkan dapat dilihat pada Gambar 5 dan Gambar 6. Gambar 5 merupakan rancangan antarmuka untuk proses membuat soal yang diujikan. Seperti yang terlihat pada Gambar 3 sebelumnya, aktivitas ini dilakukan oleh pengajar. Sementara itu Gambar 6 menunjukkan rancangan antarmuka saat pembelajar melaksanakan ujian secara daring.

Saat membuat soal, pengajar diminta untuk menuliskan pertanyaan berikut dengan pilihan-pilihan jawabannya. Selain itu, pengajar juga harus menentukan jawaban apa yang benar. Bobot nilai untuk setiap soal yang diujikan juga ditentukan agar sistem bisa memberikan nilai secara otomatis. Untuk memenuhi serangkaian proses pembuatan tersebut, rancangan antarmuka yang dibuat didominasi dengan kolom isian seperti yang terlihat pada Gambar 5.



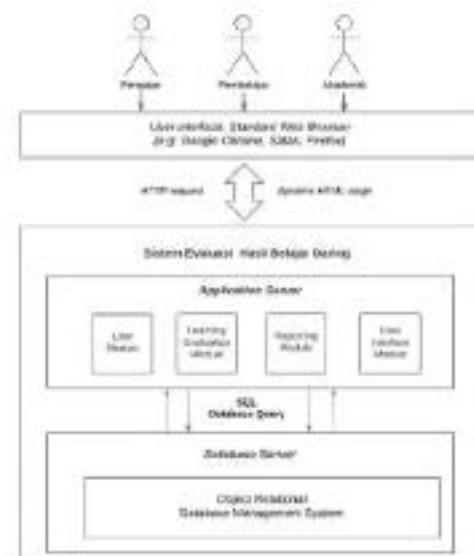
Gambar 6. Rancangan Antarmuka Halaman Ujian

Rancangan antarmuka ketika peserta ujian mengerjakan soalnya adalah seperti yang terlihat pada Gambar 6. Elemen yang ada di dalam antarmuka tersebut setidaknya ada lima buah. Pertama adalah informasi nama peserta dan mata kuliah yang diujikan. Lalu yang kedua adalah sisa waktu yang ada untuk menyelesaikan ujian. Komponen ketiga tentu saja pertanyaan ujian beserta pilihan jawaban yang tersedia. Satu tampilan layar hanya memuat satu pertanyaan saja. Oleh karena itu ada tombol navigasi untuk beralih ke pertanyaan selanjutnya atau kembali ke pertanyaan sebelumnya. Tombol navigasi ini adalah elemen yang keempat. Dan elemen terakhir adalah *progress bar* yang menunjukkan berapa persen soal yang sudah dijawab. Kelima elemen tersebut dapat dilihat pada rancangan antarmuka untuk halaman ujian di Gambar 6.

Rancangan terakhir yang dihasilkan dalam penelitian ini adalah rancangan arsitektur untuk sistemnya. Salah satu kebutuhan sistem ini adalah dapat diakses melalui internet. Untuk memenuhi kebutuhan tersebut maka sistem dibuat dalam bentuk aplikasi berbasis web. Oleh sebab itu, rancangan arsitektur yang dibuat mengacu pada rancangan aplikasi berbasis web secara umum. Gambar 7 menunjukkan rancangan arsitektur untuk sistem ujian daring hasil penelitian ini.

Seperti yang terlihat pada rancangan arsitektur pada Gambar 7., pengguna mengakses sistem menggunakan *web browser*. Dari browser tersebut suatu *http request* dikirim menuju *server*. Sistem evaluasi hasil belajar daring ini melibatkan dua *server*, yaitu *application*

*server* dan *database server*. Kode-kode program ditempatkan pada *application server*. Program yang ada di dalam sistem dirancang dalam empat *module*. Keempat *module* tersebut adalah *user module*, *learning evaluation module*, *reporting module*, dan *user interface module*. *User module* bertanggung jawab untuk segala fungsi yang berkaitan dengan pengelolaan pengguna. Sementara itu *learning evaluation module* merupakan inti dari sistem ini. *Module* tersebut memuat program untuk pelaksanaan ujian. Proses pelaporan hasil belajar ditangani oleh *reporting module*. Dan *user interface module* berisi kode-kode program untuk menampilkan antarmuka ke hadapan pengguna.



Gambar 7. Rancangan Arsitektur Sistem

Sistem ini juga membutuhkan *database server* untuk menjalankan *object relational database management system*. Apabila ada suatu *module* dalam *application server* membutuhkan data, maka ada program dari module tersebut yang mengirimkan *SQL query* ke *database server*. Hasil query tersebut dikembalikan lagi ke *module* yang bersangkutan. Lalu kemudian *user interface module* membuat halaman antarmuka yang bisa dilihat oleh pengguna melalui browser mereka. Apa yang dilihat oleh pengguna tersebut merupakan respon atas *http request* yang

mereka kirimkan sebelumnya. Hasilnya disajikan dalam bentuk *dynamic-html page* yang memungkinkan satu template antarmuka untuk menyesuaikan informasi yang disajikan.

### 3.4 High Fidelity Prototype

Semua rancangan yang sudah dihasilkan pada langkah sebelumnya, kemudian diimplementasikan dalam bentuk *high fidelity prototype*. Pengembangan *prototype* ini dilakukan untuk menunjukkan rancangan dan model kebutuhan yang dihasilkan dalam penelitian ini adalah valid. Prototipe ini dikembangkan dengan bahasa pemrograman PHP untuk *backend* aplikasinya. Sedangkan *frontend* dari aplikasinya ditulis dengan HTML dan JQuery. Tangkapan layar dari hasil pengembangan prototipe ini dapat dilihat pada Gambar 7.



Gambar 7. High Fidelity Prototype

Contoh tangkapan layar prototipe yang pada Gambar 7 merupakan pengembangan dari rancangan antarmuka yang ada pada Gambar 6. Gambar 7 merupakan rancangan final untuk antarmuka halaman ujian yang sudah dalam bentuk aplikasi berjalan. Secara fungsionalitas, prototipe yang dikembangkan sudah memungkinkan pengguna untuk melakukan proses-proses yang ada di dalam *workflow* Gambar 3.

Prototipe ini belum diujicoba untuk pelaksanaan evaluasi ujian yang sesungguhnya. Namun demikian, fungsionalitasnya sudah diuji dengan *teknik white-box testing*. Lingkup pengujiannya adalah setiap kode program yang berkaitan langsung dengan *workflow* sistem yang sebelumnya

ditunjukkan pada Gambar 3. Dari hasil pengujian ini ditemukan beberapa bug terutama pada aktivitas pengumpulan jawaban peserta ujian dan pada proses penyajian laporan. Semua bug yang ditemukan langsung diperbaiki saat itu juga karena sifat dari *white-box testing* memang membuka kode-kode programnya.

## 4. KESIMPULAN

Setelah menyelesaikan semua langkah penelitian, terdapat beberapa hal yang bisa disimpulkan. Penelitian ini berhasil mengidentifikasi 3 *business requirements*, 13 *user requirements*, dan 20 *system requirements*. Tiga kelompok *requirement* tersebut didapatkan secara berurutan dan saling terkait satu sama lain. Kebutuhan sistem yang diidentifikasi tersebut merupakan jawaban dari pertanyaan penelitian yang pertama.

*System requirement* yang dihasilkan kemudian juga dapat dibuatkan model kebutuhannya. Ada tiga *requirement model* yang dihasilkan dari penelitian ini. Pertama adalah *functional model* yang disajikan dalam bentuk *user case diagram*. Kedua adalah *process model* berupa *workflow* yang tersaji dalam bentuk *swimlane diagram*. Dan model kebutuhan terakhir adalah model data untuk sistemnya. Model data ini digambarkan dengan *entity relationship diagram* (ERD). Tiga model kebutuhan ini merupakan jawaban untuk pertanyaan penelitian yang kedua.

Rancangan yang memenuhi model kebutuhan sistem kemudian dibuat pada langkah selanjutnya. Sebanyak dua rancangan sistem berhasil dihasilkan pada tahap ini. Rancangan pertama yang dihasilkan adalah rancangan antarmuka. Rancangan antarmuka yang dihasilkan berupa *mockup* antarmuka dalam bentuk *wireframe*. Sementara itu rancangan yang kedua merupakan rancangan arsitektur untuk sistemnya. Rancangan arsitektur ini menunjukkan spesifikasi *software* dan *hardware* secara umum agar sistem berjalan dengan baik. Di dalam

rancangan arsitektur ini juga tercakup rancangan program yang terbagi menjadi empat *module*. Rancangan program tersebut dapat dijadikan acuan oleh programmer untuk mengembangkan kode-kode program untuk sistem sejenis.

Pada bagian akhir penelitian ini dilakukan pengembangan *high fidelity prototype*. Hal ini dilakukan untuk memastikan rancangan yang dihasilkan dapat dikembangkan menjadi produk aplikasi yang berjalan. Pengujian fungsionalitas dari prototipe ini telah dilakukan dengan *white-box testing*. Kesimpulan yang dihasilkan setelah pengujian ini adalah rancangan dan model kebutuhan hasil penelitian ini dapat diimplementasikan dalam bentuk prototipe sistem yang berfungsi dengan baik. Pengembangan dan pengujian lebih lanjut tetap diperlukan untuk dapat mengimplementasikan sistem ini pada skenario penggunaan yang sebenarnya.

## 5. DAFTAR PUSTAKA

1. Chakraborty, S., Sarker, S., & Sarker, S. (2010). An exploration into the process of requirements elicitation: A grounded approach. *Journal of the Association for Information Systems*, 11(4), 212-249. DOI: 10.17705/1jais.00225
2. Dennis, A., Wixom, B.H., Roth, R.M. (2014). *System analysis and design* (6th edition). Wiley.
3. Menteri Pendidikan dan Kebudayaan Republik Indonesia (Mendikbud). (2020). Surat Edaran Nomor 4 Tahun 2020 Tentang Pelaksanaan Kebijakan Pendidikan Dalam Masa Darurat Penyebaran Corona Virus Disease (COVID-19). Jakarta.
4. Preece, J., Rogers, Y., & Sharp, H. 2019. *Interaction design: beyond human-computer interaction* fifth edition. Wiley
5. Pressman, R.S., & Maxim, B.R. (2020). *Software Engineering A Practitioner's Approach* Ninth Edition. McGraw-Hill Education.
6. Tilley, S., & Rosenblatt, H.J. (2016). *System Analysis and Design* Eleventh Edition. Cengage Learning, Inc.