

**RANCANG BANGUN APLIKASI PENILAIAN ESAI
SINGKAT BERBAHASA INDONESIA DAN INGGRIS
MENGUNAKAN METODE TEST-DRIVEN
DEVELOPMENT**

TUGAS AKHIR

Diajukan sebagai syarat menyelesaikan jenjang strata Satu (S-1)
di Program Studi Teknik Informatika, Jurusan Teknologi,
Produksi dan Industri, Institut Teknologi Sumatera

Oleh:

GEIZKA ROZILIA RUICOSTA

119140114



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI, PRODUKSI DAN INDUSTRI
INSTITUT TEKNOLOGI SUMATERA
LAMPUNG SELATAN**

2023

DAFTAR ISI

BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	4
1.4 Batasan Masalah	4
1.5 Manfaat Penelitian	4
1.6 Sistematika Penulisan	5
1.6.1 BAB I PENDAHULUAN	5
1.6.2 BAB II TINJAUAN PUSTAKA	5
1.6.3 BAB III METODE PENELITIAN	5
BAB II TINJAUAN PUSTAKA	6
2.1 Tinjauan Pustaka	6
2.2 Dasar Teori	10
2.2.1 Test-Driven Development	10
2.2.2 Use Case Diagram	12
2.2.3 Activity Diagram	13
2.2.4 Website	14
2.2.5 JavaScript	15
2.2.6 Node.js	15
2.2.7 Express.js	15
2.2.8 JavaScript Object Notation	16
2.2.9 Esai Singkat	17
2.2.10 Python	17
2.2.11 System Usability Scale	17

2.2.12	Black Box Testing	19
2.2.13	Peramban Web	20
2.2.14	Peluncuran Perangkat Lunak	20
BAB III METODE PENELITIAN		21
3.1	Alur Penelitian	21
3.2	Penjabaran Langkah Penelitian	21
3.2.1	Identifikasi Masalah	22
3.2.2	Studi Literatur	22
3.2.3	Penerapan <i>Test-Driven Development</i>	22
3.2.4	Pengujian	23
3.3	Alat dan Bahan Tugas Akhir	23
3.3.1	Alat	23
3.3.2	Bahan	24
3.4	Metode Pengembangan	24
3.4.1	Kebutuhan	25
3.4.1.1	Wawancara	25
3.4.1.2	User Story	27
3.4.1.3	Kebutuhan Fungsional	27
3.4.1.4	Kebutuhan Non-Fungsional	28
3.4.1.5	Use Case Diagram	28
3.4.1.6	Activity Diagram	29
3.4.1.7	Mockup	31
3.4.2	Pembuatan Uji Coba (Test Writing)	35
3.4.3	Pembuatan Kode (Coding)	36
3.4.4	Refaktor (Refactor)	36
3.5	Rancangan Pengujian	36

3.5.1	Responsiveness	36
3.5.2	System Usability Scale	37
3.5.3	Black Box	37
3.6	Peluncuran	40

DAFTAR TABEL

Tabel 2.1 Perbandingan Referensi	8
Tabel 2.2 Simbol <i>Use Case Diagram</i>	13
Tabel 2.3 Simbol <i>Activity Diagram</i>	14
Tabel 2.4 Standar hasil SUS [13]	19
Tabel 3.1 Hasil Wawancara	26
Tabel 3.2 Kumpulan User Story	27
Tabel 3.3 Kebutuhan Fungsional	28
Tabel 3.4 Kebutuhan Non-Fungsional	28
Tabel 3.5 Skenario Uji Coba	35
Tabel 3.6 Kuesioner <i>System Usability Scale</i>	37
Tabel 3.7 Daftar pengujian Black Box	38

DAFTAR GAMBAR

Gambar 2.1 SDLC Test-Driven Development (TDD) [20]	11
Gambar 2.2 SUS <i>Score Percentile Rank</i> [13]	19
Gambar 3.1 <i>Flowchart</i> Penelitian	21
Gambar 3.2 Alur Penelitian <i>Test-Driven Development</i>	25
Gambar 3.3 <i>Use Case Diagram</i> Aplikasi	29
Gambar 3.4 <i>Activity Diagram</i> Aplikasi	30
Gambar 3.5 <i>Mockup</i> Halaman Home	31
Gambar 3.6 Mockup Metode Penilaian Tunggal	32
Gambar 3.7 <i>Mockup</i> Hasil Metode Penilaian Tunggal	32
Gambar 3.8 <i>Mockup</i> Metode Penilaian Jamak	33
Gambar 3.9 <i>Mockup</i> Hasil Metode Penilaian Jamak	34
Gambar 3.10 <i>Mockup</i> Halaman Bantuan	34

DAFTAR RUMUS

Rumus perhitungan SUS (1).....	18
--------------------------------	----

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Berdasarkan data statistik dari Perserikatan Bangsa-Bangsa (*United Nations*), sejak bulan November 2022 jumlah populasi manusia telah menyentuh angka delapan milyar dengan negara Indonesia berada di peringkat 4 dengan angka 280 juta jiwa [1][2]. Kemudian pada tahun 2021, Badan Pusat Statistik menyatakan bahwa penduduk di Indonesia terdiri dari 22 juta jiwa anak-anak dengan rentang umur 0 s/d 4 tahun dan 22 juta jiwa anak-anak dengan rentang umur 5 s/d 9 tahun dengan total anak-anak dengan umur dibawah 10 tahun berjumlah 44 juta jiwa yang merupakan 15.71% dari total populasi penduduk di Indonesia [3].

Dengan kemungkinan banyaknya pelajar yang akan bersekolah, maka teknologi di bidang pendidikan juga diperlukan untuk meningkatkan efisiensi kegiatan belajar mengajar di Indonesia, khususnya pada teknik penilaian esai karena esai memperbolehkan pelajar untuk memberikan jawaban rinci terkait pertanyaan yang diberikan oleh pengajar [4]. Menurut studi sosial yang dilakukan pada tahun 2018, semakin banyak esai yang harus dikoreksi oleh pengajar, maka kualitas penilaian juga akan menurun dan tidak lagi objektif (bias) yang menyebabkan evaluasi penilaian pengajar tidak akurat [5].

Permasalahan efisiensi waktu pada penilaian esai dapat dilakukan apabila evaluasi esai pelajar dilakukan oleh beberapa tim pengajar sehingga waktu yang diperlukan untuk menilai seluruh esai dapat terbagi dengan banyaknya jumlah tim pengajar. Solusi pertama walau dapat mempersingkat waktu yang dibutuhkan untuk melakukan evaluasi terhadap jawaban esai pelajar, tetap memiliki permasalahan terkait subjektivitas yang dapat mempengaruhi akurasi penilaian yang dilakukan oleh tim pengajar. Selain itu, setiap pengajar yang melakukan evaluasi juga mungkin meyakini jawaban yang berbeda terkait sebuah pertanyaan dalam suatu esai, hal ini dapat menyebabkan penilaian tim pengajar semakin tidak akurat. Solusi lainnya adalah menggunakan teknologi untuk mengevaluasi esai pelajar berdasarkan jawaban

pengajar untuk menilai seberapa sesuai jawaban tersebut. Teknologi dapat berupa aplikasi yang terpasang pada sebuah perangkat tertentu (aplikasi *mobile* atau *desktop*) atau berupa aplikasi *online* yang dapat diakses melalui internet (*website*). Solusi ini tentunya akan memberikan penilaian lebih objektif terhadap jawaban esai yang dievaluasi.

Aplikasi ini kemudian juga harus dapat diakses dimanapun dan kapanpun agar penggunaan aplikasi tidak terbatas di satu perangkat saja. Namun, pengembangan aplikasi *mobile* perlu disesuaikan untuk setiap *platform* yang akan semakin membatasi dimana aplikasi dapat digunakan [6]. Maka dari itu, aplikasi berbasis *website* adalah solusi terbaik untuk permasalahan ini karena aplikasi akan dapat digunakan kapan saja dan dimana saja, tanpa membatasi perangkat apa yang dapat mengakses aplikasi tersebut.

Aplikasi akan dikembangkan mengikuti siklus hidup perangkat lunak *Test-Driven Development* (TDD) yang merupakan salah satu *Software Development Life-Cycle* (SDLC) *Agile* [7]. Praktik TDD dimulai dengan memikirkan cara menguji sebuah fungsi. Programmer kemudian menulis kode yang dapat memenuhi pengujian [8]. Dibanding dengan *eXtreme Programming* (XP), dan *Personal eXtreme Programming* (PXP), yang juga termasuk dalam SDLC *Agile*, TDD terbukti dapat menghasilkan aplikasi yang sangat kecil kemungkinannya untuk memiliki *bug* dan *error* karena pada setiap fungsi yang akan dikembangkan perlu dibuat syarat keberhasilan pengujian terlebih dahulu, kemudian jika kode yang dituliskan berhasil memenuhi syarat keberhasilan pengujian tersebut, pengembang dapat melanjutkan pengembangan fungsi selanjutnya [9].

Aplikasi yang dikembangkan akan dilakukan pada aplikasi yang dikembangkan untuk memastikan kelayakannya menggunakan dua metode pengujian, yaitu *System Usability Scale* (SUS) dan *Black Box*. SUS memiliki tingkat keabsahan dan keandalan tinggi untuk mengukur tingkat kemudahan perangkat lunak saat digunakan [10][11]. SUS digunakan karena terfokus pada kegunaan yang dirasakan pengguna melalui 10 kuesioner yang dapat diisi dengan cepat, dibanding *Post-Study System*

Usability Questionnaire (PSSUQ) yang lebih terfokus untuk menilai kenyamanan yang dirasakan pengguna saat menggunakan sebuah sistem [12]. SUS juga tidak membutuhkan biaya lebih besar untuk melakukan pengujiannya dibanding dengan *Heuristic Evaluation* (HE) yang merupakan salah satu metode pengujian perangkat lunak lainnya [13]. Sedangkan pengujian *Black Box* dilakukan untuk menguji tampilan, fungsionalitas, masukan, dan luaran perangkat lunak tanpa harus mengetahui bagaimana baris-per-baris program kode perangkat lunak bekerja dengan harapan bahwa perangkat lunak yang dikembangkan bekerja semestinya [14]. Pada penelitian ini, SUS digunakan karena pengembang ingin memfokuskan aplikasi pada sisi kegunaannya.

Dengan ini, akan dikembangkan aplikasi berbasis *website* yang dapat membantu pengajar dalam menilai jawaban esai pelajar berdasarkan jawaban yang sudah diberikan oleh pengajar untuk dijadikan landasan untuk program menilai jawaban pelajar. Berdasarkan wawancara yang dilakukan pada beberapa guru yang merupakan calon pengguna aplikasi yang akan dibangun, aplikasi akan memiliki dua fitur utama, yaitu untuk menilai satu jawaban esai singkat dan menilai banyak jawaban sekaligus, hal ini akan mengurangi lebih banyak beban kerja pengajar karena tidak perlu memasukkan jawaban pelajar satu-persatu. Fitur lainnya yaitu pengajar dapat memilih bahasa esai yang ingin diperiksa, Indonesia atau Inggris.

1.2 Rumusan Masalah

Untuk menyelesaikan permasalahan yang sudah dipaparkan pada latar belakang, maka didapat rumusan masalah sebagai berikut:

1. Bagaimana implementasi aplikasi penilaian esai singkat menggunakan metode *test-driven development* dapat dilakukan?
2. Bagaimana menerapkan *pre-trained machine learning model* untuk menilai esai singkat pada sebuah aplikasi?
3. Bagaimana memastikan hasil penilaian pada aplikasi sama dengan penilaian pada model *machine learning* berdasarkan skor yang dihasilkan?

1.3 Tujuan Penelitian

Tujuan dilakukannya penelitian tugas akhir ini berdasarkan latar belakang dan rumusan masalah adalah:

1. Mengetahui implementasi metode *test-driven development* pada pembangunan aplikasi implementasi penilaian esai singkat.
2. Menerapkan *pre-trained machine learning model* python pada aplikasi berbasis JavaScript.
3. Memastikan hasil penilaian dari aplikasi sama dengan hasil penilaian pada model *machine learning* berdasarkan skor.

1.4 Batasan Masalah

Batasan masalah yang ada pada penelitian tugas akhir ini yaitu:

1. Aplikasi berbasis web hanya dapat digunakan melalui *browser*.
2. Aplikasi hanya dapat memproses penilaian esai yang diketik, bukan ditulis tangan.
3. Aplikasi hanya dapat menerima masukan berupa teks untuk jawaban guru.
4. Aplikasi hanya dapat menerima masukan berupa teks atau *file excel* (xls) berisi kumpulan jawaban untuk jawaban murid.
5. Aplikasi dapat melakukan penilaian dengan teks minimal 1 frasa hingga 100 kata.
6. Aplikasi hanya dapat melakukan penilaian terhadap 1 pertanyaan pada sekali pemakaian.
7. Penelitian ini terfokus pada implementasi aplikasi menggunakan model *machine learning* penilaian esai bahasa Indonesia dan bahasa Inggris yang sudah ada.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat meningkatkan efisiensi kegiatan belajar-mengajar (KBM) pada seluruh jenjang pendidikan dengan mempermudah proses penilaian esai singkat pada soal dengan jawaban berbahasa Indonesia atau Inggris.

1.6 Sistematika Penulisan

1.6.1 BAB I PENDAHULUAN

Pada bab ini membahas mengenai latar belakang dalam pembuatan proyek tugas kerja praktik, rumusan masalah, tujuan, manfaat, ruang lingkup, metodologi dan sistematika penulisan dari laporan kerja praktik.

1.6.2 BAB II TINJAUAN PUSTAKA

Pada bab ini menjelaskan mengenai landasan teori atau pustaka yang digunakan dalam membuat laporan ataupun dalam pembuatan sistem yang dibuat.

1.6.3 BAB III METODE PENELITIAN

Pada bab ini menjelaskan langkah-langkah dilakukannya penelitian untuk mencapai hasil penelitian yang diharapkan.

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Pada penelitian ini, ada beberapa referensi yang dijadikan landasan pendukung untuk pelaksanaan setiap langkah penelitian. Referensi yang digunakan merupakan penelitian terdahulu yang akan menjadi acuan dalam membangun perangkat lunak berupa aplikasi berbasis web. Pada Tabel 2.1 berikut dapat diperhatikan referensi yang digunakan pada penelitian ini.

Aplikasi yang dikembangkan pada penelitian ini akan dibuat berbasis *website* menggunakan metode siklus hidup pengembangan perangkat lunak (SDLC) *Test-Driven Development* (TDD) yang merupakan salah satu bentuk SDLC *Agile* seperti yang sudah dilakukan pada penelitian sebelumnya dengan judul *Comparative Study of Test-Driven Development (TDD), Behavior-Driven Development (BDD) and Acceptance Test-Driven Development (ATDD)* yang dilakukan oleh Myint Myint Moe [7]. Pada penelitian yang dilakukannya, didapatkan bahwa TDD digunakan saat pengembang melakukan pengujian, hal ini berlaku untuk memastikan seluruh fitur dan fungsi yang dikembangkan pada aplikasi ini terbebas dari error dan bug.

Kemudian pada penelitian yang dilakukan oleh Xinyu Chang dan Jing Li dengan judul *Improvement of excel data processing function based on Spring MVC Framework* dijelaskan bahwa penggunaan JavaScript Object Notation (JSON) untuk memproses data dari excel agar dapat diproses lebih cepat [15]. Hal ini akan digunakan pada penelitian tugas akhir kali ini pada aplikasi saat melakukan penilaian otomatis untuk jawaban secara massal.

Penelitian ini juga akan menggunakan teknologi Express.js karena memiliki performa yang cepat dibandingkan pesaingnya yaitu Ktor sebagaimana yang telah diteliti pada penelitian sebelumnya dengan judul *Express.js and Ktor web server performance A comparative study* yang dilakukan oleh Isac Glantz dan Hampus Hurtig pada tahun 2022 [16].

Untuk memeriksa jawaban secara otomatis, akan digunakan model pemrosesan bahasa alami yang sudah terlatih berupa model *Python* dalam bahasa Inggris dan Indonesia. Diperlukan penelitian pendukung yang dapat menjembatani antara bahasa JavaScript yang akan digunakan pada website dan Python yang akan menjalankan pemeriksaan jawaban, penelitian tersebut adalah TENSORFLOW.JS: MACHINE LEARNING FOR THE WEB AND BEYOND [17]. Dengan ini, aplikasi yang dikembangkan akan menggunakan *Node.js* sebagai teknologi penghubung *JavaScript* dan *Python*.

Model yang digunakan dapat membandingkan jawaban siswa terhadap guru dalam bahasa Inggris dan Indonesia menggunakan teknik sebagaimana dilakukan pada penelitian oleh Svanhvít Ingólfssdóttir tentang *lemmatization* untuk bahasa Islandia dengan judul Nefnir: A high accuracy lemmatizer for Icelandic [18].

Aplikasi yang dikembangkan juga akan menggunakan basis data untuk menyimpan jawaban khususnya jawaban yang ingin diperiksa otomatis secara masal menggunakan basis data non-SQL MongoDB. MongoDB digunakan karena dapat melakukan ekspor data berupa JSON yang selanjutnya akan digunakan untuk memeriksa jawaban satu-persatu pada model *Python*. Kemudian, menggunakan basis data non-SQL memiliki performa lebih cepat daripada basis data SQL khususnya pada eksekusi *Query*. Hal ini didukung oleh penelitian berjudul Performance analysis of NoSQL and relational databases with MongoDB and MySQL yang dilakukan oleh Benymol Jose dan Sajimon Abraham pada tahun 2020 [19].

Tabel 2.1 Perbandingan Referensi

No	Judul	Permasalahan	Metode	Hasil	Perbandingan
1	TENSORFLOW.JS: MACHINE LEARNING FOR THE WEB AND BEYOND (2019)	<i>Library</i> pembelajaran mesin biasanya ditulis dalam bahasa <i>Python</i> atau C++. Namun, jumlah pengguna <i>JavaScript</i> untuk frontend dan backend semakin bertambah. Jurnal ini berisi panduan untuk menjembatani <i>Python</i> dengan <i>JavaScript</i>	-API (<i>Application Programming Interface</i>)	Menggunakan <i>Layers API</i> , didapatkan performa waktu sebesar - 3426 ms dengan <i>JavaScript</i> biasa dengan 1x percepatan, - 10 ms dengan <i>WebGL</i> dengan 342x percepatan, - 3 ms dengan <i>Node.js</i> dengan 1105x percepatan.	Aplikasi yang akan dikembangkan akan menggunakan API dibangun dengan <i>Node.js</i> tanpa menggunakan <i>TensorFlow</i> .
2	Improvement of excel data processing function based on Spring MVC Framework (2022)	<i>Excel</i> yang sangat umum digunakan, masih dinilai kurang pada pemrosesan sekelompok data dan verifikasi data. Dengan berkembangnya bidang sains dan teknologi, diperlukan pemrosesan data berskala besar dan berkelanjutan.	-JSON(<i>JavaScript Object Notation</i>)	Berdasarkan <i>Spring MVC</i> + <i>EasyUI</i> dengan pengembangan <i>Java J2EE IDE</i> . Diciptakan <i>kit</i> pengembangan yang bisa menkonversi <i>excel</i> dan melakukan ekspor dan impor sekumpulan data dengan menkonversinya menjadi JSON.	Excel akan dikonversi menjadi bentuk JSON menggunakan teknik <i>spawn JavaScript</i> .
3	Nefnir: A high accuracy lemmatizer for Icelandic (2019)	Mencari bentuk morfologi dasar pada sebuah kumpulan tulisan dengan bahasa yang kaya akan morfologi.	- <i>Nefnir</i> - <i>part-of-speech</i>	<i>Nefnir</i> meraih akurasi sebesar 99.55% untuk teks yang ditandai dengan benar. <i>Part-of-speech</i> (PoS) mendapat akurasi sebesar 96.88% dengan teks yang ditandai dengan <i>PoS tagger</i> .	Model yang digunakan pada penelitian ini dicapai menggunakan teknik <i>stemming</i> dan <i>lemmatization</i> .

No	Judul	Permasalahan	Metode	Hasil	Perbandingan
4	Express.js and Ktor web server performance A comparative study (2022)	Membandingkan dua <i>framework</i> web berdasarkan waktu merespon untuk membantu <i>developer</i> memilih antara <i>Express.js</i> dan <i>Ktor</i>	Melakukan test terhadap waktu respon menggunakan database melalui <i>Object Relational Mapper</i> (ORM) <i>Sequelize</i> untuk <i>Express.js</i> , dan <i>Exposed</i> untuk <i>Ktor</i>	<i>Express.js</i> memiliki waktu respon yang lebih baik (3 ms) secara keseluruhan daripada <i>Ktor</i> (106 ms). Namun penggunaan <i>Object Relational Mapper</i> pada <i>Ktor</i> lebih berpengaruh pada hasil daripada <i>Express.js</i> .	Aplikasi yang akan dikembangkan pada penelitian ini akan dibangun menggunakan <i>Express.js</i> sebagai framework modul HTTP dari <i>Node.js</i> .
5	Performance analysis of NoSQL and relational databases with MongoDB and MySQL (2020)	Basis data relasional dinilai tidak efektif untuk bekerja dengan bervariasi informasi yang besar, karena basis data relasional seperti <i>MySQL</i> menyimpan data secara terorganisir.	-Basis data non-relasional (<i>NoSQL</i>) seperti MongoDB.	Berdasarkan waktu eksekusi <i>query</i> untuk <i>insert</i> , menggunakan MySQL membutuhkan waktu 302156 ms, sedangkan saat menggunakan MongoDB hanya memerlukan waktu 56985 ms.	Aplikasi akan menggunakan basis data non-SQL MongoDB.
6	Comparative Study of Test-Driven Development (TDD), Behavior-Driven Development (BDD) and Acceptance Test-Driven Development (ATDD) (2019)	Membandingkan perbedaan antara <i>Test-Driven Development</i> (TDD), <i>Behavior-Driven Development</i> (BDD), dan <i>Acceptance Test-Driven Development</i> (ATDD) dalam pengembangan perangkat lunak dengan lingkungan pengembangan yang berbeda.	Studi banding metode pengembangan perangkat lunak.	TDD digunakan saat pengembang menulis dan menjalankan pengujian. BDD merincikan perilaku fitur menggunakan bahasa yang dapat dimengerti semua orang yang terkait dalam pembangunan. ATDD membuat implementasi lebih efektif.	SDLC TDD akan digunakan pada penelitian ini untuk mengembangkan aplikasi penilaian esai otomatis guna meminimalisir kemungkinan adanya <i>bug</i> pada tahap penulisan kode.

Pada Tabel 2.1 sebelumnya, dapat diperhatikan beberapa tinjauan pustaka yang menjadi landasan pada penelitian ini lengkap dengan permasalahan, metode, hasil, dan perbedaan penelitian penulis dengan landasan teori penelitian sebelumnya.

2.2 Dasar Teori

Ada beberapa teori yang akan digunakan untuk mendukung penelitian tugas akhir ini. Teori yang digunakan akan menjadi landasan, dan memperkuat pemahaman selama penelitian ini dilakukan. Berikut adalah teori yang digunakan pada penelitian tugas akhir ini:

2.2.1 Test-Driven Development

Test-Driven Development (TDD) merupakan metode pembangunan perangkat lunak *agile* yang dikenalkan oleh *Extreme Programming* (XP). TDD menggunakan pendekatan pengujian unit yang terfokus pada pengembang [7]. TDD bekerja dengan cara menuliskan kode pengujian dan fungsionalitas hingga pengujian berhasil, kemudian menuliskan kode pengujian dan fungsionalitas selanjutnya [9]. Metode ini meningkatkan kualitas produk perangkat lunak yang dikembangkan dan produktifitas pengembang.

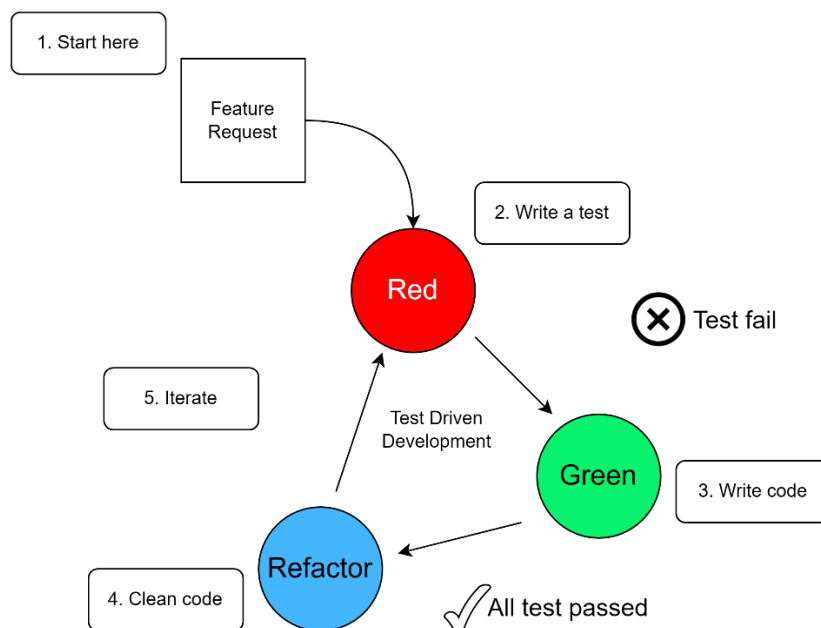
Studi kasus pada tahun 2019 menyebutkan kelebihan saat menggunakan metode pengembangan perangkat lunak TDD sebagai berikut:

1. Membantu mencegah kecacatan produk
2. Membantu dokumentasi kode dengan contoh eksekusi
3. Membantu programmer untuk sangat memahami kode mereka
4. Mendukung refaktor sebagai kebutuhan dan perubahan desain
5. Mendorong desain yang lebih baik
6. Menyediakan peringatan awal terkait masalah desain
7. Membuat rangkaian uji regresi otomatis
8. Programmer mempelajari cara menulis jenis pengujian lain
9. Mendorong langkah kecil dan prinsip bahwa lebih baik untuk menjaga sistem tetap bekerja

Kemudian, pada studi yang sama juga menyebutkan kekurangan saat menggunakan metode pengembangan perangkat lunak TDD, yaitu sebagai berikut:

1. Sulit untuk dipelajari
2. Sulit untuk diaplikasikan kepada kode yang telah dibuat sebelumnya dari orang lain (*legacy code*)
3. Banyak kesalahpahaman yang mempersulit programmer untuk mempelajarinya.

Test-Driven Development (TDD) memiliki siklus pengembangan perangkat lunak seperti Gambar 2.1 berikut ini:



Gambar 2.1 SDLC Test-Driven Development (TDD) [20]

1. Langkah pertama yaitu membaca, memahami, dan memproses fitur atau bug yang diminta.
2. Berdasarkan kebutuhan yang diminta, unit pengujian akan dibuat dan dijalankan dengan kode pada langkah 3.
3. Membuat dan mengimplementasikan kode yang memenuhi kebutuhan pengujian. Saat dijalankan, semua pengujian harus lolos, jika tidak, langkah ini akan diulang hingga berhasil.
4. Apabila sudah berhasil, maka kode akan dibersihkan/dirapikan (*refactoring*).

5. Ulang dari langkah 1 dengan permintaan fitur atau bug yang baru.

Alur kerja *Test-Driven Development* (TDD) sebagaimana dilihat pada Gambar 2.1 disebut juga sebagai *Red-Green-Refactoring* (Merah-Hijau-Refaktor), yang merupakan status dari pengujian dalam setiap siklusnya [20].

Penelitian tugas akhir ini akan menggunakan siklus hidup pembangunan perangkat lunak (SDLC) *Test-Driven Development* (TDD) karena kebutuhan yang dapat bervariasi dari aplikasi yang akan dikembangkan baik dari fitur maupun kebutuhan utama. Dengan menggunakan TDD, aplikasi yang dikembangkan dapat diminimalisir adanya *error* atau *bug* dengan melakukan unit pengujian untuk setiap langkah pembangunannya[8]. Pengembangan aplikasi penilaian esai singkat otomatis ini juga tidak menggunakan *legacy code* pada sisi yang dikerjakan oleh penulis, sehingga tidak perlu khawatir dengan kekurangan pada penerapan TDD yang telah disebutkan sebelumnya tentang kesulitan penerapan TDD pada *legacy code* [7].

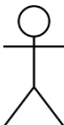
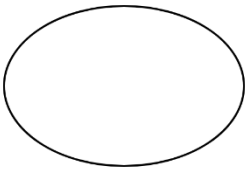
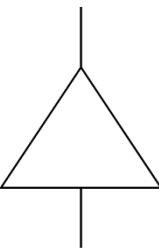



2.2.2 Use Case Diagram

Use Case Diagram adalah salah satu diagram dalam *Unified Modeling Language*. *Use Case Diagram* merupakan diagram perilaku yang mendeskripsikan kebutuhan fungsional pada sebuah perangkat lunak. Diagram ini juga digunakan untuk memahami bagaimana sebuah sistem seharusnya bekerja [21]. Berikut ini merupakan manfaat dari digunakannya *Use Case Diagram*, yaitu:

1. Mendapatkan tampilan luar dari suatu sistem.
2. Mendapatkan seluruh kebutuhan sistem.
3. Mengenali faktor-faktor yang dapat mempengaruhi sistem secara eksternal maupun internal.
4. Mendemonstrasikan interaksi antara sistem dan aktor.

Pada Tabel 2.2 dibawah ini, dapat diperhatikan nama dan deskripsi dari simbol yang akan digunakan dalam sebuah *Use Case Diagram*.

Tabel 2.2 Simbol *Use Case Diagram*

Simbol	Nama	Deskripsi
	Aktor	Peranan eksternal yang berinteraksi dengan sistem.
	Use Case	Fungsionalitas atau kebutuhan yang akan diimplementasikan pada sebuah sistem.
	Generalisasi	Merepresentasikan hubungan antara aktor dengan aktor, maupun <i>use case</i> dengan <i>use case</i> lainnya.
	Asosiasi	Merepresentasikan komunikasi dua arah antara aktor dengan <i>use case</i> , setiap kasus dimulai dengan aktor utama yang harus mendapatkan respon dari <i>use case</i> .
	Include	Merepresentasikan hubungan antara dua <i>use case</i> . Saat <i>use case</i> A membutuhkan <i>use case</i> B untuk menyelesaikan tugasnya, walau <i>use case</i> B dapat menyelesaikan tugasnya sendiri.
	Extend	Merepresentasikan hubungan antara dua <i>use case</i> . Saat <i>use case</i> A mungkin memerlukan <i>use case</i> B untuk menyelesaikan tugasnya, namun <i>use case</i> B tidak bisa ada sendiri.




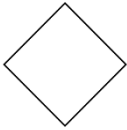
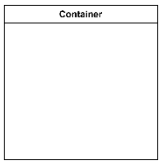
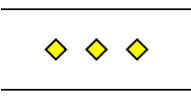
Penelitian ini akan menggunakan *Use Case Diagram* untuk memperjelas skenario dan kebutuhan dari perilaku sistem yang akan dikembangkan. Hal ini dilakukan untuk memastikan bahwa sistem yang akan dikembangkan memenuhi skenario dan perilaku sistem yang telah diajukan.

2.2.3 Activity Diagram

Sama seperti *Use Case Diagram*, *Activity Diagram* merupakan salah satu diagram dalam *Unified Modeling Language*. *Activity Diagram* dapat menampilkan alur

kegiatan dari sebuah sistem perangkat lunak [22]. *Activity Diagram* biasanya berisi aktifitas, transaksi, keputusan, swimlane, dan aktifitas paralel [23]. Berdasarkan itu, simbol, nama, dan deskripsi dari *Activity Diagram* dapat dilihat pada Tabel 2.3 berikut ini.

Tabel 2.3 Simbol *Activity Diagram*

Simbol	Nama	Deskripsi
	Start	Menunjukkan kondisi awal / dimulainya sebuah aktifitas
	End	Menunjukkan berakhirnya semua kondisi dari sebuah aktifitas
	Aktifitas	Menunjukkan aktifitas dari sebuah proses
	Keputusan	Menunjukkan kondisi percabangan dari 1 masukkan dengan banyak keluaran
	Swimlane	Mengorganisir aktifitas berdasarkan peran dari setiap proses
	Aktifitas Paralel	Menunjukkan proses yang berjalan secara paralel, biasanya terdiri dari dua aktifitas yang dijalankan bersamaan

Penelitian ini akan menggunakan *Activity Diagram* untuk menggambarkan alur aktifitas dari sistem perangkat lunak yang akan dikembangkan.

2.2.4 Website

Website adalah keseluruhan halaman-halaman web yang terdapat dari sebuah domain yang mengandung informasi [24]. Layanan yang terdapat pada *website* dapat diakses melalui jaringan internet, dengan permintaan layanan dieksekusi melalui sistem jarak jauh[25].

2.2.5 JavaScript

JavaScript adalah bahasa pemrograman yang populer, digunakan bukan hanya untuk web sisi klien, tapi juga sisi *server* [26]. *JavaScript* juga memiliki tipe dinamis, yang artinya pembangun tidak perlu menspesifikasikan tipe dalam kodenya. Hal ini yang membuat IDE *JavaScript* sering gagal mensugestikan tipe yang akurat karena tipe dari elemen kode tidak diketahui hingga memiliki isi. Namun kekurangan ini bisa diatasi dengan membuat kode yang bagus dan tidak ambigu.

Pada penelitian tugas akhir ini, *JavaScript* akan digunakan untuk membangun aplikasi berbasis website penilaian esai singkat pada sisi klien dan juga sisi *server* dengan *Node.js*.

2.2.6 Node.js

Node.js adalah *run timer* sisi server *JavaScript* yang digunakan pada *browser* untuk memproses HTTP. *Node.js* memungkinkan penulis untuk membuat *backend* dari sebuah *website* yang dapat menerima *request GET* dan *POST* menggunakan bahasa pemrograman *JavaScript* [27].

Node.js memiliki *package manager* yaitu NPM, gudang perangkat lunak terbesar di dunia dengan jutaan paket yang dapat digunakan pada perangkat lunak siapapun menjadi bantuan (*dependencies*).

Penelitian ini menggunakan *Node.js* sebagai *run timer* sisi *server* untuk aplikasi berbasis *website* yang akan dibangun karena dapat menangani lebih banyak permintaan dalam satuan waktu dibanding PHP [28]. Hal ini membuat *Node.js* lebih ideal untuk membangun *website* dengan masukan dan keluaran yang intens. Selain itu, *Node.js* juga *developer friendly*, yang mana mudah dipahami karena cara penggunaannya yang konsisten.

2.2.7 Express.js

Pada teori sebelumnya, dapat diketahui bahwa *Node.js* merupakan *run timer JavaScript* dan dapat menerima permintaan HTTP *GET* dan *POST*. *Express.js*

merupakan paket NPM berupa *framework* untuk mempermudah penggunaan modul HTTP pada *Node.js* [29] sehingga penulisan kode untuk melakukan permintaan HTTP dapat dilakukan lebih cepat.

Sebagai *framework* untuk menangani permintaan HTTP, performa *Express.js* terbilang cepat yaitu hanya memerlukan waktu 3 ms untuk memproses permintaan *Object Relational Model* [16], dan memerlukan waktu rata-rata 17 ms untuk memproses halaman dinamis dari permintaan sebanyak 200 pengguna.

Atas alasan tersebut, perangkat lunak yang akan dikembangkan dalam penelitian ini akan menggunakan *Express.js* untuk menangani permintaan HTTP sebagai *framework* dari *Node.js* karena kemampuan paket NPM ini diperlukan untuk memproses nilai dan jawaban pada aplikasi website penilaian esai singkat.

2.2.8 JavaScript Object Notation

JavaScript Object Notation (JSON) adalah format data dengan penulisan seperti tipe data objek pada bahasa pemrograman *JavaScript*. JSON merupakan format data berisi objek atau kumpulan objek-objek dalam suatu objek yang ringan, mudah dibaca dan ditulis oleh manusia, serta dibuat dan diterjemahkan dengan komputer. Format data ini merupakan bagian dari bahasa pemrograman *JavaScript* [30].

JSON umumnya berisikan sebuah objek yang dapat berisi banyak objek lain. Dalam format dokumen ini, objek harus dimulai dan diakhiri oleh kurung keriting ({ }), kemudian setiap objek dalam JSON dipisah dengan simbol koma (,) setelah kurung keriting tutup (}) [30].

Untuk menggunakan dokumen JSON pada aplikasi website khususnya yang menggunakan *JavaScript*, objek dapat diakses dengan menggunakan dokumen JSON dan menggunakan simbol titik (.) untuk mengakses setiap sub-objeknya.

Pada aplikasi web yang akan dikembangkan pada penelitian ini, akan ada fitur untuk mengunggah seluruh jawaban pelajar berupa dokumen *excel* ke aplikasi website

penilaian esai. Dokumen *excel* kemudian akan diubah menjadi bentuk JSON untuk selanjutnya diproses kedalam penilaian.

2.2.9 Esai Singkat

Esai merupakan karangan yang membahas suatu masalah berdasarkan sudut pandang penulis, pada kasus ini yaitu jawaban dari pertanyaan yang diberikan oleh pengajar menurut pelajar. Esai merupakan salah satu metode penilaian untuk mengukur tingkat pemahaman pelajar pada suatu topik yang kerap diberikan pada saat ujian [31].

Esai singkat menurut Steven Burrows pada tahun 2014 merupakan sebuah teks deskriptif tentang suatu topik khusus yang terdiri dari sebuah frasa atau paling banyak 100 kata [32].

2.2.10 Python

Python, seperti *JavaScript*, merupakan salah satu bahasa pemrograman yang banyak digunakan oleh pembangun perangkat lunak di bidang teknologi. Bedanya, *Python* lebih banyak digunakan untuk sains data dengan kapabilitasnya untuk pembelajaran mesin [33].

Python merupakan bahasa pemrograman yang banyak dipilih bagi praktisi pemrosesan bahasa alami, yang merupakan cabang dari kecerdasan buatan. *Python* memiliki *library* yang dapat mendukung pemrosesan bahasa alami yang mencakup banyak bahasa manusia seperti *Stanza* [34].

Aplikasi penilaian esai singkat yang akan dikembangkan penulis akan menggunakan model terlatih berbasis *Python* untuk melakukan proses penilaian dan perbandingan jawaban.

2.2.11 System Usability Scale

Dalam membangun sebuah sistem perangkat lunak, diperlukan adanya skala standar yang dapat digunakan untuk memastikan bahwa sistem yang dikembangkan dapat

digunakan oleh pengguna sebaik-baiknya. *System Usability Scale* (SUS) adalah kuesioner yang digunakan untuk mengukur tingkat kemudahan perangkat lunak saat digunakan [10]. SUS adalah alat psikometrik gratis yang digunakan diseluruh dunia dengan keabsahan dan keandalan tinggi [11].

Pada penelitian ini, penulis menggunakan SUS untuk mengukur tingkat kemudahan pengguna dalam menggunakan aplikasi penilaian esai singkat otomatis berbasis web melalui antarmuka yang akan disediakan untuk mengoperasikan aplikasi yang akan dibangun.

Kuesioner SUS akan diajukan kepada sejumlah pengguna relawan yang tertarik untuk menggunakan aplikasi penilaian esai otomatis berbasis web yang akan dikembangkan pada penelitian tugas akhir ini. Kemudian hasil dari kuesioner SUS ini akan berupa rata-rata skor dari semua responden dengan rentang nilai dari 0 s/d 100. Nilai prinsip SUS adalah untuk menyediakan skor referensi dari pandangan peserta terhadap kegunaan sebuah produk / sistem [35].

Nilai akan diperoleh dengan cara memisahkan skor pertanyaan dengan nomor ganjil dari nomor genap dengan kalkulasi pada rumus (1) berikut ini :

$$\bar{x} = \frac{\sum x}{n}$$

Rumus perhitungan SUS (1)

Keterangan:

\bar{x} : skor rata-rata

$\sum x$: jumlah skor SUS

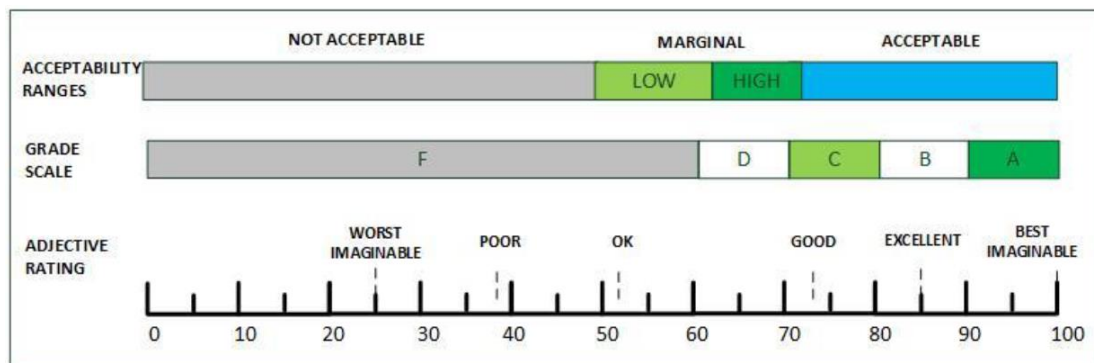
n : jumlah responden

Kemudian pada Tabel 2.4 berikut ini, dapat dilihat interpretasi dari setiap nilai yang diperoleh kalkulasi pengujian SUS.

Tabel 2.4 Standar hasil SUS [13]

Skor SUS	Nilai	Sifat Nilai
>80.3	A	Sangat Baik
74 – 80.3	B	Baik
68 - 74	C	Oke
51 – 68	D	Buruk
<51	F	Sangat Buruk

Melalui standar hasil SUS pada Tabel 2.4, aplikasi penilaian esai otomatis berbasis web yang dikembangkan sebaiknya meraih minimal nilai C untuk dapat dinyatakan berhasil dan dapat digunakan oleh banyak pengguna sebagaimana terdapat pada Gambar 2.2 berikut ini [13].



Gambar 2.2 SUS Score Percentile Rank [13]

2.2.12 Black Box Testing

Teknik pengujian selanjutnya yaitu teknik pengujian *Black Box*. Teknik pengujian aplikasi ini membahas aplikasi perangkat lunak dari sisi luar seperti tampilan, fungsionalitas, masukan, dan keluaran [14].

Aplikasi yang akan dikembangkan dari penelitian ini akan diuji menggunakan pengujian *Black Box*. Aplikasi akan diuji oleh sejumlah relawan untuk menguji masukan dan keluaran dari aplikasi ini dengan maksud untuk memastikan bahwa aplikasi yang dibangun bekerja semestinya.

2.2.13 Peramban Web

Dengan semakin bertambahnya kemampuan teknis internet, bertambah juga kebutuhan untuk sebuah aplikasi web yang mempengaruhi berubahnya cara sebuah halaman web untuk menampilkan sebuah data dan informasi [36].

JavaScript telah mencapai platform web secara luas dan menjadi bahasa pemrograman multifungsi dengan adanya ES6 yang diimplementasikan pada mesin peramban web sejak tahun 2017 [37].

2.2.14 Peluncuran Perangkat Lunak

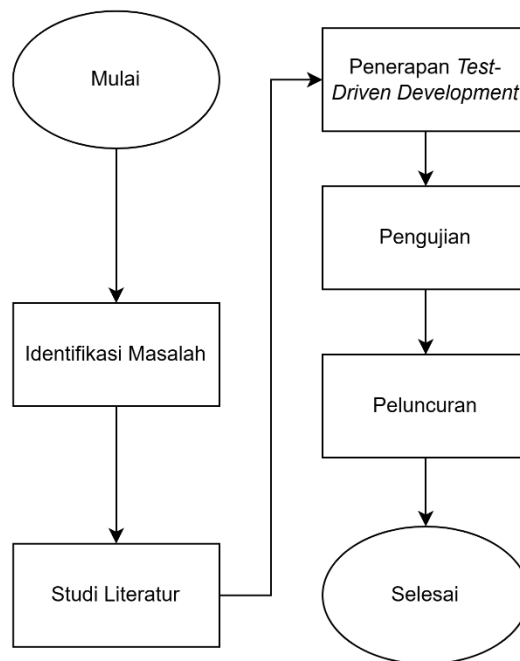
Peluncuran (*deployment*) perangkat lunak merupakan proses eksekusi dari sebuah perangkat lunak. Peluncuran adalah aktifitas dari sebuah perangkat lunak setelah proses pembangunan perangkat lunak [38]. Peluncuran dilakukan untuk mendapatkan timbal balik berupa perbaruan, konfigurasi ulang, adaptasi, penonaktifan, dan peluncuran ulang pada sebuah perangkat lunak [39].

BAB III

METODE PENELITIAN

3.1 Alur Penelitian

Penelitian tugas akhir ini akan dilakukan membangun aplikasi penilaian esai singkat otomatis berbasis web dengan metode pengembangan *Test-Driven Development* (TDD) dan diuji dengan metode dua metode pengujian, yakni *System Usability Scale* (SUS) dan *Black Box*. Alur penelitian dapat dilihat pada *flowchart* pada Gambar 3.1 dibawah ini.



Gambar 3.1 *Flowchart* Penelitian

3.2 Penjabaran Langkah Penelitian

Dapat diperhatikan melalui Gambar 3.1 sebelumnya bahwa pada penelitian ini memiliki 4 fase sebagaimana pada gambar tersebut. Penjelasan setiap fase penelitian ini adalah sebagai berikut:

3.2.1 Identifikasi Masalah

Pada tahap ini, peneliti akan menyediakan alasan dilakukannya penelitian dengan mengumpulkan permasalahan nyata yang memerlukan tindakan dan dapat diselesaikan dengan disiplin teknik informatika.

3.2.2 Studi Literatur

Dalam sebuah penelitian, diperlukan pengetahuan landasan secara teoritis terhadap permasalahan yang dihadapi dan ingin ditindak. Pengetahuan teoritis ini dapat diperoleh melalui jurnal penelitian terdahulu maupun buku yang berisi materi yang dibutuhkan. Jurnal penelitian terdahulu dan buku yang digunakan akan memiliki kesimpulan terkait satu atau lebih teori yang akan diperlukan pada penelitian ini. Kemudian teori yang sudah dikumpulkan dan dijadikan landasan akan menjadi arahan kemana kemungkinan penelitian yang dilakukan akan berakhir.

3.2.3 Penerapan *Test-Driven Development*

Penelitian ini akan dilakukan menggunakan metode siklus hidup perangkat lunak (SDLC) *Test-Driven Development* (TDD) dengan harapan menjadi solusi atas permasalahan yang ingin diselesaikan. Dengan ini, penelitian akan menekankan pengembangan melalui beberapa kebutuhan yang dijadikan pengujian atas dibuatnya fitur maupun fungsi aplikasi yang dikembangkan. Dengan metode ini, aplikasi yang dikembangkan dapat dipastikan kelayakannya dalam memenuhi semua fitur dan fungsi berdasarkan pengujian sebelumnya.

Telah diketahui sebelumnya bahwa TDD memiliki 3 fase utama yaitu membuat pengujian, membuat kode yang berhasil memenuhi pengujian tersebut, lalu *refactor* kode yang telah berhasil tersebut. Tahap lainnya seperti pemahaman terhadap fitur atau *bug* yang ingin diuji sebelum masuk ke tiga fase utama, dan tahap iterasi atau tindakan terhadap fitur atau *bug* lain yang ingin diuji dan diselesaikan apabila pengujian sebelumnya sudah berhasil tanpa error. Apabila seluruh fitur dan *bug* yang ingin ditindak sudah selesai dibangun, maka aplikasi siap memasuki tahap pengujian

3.2.4 Pengujian

Aplikasi yang dikembangkan pada penelitian tugas akhir ini akan diuji menggunakan *System Usability Scale* (SUS) dan *Black Box* untuk memastikan apakah aplikasi penilaian esai otomatis berbasis web yang dikembangkan dapat meningkatkan proses penilaian esai singkat menjadi lebih efisien. Pengujian dilakukan dengan melakukan perbandingan subjektif antara nilai yang diusulkan dari aplikasi dengan nilai yang diperiksa oleh pengajar suatu subjek terhadap jawaban esai dari pertanyaan esai yang diberikan pengajar tersebut. Apabila hasil nilai akhir dari aplikasi dan pribadi pengajar kurang lebih sama, dapat dipastikan bahwa aplikasi dapat bekerja dengan baik.

3.3 Alat dan Bahan Tugas Akhir

Alat dan bahan yang dibutuhkan untuk mendukung jalannya penelitian tugas akhir ini. Alat dan bahan yang dimaksud dapat dilihat sebagai berikut ini:

3.3.1 Alat

Alat yang mendukung dan dibutuhkan dalam pelaksanaan penelitian tugas akhir ini adalah berikut ini:

1. *Notebook* dengan spesifikasi minimum, sistem operasi Windows 8, *processor* Intel Core i3 2330M CPU @ 2,2 GHz, memori 4GB DDR3, grafis NVIDIA GeForce GT 610 (4GB), *hardisk* 500GB. Pada tugas akhir ini digunakan komputer dengan spesifikasi minimum sistem operasi Windows 10, *processor* AMD Ryzen 5 2500U @ 2 GHz, memori 16GB DDR4, grafis AMD Radeon Vega 8, dan *Solid-State Drive* 512GB.
2. *Smartphone* dengan spesifikasi tipe minimum, OS Android OS v4.1.2 (Jelly Bean), CPU Dual-core 800 MHz, GPU Mali-400, Internal 4 GB, 768 MB RAM. Pada tugas akhir ini digunakan OS Android 9.0 (Pie), CPU Octa-core 2.3GHz, GPU PowerVR GE8320, Internal 128 GB, 4GB RAM.
3. *Node.js* dengan versi minimal 16.16.0. Pada tugas akhir ini menggunakan *Node.js* dengan versi 18.12.1.

4. *Visual Studio Code* (VSCode) untuk menulis kode program menggunakan versi 1.74.2. Versi dari VSCode dapat berubah seiring adanya pembaruan dari pembangun aplikasi VSCode tersebut sewaktu-waktu.

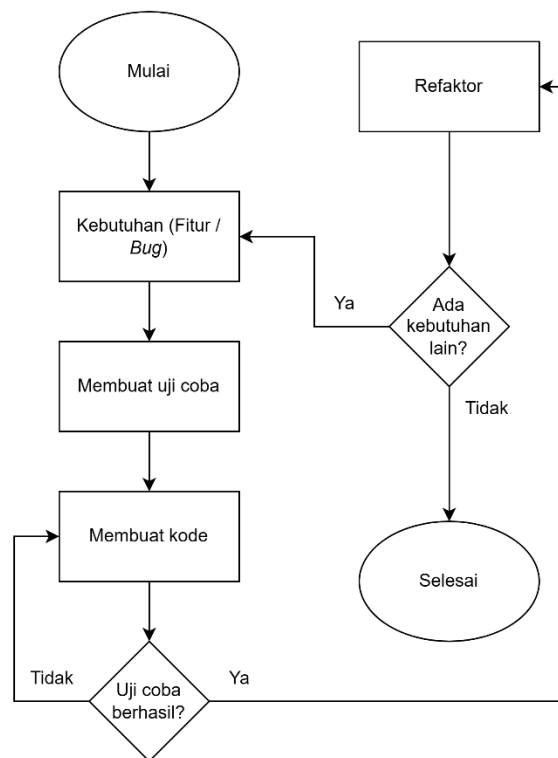
3.3.2 Bahan

Bahan yang digunakan / diperlukan untuk melakukan penelitian tugas akhir ini adalah sebagai berikut:

1. Model Terlatih untuk memproses penilaian esai singkat berbahasa Indonesia.
2. Dataset dari model terlatih penilaian esai singkat untuk bahasa Indonesia
3. Model Terlatih untuk memproses penilaian esai singkat berbahasa Inggris.
4. Dataset dari model terlatih penilaian esai singkat untuk bahasa Inggris

3.4 Metode Pengembangan

Dalam pengembangan aplikasi penilaian esai singkat otomatis berbasis web pada penelitian ini, akan digunakan salah satu metode pengembangan siklus hidup perangkat lunak *Agile*, khususnya *Test-Driven Development* (TDD). Alur pengembangan perangkat lunak menggunakan TDD pada penelitian ini dapat diperhatikan pada Gambar 3.2 dibawah ini.



Gambar 3.2 Alur Penelitian *Test-Driven Development*

Berdasarkan diagram alir yang dapat dilihat pada Gambar 3.2, TDD memiliki tahapan berupa kebutuhan, pembuatan uji coba, pembuatan kode, refaktor, dan peluncuran. Kemudian dibawah ini merupakan penjelasan dari masing-masing tahapan pada metode pengembangan TDD:

3.4.1 Kebutuhan

Tahap ini merupakan tahap pertama dari metode pengembangan TDD. Pada tahap ini, pengembang diharapkan dapat mengerti kebutuhan yang diminta untuk dapat dilakukan oleh perangkat lunak yang sedang dikembangkan.

3.4.1.1 Wawancara

Dalam penelusuran kebutuhan fungsi untuk aplikasi yang akan dibangun pada tugas akhir ini, dibutuhkan informasi dari calon pengguna yaitu pengajar yang sudah biasa melakukan penilaian esai pada pelajar. Penulis melakukan wawancara kepada tiga orang pengajar yaitu bapak Sukma Erawan, S.Kom sebagai guru SMA Negeri 68 Jakarta, bapak Alan Syahwandi, S.Pd.I sebagai guru SMP Negeri 3 Bogor, ibu Ajeng

Yeni Ratnasari, S.Pd selaku guru Pondok Pesantren Alfassalam. Tabel 3.1 berikut merupakan pertanyaan dan jawaban dari wawancara yang dilakukan.

Tabel 3.1 Hasil Wawancara

No	Pertanyaan	Jawaban
1	Apabila ada aplikasi penilai esai otomatis, apa fungsi/fitur yang ibu/bapak harapkan dari aplikasi tersebut?	Aplikasi harus bisa menentukan kata kunci dari jawaban yang mana bisa digunakan untuk penilaian. Aplikasi juga harus bisa menilai sekaligus banyak karena apabila tidak, akan tetap memakan energi seperti menilai esai biasa
2	Apakah ada permasalahan terkait jumlah pelajar dan pengajar?	Sebagai pembuat soal, biasanya perlu waktu lama untuk mengoreksi 300 jawaban hanya untuk satu soal, maka perlu alat bantu untuk mempercepat ini.
3	Apa ada fungsi/fitur lain yang kira-kira diperlukan?	Aplikasi harus bisa menilai jawaban dari singkatan, karena tidak seluruh jawaban akan menggunakan kata formal untuk setiap kalimatnya, khususnya untuk pelajar yang masih SD atau SMP.
4	Terkait efisiensi, apa yang sekiranya dibutuhkan untuk sebuah aplikasi penilaian esai otomatis?	Aplikasi harus bisa menilai banyak jawaban sekaligus. Kalau bisa terintegrasi kepada media yang sering digunakan untuk jawaban esai seperti Google Form
5	Apakah ada fitur terkait efektifitas aplikasi yang dibutuhkan?	Aplikasi harus bisa menampilkan perbandingan jawaban pengajar dan pelajar serta bobot nilai dari aplikasinya. Karena nilai untuk sebuah soal esai itu tidak satu dan nol, bisa saja nilainya setengah (50% benar).
6	Aplikasi sebaiknya dapat diakses hanya pada komputer atau bebas?	Untuk utamanya pada komputer, karena pengajar biasanya menilai dari komputer sekolah, namun sebaiknya dibuat juga untuk versi ponsel untuk mengantisipasi adanya kebutuhan khusus saat menggunakan aplikasi tersebut, seperti saat ada pelajar yang ketinggalan jawabannya untuk dinilai bersama yang lain.

3.4.1.2 User Story

Tahap ini dilakukan untuk mendapatkan fungsi dan fitur utama pada aplikasi yang dikembangkan pada kebutuhan fungsional dan non-fungsional. *User story* didapatkan dengan melakukan wawancara kepada calon pengguna sistem dengan mengetahui apa yang diinginkan dan mengapa mereka menginginkan hal tersebut. *User story* dari hasil wawancara yang dilakukan dapat dilihat pada Tabel 3.2 dibawah ini:

Tabel 3.2 Kumpulan User Story

No	Role	Objective	Alasan
1	Guru	Dapat melakukan penilaian secara masal	Aplikasi harus bisa dilakukan sekaligus agar cepat
2	Guru	Dapat melakukan penilaian untuk satu jawaban	Aplikasi bisa dipakai untuk menilai satu esai apabila ada kebutuhan khusus
3	Guru	Dapat menilai esai dengan pembobotan nilai dan melihat bobot nilai setiap jawaban	Nilai kebenaran esai tidak selalu 1 atau 0, karena ada esai yang mungkin hanya benar 50 persen.

Berdasarkan wawancara yang dilakukan, calon pengguna membutuhkan aplikasi yang bisa melakukan penilaian esai secara masal untuk sehingga pengajar tidak perlu melakukan aktifitas yang berulang (menyalin dan menempel jawaban) untuk setiap jawaban pelajar dan penilaian esai secara tunggal dengan memasukkan satu pasangan jawaban pengajar dan pelajar untuk kasus khusus (seperti esai yang telat dikumpul oleh pelajar). Aplikasi juga diharapkan dapat menampilkan bobot nilai jawaban pelajar karena menurut narasumber nilai pada esai tidak benar atau salah seluruhnya, melainkan ada bobot pada seberapa benar jawaban esai tersebut. Kebutuhan ini diuraikan menjadi kebutuhan teknis yang dijelaskan pada sub-bab 3.4.1.2 Kebutuhan Fungsional.

3.4.1.3 Kebutuhan Fungsional

Pada pengimplementasian aplikasi penilaian esai otomatis berbasis *website* ini, perlu beberapa kebutuhan fungsional yang harus dipenuhi untuk aplikasi dapat bekerja

semestinya. Kebutuhan fungsional aplikasi berbasis *website* dapat dilihat pada Tabel 3.3 berikut ini.

Tabel 3.3 Kebutuhan Fungsional

ID	Deskripsi
F0	Aplikasi <i>website</i> dapat menentukan algoritma penilaian yang digunakan berdasarkan permintaan pengguna melalui tombol
F1A	Aplikasi <i>website</i> dapat menerima masukkan dua buah jawaban berupa teks melalui <i>input form</i>
F1B	Aplikasi <i>website</i> dapat melakukan penilaian otomatis dari kedua masukkan pada fungsi dengan ID F1A
F2A	Aplikasi <i>website</i> dapat menerima masukkan dokumen dengan ekstensi xls, sebuah jawaban berupa teks, nama kolom nama pelajar berupa teks, dan nama kolom jawaban pelajar berupa teks.
F2B	Aplikasi <i>website</i> dapat melakukan penilaian berdasarkan masukkan pada fungsi dengan ID F2A

3.4.1.4 Kebutuhan Non-Fungsional

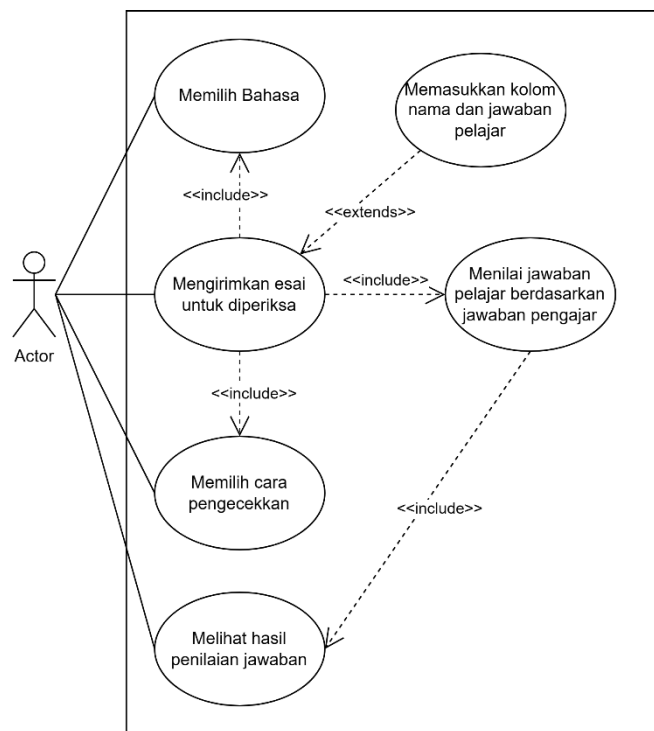
Untuk meningkatkan kenyamanan pengguna saat menggunakan aplikasi penilaian esai otomatis berbasis *website* ini, diperlukan beberapa kebutuhan non-fungsional yang ingin dikembangkan. Kebutuhan ini dapat meningkatkan retensi pengguna agar lebih nyaman saat menggunakan aplikasi ini. Pada Tabel 3.4 dibawah ini dapat dilihat kebutuhan non-fungsional yang akan dipenuhi.

Tabel 3.4 Kebutuhan Non-Fungsional

ID	Deskripsi
NF1	Aplikasi <i>website</i> dapat digunakan pada berbagai perangkat, komputer dan mobile
NF2	Aplikasi <i>website</i> dapat dijalankan pada peramban web yang mendukung ES6
NF3	Aplikasi <i>website</i> memiliki tampilan yang menyesuaikan dengan ukuran layar perangkat yang digunakan (responsif)

3.4.1.5 Use Case Diagram

Use Case Diagram adalah diagram yang digunakan untuk menggambarkan sebuah interaksi yang dapat terjadi antara pengguna dan sistem. Pada Gambar 3.3 dibawah ini dapat diperhatikan interaksi pengguna pada sistem yang terjadi pada sistem yang hendak dibangun.

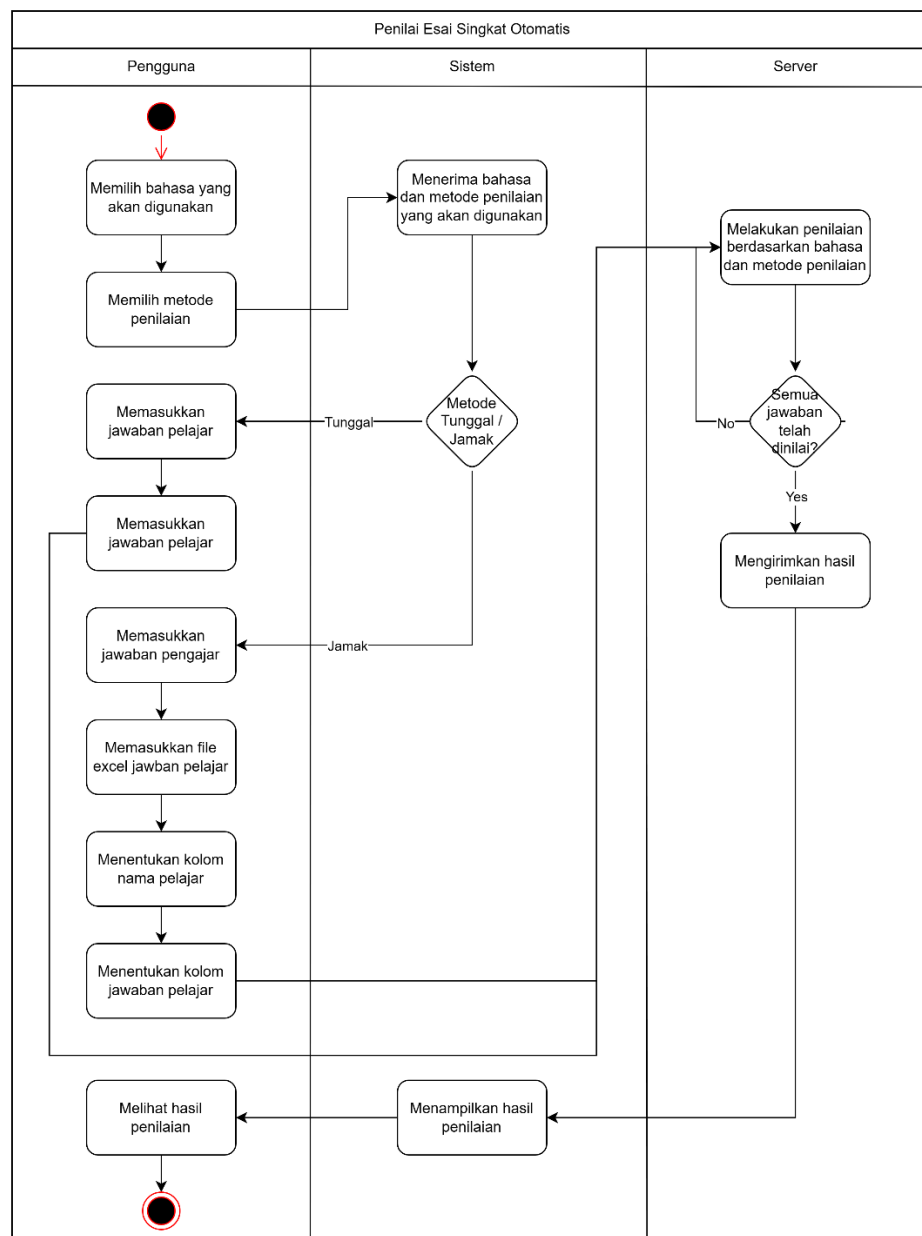


Gambar 3.3 *Use Case Diagram* Aplikasi

Pada Gambar 3.3 tersebut, aktor merupakan pengajar yang ingin melakukan pengecekan jawaban esai milik pelajar. Pengguna dapat melihat hasil jawaban yang diperiksa dengan mengirimkan esai yang ingin diperiksa, kemudian memilih bahasa dan teknik pengecekan yang diinginkan.

3.4.1.6 Activity Diagram

Activity diagram digunakan untuk menggambarkan alur pekerjaan dari aktivitas pada suatu sistem. Pada gambar 3.4 berikut ini dapat dilihat *activity diagram* dari aplikasi yang akan dibangun pada penelitian ini.

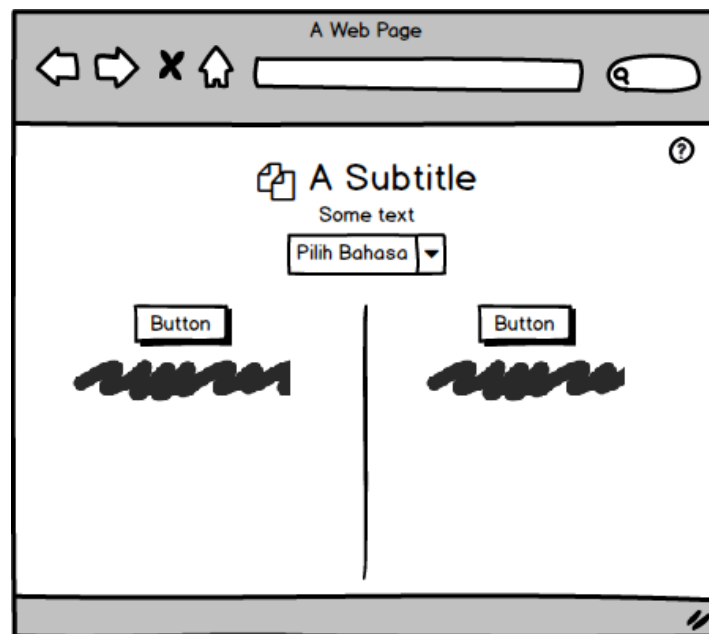


Gambar 3.4 Activity Diagram Aplikasi

Pada Gambar 3.4 tersebut dapat diperhatikan bahwa aplikasi dimulai dengan pengguna memilih bahasa dan metode penilaian yang akan digunakan. Kemudian berdasarkan pilihan tersebut, sistem akan menentukan masukkan yang dapat dilakukan oleh pengguna. Apabila semua masukkan telah diisi, kemudian sistem akan meminta *server* melakukan penilaian yang diminta oleh pengguna berdasarkan bahasa dan metode penilaian. Setelah *server* selesai melakukan penilaian, *server* akan mengirimkan hasil penilaian untuk ditampilkan pada aplikasi agar dapat dilihat oleh pengguna.

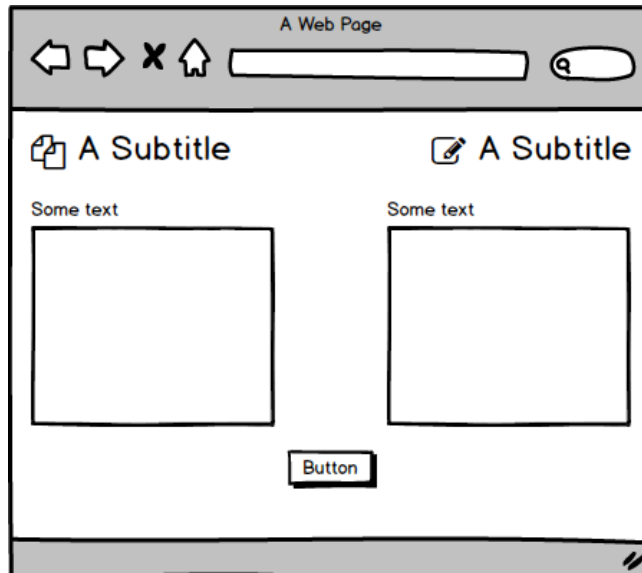
3.4.1.7 Mockup

Untuk pengguna dapat menggunakan aplikasi web yang akan dibangun, perlu dibuat sebuah desain antarmuka agar dapat mengoperasikan aplikasinya. Perancangan antarmuka ini dilakukan pada tahap kebutuhan sebagai perencanaan terhadap gambaran aplikasi yang akan dikembangkan. Rancangan antarmuka aplikasi dapat dilihat sebagai berikut.



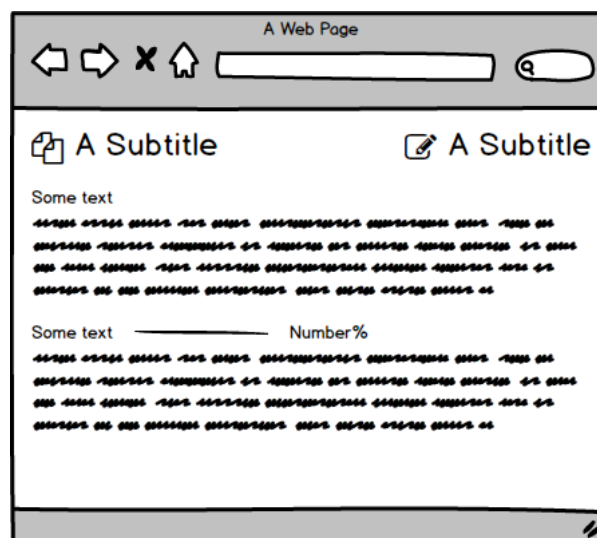
Gambar 3.5 *Mockup* Halaman Home

Saat pengguna mengakses halaman *website*, pengguna akan ditunjukkan halaman beranda seperti pada Gambar 3.5 dimana pengguna dapat memilih bahasa yang akan digunakan, dan metode penilaian yang akan digunakan dengan sebelah kiri merupakan metode tunggal dan sebelah kanan merupakan metode jamak. Pengguna juga dapat mengakses halaman bantuan dengan menekan logo tanda tanya yang ada pada pojok kanan atas halaman beranda web.



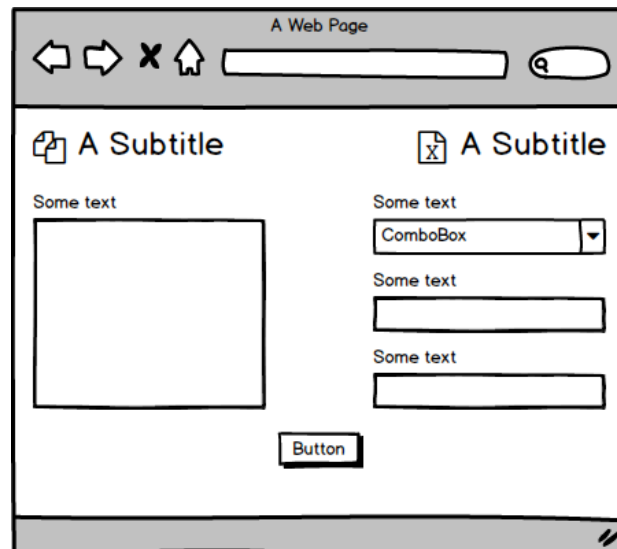
Gambar 3.6 Mockup Metode Penilaian Tunggal

Pada Gambar 3.6 merupakan tampilan yang akan dilihat pengguna apabila memilih metode penilaian tunggal. Pada halaman ini, pengguna sebagai pengajar dapat mengisi jawaban pengajar di sebelah kiri dan jawaban pelajar untuk dinilai di sebelah kanan.



Gambar 3.7 Mockup Hasil Metode Penilaian Tunggal

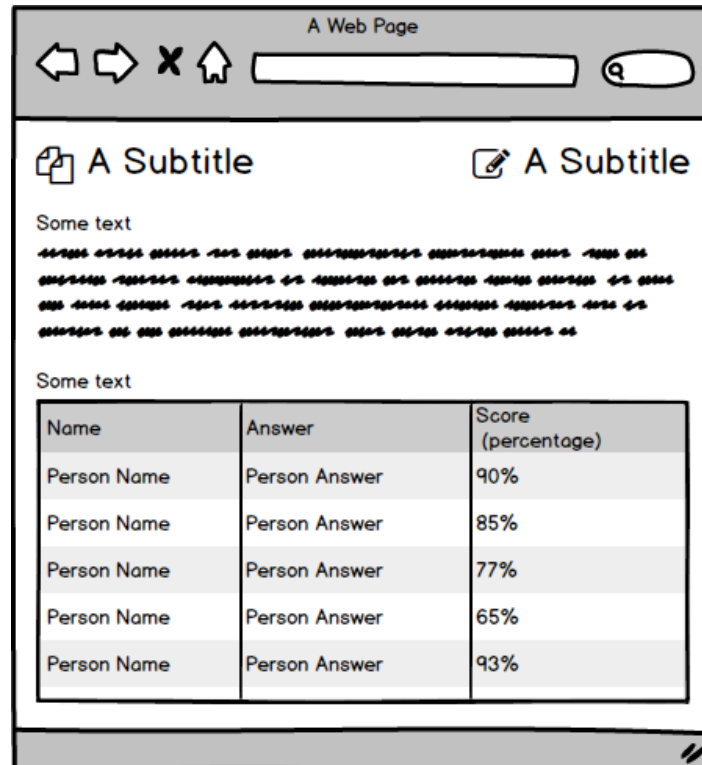
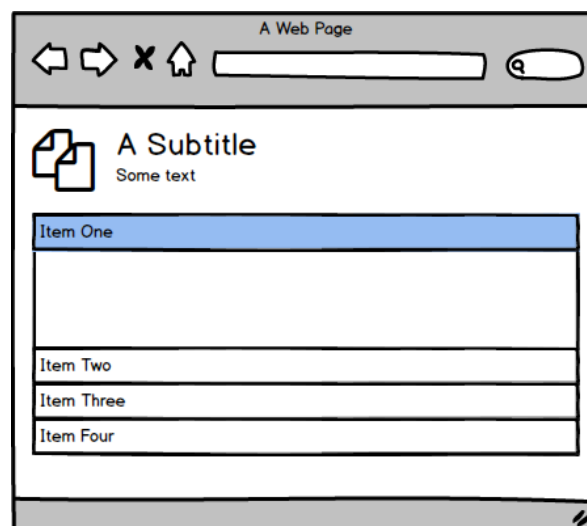
Setelah memproses penilaian otomatis, aplikasi akan mengarahkan pengguna ke halaman hasil penilaian dengan metode penilaian 1 sebagaimana ditunjukkan pada Gambar 3.7 dimana pengguna dapat melihat jawaban pengajar dan jawaban pelajar dengan persentase hasil jawabannya.



Gambar 3.8 *Mockup* Metode Penilaian Jamak

Pada Gambar 3.8 diatas merupakan halaman yang akan ditampilkan apabila pengguna memilih metode penilaian secara jamak. Pada halaman ini, pengguna dapat mengisi jawaban pengajar dan memasukkan dokumen *excel* (xls) pada aplikasi dan menentukan judul kolom berisi nama dan jawaban pelajar untuk diperiksa terhadap jawaban pengajar.

Gambar 3.9 merupakan halaman yang akan ditampilkan kepada pengguna untuk melihat hasil penilaian otomatis oleh aplikasi. Aplikasi akan menampilkan halaman yang akan menunjukkan jawaban pengajar pada bagian atas dan tabel berisi nama, jawaban, dan persentase nilai pelajar.

Gambar 3.9 *Mockup* Hasil Metode Penilaian JamakGambar 3.10 *Mockup* Halaman Bantuan

Apabila pada halaman beranda seperti pada Gambar 3.5 pengguna memutuskan untuk menekan logo tanda tanya, maka pengguna akan diarahkan ke halaman bantuan seperti pada Gambar 3.10. Pada halaman ini, pengguna dapat melihat cara menggunakan aplikasi penilaian esai otomatis.

3.4.2 Pembuatan Uji Coba (Test Writing)

Pada tahap ini, pengembang membuat uji coba lengkap dengan ekspektasi berupa hasil dari pengujian tersebut. Pengembang membuat uji coba berdasarkan satu kebutuhan yang telah didapat dari tahap sebelumnya.

Aplikasi yang dikembangkan akan memiliki 4 skenario uji coba yang harus berhasil setiapnya dengan ekspektasi pada Tabel 3.5 berikut ini:

Tabel 3.5 Skenario Uji Coba

Skenario	Masukkan Jawaban Pengajar	Masukkan Jawaban Pelajar	Ekspektasi Keluaran
SC1	Teks untuk uji coba jawaban tunggal “To sumulate the behaviour of portions of the desired software product.”	Teks untuk uji coba tunggal “Simulating the behavior of only a portion of the desired software product.”	71%
SC2	Teks untuk uji coba masal “To sumulate the behaviour of portions of the desired software product.”	Dokumen CSV berisi jawaban pelajar (“dataset.csv”)	Persentase Hasil Penilaian
SC3	Teks untuk uji coba tunggal “Fungsi karbohidrat adalah sebagai pemasok energi, dapat memperlancar proses pada pencernaan, memberikan efek kenyang dengan kandungan selulosa-nya dan penyeimbang asam dan basa dalam tubuh “	Teks untuk uji coba tunggal “Karbohidrat mempunyai peran penting untuk proses metabolisme atau proses yang berfungsi sebagai penyeimbang asam dan basa dalam tubuh. Karbohidrat dapat mencegah terjadinya ketidaksempurnaan proses oksidasi lemak. Fungsi karbohidrat yang utama adalah sebagai pemasok energi. Makanan berkarbohidrat dan serat tinggi dapat membantu memperlancar proses pada pencernaan. ”	83%

3.4.3 Pembuatan Kode (Coding)

Pada tahap ini, pengembang akan membuat fungsi pada perangkat lunak yang dikembangkan yang dapat memenuhi uji coba pada tahapan sebelumnya. Apabila ekspektasi hasil uji coba tidak dapat terpenuhi, maka kode dianggap gagal dan harus dibuat ulang hingga berhasil meraih ekspektasi yang diharapkan dari uji coba tersebut.

3.4.4 Refaktor (Refactor)

Saat kode untuk suatu fungsi berdasarkan suatu uji coba sudah berhasil dibuat, maka akan dilakukan pembersihan kode. Pembersihan disini berarti kode akan ditulis sedemikian rupa agar tidak bersifat ambigu bagi pengembang lain yang mungkin akan melanjutkan pengembangan perangkat lunak yang sedang dikembangkan saat ini di masa depan. Setelah melakukan refaktorisasi kode, pengembang dapat kembali ke tahap kebutuhan dan mengulang seluruh tahapan ini dengan kebutuhan yang berbeda hingga semua kebutuhan berhasil dibuat.

3.5 Rancangan Pengujian

Setelah aplikasi yang dikembangkan pada penelitian ini diluncurkan, akan dilakukan pengujian untuk memastikan bahwa aplikasi yang telah dikembangkan layak sebelum dapat digunakan sepenuhnya oleh publik. Pengujian yang dimaksud adalah sebagai berikut:

3.5.1 Responsiveness

Aplikasi berbasis website yang dibangun harus bersifat *responsive*, yaitu kemampuan *website* yang dibangun untuk beradaptasi dengan setiap resolusi layar yang digunakan pengguna [18] agar pengguna website selalu nyaman dengan tampilan *websitenya*.

Pengujian terkait *responsiveness* ini dilakukan dengan cara mengakses *website* menggunakan semua ukuran layar yang memungkinkan untuk mencari kemungkinan tampilan yang tumpang-tindih yang dapat mengurangi kenyamanan pengguna *website* saat melihat tampilan *website*.

3.5.2 System Usability Scale

Seperti yang telah dilansir pada Bab 2, aplikasi yang dihasilkan penelitian ini akan diuji menggunakan SUS pada sekelompok orang untuk mendapatkan hasil kegunaan aplikasi dari penelitian yang dilakukan. Dengan itu, pada Tabel 3.6 berikut adalah daftar kuesioner yang akan diajukan kepada pengguna setelah menggunakan aplikasi yang bersangkutan.

Tabel 3.6 Kuesioner *System Usability Scale*

No	Pertanyaan
1	Saya berpikir akan menggunakan aplikasi penilaian esai otomatis berbasis web ini lagi
2	Saya merasa aplikasi penilaian esai otomatis berbasis web ini sulit digunakan
3	Saya merasa aplikasi penilaian esai otomatis berbasis web ini mudah digunakan
4	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan aplikasi penilaian esai otomatis berbasis web ini
5	Saya merasa fitur-fitur aplikasi penilaian esai otomatis berbasis web ini berjalan dengan semestinya
6	Saya merasa banyak hal yang tidak konsisten / tidak serasi pada aplikasi penilaian esai otomatis berbasis web ini
7	Saya merasa orang lain akan memahami cara menggunakan aplikasi penilaian esai otomatis berbasis web ini dengan cepat
8	Saya merasa aplikasi penilaian esai otomatis berbasis web ini membingungkan
9	Saya merasa tidak ada hambatan dalam menggunakan aplikasi penilaian esai otomatis berbasis web ini
10	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan aplikasi penilaian esai otomatis berbasis web ini

3.5.3 Black Box

Pada waktu yang bersamaan, akan dilakukan pengujian Black Box dari sejumlah pengguna sukarela untuk memastikan bahwa aplikasi yang dikembangkan bekerja sesuai dengan yang diharapkan. Pada tahap ini, penulis mungkin akan mendapatkan kecacatan program (*bug*), namun dengan ini penulis dapat memperbaiki aplikasi terkait dan menyempurnakannya. Daftar pengujian Black Box dapat dilihat secara rinci pada Tabel 3.7 dibawah ini.

Tabel 3.7 Daftar pengujian Black Box

PARENT ID	ID	Pengujian	Ekspektasi
F1A	F1A-1	Aplikasi web dapat menerima masukkan teks sebagai jawaban pengajar	Aplikasi web berhasil menyimpan teks sebagai jawaban pengajar
	F1A-2	Aplikasi web dapat menerima masukkan teks sebagai jawaban pelajar	Aplikasi web berhasil menerima teks sebagai jawaban pelajar
F1B	F1B-1	Aplikasi web dapat menerima masukkan dari fungsi F1A untuk penilaian otomatis tunggal bahasa Inggris	Aplikasi web berhasil menerima masukkan dari fungsi F1A untuk digunakan pada penilaian otomatis tunggal bahasa Inggris
	F1B-2	Aplikasi web dapat menerima masukka dari fungsi F1A untuk penilaian otomatis tunggal bahasa Indonesia	Aplikasi berhasil menerima masukkan dari fungsi F1A untuk digunakan pada penilaian otomatis tunggal bahasa Indonesia
F0	F0	Aplikasi dapat menggunakan model yang tepat untuk melakukan penilaian berdasarkan bahasa yang dipilih pengguna	Aplikasi web berhasil menerima permintaan bahasa yang digunakan untuk penilaian dan menggunakan bahasa tersebut untuk penilaian.
F2A	F2A-1	Aplikasi web dapat menerima masukkan berupa dokumen berbentuk .xls sebagai kumpulan jawaban pelajar	Aplikasi web berhasil menyimpan dokumen .xls sebagai kumpulan jawaban pelajar
	F2A-2	Aplikasi web dapat menerima masukkan teks sebagai kolom nama pada kumpulan jawaban pelajar	Aplikasi web berhasil menyimpan nama kolom yang berisi nama pelajar pada dokumen .xls
	F2A-3	Aplikasi web dapat menerima masukkan teks sebagai kolom jawaban pada kumpulan jawaban pelajar	Aplikasi web berhasil menyimpan nama kolom yang berisi jawaban pelajar pada dokumen .xls
	F2A-4	Aplikasi web dapat mendapatkan pasangan nama dan jawaban pada kumpulan jawaban pelajar sebagaimana ditentukan pada pengujian F2A-2 dan F2A-3	Aplikasi web berhasil menentukan pasangan nama dan jawaban pada dokumen .xls untuk proses penilaian masal
F2B	F2B-1	Aplikasi web dapat membuat dokumen .csv berdasarkan pengujian F1A-1, dan F1A-2 atau F2A-1 agar cocok dengan	Aplikasi web berhasil membuat dokumen .csv dari dokumen .xls, nama, dan jawaban pelajar untuk

PARENT ID	ID	Pengujian	Ekspektasi
		syarat penilaian pada algoritma penilaian otomatis bahasa Inggris	diproses pada model penilaian otomatis bahasa Inggris
	F2B-3	Algoritma penilaian otomatis bahasa Inggris dapat membuat dokumen .json setelah melakukan penilaian otomatis	Algoritma penilaian berhasil membuat dokumen .json sebagai hasil penilaian otomatis model bahasa Inggris
	F2B-5	Aplikasi dapat menampilkan hasil penilaian bahasa Inggris berdasarkan data .json yang dihasilkan oleh algoritma penilaian bahasa Inggris pada halaman web	Aplikasi web berhasil menggunakan dokumen .json yang telah dihasilkan penilaian otomatis bahasa Inggris sebelumnya untuk ditampilkan pada halaman web
	F2B-2	Aplikasi web dapat membuat dokumen .json berdasarkan pengujian F1A-1, dan F1A-2 atau F2A-1 agar cocok dengan syarat penilaian pada algoritma penilaian otomatis bahasa Indonesia	Aplikasi web berhasil membuat dokumen .json setelah menentukan pasangan nama dan jawaban pada dokumen .xls
	F2B-4	Algoritma penilaian otomatis bahasa Indonesia dapat membuat dokumen .json setelah melakukan penilaian otomatis	Algoritma penilaian otomatis bahasa Indonesia berhasil membuat dokumen .json untuk ditampilkan pada halaman web
F2	F2	Aplikasi penilaian otomatis dapat menampilkan hasil penilaian ke halaman web berdasarkan dokumen .json yang dihasilkan oleh algoritma penilaian otomatis.	Dokumen .json yang merupakan hasil penilaian otomatis dari algoritma bahasa Indonesia maupun bahasa Inggris berhasil ditampilkan pada halaman web
NF1	NF1-1	Aplikasi dapat dijalankan pada ponsel	Aplikasi web berhasil diakses dan digunakan pada ponsel
	NF1-2	Aplikasi dapat dijalankan pada komputer	Aplikasi web berhasil diakses dan digunakan pada komputer
NF2	NF2	Aplikasi dapat dijalankan pada peramban web yang mendukung ES6	Aplikasi web berhasil diakses dan dijalankan pada peramban web yang telah mendukung ES6
NF3	NF3	Aplikasi tidak memiliki tampilan tumpang-tindih pada resolusi	Aplikasi web berhasil beradaptasi pada setiap

PARENT ID	ID	Pengujian	Ekspektasi
		yang berbeda	resolusi dimana perlu ditampilkan tanpa adanya komponen yang tumpang-tindih
SC1	SC1	Hasil penilaian pada aplikasi untuk satu jawaban esai bahasa Inggris harus sama dengan hasil penilaian pada algoritma	Aplikasi web berhasil menunjukkan hasil penilaian dengan skor yang sama dengan hasil penilaian dengan algoritma bahasa Inggris untuk penilaian satu jawaban
SC2	SC2	Hasil penilaian pada aplikasi untuk banyak jawaban esai bahasa Inggris harus sama dengan hasil penilaian pada algoritma	Aplikasi web berhasil menunjukkan hasil penilaian dengan skor yang sama dengan hasil penilaian dengan algoritma bahasa Inggris untuk penilaian masal
SC3	SC3	Hasil penilaian pada aplikasi untuk satu jawaban esai bahasa Indonesia harus sama dengan hasil penilaian pada algoritma	Aplikasi web berhasil menunjukkan hasil penilaian dengan skor yang sama dengan hasil penilaian dengan algoritma bahasa Indonesia untuk penilaian satu jawaban

3.6 Peluncuran

Apabila semua kebutuhan terhadap perangkat lunak yang dibangun sudah dibuat dan telah teruji bekerja dengan baik, maka akan dilakukan tahap peluncuran, yaitu kondisi dimana perangkat lunak yang dikembangkan sudah siap digunakan secara umum.

DAFTAR PUSTAKA

- [1] U. Nations, "Population." <https://www.un.org/en/global-issues/population> (accessed Dec. 25, 2022).
- [2] B. P. Statistik, "Jumlah Penduduk Pertengahan Tahun (Ribuan Jiwa), 2020-2022." <https://www.bps.go.id/indikator/12/1975/1/jumlah-penduduk-pertengahan-tahun.html> (accessed Dec. 26, 2022).
- [3] B. P. Statistik, "Jumlah Penduduk Menurut Kelompok Umur dan Jenis Kelamin, 2021." https://www.bps.go.id/indikator/indikator/view_data_pub/0000/api_pub/YW40a21pdTU1cnJxOGt6dm43ZEdoZz09/da_03/1 (accessed Dec. 26, 2022).
- [4] F. Oktariano and H. Hastuti, "Buku Panduan penulisan Esai Berdasarkan Analisis Historical Thinking," vol. 2, no. 4, 2020.
- [5] E. Amorim, M. Cançado, and A. Veloso, "Automated Essay Scoring in the Presence of Biased Ratings," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana, 2018, pp. 229–237. doi: 10.18653/v1/N18-1021.
- [6] D. Fortunato and J. Bernardino, "Progressive web apps: An alternative to the native mobile Apps," in *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*, Cáceres, Jun. 2018, pp. 1–6. doi: 10.23919/CISTI.2018.8399228.
- [7] M. M. Moe and University of Computer Studies, Hpa-An, Kayin State, Myanmar, "Comparative Study of Test-Driven Development TDD, Behavior-Driven Development BDD and Acceptance Test-Driven Development ATDD," *Int. J. Trend Sci. Res. Dev.*, vol. Volume-3, no. Issue-4, pp. 231–234, Jun. 2019, doi: 10.31142/ijtsrd23698.
- [8] B. George and L. Williams, "A structured experiment of test-driven development," *Inf. Softw. Technol.*, vol. 46, no. 5, pp. 337–342, Apr. 2004, doi: 10.1016/j.infsof.2003.09.011.
- [9] J. B. Ibarra *et al.*, "Development of the Low Cost Classroom Response System Using Test-Driven Development Approach and Analysis of the Adaptive Capability of Students Using Sequential Minimal Optimization Algorithm," in *2019 IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA)*, Tokyo, Japan, Apr. 2019, pp. 689–693. doi: 10.1109/IEA.2019.8714889.
- [10] D. Pal and V. Vanijja, "Perceived usability evaluation of Microsoft Teams as an online learning platform during COVID-19 using system usability scale and technology acceptance model in India," *Child. Youth Serv. Rev.*, vol. 119, p. 105535, Dec. 2020, doi: 10.1016/j.childyouth.2020.105535.
- [11] P. Vlachogianni and N. Tselios, "Perceived usability evaluation of educational technology using the System Usability Scale (SUS): A systematic review," *J. Res. Technol. Educ.*, vol. 54, no. 3, pp. 392–409, May 2022, doi: 10.1080/15391523.2020.1867938.
- [12] J. R. Lewis, "Psychometric Evaluation of the PSSUQ Using Data from Five Years of Usability Studies," *Int. J. Hum.-Comput. Interact.*, vol. 14, no. 3–4, pp. 463–488, Sep. 2002, doi: 10.1080/10447318.2002.9669130.

- [13] U. Ependi, T. B. Kurniawan, and F. Panjaitan, "SYSTEM USABILITY SCALE VS HEURISTIC EVALUATION: A REVIEW," *Simetris J. Tek. Mesin Elektro Dan Ilmu Komput.*, vol. 10, no. 1, pp. 65–74, Apr. 2019, doi: 10.24176/simet.v10i1.2725.
- [14] T. Hidayat and M. Muttaqin, "Pengujian Sistem Informasi Pendaftaran dan Pembayaran Wisuda Online menggunakan Black Box Testing dengan Metode Equivalence Partitioning dan Boundary Value Analysis," vol. 6, 2018.
- [15] xinyu chang and jing Li, "Improvement of Excel data processing function based on Spring MVC framework," in *Third International Conference on Computer Science and Communication Technology (ICCSCT 2022)*, Beijing, China, Dec. 2022, p. 25. doi: 10.1117/12.2661778.
- [16] I. Glantz and H. Hurtig, "Express.js and Ktor web server performance A comparative study".
- [17] D. Smilkov *et al.*, "TensorFlow.js: Machine Learning for the Web and Beyond".
- [18] S. L. Ingólfssdóttir, H. Loftsson, J. F. Daðason, and K. Bjarnadóttir, "Nefnir: A high accuracy lemmatizer for Icelandic." arXiv, Jul. 27, 2019. Accessed: Jan. 05, 2023. [Online]. Available: <http://arxiv.org/abs/1907.11907>
- [19] B. Jose and S. Abraham, "Performance analysis of NoSQL and relational databases with MongoDB and MySQL," *Mater. Today Proc.*, vol. 24, pp. 2036–2043, 2020, doi: 10.1016/j.matpr.2020.03.634.
- [20] G. Steinfeld, "5 steps of test-driven development," *IBM Developer*, Feb. 06, 2020. Accessed: Jan. 30, 2023. [Online]. Available: <https://developer.ibm.com/articles/5-steps-of-test-driven-development/>
- [21] Institut Teknologi Sepuluh Nopember *et al.*, "A Different Approach on Automated Use Case Diagram Semantic Assessment," *Int. J. Intell. Eng. Syst.*, vol. 14, no. 1, pp. 496–505, Feb. 2021, doi: 10.22266/ijies2021.0228.46.
- [22] Dept. of Computer Science & Engineering, BITM, VTU, Ballari, India., Dr. R. N. Kulkarni, C. K. Srinivasa, and Dept. of Computer Science & Engineering, BITM, VTU, Ballari, India., "Novel approach to transform UML Sequence diagram to Activity diagram," *J. Univ. Shanghai Sci. Technol.*, vol. 23, no. 07, pp. 1247–1255, Jul. 2021, doi: 10.51201/JUSST/21/07300.
- [23] M. Shirole and R. Kumar, "Constrained permutation-based test scenario generation from concurrent activity diagrams," *Innov. Syst. Softw. Eng.*, vol. 17, no. 4, pp. 343–353, Dec. 2021, doi: 10.1007/s11334-021-00389-4.
- [24] A. Firdaus, S. Widodo, A. Sutrisman, S. G. F. Nasution, and R. Mardiana, "RANCANG BANGUN SISTEM INFORMASI PERPUSTAKAAN MENGGUNAKAN WEB SERVICE PADA JURUSAN TEKNIK KOMPUTER POLSRI," vol. 5, 2019.
- [25] A. Ismail and K. S. Kuppasamy, "Web accessibility investigation and identification of major issues of higher education websites with statistical measures: A case study of college websites," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 3, pp. 901–911, Mar. 2022, doi: 10.1016/j.jksuci.2019.03.011.
- [26] R. S. Malik, J. Patra, and M. Pradel, "NL2Type: Inferring JavaScript Function Types from Natural Language Information," in *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, Montreal, QC, Canada, May 2019, pp. 304–315. doi: 10.1109/ICSE.2019.00045.

- [27] M. Shcherbakov, M. Balliu, and C.-A. Staicu, “Silent Spring: Prototype Pollution Leads to Remote Code Execution in Node.js”.
- [28] H. Brar, T. Kaur, and Y. Rajoria, “The Better Comparison between PHP, Python-web & Node.js,” *web ...*
- [29] A. Romanelli, S. Serbout, and C. Pautasso, “ExpressO: From Express.js implementation code to OpenAPI interface descriptions”.
- [30] G. Langdale and D. Lemire, “Parsing gigabytes of JSON per second,” *VLDB J.*, vol. 28, no. 6, pp. 941–960, Dec. 2019, doi: 10.1007/s00778-019-00578-5.
- [31] S. Hamza, M. Sarosa, and P. B. Santoso, “Sistem Koreksi Soal Essay Otomatis Dengan Menggunakan Metode Rabin Karp,” vol. 7, no. 2, 2013.
- [32] S. Burrows, I. Gurevych, and B. Stein, “The Eras and Trends of Automatic Short Answer Grading,” *Int. J. Artif. Intell. Educ.*, vol. 25, no. 1, pp. 60–117, Mar. 2015, doi: 10.1007/s40593-014-0026-8.
- [33] J. Hao and T. K. Ho, “Machine Learning Made Easy: A Review of *Scikit-learn* Package in Python Programming Language,” *J. Educ. Behav. Stat.*, vol. 44, no. 3, pp. 348–361, Jun. 2019, doi: 10.3102/1076998619832248.
- [34] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning, “Stanza: A Python Natural Language Processing Toolkit for Many Human Languages.” arXiv, Apr. 23, 2020. Accessed: Jan. 09, 2023. [Online]. Available: <http://arxiv.org/abs/2003.07082>
- [35] A. Bangor, P. T. Kortum, and J. T. Miller, “An Empirical Evaluation of the System Usability Scale,” *Int. J. Hum.-Comput. Interact.*, vol. 24, no. 6, pp. 574–594, Jul. 2008, doi: 10.1080/10447310802205776.
- [36] A. Popov, J. Bilokin, T. Solianyk, and K. Vasylchenko, “Development of the system to provide cross-browser compatibility of web application,” in *2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Kyiv, Ukraine, May 2018, pp. 117–122. doi: 10.1109/DESSERT.2018.8409111.
- [37] K. Paltoglou, V. E. Zafeiris, N. A. Diamantidis, and E. A. Giakoumakis, “Automated refactoring of legacy JavaScript code to ES6 modules,” *J. Syst. Softw.*, vol. 181, p. 111049, Nov. 2021, doi: 10.1016/j.jss.2021.111049.
- [38] A. Dearle, “Software Deployment, Past, Present and Future,” in *Future of Software Engineering (FOSE '07)*, Minneapolis, MN, May 2007, pp. 269–284. doi: 10.1109/FOSE.2007.20.
- [39] R. S. Hall, D. Heimbigner, and A. L. Wolf, “A Cooperative Approach to Support Software Deployment Using the Software Dock”.