**RANCANG BANGUN APLIKASI PENILAIAN ESAI SINGKAT BERBAHASA INDONESIA DAN INGGRIS MENGGUNAKAN 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**

# LEMBAR PENGESAHAN

Tugas Akhir dengan judul “Tulis Judul Disini” adalah benar dibuat oleh saya sendiri dan belum pernah dibuat dan diserahkan sebelumnya, baik sebagian ataupun seluruhnya, baik oleh saya ataupun orang lain, baik di Institut Teknologi Sumatera maupun di institusi pendidikan lainnya.

|  |  |
| --- | --- |
| Lampung Selatan, DD-MM-YYYY  Penulis, | PHOTO BERWARNA |
| Geizka Rozilia Ruicosta  NIM. 119140114 |  |

Diperiksa dan disetujui oleh,

Pembimbing Tanda Tangan

1. Nama Pembimbing 1 + Gelar

NIP. XXXXXX ………………

2. Nama Pembimbing 2 + Gelar

NIP. XXXXXX ………………

Penguji Tanda Tangan

1. Nama Penguji 1 + Gelar

NIP. XXXXXXXXXXXX ………………

2. Nama Penguji 2+ Gelar

NIP. XXXXXXXXXXXX ………………

Disahkan oleh,

Koordinator Program Studi Teknik Informatika

Jurusan Teknologi, Produksi dan Industri

Institut Teknologi Sumatera

Nama Kaprodi + Gelar

NIP. XXXXXXXXXXXXXX

# HALAMAN PERNYATAAN ORISINALITAS

**Tugas Akhir dengan judul “TULIS JUDUL DISINI” adalah karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan benar.**

**Nama : Geizka Rozilia Ruicosta**

**NIM : 11940114**

**Tanda Tangan : ………………………….**

**Tanggal : ………………………….**

# HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai civitas akademik Institut Teknologi Sumatera, saya yang bertanda tangan di bawah ini:

Nama : Geizka Rozilia Ruicosta

NIM : 119140114

Program Studi : Teknik Informatika

Jurusan : Jurusan Teknologi, Produksi dan Industri

Jenis Karya : Tugas Akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Institut Teknologi Sumatera **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty Free Right*)** atas karya ilmiah saya yang berjudul:

**TULIS JUDUL DISINI**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini, Institut Teknologi Sumatera berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di Lampung Selatan

Pada tanggal DD Bulan YYYY

Yang menyatakan,

Geizka Rozilia Ruicosta

**KATA PENGANTAR**

Puji syukur kehadirat Allah SWT atas limpahan rahmat, karunia, serta petunjuk-Nya sehingga penyusunan tugas akhir ini telah terselesaikan dengan baik. Dalam penyusunan tugas akhir ini penulis telah banyak mendapatkan arahan, bantuan, serta dukungan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis mengucapan terima kasih kepada:

1. <isi dengan nama Rektor ITERA>
2. <isi dengan nama Kajur JTPI>
3. <isi dengan nama Kaprodi IF>
4. <isi dengan nama Sesprodi IF>
5. <isi dengan nama Koordinator TA>
6. <isi dengan nama Dosen Pembimbing>
7. Kedua Orang Tua, kakak dan adik yang selalu memberikan arahan selama belajar dan menyelesaikan tugas akhir ini.
8. <isi dengan nama orang lainnya>

Akhir kata penulis berharap semoga tugas akhir ini dapat memberikan manfaat bagi kita semua, amin. [Contoh]

**RINGKASAN**

Judul TA

Nama Mahasiswa

Halaman Ringkasan berisi uraian singkat tentang latar belakang masalah, rumusan masalah, tujuan, metodologi penelitian, hasil dan analisis data, serta kesimpulan dan saran. Isi ringkasan tidak lebih dari 1500 kata (sekitar 3 halaman).

**ABSTRAK**

Judul TA

Nama Mahasiswa

Halaman ABSTRAK berisi uraian tentang latar belakang, tujuan, metodologi penelitian, hasil / kesimpulan. Ditulis dalam BAHASA INDONESIA tidak lebih dari 250 kata, dengan jarak antar baris satu spasi.

Pada akhir abstrak ditulis kata “Kata Kunci” yang dicetak tebal, diikuti tanda titik dua dan kata kunci yang tidak lebih dari 5 kata. Kata kunci terdiri dari kata-kata yang khusus menunjukkan dan berkaitan dengan bahan yang diteliti, metode/instrumen yang digunakan, topik penelitian. Kata kunci diketik pada jarak dua spasi dari baris akhir isi abstrak.

**Kata Kunci : Penambangan Data, Kecerdasan Buatan, Lampung Selatan**

**ABSTRACT**

Judul TA (Bahasa Inggris)

Nama Mahasiswa

Halaman ABSTRACT berisi uraian tentang latar belakang, tujuan, metodologi penelitian, hasil / kesimpulan. Ditulis dalam BAHASA INGGRIS tidak lebih dari 250 kata, dengan jarak antar baris satu spasi. Secara khusus, kata dan kalimat pada halaman ini tidak perlu ditulis dengan huruf miring meskipun menggunakan Bahasa Inggris, kecuali terdapat huruf asing lain yang ditulis dengan huruf miring (misalnya huruf Latin atau Greek, dll).

Pada akhir abstract ditulis kata “Keywords” yang dicetak tebal, diikuti tanda titik dua dan kata kunci yang tidak lebih dari 5 kata. Keywords terdiri dari kata-kata yang khusus menunjukkan dan berkaitan dengan bahan yang diteliti, metode/instrumen yang digunakan, topik penelitian. Keywords diketik pada jarak dua spasi dari baris akhir isi abstrak.

**Keywords : Data Mining, Artificial Intelligence, Lampung Selatan**

**DAFTAR ISI**

[LEMBAR PENGESAHAN i](#_Toc144318476)

[HALAMAN PERNYATAAN ORISINALITAS ii](#_Toc144318477)

[HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS iii](#_Toc144318478)

[BAB I PENDAHULUAN 1](#_Toc144318479)

[1.1 Latar Belakang Masalah 1](#_Toc144318480)

[1.2 Rumusan Masalah 3](#_Toc144318481)

[1.3 Tujuan Penelitian 4](#_Toc144318482)

[1.4 Batasan Masalah 4](#_Toc144318483)

[1.5 Manfaat Penelitian 4](#_Toc144318484)

[1.6 Sistematika Penulisan 5](#_Toc144318485)

[1.6.1 BAB I PENDAHULUAN 5](#_Toc144318486)

[1.6.2 BAB II TINJAUAN PUSTAKA 5](#_Toc144318487)

[1.6.3 BAB III METODE PENELITIAN 5](#_Toc144318488)

[BAB II TINJAUAN PUSTAKA 6](#_Toc144318489)

[2.1 Tinjauan Pustaka 6](#_Toc144318490)

[2.2 Dasar Teori 10](#_Toc144318491)

[2.2.1 Test-Driven Development 10](#_Toc144318492)

[2.2.2 Use Case Diagram 12](#_Toc144318493)

[2.2.3 Activity Diagram 14](#_Toc144318494)

[2.2.4 Website 14](#_Toc144318495)

[2.2.5 JavaScript 15](#_Toc144318496)

[2.2.6 Node.js 15](#_Toc144318497)

[2.2.7 Express.js 16](#_Toc144318498)

[2.2.8 JavaScript Object Notation 16](#_Toc144318499)

[2.2.9 Esai Singkat 17](#_Toc144318500)

[2.2.10 Python 17](#_Toc144318501)

[2.2.11 System Usability Scale 18](#_Toc144318502)

[2.2.12 Black Box Testing 19](#_Toc144318503)

[2.2.13 Unit Testing 20](#_Toc144318504)

[2.2.14 Peramban Web 20](#_Toc144318505)

[2.2.15 Peluncuran Perangkat Lunak 20](#_Toc144318506)

[BAB III METODE PENELITIAN 21](#_Toc144318507)

[3.1 Alur Penelitian 21](#_Toc144318508)

[3.2 Penjabaran Langkah Penelitian 21](#_Toc144318509)

[3.2.1 Identifikasi Masalah 22](#_Toc144318510)

[3.2.2 Studi Literatur 22](#_Toc144318511)

[3.2.3 Penerapan *Test-Driven Development* 22](#_Toc144318512)

[3.2.4 Integrasi Aplikasi dan Model 23](#_Toc144318513)

[3.2.5 Pengujian 24](#_Toc144318514)

[3.3 Alat dan Bahan Tugas Akhir 24](#_Toc144318515)

[3.3.1 Alat 24](#_Toc144318516)

[3.3.2 Bahan 25](#_Toc144318517)

[3.4 Metode Pengembangan 25](#_Toc144318518)

[3.4.1 System Requirement Specification 26](#_Toc144318519)

[3.4.2 Pembuatan Uji Coba (Test Writing) 37](#_Toc144318520)

[3.4.3 Pembuatan Kode (Coding) 38](#_Toc144318521)

[3.4.4 Refaktor (Refactor) 38](#_Toc144318522)

[3.5 Rancangan Pengujian 38](#_Toc144318523)

[3.5.1 Unit Testing 38](#_Toc144318524)

[3.5.2 System Usability Scale 39](#_Toc144318525)

[3.5.3 Black Box 40](#_Toc144318526)

[3.6 Peluncuran 43](#_Toc144318527)

[BAB IV HASIL DAN PEMBAHASAN 44](#_Toc144318528)

[4.1 Hasil Pengembangan 44](#_Toc144318529)

[4.1.1 Pengembangan Frontend 45](#_Toc144318530)

[4.1.2 46](#_Toc144318531)

[4.2 Hasil Pengujian 46](#_Toc144318532)

[4.3 Analisis Hasil Penelitian 46](#_Toc144318533)

[4.3.1 Analisis Hasil Data 1 47](#_Toc144318534)

[4.3.2 Analisis Hasil Data 2 47](#_Toc144318535)

[4.4 Pembahasan 47](#_Toc144318536)

[4.5 Pengujian 47](#_Toc144318537)

[BAB V KESIMPULAN DAN SARAN 48](#_Toc144318538)

[5.1 Kesimpulan 48](#_Toc144318539)

[5.2 Saran 48](#_Toc144318540)

**DAFTAR TABEL**

[Tabel 2.1 Perbandingan Referensi 8](#_Toc135083386)

[Tabel 2.2 Simbol *Use Case Diagram* 13](#_Toc135083387)

[Tabel 2.3 Simbol *Activity Diagram* 14](#_Toc135083388)

[Tabel 2.4 Standar hasil SUS [13] 19](#_Toc135083389)

[Tabel 3.1 Hasil Wawancara 26](#_Toc135083390)

[Tabel 3.2 Kumpulan User Story 27](#_Toc135083391)

[Tabel 3.3 Kebutuhan Fungsional 28](#_Toc135083392)

[Tabel 3.4 Kebutuhan Non-Fungsional 28](#_Toc135083393)

[Tabel 3.5 Skenario Uji Coba 35](#_Toc135083394)

[Tabel 3.6 Unit Testing 36](#_Toc135083395)

[Tabel 3.7 Kuesioner *System Usability Scale* 37](#_Toc135083396)

[Tabel 3.8 Daftar pengujian Black Box 38](#_Toc135083397)

**DAFTAR GAMBAR**

[Gambar 2.1 SDLC Test-Driven Development (TDD) [20] 11](#_Toc135083424)

[Gambar 2.2 SUS *Score Percentile Rank* [13] 19](#_Toc135083425)

[Gambar 3.1 *Flowchart* Penelitian 21](#_Toc135083426)

[Gambar 3.2 Alur Penelitian *Test-Driven Development* 25](#_Toc135083427)

[Gambar 3.3 *Use Case Diagram* Aplikasi 29](#_Toc135083428)

[Gambar 3.4 *Activity Diagram* Aplikasi 30](#_Toc135083429)

[Gambar 3.5 *Mockup* Halaman Home 31](#_Toc135083430)

[Gambar 3.6 Mockup Metode Penilaian Tunggal 32](#_Toc135083431)

[Gambar 3.7 *Mockup* Hasil Metode Penilaian Tunggal 32](#_Toc135083432)

[Gambar 3.8 *Mockup* Metode Penilaian Jamak 33](#_Toc135083433)

[Gambar 3.9 *Mockup* Hasil Metode Penilaian Jamak 34](#_Toc135083434)

[Gambar 3.10 *Mockup* Halaman Bantuan 34](#_Toc135083435)

**DAFTAR RUMUS**

[Rumus perhitungan SUS (1) 18](#_Toc130882362)

**DAFTAR LAMPIRAN**

[LAMPIRAN 1 Isi Lampiran 11](#_2lwamvv)

# BAB I PENDAHULUAN

## 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 *Integration Testing*, *System Usability Scale* (SUS) dan *Black Box*. *Integration Testing* dilakukan untuk memastikan bahwa setiap fungsi yang dibangun pada aplikasi telah memenuhi syarat kebutuhannya [10]. SUS memiliki tingkat keabsahan dan keandalan tinggi untuk mengukur tingkat kemudahan perangkat lunak saat digunakan [11][12]. SUS digunakan karena terfokus pada kegunaan yang dirasakan pengguna melalui 10 kuesioner yang dapat diisi dengan cepat, dibanding *Post-Study System Usability Questionaire* (PSSUQ) yang lebih terfokus untuk menilai kenyamanan yang dirasakan pengguna saat menggunuakan sebuah sistem [13]. SUS juga tidak membutuhkan biaya lebih besar untuk melakukan pengujiannya dibanding dengan *Heuristic Evaluation* (HE) yang merupakan salah satu metode pengujian perangkat lunak lainnya [14]. Sedangkan pengujian *Black Box* dilakukan untuk menguji tampilan, fungsionalitas, masukkan, 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 [15]. 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.

## 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?
      4. Bagaimana aplikasi penilaian esai singkat dapat digunakan?

## Tujuan Penelitian

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

Mengetahui implementasi metode *test-driven development* pada pembangunan aplikasi implementasi penilaian esai singkat.

Menerapkan *pre-trained machine learning model* python pada aplikasi berbasis JavaScript.

Memastikan hasil penilaian dari aplikasi sama dengan hasil penilaian pada model *machine learning* berdasarkan skor.

Mengetahui penggunan aplikasi penilaian esai singkat.

## 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 masukkan berupa teks untuk jawaban guru.
4. Aplikasi hanya dapat menerima masukkan berupa teks atau berkas *comma-separated values* (CSV) berisi kumpulan jawaban untuk jawaban murid.
5. Aplikasi hanya dapat melakukan penilaian terhadap 1 pertanyaan pada sekali pemakaian.
6. Penelitian ini terfokus pada implementasi aplikasi menggunakan model *machine learning* penilaian esai bahasa Indonesia dan bahasa Inggris yang sudah ada.

## 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.

## Sistematika Penulisan

### 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.

### BAB II TINJAUAN PUSTAKA

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

### BAB III METODE PENELITIAN

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

# BAB II TINJAUAN PUSTAKA

## 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 dilakukannnya, 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 [16]. Hal ini akan digunakan pada penelitian tugas akhir kali ini pada aplikasi saat melakukan penilaian otomatis untuk jawaban secara massal dengan format berkas CSV.

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 [17].

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 [18]. 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 Ingólfsdóttir tentang *lemmatization* untuk bahasa Islandia dengan judul Nefnir: A high accuracy lemmatizer for Icelandic [19].

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 [20].

1. Perbandingan Refrensi

| **No** | **Judul** | **Permasalahan** | **Metode** | **Hasil** | **Perbandingan** |
| --- | --- | --- | --- | --- | --- |
| 1 | TENSORFLOW.JS: MACHINE LEARNING FOR THE WEB AND BEYOND  (2019) | *Library* pembelajaran mesin biasanya ditulis dalam bahasa *Python* atau C++. Namun, jumlah pengguna *JavaScript* untuk frontend dan backend semakin bertambah. Jurnal ini berisi panduan untuk menjembatani *Python* dengan *JavaScript* | -API (*Application Programming Interface*) | Menggunakan *Layers API*, didapatkan performa waktu sebesar  - 3426 ms dengan *JavaScript* biasa dengan 1x percepatan,  - 10 ms dengan *WebGL* dengan 342x percepatan,  - 3 ms dengan *Node.js* dengan 1105x percepatan. | Aplikasi yang akan dikembangkan akan menggunakan API dibangun dengan *Node.js* tanpa menggunakan *TensorFlow*. |
| 2 | Improvement of excel data processing function based on Spring MVC  Framework  (2022) | *Excel* 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(*JavaScript Object Notation*) | Berdasarkan *Spring MVC + EasyUI* dengan pengembangan *Java J2EE IDE*. Diciptakan *kit* pengembangan yang bisa menkonversi *excel* dan melakukan ekspor dan impor sekumpulan data dengan menkonversinya menjadi JSON. | CSV akan dikonversi menjadi bentuk JSON menggunakan teknik *spawn* *JavaScript*. |
| 3 | Nefnir: A high accuracy lemmatizer for Icelandic  (2019) | Mencari bentuk morfologi dasar pada sebuah kumpulan tulisan dengan bahasa yang kaya akan morfologi. | -*Nefnir*  -*part-of-speech* | *Nefnir* meraih akurasi sebesari 99.55% untuk teks yang ditandai dengan benar. *Part-of-speech* (PoS) mendapat akurasi sebesar 96.88% dengan teks yang ditandai dengan *PoS tagger*. | Model yang digunakan pada penelitian ini dicapai menggunakan teknik *stemming* dan *lemmatization*. |
| 4 | Express.js and Ktor web server performance A comparative study  (2022) | Membandingkan dua *framework* web berdasarkan waktu merespon untuk membantu *developer* memilih antara *Express.js* dan *Ktor* | Melakukan test terhadap waktu respon menggunakan database melalui *Object Relational Mapper* (ORM) *Sequelize* untuk *Express.js*, dan *Exposed* untuk *Ktor* | *Express.js* memiliki waktu respon yang lebih baik (3 ms) secara keseluruhan daripada *Ktor* (106 ms). Namun penggunaan *Object Relational Mapper* pada *Ktor* lebih berpengaruh pada hasil daripada *Express.js*. | Aplikasi yang akan dikembangkan pada penelitian ini akan dibangun menggunakan *Express.js* sebagai framework modul HTTP dari *Node.js*. |
| 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 *MySQL* menyimpan data secara terorganisir. | -Basis data non-relasional (*NoSQL*) seperti MongoDB. | Berdasarkan waktu eksekusi *query* untuk *insert*, 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 *Test-Driven Development* (TDD), *Behavior-Driven Development* (BDD), dan *Acceptance Test-Driven Development* (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 *bug* 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.

## 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:

### Test-Driven Development

*Test-Driven Development* (TDD) merupakan metode pengembangan 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]. Kode pengujian yang digunakan dapat berupa *integration testing* atau *scenario testing*. *Scenario testing* digunakan untuk memastikan bahwa fungsi pada aplikasi bekerja pada kondisi spesifik yang biasa digunakan dengan metode pengujian lain [21], sedangkan *integration testing* digunakan untuk memastikan bahwa sebuah fungsi pada aplikasi dapat bekerja sesuai ekspektasi secara keseluruhan. 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:



1. SDLC Test-Driven Development (TDD) [22]
2. Langkah pertama yaitu membaca, memahami, dan memproses fitur atau bug yang diminta.
3. Berdasarkan kebutuhan yang diminta, unit pengujian akan dibuat dan dijalankan dengan kode pada langkah 3.
4. Membuat dan mengimplementasikan kode yang memenuhi kebutuhan pengujian. Saat dijalankan, semua pengujian harus lolos, jika tidak, langkah ini akan diulang hingga berhasil.
5. Apabila sudah berhasil, maka kode akan dibersihkan/dirapikan (*refactoring*).
6. 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 [22].

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].

### 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 [23]. 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*.

1. 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 | Merepresentasiskan hubungan antara aktor dengan aktor, maupun *use case* dengan *use case* lainnya. |
|  | Asosiasi | Merepresentasikan komunikasi dua arah antara aktor dengan *use case*, setiap kasus dimulai dengan aktor utama yang harus mendapatkan respon dari *use case*. |
|  | Include | Merepresentasikan hubungan antara dua *use case*. Saat *use case* A membutuhkan *use case* B untuk menyelesaikan tugasnya, walau *use case* B dapat menyelesaikan tugasnya sendiri. |
|  | Extend | Merepresentasikan hubungan antara dua *use case*. Saat *use case* A mungkin memerlukan *use case* B untuk menyelesaikan tugasnya, namun *use case* 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.

### 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 [24]. *Activity Diagram* biasanya berisi aktifitas, transaksi, keputusan, swimlane, dan aktifitas paralel [25]. Berdasarkan itu, simbol, nama, dan deskripsi dari *Activity Diagram* dapat dilihat pada Tabel 2.3 berikut ini.

1. 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.

### Website

*Website* adalah keseluruhan halaman-halaman web yang terdapat dari sebuah domain yang mengandung informasi [26]. Layanan yang terdapat pada *website* dapat diakses melalui jaringan internet, dengan permintaan layanan dieksekusi melalui sistem jarak jauh[27].

### JavaScript

*JavaScript* adalah bahasa pemrograman yang populer, digunakan bukan hanya untuk web sisi klien, tapi juga sisi *server* [28]. *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*.

### 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* [29].

*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 [30]. Hal ini membuat *Node.js* lebih ideal untuk membangun *website* dengan masukkan dan keluaran yang intens. Selain itu, *Node.js* juga *developer friendly*, yang mana mudah dipahami karena cara penggunaannya yang konsisten.

### 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* [31] 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* [17], 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.

### 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* [32].

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 (}) [32].

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 berkas CSV ke aplikasi website penilaian esai. Berkas CSV kemudian akan diubah menjadi bentuk JSON untuk selanjutnya diproses kedalam penilaian.

### Esai Singkat

Esai merupakan karangan yang membahas suatu masalaah 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 [33].

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 [34].

### 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 [35].

*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* [36].

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

### 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 [11]. SUS adalah alat psikometrik gratis yang digunakan diseluruh dunia dengan keabsahan dan keandalan tinggi [12].

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 [37].

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

Rumus perhitungan SUS (1)

Keterangan:

*α* : skor pertanyaan bernomor ganjil

*b* : skor pertanyaan bernomor genap

: skor rata-rata

: jumlah responden

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

1. Standar Hasil SUS [14]

| **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 [14].



1. SUS *Score Percentile Rank* [14]

### 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, masukkan, dan luaran [15].

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

### Integration Testing

Untuk meningkatkan kepastian bahwa suatu program berfungsi sebagaimana diinginkan, perlu proses validasi untuk melakukannya. Proses validasi dapat dilakukan menggunakan Integration Testing karena dapat membuktikan bahwa sebuah fungsi pada program telah memenuhi spesifikasinya [10]. Integration Testing dilakukan dengan memastikan bahwa komponen-komponen pada program saling cocok dan memberikan hasil sesuai harapan [38].

### 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 [39].

*JavaScript* telah mencapai platform web secara luas dan menjadi bahasa pemrograman multifungsi dengan adanya ES6 yang diimplementasikan pada mesin peramban web sejak tahun 2017 [40].

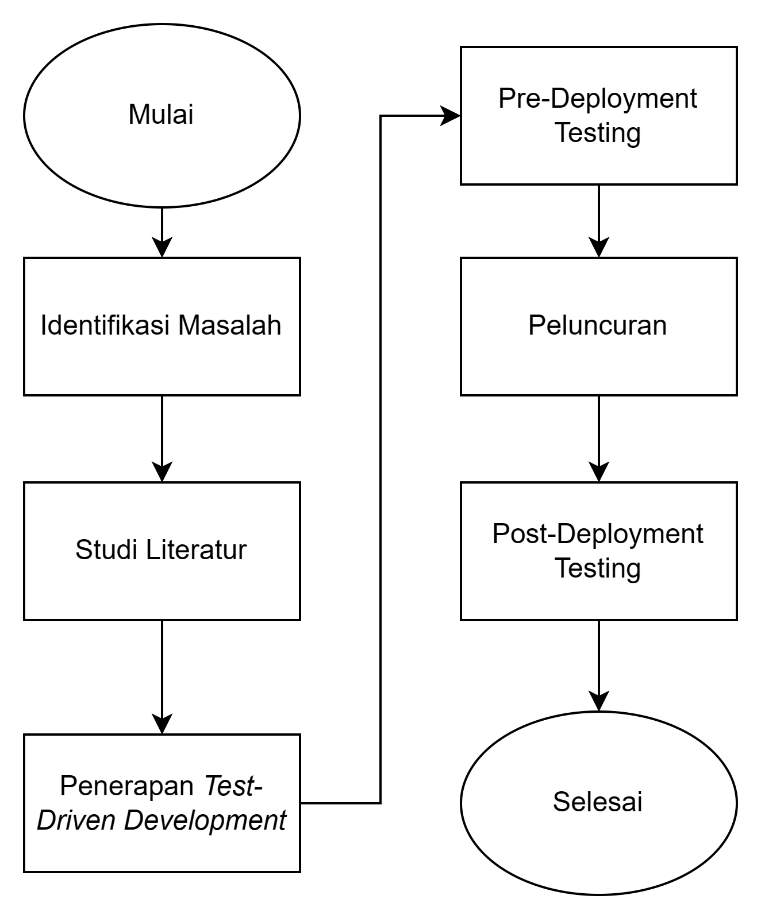
### Peluncuran Perangkat Lunak

Peluncuran (*deployment*) perangkat lunak merupakan proses eksekusi dari sebuah perangkat lunak. Peluncuran adalah aktifitas dari sebuah perangkat lunak setelah proses pembangunan perangkat lunak [41]. Peluncuran dilakukan untuk mendapatkan timbal balik berupa perbaruan, konfigurasi ulang, adaptasi, penonaktifan, dan peluncuran ulang pada sebuah perangkat lunak [42].

# BAB III METODE PENELITIAN

## 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.



1. *Flowchart* Penelitian

## 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:

### 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.

### 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.

### 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

### Integrasi Aplikasi dan Model

Pada penelitian ini, aplikasi yang akan dikembangkan akan menggunakan dua model pembelajaran mesin yang telah dilatih sebelumnya. Dua model tersebut yakni dalam bahasa Indonesia dan bahasa Inggris. Alur integrasi aplikasi dengan model yang digunakan dapat dilihat pada Gambar 3.2 berikut ini.



1. Alur Integrasi Aplikasi dan Model

Integrasi model pembelajaran mesin pada aplikasi digunakan dengan pengguna memilih bahasa yang akan digunakan pada penilaian yang dilakukannya, dengan ini aplikasi akan memilih model dengan bahasa yang tepat.

Selanjutnya aplikasi akan menjalankan model terkait dengan masukkan dari pengguna untuk diproses. Setelah model selesai melakukan penilaian otomatis, model akan menghasilkan objek dan mengembalikannya pada aplikasi. Kemudian aplikasi akan menggunakan objek untuk menampilkan hasil penilaian kepada pengguna untuk dilihat atau diunduh berupa PDF.

### 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.

## 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:

### 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 2GB 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. *Python* dengan versi minimal 3.11.5.
5. *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.

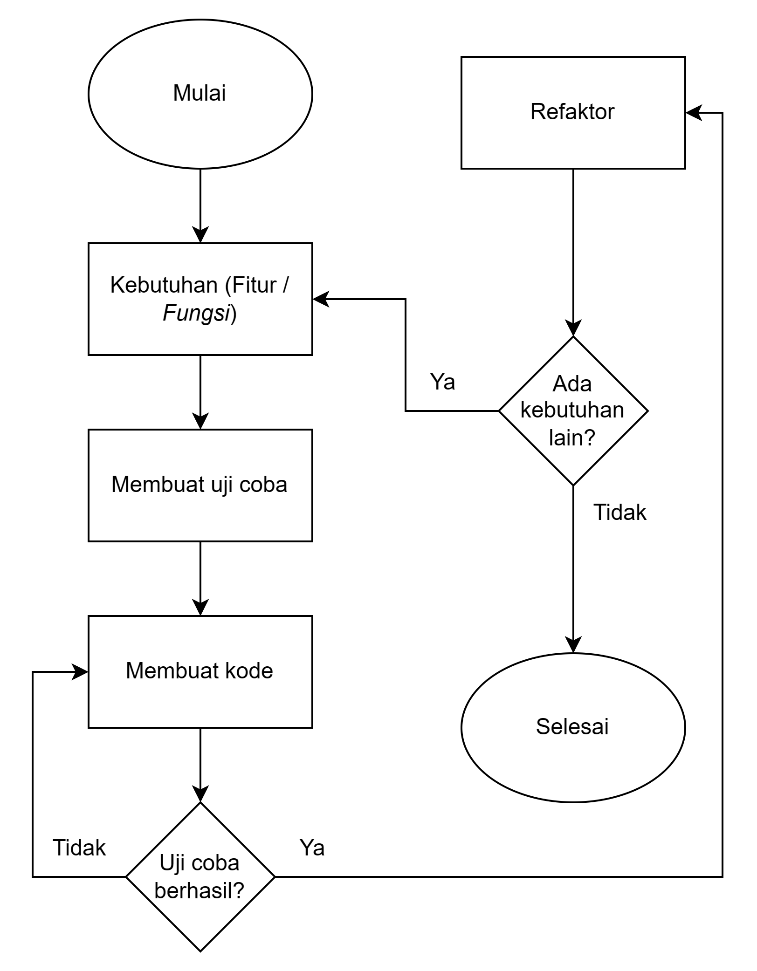
### Bahan

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

## 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 3.3 dibawah ini.



1. Alur Penelitian *Test-Driven Development*

Berdasarkan diagram alir yang dapat dilihat pada Gambar 3.3, 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:

### Kebutuhan Fitur / Fungsi

Tahap ini merupakan tahap pertama dari metode pengembangan TDD guna mendapatkan kebutuhan atau fungsi pada aplikasi yang akan dibangun. Pada tahap ini, pengembang diharapkan dapat mengerti kebutuhan yang diminta untuk dapat dilakukan oleh perangkat lunak yang sedang dikembangkan.

#### 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 sepuluh guru/dosen. Pada Tabel 3.1 dapat dilihat data narasumber, dan pada Tabel 3.2 merupakan pertanyaan dan jawaban dari wawancara yang dilakukan.

1. Data Narasumber

| **No** | **Nama Responden** | **Peran** | **Lokasi** |
| --- | --- | --- | --- |
| 1 | Zamroni Riza S.E | Guru | SMA Sapta Kharisma Jakarta |
| 2 | Dr. Dian Mustika Dewi Priyandari, S. Sos., M.M. | Dosen | Universitas Suryadarma |
| 3 | Sukma Erawan, S.Kom | Guru | SMA Negeri 68 Jakarta |
| 4 | Alan Syahwandi, S.Pd.I | Guru | SMP Negeri 3 Bogor |
| 5 | Ajeng Yeni Ratnasari, S.Pd | Guru | Pondok Pesantren Alfassalam |
| 6 | Dita Putri Bestari S.Pd | Guru | SMA Negeri 65 Jakarta |
| 7 | Ayu Nulantika S.Pd | Guru | SMA Negeri 65 Jakarta |
| 8 | Virgawaty Dewy, S.Pd | Guru | SMA Negeri 65 Jakarta |
| 9 | Ahmad Arfandi, S.Kom | Guru | SMA Negeri 65 Jakarta |
| 10 | Muhammad Lutfi, S.Pd | Guru | SMA Negeri 113 Jakarta |

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. |

#### 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 3.3 dibawah ini:

1. Kumpulan *User Story*

| **No** | **Nama** | **Objective** | **Alasan** |
| --- | --- | --- | --- |
|  | Zamroni Riza S.E | Dapat melakukan penilaian secara masal | Aplikasi harus bisa dilakukan sekaligus agar proses penilaian bisa cepat |
|  | Alan Syahwandi, S.Pd.I |
|  | Ajeng Yeni Ratnasari, S.Pd |
|  | Dita Putri Bestari S.Pd | 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. |
|  | Ayu Nulantika S.Pd |
|  | Muhammad Lutfi, S.Pd |
|  | Dr. Dian Mustika Dewi Priyandari, S. Sos., M.M. | Dapat dipakai langsung dari aplikasi ujian | Aplikasi harus dapat dipakai langsung dari aplikasi yang sering digunakan ujian esai |
|  | Ahmad Arfandi, S.Kom | Dapat melakukan penilaian untuk satu jawaban | Aplikasi bisa dipakai untuk menilai satu esai apabila ada kebutuhan khusus |
|  | Virgawaty Dewy, S.Pd |
|  | Sukma Erawan, S.Kom |

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. Kemudian untuk penilaian esai secara masal harus menggunakan tipe dokumen yang sama dari aplikasi ujian pada umumnya. Kebutuhan ini diuraikan menjadi kebutuhan teknis yang dijelaskan pada sub-bab 3.4.1.2 Kebutuhan Fungsional.

#### 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 3.4 berikut ini.

1. Kebutuhan Fungsional

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

#### 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 3.5 dibawah ini dapat dilihat kebutuhan non-fungsional yang akan dipenuhi.

1. Kebutuhan Non-Fungsional

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

#### Use Case Diagram

*Use Case Diagram* adalah diagram yang digunakan untuk menggambarkan sebuah interaksi yang dapat terjadi antara pengguna dan sistem. Pada Gambar 3.4 dibawah ini dapat diperhatikan interaksi pengguna pada sistem yang terjadi pada sistem yang hendak dibangun.

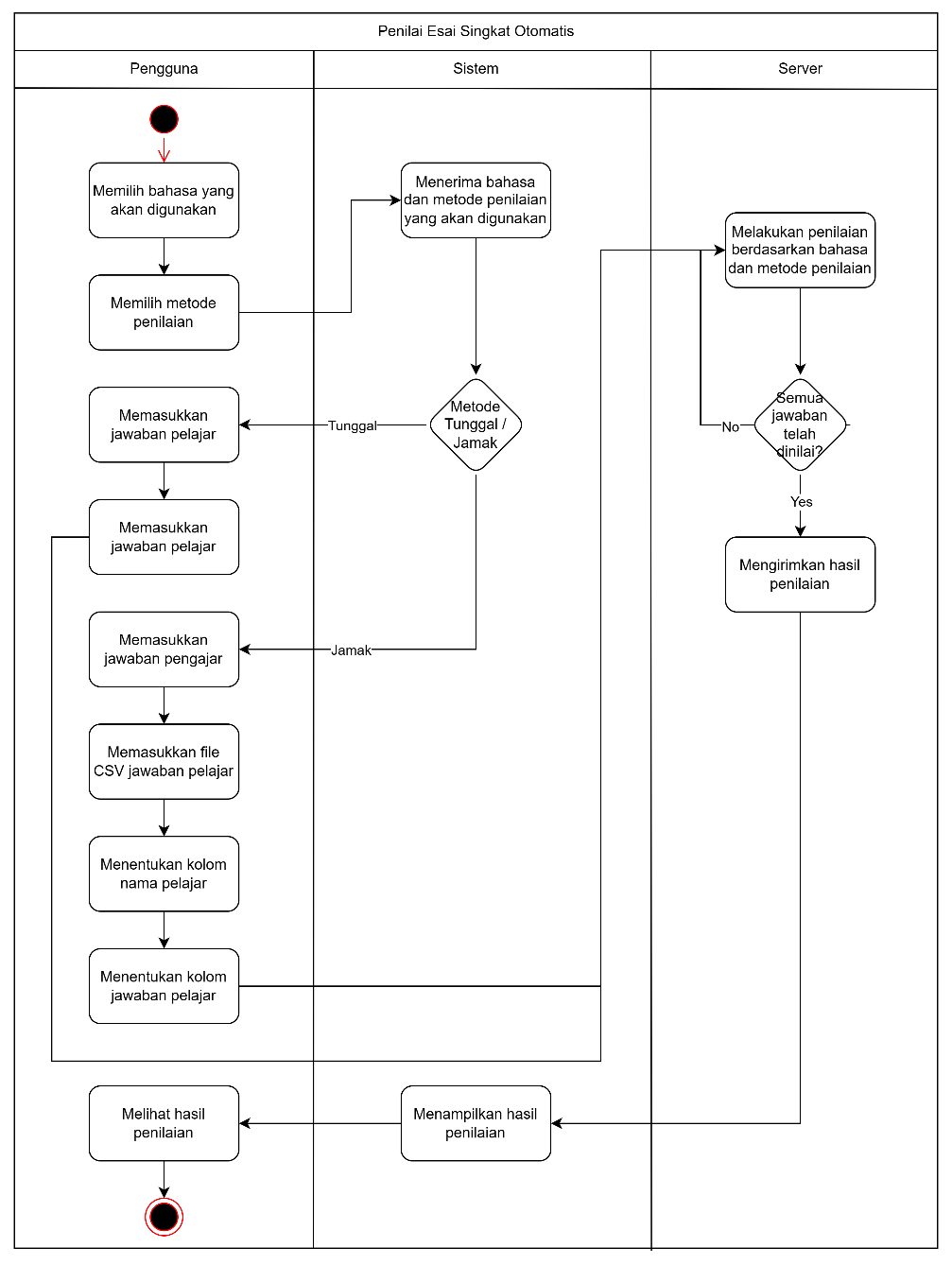


1. *Use Case Diagram* Aplikasi

Pada Gambar 3.4 tersebut, aktor merupakan pengajar yang ingin melakukan pengecekkan jawaban esai milik pelajar. Pengguna dapat melihat hasil jawaban yang diperiksa dengan mengirimkan esai yang ingin diperiksa, kemudian memilih bahasa dan teknik pengecekkan yang diinginkan.

#### Activity Diagram

*Activity diagram* digunakan untuk menggambarkan alur pekerjaan dari aktivitas pada suatu sistem. Pada Gambar 3.5 berikut ini dapat dilihat *activity diagram* dari aplikasi yang akan dibangun pada penelitian ini.



1. *Activity Diagram* Aplikasi

Pada Gambar 3.5 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.

#### 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.



1. *Mockup* Halaman Home

Saat pengguna mengakses halaman *website*, pengguna akan ditunjukkan halaman beranda seperti pada Gambar 3.6 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.



1. *Mockup* Metode Penilaian Tunggal

Pada Gambar 3.7 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.



1. *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 3.8 dimana pengguna dapat melihat jawban pengajar dan jawaban pelajar dengan persentase hasil jawabannya.



1. *Mockup* Metode Penilaian Jamak

Pada Gambar 3.9 diatas merupakan halaman yang akan ditampilkan apabila pengguna memilih metode penilaian secara jamak. Pada halaman ini, pengguna dapat mengisi jawaban pengajar dan memasukkan berkas CSV pada aplikasi dan menentukan judul kolom berisi nama dan jawaban pelajar untuk diperiksa terhadap jawaban pengajar.

Gambar 3.10 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.



1. *Mockup* Hasil Metode Penilaian Jamak



1. *Mockup* Halaman Bantuan

Apabila pada halaman beranda seperti pada Gambar 3.6 pengguna memutuskan untuk menekan logo tanda tanya, maka pengguna akan diarahkan ke halaman bantuan seperti pada Gambar 3.11. Pada halaman ini, pengguna dapat melihat cara menggunakan aplikasi penilaian esai otomatis.

### Pembuatan Uji Coba (Scenario Testing)

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 3 skenario uji coba yang harus berhasil setiapnya dengan ekspektasi pada Tabel 3.6 berikut ini:

1. Skenario Uji Coba

|  |  |  |  |
| --- | --- | --- | --- |
| **Skenario** | **Masukkan Jawaban Pengajar** | **Masukkan Jawaban Pelajar** | **Ekspektasi Keluaran** |
| SC1 | Teks untuk uji coba jawaban tunggal “The testing stage can influence both the coding stage (phase 5) and the solution refinement stage (phase 7)” | Teks untuk uji coba tunggal “The implementation phase and the maintenance phase are effected” | 1.128299 dari 5.0 |
| 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. ” | 55 dari 100 |

### 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.

### Refaktor (Refactor)

Saat kode untuk suatu fungsi berdasarkan suatu uji coba sudah berhasil dibuat, maka akan dilakukan pembersihan kode. Pembersihan disini berarti kode akan tulis 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.

## Pre-Deployment Testing (Integration Testing)

Pengujian ini dibuat untuk memastikan bahwa komponen-komponen berbeda pada program berjalan semestinya berdasarkan skenario yang telah dibuat sebelumnya. Akan dibuat sebuah fungsi pada *server* untuk dapat memproses skenario pengujian dengan rancangan pada Tabel 3.7 berikut:

1. *Integration Testing*

| **ID** | **Metode** | **Input 1** | **Input 2** | **Output** |
| --- | --- | --- | --- | --- |
| IT1 | ScenarioEn(req, res) | Teks jawaban pengajar berbahasa Inggris. | Teks jawaban pelajar berbahasa Inggris. | JSON hasil penilaian tunggal berbahasa Inggris |
| IT2 | ScenarioEnTab(req, res) | To sumulate the behaviour of portions of the desired software product. | dataset.csv | JSON hasil penilaian jamak berbahasa Inggris. |
| IT3 | ScenarioId(req, res) | Teks jawaban pengajar berbahasa Indonesia. | Teks jawaban pelajar berbahasa Indonesia. | Hasil penilaian tunggal berbahasa Indonesia |

Pada Tabel 3.7 tersebut, akan dibuat 3 fungsi yang masing-masing dapat menerima teks jawaban pengajar dalam bahasa Indonesia atau Inggris, kemudian menerima jawaban pelajar untuk diperiksa melalui teks atau berkas CSV. Fungsi tersebut selanjutnya akan memproses kedua masukkan, dan menampilkan persentase hasil penilaian jawaban pada *console*.

## Peluncuran

Apabila semua kebutuhan terhadap perangkat lunak yang dibangun sudah dibuat dan telah teruji bekerja dengan baik berdasasrkan pengujian sebelumya, maka akan dilakukan tahap peluncuran, yaitu kondisi dimana perangkat lunak yang dikembangkan sudah siap digunakan secara umum. Tahap ini dilakukan untuk mendapatkan timbal balik dari pengguna terkait perangkat lunak yang dikembangkan.

## Post-Deployment Testing

Setelah aplikasi diluncurkan, maka akan didapatkan timbal balik dari pengguna untuk pengujian SUS dan *Blackbox*. Dengan ini dapat dilakukan perbaikan apabila terdapat bug yang tidak terdeteksi selama pembangunan aplikasi dan dapat diluncurkan kembali. Apabila tahap ini telah membuahkan hasil yang memuaskan, maka penelitian ini dapat dianggap selesai.

### 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 3.8 berikut adalah daftar kuesioner yang akan diajukan kepada pengguna setelah menggunakan aplikasi yang bersangkutan.

1. 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 |

### Black Box

Pada waktu yang bersamaan, akan dilakukan pengujian Black Box 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 3.9 dibawah ini.

1. Daftar Pengujian Fungsional 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 berkas berbentuk CSV sebagai kumpulan jawaban pelajar | Aplikasi web berhasil menyimpan berkas CSV 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 berkas CSV |
| 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 berkas CSV |
| 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 berkas CSV 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 syarat penilaian pada algoritma penilaian otomatis bahasa Inggris | Aplikasi web berhasil membuat berkas CSV dari berkas CSV, nama, dan jawaban pelajar untuk 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 berkas CSV |
| 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 |

Pada Tabel 3.9 tersebut dapat diperhatikan ID dan PARENT ID untuk setiap pengujian *black box*. PARENT ID melambangkan bahwa pengujian memiliki fokus utama untuk suatu pengujian yang dipecah menjadi beberapa pengujian lainnya yang dilambangkan sebagai ID.

PARENT ID F1A merupakan pengujian untuk memastikan bahwa aplikasi web dapat menerima masukkan teks dari pengguna sebagai variabel pada *backend* untuk dilakukan penilaian jawaban.

PARENT ID F1B merupakan pengujian untuk memastikan bahwa masukkan teks dari F1A dapat digunakan sebagai variabel penilaian pada model *machine learning* yang akan digunakan yang penentuannya merupakan pengujian dari ID F0.

Selanjutnya, PARENT ID F2A adalah sekumpulan pengujian untuk menguji fungsi penilaian jamak melalui aplikasi web yang akan dikembangkan. Pengujian F2A memastikan bahwa *backend* akan menerima berkas CSV dan rinciannya yang akan digunakan untuk memproses penilaian jamak untuk setiap model *machine learning* terpilih yang mana pengujiannya merupakan bagian dari PARENT ID F2B.

Setelah pengujian fungsional, dilakukan juga pengujian non-fungsional pada aplikasi yang dikembangkan penelitian ini dengan pengujian sebagai berikut.

1. Daftar Pengujian Non Fungsional Black Box

| **PARENT ID** | **ID** | **Pengujian** | **Ekspektasi** |
| --- | --- | --- | --- |
| 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 yang berbeda | Aplikasi web berhasil beradaptasi pada setiap resolusi dimana perlu ditampilkan tanpa adanya komponen yang tumpang-tindih |

Pada Tabel 3.10, PARENT ID NF1 merupakan pengujian untuk tampilan yang dapat berubah (*responsive*) pada ukuran layar apapun yang sifatnya non-fungsional.

# BAB IV HASIL DAN PEMBAHASAN

## Hasil Pengembangan Metode TDD

Metode pengembangan *Test-Driven Development (TDD)* adalah salah satu metode pengembangan software *Agile* yang dikenalkan oleh *eXtreme Programming (XP)*. Metode TDD ini dapat menghasilkan perangkat lunak yang terbebas dari *bug* atau *error*. Hal ini dikarenakan TDD mengharuskan pengembang untuk berhasil menerapkan satu fitur / fungsi terlebih dahulu, kemudian apabila berhasil melewati berbagai uji coba akan dilanjutkan ke fitur / fungsi selanjutnya.

Pada TDD, pengembang akan melewati tiga tahap utama yang disebut sebagai tahap *red-green-refactoring*. Pada tahap *red*, pengembang akan menuliskan kode untuk meloloskan sebuah pengujian untuk suatu fitur / fungsi hingga berhasil. Apabila berhasil, maka pengembang akan masuk ke tahap *green* yaitu dimana kode yang dituliskan sudah berhasil melewati uji coba suatu fitur / fungsi. Setelahnya, pengembang akan masuk ke tahap *refactoring* dimana pengembang harus membersihkan kode yang telah dikembangkan dengan membuang fungsi atau baris kode yang tidak dipakai sehingga menghasilkan perangkat lunak dengan *clean code*. Setelah tahap *refactoring*, pengembang bisa langsung masuk lagi ke tahap *red* dan memulai siklus baru untuk fitur / fungsi baru pada perangkat lunak yang dikembangkan.

Aplikasi web yang dikembangkan dapat dicapai dengan memenuhi kebutuhan fungsional dan kebutuhan non-fungsional yang didapat setelah mengumpulkan *user story* dari calon pengguna aplikasi yang dikembangkan sebagaimana dijabarkan pada bab sebelumnya. Kemudian dengan ini didapatkan rancangan perangkat lunak berupa *use case diagram*, *activity diagram* dan *mockup* sebagai rancangan antarmuka.

Penelitian ini dilakukan dengan maksud menghasilkan aplikasi yang dapat mempermudah para pengajar dalam menilai jawaban esai pelajar, mengetahui banyaknya jumlah pelajar yang perlu diperiksa oleh satu guru yang dapat memakan waktu lama. Aplikasi yang dikembangkan penulis memiliki *frontend*, *backend*, dan 2 *model machine learning* yang dipakai untuk melakukan penilaian esai pada *backend*.

Dengan ini, pengembangan perangkat lunak yang dilakukan dapat dilihat secara rinci pada sub-bab berikut ini.

### Pengembangan Frontend

*Frontend* yang dikembangkan digunakan sebagai antarmuka agar pengguna bisa memilih bahasa yang hendak digunakan, memilih cara penilaian, memasukkan kunci jawaban dan jawaban yang ingin diperiksa, dan memulai proses penilaian. Setelah pengguna memulai proses tombol penilaian, *frontend* akan mengirimkan data yang diberikan oleh pengguna ke *backend* untuk melakukan proses penilaian. Gambar yang dilampirkan dibawah ini adalah hasil akhir dari *frontend* aplikasi penilai esai otomatis berdasarkan tampilan *desktop* dan *mobile*.

* + - 1. **Frontend Halaman Home**

Pada Gambar 4.1 dapat dilihat *frontend* dari halaman utama aplikasi penilaian esai otomatis. Pada halaman ini, pengguna dapat memilih bahasa yang ingin digunakan untuk memeriksa esai, memilih metode penilaian esai, ataupun mengakses halaman bantuan.



1. *Frontend* Halaman Home
   * + 1. **Frontend Halaman Metode Penilaian Tunggal**

Pengguna dapat memilih “Cek satu jawaban” pada halaman home pada Gambar 4.1, kemudian pengguna akan diarahkan ke halaman metode penilaian tunggal untuk memasukkan 1 jawaban pengajar dan 1 jawaban pelajar untuk diperiksa seperti pada Gambar 4.2. Pengguna juga dapat melihat bahasa yang digunakan di pojok kanan atas halaman untuk memastikan pengguna tidak salah pilih bahasa yang digunakan untuk menilai esai.



1. *Frontend* Halaman Metode Penilaian Tunggal
   * + 1. **Frontend Halaman Hasil Metode Penilaian Tunggal**

Setelah menekan tombol “periksa” pada halaman Gambar 4.2, pengguna akan diarahkan ke halaman hasil penilaian seperti pada Gambar 4.3 untuk melihat hasil penilaian dari metode penilaian tunggal.



1. *Frontend* Halaman Hasil Metode Penilaian Tunggal
   * + 1. **Frontend Halaman Metode Penilaian Jamak**

Apabila pengguna ingin melakukan penilaian jamak, pengguna dapat memilih “Cek banyak jawaban” pada halaman home, dan pengguna akan diarahkan ke halaman metode penilaian jamak seperti pada Gambar 4.4. Pada halaman ini, pengguna dapat mengisi jawaban pengajar, kemudian mengunggah file CSV yang berisi nama dan jawaban pelajar, kemudian menspesifikasikannya pada kolom yang disediakan. Setelahnya, pengguna dapat menekan tombol “periksa”.



1. *Frontend* Halaman Metode Penilaian Jamak
   * + 1. **Frontend Halaman Hasil Metode Penilaian Jamak**

Setelah menekan tombol “periksa”, pengguna akan diarahkan ke halaman hasil metode penilaian jamak seperti pada Gambar 4.5. Di halaman ini, pengguna dapat melihat tabel pasangan nama, jawaban, dan skor esai pelajar serta mengunduhnya dalam format PDF dengan menekan tombol “Export to PDF”.



1. *Frontend* Halaman Hasil Metode Penilaian Jamak
   * + 1. **Frontend Halaman Bantuan**

Apabila pengguna mengalami kesulitan dalam menggunakan aplikasi yang telah dibangun oleh penulis, pengguna dapat menekan tombol “?” yang ada di pojok kanan atas pada halaman home seperti pada Gambar 4.1, dan akan diarahkan ke halaman bantuan seperti pada Gambar 4.6. Pada halaman bantuan ini, pengguna dapat menemukan pertanyaan, jawaban, dan tutorial cara menggunakan aplikasi penilai esai otomatis ini.



1. *Frontend* Halaman Bantuan

### Pengembangan Backend

*Backend* akan berperan untuk menerima dan memproses permintaan dari *frontend*, baik permintaan rute (*routing*) maupun memproses penilaian. Saat memproses rute, *backend* akan mengarahkan pengguna ke halaman yang dituju oleh rute tersebut baik dari permintaan *get* atau *post*. Khususnya pada saat memeriksa jawaban esai, *backend* akan menggunakan model *machine learning* melalui metode *child process*.

* + - 1. **Rute Untuk Halaman Home**

Pada saat mengakses aplikasi, pengguna akan diarahkan ke halaman utama / *home*. Backend disini berperan untuk menerima permintaan *get* ke rute “/”. Aksi yang dilakukan rute ini hanyalah menampilkan halaman home kepada pengguna sebagaimana pada Tabel 4.1.

|  |
| --- |
| **app.get("/", (req, res) => {**  **res.render("home/home");**  **});** |

1. Kode Program Rute /
   * + 1. **Rute Untuk Mengakses Halaman Bantuan**

Apabila pengguna menekan tombol bantuan, *backend* akan menerima permintaan *get* ke rute “/help” sebagaimana dilihat pada Tabel 4.2. Rute ini hanya akan menampilkan halaman bantuan.

|  |
| --- |
| app.get("/", (req, res) => {  res.render("home/home");  }); |

1. Kode Program Rute /help
   * + 1. **Rute Untuk Menentukan Pilihan**

Pada halaman *home*, pengguna dapat menetukan bahasa esai yang akan digunakan untuk melakukan penilaian dan metode penilaian esainya melalui permintaan *post* ke rute “/option”. Pada Tabel 4.3, backend akan memberikan pengguna halaman sesuai dengan permintaannya pada saat memilih di halaman *home*.

|  |
| --- |
| app.route("/option").post((req, res) => {  const obj = {  };  if (obj.language === "Indonesia") {  } else if (obj.language === "Inggris") {  } else {  res.redirect("/");  }  }); |

1. Kode Program Rute /option
   * + 1. **Rute Untuk Mengakses Halaman Penilaian Tunggal Bahasa Inggris**

Apabila pengguna menginginkan untuk memeriksa esai dengan bahasa Inggris dan metode penilaian tunggal, maka pengguna akan mengunggah jawaban pengajar dan pelajar. Pada tahap ini, *backend* mendapatkan permintaan *post* pada rute “/ens” dan menjalankan model *machine learning* bahasa Inggris untuk melakukan penilaian tersebut. Setelahnya, *backend* akan menyajikan hasil penilaian kepada pengguna. Rinci dari rute ini dapat dilihat pada Tabel 4.4.

|  |
| --- |
| app.post("/ens", (req, res) => {  const keyAnswer = req.body.teacherAnswer;  const answer = req.body.studentAnswer;  const language = req.body.language;  const docid = req.body.docid;  let Dataset = `keyAnswer,studentAnswer\n"${keyAnswer}","${answer}"`;  let scoreLoad;  fs.writeFileSync(docid + ".csv", Dataset, "utf8");  let options = {  };  PythonShell.run("models/english/aprilModel.py", options)  .then((messages) => {  })  .then(() => {  })  .then(() => {  })  .catch((err) => {  });  }); |

1. Kode Program Rute /ens
   * + 1. **Rute Untuk Mengakses Halaman Penilaian Jamak Bahasa Inggris**

Apabila pengguna menginginkan untuk memeriksa esai dengan bahasa Inggris dan metode penilaian jamak, maka pengguna akan mengunggah jawaban pengajar, berkas CSV jawaban pelajar, dan nama kolom untuk nama dan jawaban. Pada tahap ini, *backend* mendapatkan permintaan *post* pada rute “/enb” dan menjalankan model *machine learning* bahasa Inggris untuk melakukan penilaian tersebut. Setelahnya, *backend* akan menyajikan hasil penilaian kepada pengguna. Rinci dari rute ini dapat dilihat pada Tabel 4.5.

|  |
| --- |
| app.post("/enb", upload.single("studentAnswers"), (req, res) => {  const keyAnswer = req.body.teacherAnswer;  const studentNames = req.body.nameColumns;  const answerColumns = req.body.answerColumns;  const language = req.body.language;  let docid = bulkDocumentID;  let scoreLoad;  CSVToJSON()  .fromFile("./" + docid + ".csv")  .then((source) => {  });  }); |

1. Kode Program Rute /enb
   * + 1. **Rute Untuk Mengakses Halaman Penilaian Tunggal Bahasa Indonesia**

Apabila pengguna menginginkan untuk memeriksa esai dengan bahasa Indonesia dan metode penilaian tunggal, maka pengguna akan mengunggah jawaban pengajar dan pelajar. Pada tahap ini, *backend* mendapatkan permintaan *post* pada rute “/ids” dan menjalankan model *machine learning* bahasa Indonesia untuk melakukan penilaian tersebut. Setelahnya, *backend* akan menyajikan hasil penilaian kepada pengguna. Rinci dari rute ini dapat dilihat pada Tabel 4.6.

|  |
| --- |
| app.post("/ids", (req, res) => {  const { language, teacherAnswer, studentAnswer, docid } = req.body;  let option = {  };  PythonShell.run("./models/indonesia/ASAG.py", option).then((messages) => {  });  }); |

1. Kode Program Rute /ids
   * + 1. **Rute Untuk Mengakses Halaman Penilaian Jamak Bahasa Indonesia**

Apabila pengguna menginginkan untuk memeriksa esai dengan bahasa Indonesia dan metode penilaian jamak, maka pengguna akan mengunggah jawaban pengajar, berkas CSV jawaban pelajar, dan nama kolom untuk nama dan jawaban. Pada tahap ini, *backend* mendapatkan permintaan *post* pada rute “/idbulk” dan menjalankan model *machine learning* bahasa Indonesia untuk melakukan penilaian tersebut. Setelahnya, *backend* akan menyajikan hasil penilaian kepada pengguna. Rinci dari rute ini dapat dilihat pada Tabel 4.7.

|  |
| --- |
| app.post("/idbulk", upload.single("studentAnswers"), async (req, res) => {  const { teacherAnswer, nameColumns, answerColumns, language } = req.body;  let docid = bulkDocumentID;  let idbulk1 = fs.readFileSync("public/pages/bulkID1.html", "utf-8");  let idbulk2 = fs.readFileSync("public/pages/bulkID2.html", "utf-8");  let idbulk3 = fs.readFileSync("public/pages/bulkID3.html", "utf-8");  res.write(idbulk1);  res.write("<p>");  res.write(teacherAnswer);  res.write("</p>");  res.write(idbulk2);  const sources = await CSVToJSON().fromFile("./" + docid + ".csv");  if (  sources[0][nameColumns] == undefined &&  sources[0][answerColumns] == undefined  ) {  } else {  }  }); |

1. Kode Program Rute /idbulk
   * + 1. **Konversi Hasil Penilaian Menjadi PDF**

Khusus untuk fungsi penilaian jamak dimana pengguna dapat melihat banyak hasil penilaian esai pelajar, peneliti membuat fitur untuk dapat menyimpan hasil penilaian dengan bentuk PDF.

|  |
| --- |
| document.getElementById("print").addEventListener("click", () => {  document.getElementById("print").style.opacity = 0;  window.print();  document.getElementById("print").style.opacity = 1;  }); |

1. Kode Konversi Hasil Penilaian Menjadi PDF

Pada kode tersebut, *JavaScript* dari *frontend* hasil penilaian dapat mendeteksi saat pengguna menekan tombol “Export to PDF” dengan Id “print”, untuk menjalankan perintah cetak halaman. Namun sebelumnya, akan disembunyikan terlebih dahulu tombol “Export to PDF” yang diklik pengguna agar tidak muncul pada hasil penilaian yang telah dicetak, kemudian setelahnya akan ditampilkan lagi.

* + - 1. **Konversi Berkas CSV Menjadi JSON**

Sebelum berkas CSV unggahan pengguna dapat digunakan, berkas CSV tersebut perlu diubah menjadi JSON untuk dapat digunakan isinya. Hal ini dilakukan agar *backend* dapat menambahkan kolom baru pada berkas CSV yang ingin diperiksa, khususnya untuk menentukan kolom jawaban dan nama pelajar.

|  |
| --- |
| CSVToJSON()  .fromFile("./" + docid + ".csv")  .then((source) => {  for (let i = 0; i < source.length; i++) {  source[i].keyAnswer = keyAnswer;  source[i].studentAnswer = source[i][answerColumns];  source[i].studentName = source[i][studentNames];  }  const csv = JSONToCSV(source);  fs.writeFileSync(docid + ".csv", csv); |

1. Kode Konversi Berkas CSV ke JSON

Kode program diatas menunjukkan konversi berkas CSV menjadi JSON, dimana konten akan ditambah dengan informasi jawaban pelajar, jawaban pelajar, dan nama pelajar. Setelah dilakukan hal tersebut, JSON akan dikonversi kembali menjadi berkas CSV dengan informasi tambahan yang cocok untuk dilakukan pemeriksaan pada model *machine learning*.

### Child-Process Model Machine Learning

Setelah *backend* menerima jawaban pengajar dan jawaban pelajar untuk diperiksa, *backend* akan mengirimkan jawaban pengajar dan jawban pelajar tersebut ke model *machine learning* untuk diperiksa melalui metode *child process* sesuai dengan bahasa yang pengguna pilih sebelumnya pada *frontend*. Model *machine learning* kemudian akan melakukan penilaian esai otomatis dan akan menghasilkan *output* yang akan dikembalikan kepada *backend*. Setelah *backend* menerima hasil penilaian dari model *machine learning*, maka akan dikirimkan *frontend* dengan hasil penilaian untuk dapat dilihat oleh pengguna.

1. **Model Machine Learning Bahasa Inggris**

Model *machine learning* bahasa Inggris menerima nama CSV berisi jawaban pengajar dan jawaban pelajar dari *backend* melalui argumen sistem. Model bahasa Inggris memeriksa dokumen CSV yang dibuat oleh *backend* untuk diproses seperti pada Tabel 4.8.

|  |
| --- |
| full\_df = pd.read\_csv(f"{sys.argv[1]}.csv")  full\_df |

1. Model Bahasa Inggris Menerima Nama Berkas CSV

Apabila model telah selesai melakukan penilaian, maka model akan mengirimkan kembali hasil penilaian dalam bentuk JSON ke backend seperti pada Tabel 4.9.

|  |
| --- |
| print(full\_df.to\_json()) |

1. Model Bahasa Inggris Mengirim Hasil Penilaian
2. **Model Machine Learning Bahasa Indonesia**

Model *machine learning* bahasa Indonesia menerima jawaban pengajar dan jawaban pelajar berupa teks langsung dari *backend* melalui argumen sistem. Hal ini dapat dilihat pada Tabel 4.10.

|  |
| --- |
| # print("Masukkan Kunci Jawaban:")  kj = sys.argv[1]  # print("Masukkan Jawaban Siswa:")  jawaban = sys.argv[2] |

1. Model Bahasa Indonesia Menerima Teks Jawaban

Setelah melakukan penilaian, model akan menuliskan berkas teks berisi hasil penilaian dan sebuah *output* “done” sebagai penanda bahwa proses penilaian telah selesai untuk *backend*. Hal ini dapat dilihat pada Tabel 4.11 berikut ini.

|  |
| --- |
| newfile.write(f"{sctScore}\n")  newfile.close()  print("done") |

1. Model Bahasa Indonesia Selesai Melakukan Penilaian

### Hasil Scenario Testing

*Scenario testing* adalah uji coba yang dilakukan yang termasuk dalam cakupan metode pengembangan yang digunakan, yakni *Test-Driven Development (TDD)*. *Scenario testing* ini dilakukan untuk memastikan bahwa fitur yang dikembangkan bekerja sesuai dengan yang diinginkan. Pada kasus ini, *scenario testing* dilakukan untuk memastikan bahwa hasil keluaran pada saat melakukan penilaian esai otomatis pada model *machine learning* sama dengan hasil keluaran pada aplikasi pada masukkan spesifik. Hal ini membuktikan bahwa selama proses implementasi model *machine learning*, keaslian model tetap terjaga dan tidak terjadi adanya perubahan atas pengaruh implementasi pada aplikasi web yang dikembangkan pada penelitian ini. Dengan itu, pada Tabel 4.12 berikut dapat diperhatikan hasil *scenario testing* yang dilakukan pada pengembangan aplikasi terkait.

1. Hasil Scenario Testing

| **Skenario** | **Masukkan Jawaban Pengajar** | **Masukkan Jawaban Pelajar** | **Ekspektasi Keluaran** | **Keluaran** | **Sukses** |
| --- | --- | --- | --- | --- | --- |
| SC1 | Teks untuk uji coba jawaban tunggal “The testing stage can influence both the coding stage (phase 5) and the solution refinement stage (phase 7)” | Teks untuk uji coba tunggal “The implementation phase and the maintenance phase are effected” | 1.128299 dari 5.0 | 22% | Ya |
| SC2 | Teks untuk uji coba masal “To sumulate the behaviour of portions of the desired software product.” | Dokumen CSV berisi jawaban pelajar (“dataset.csv”) | Hasil Penilaian dari 0.0 sampai dengan 5.0 | Persentase Hasil Penilaian | Ya |
| 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. ” | 55 dari 100 | 55% | Ya |

Pada Tabel 4.12, dapat dilihat bahwa uji coba skenario berhasil untuk memenuhi ekspektasi awal. Hasil ekspektasi keluaran adalah hasil penilaian yang dilakukan pada model *machine learning* langsung, sedangkan keluaran adalah hasil penilaia yang dilakukan melalui implementasi aplikasi saat tahap pembuatan fungsi. Ekspektasi yang dipenuhi merupakan sinkronnya hasil keluaran penilaian esai melalui aplikasi maupun langsung melalui model *machine learning* yang digunakan.

Pada SC1, peneliti memberikan masukkan dua teks jawaban pengajar dan jawaban sebagaimana tertera pada Tabel 4.12. Model *machine learing* bahasa Inggris kemudian mengaplikasikan *Term Frequency – Inverse Document Frequency* (TF-IDF) untuk menghasilkan bobot nilai dari setiap kata pada kedua teks tersebut. Setelahnya, model akan mengaplikasikan *Cosine Similarity* dengan nilai dikalikan dengan 5 pada kedua teks yang telah diterapkan TF-IDF sebelumnya, yang berarti hasil penilaian berupa angka dari 0.0 hingga 5.0. Dengan ini, hasil menunjukkan 1.128299 pada model dan 22% pada aplikasi, dengan mengetahui bahwa model *machine learning* melakukan penilaian dari rentang angka 0.0 s/d 5.0, maka keluaran hasil penilaian identik dikarenakan 1.128299 merupakan 22% dari 5.0. Dengan ini dapat dinyatakan bahwa tidak ada keaslian yang berubah saat mengimplemetasikan model *machine learning* bahasa Inggris pada pengembangan aplikasi untuk metode penilaian tunggal.

Kemudian pada SC2 peneliti juga melakukan uji coba pada model *machine learning* berbahasa Inggris, namun kali ini dengan metode penilaian jamak. Ekspektasi bukanlah nilai satu jawaban, namun nilai banyak jawaban hasil penilaian model tersebut berupa JSON. Dan pada hasil pegujian skenario yang didapat, aplikasi berhasil melakukan penilaian dengan metode jamak menggunakan berkas CSV bawaan dari model *machine learning* bahasa Inggris yang digunakan.

Selanjutnya pada SC3 juga dilakuka uji coba terhadap model *machine learning* penilai esai otomatis berbahasa Indonesia. Peneliti memberi masukkan jawaban pengajar dan jawaban pelajar untuk diperiksa, kemudian membandingkan keluaran hasil jawabannya untuk dibandingkan. Hasil percobaan ini berhasil karena keluaran dari pengecekkan kedua paragraf tersebut identik. Nilai 55 dari model *machine learning* didapat dengan mengaplikasikan teknik TF-IDF dan *Cosine Similarity*, namun dengan skor dikalikan dengan 100 yang artinya hasil penilaian berada dari rentang 0 hingga 100. Dengan hasil 55% dari aplikasi dan 55 dari model *machine learning*, dapat dinyatakan bahwa hasil implementasi model pada aplikasi tidak memiliki perubahan yang dapat mempengaruhi hasil penilaian esai singkat.

Untuk model *machine learning* bahasa Indonesia, tidak dilakukan percobaan skenario penilaian jamak karena pada dasarnya model yang digunakan hanya dapat menilai secara tunggal. Namun pada aplikasi yang dikembangkan, dapat dilakukan penilaian secara jamak dengan melakukan penilaian tunggal secara berulang.

## Hasil Pre-Deployment Testing (Integration Testing)

Untuk memastikan bahwa aplikasi yang dikembangkan oleh penelitian ini berfungsi semestinya sebelum diluncurkan untuk pemakaian umum, perlu adanya tolak ukur untuk memastikan bahwa aplikasi akan bekerja sesuai dengan ekspektasi awal. Dengan itu, pada bab 3 telah dibuat uji coba yang harus diloloskan saat mengembangkan aplikasi ini. Berikut merupakan hasil uji coba tersebut.

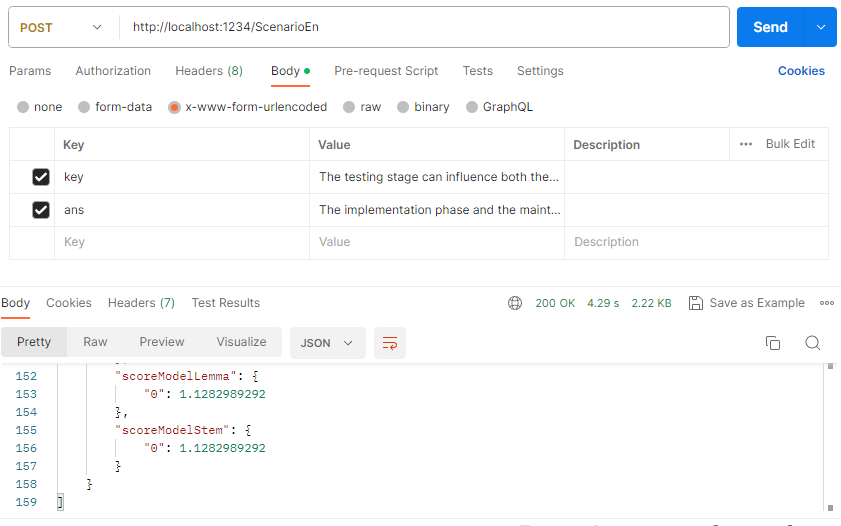
Diluar cakupan metode pengembangan *Test-Driven Development (TDD)*, dilakukan juga *Integration Testing* untuk memastikan aplikasi dapat berjalan dalam kondisi umum dengan hasil keluaran yang mirip seperti *scenario testing* namun tidak memiliki masukkan dan keluaran spesifik. Pada Tabel 4.13 dibawah ini dapat dilihat hasil pengujian unit yang telah dilakukan beserta hasilnya.

1. Hasil Integration Testing

| **ID** | **Metode** | **Input 1** | **Input 2** | **Output** | **Sukses** |
| --- | --- | --- | --- | --- | --- |
| IT1 | ScenarioEn(req, res) | Teks jawaban pengajar berbahasa Inggris. | Teks jawaban pelajar berbahasa Inggris. | JSON hasil penilaian tunggal berbahasa Inggris | Ya |
| IT2 | ScenarioEnTab(req, res) | To sumulate the behaviour of portions of the desired software product. | dataset.csv | JSON hasil penilaian jamak berbahasa Inggris. | Ya |
| IT3 | ScenarioId(req, res) | Teks jawaban pengajar berbahasa Indonesia. | Teks jawaban pelajar berbahasa Indonesia. | Hasil penilaian tunggal berbahasa Indonesia | Ya |

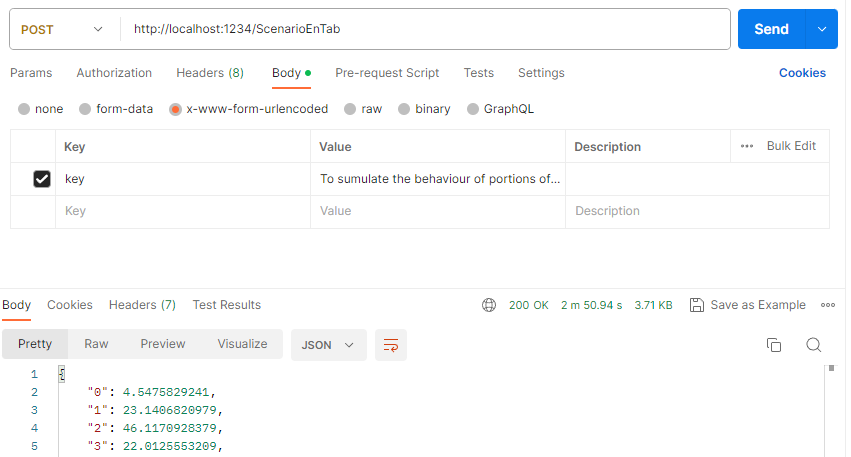
*Integration testing* pada Tabel 4.13 diatas dilakukan menggunakan aplikasi *Postman*, yaitu aplikasi yang membantu untuk melakukan permintaan API. Pada tahap pengujian, *integration testing* dilakukan dengan melakukan permintaan *post* pada API sebagaimana dispesifikasikan pada Tabel 4.13.

Pada IT1, aplikasi *Postman* melakukan permintaan *post* ke API “/ScenarioEn” dengan *body* menggunakan format *x-www-form-urlencoded* dengan *key “key”* dan *“ans”* merupakan teks pada UT1 *scenario testing* sebelumnya. API kemudian akan mencari *“key”* dan *“ans”* dan mendapatkan isi dari kedua *key* tersebut untuk melakukan penilaian dan mengirimkannya kembali kepada aplikasi *Postman*.



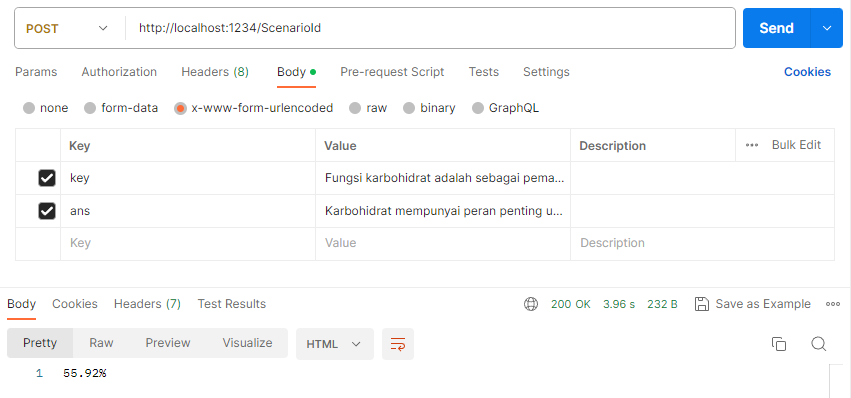
1. Hasil IT1

Pada IT2, aplikasi *Postman* melakukan permintaan *post* ke API “/ScenarioEnTab” dengan *body* menggunakan format *x-www-form-urlencoded* dengan *key “key”* berisi teks berdasarkan IT2 *scenario testing* sebelumnya. Pada *integration test* ini, API melakukan penilaian terhadap teks yang didapat dari *“key*” untuk melakukan penilaian dengan berkas CSV bawaan dari model *machine learning* seperti yang telah disebut sebelumnya pada *scenario testing*. Setelah berhasil, API akan mengembalikan hasil penilaian esai ke aplikasi *Postman* sebagai respon.



1. Hasil IT2

Kemudian pada IT3, aplikasi *Postman* melakukan permintaan *post* ke API “/ScenarioId” dengan *body* menggunakan format *x-www-form-urlencoded* dengan *key “key”* dan *“ans”* merupakan teks pada IT3 *scenario testing* sebelumnya. API kemudian akan mencari *“key”* dan *“ans”* dan mendapatkan isi dari kedua *key* tersebut untuk melakukan penilaian dan mengirimkannya kembali kepada aplikasi *Postman*.



1. Hasil IT3

Dengan berhasilnya *integration testing*, maka *scenario testing* juga berhasil dilakukan karena keduanya merupakan uji coba yang sinkron untuk suatu kasus umum dan spesifik. Maka dari itu, pengujian aplikasi penilaian esai otomatis ini dinyatakan berhasil memenuhi syarat dan ekspektasi awal.

## Hasil Peluncuran

Setelah melakukan pengembangan aplikasi untuk penelitian ini, aplikasi yang telah dikembangkan diluncurkan untuk dapat digunakan oleh banyak pengguna. Aplikasi diluncurkan dengan nama Easysay.

Aplikasi ini diluncurkan pada server IdCloudHost menggunakan *virtual machine* dengan *Operating System* Ubuntu, 2 VCPU, 2048 MB RAM, dan memori sebesar 20 GB. *Virtual Machine* pada IdCloudHost juga diatur untuk dapat menjalankan *Node.js* dan *Python* agar aplikasi bisa dijalankan sebagaimana di komputer pengembang. Dengan ini, aplikasi Easysay dapat diakses melalui peramban web pada tautan https://shercosta.github.io/easysay/.

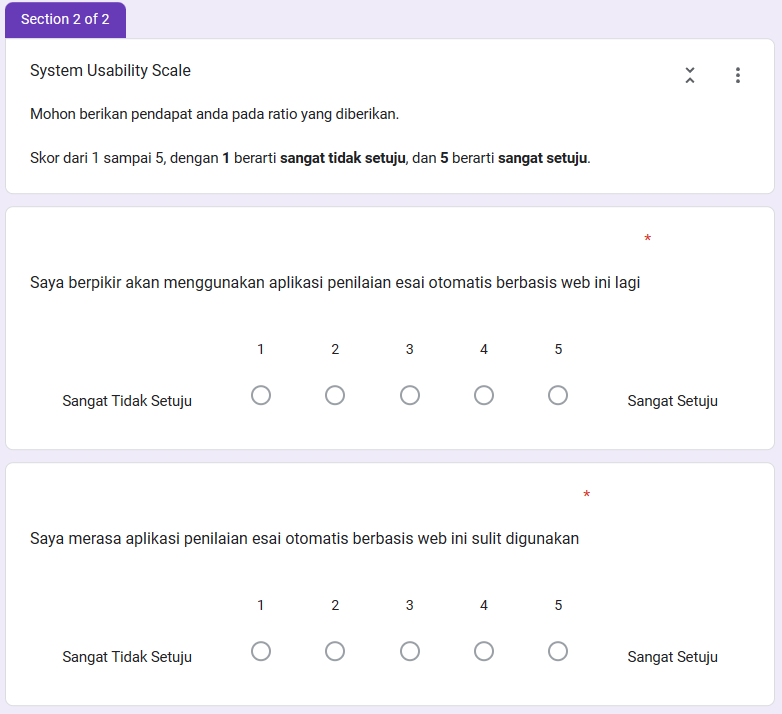
Apabila pada pengujian setelah peluncuran yang dilakukan, didapatkan *bug* atau *error* yang perlu diperbaiki, akan dilakukan peluncuran ulang terhadap aplikasi Easysay dengan spesifikasi server yang sama.

## Hasil Post-Deployment Testing

Setelah melakukan peluncuran aplikasi, peneliti melakukan pengujian lebih lanjut kepada narasumber awal dan pengguna lainnya untuk memastikan bahwa aplikasi yang dikembangkan memenuhi kebutuhan pengguna dan dapat berjalan dengan baik. Berikut merupakan pengujian yang dilakukan.

### Hasil Pengujian System Usability Scale

Setelah aplikasi Easysay dilucurkan, peneliti menyebarkan aplikasi agar dapat digunakan dan dinilai oleh pengguna, khususnya narasumber yang terlibat pada Tabel 3.1 sebelumnya. Peneliti juga meminta narasumber terkait untuk menyebarkan aplikasi tersebut dan kuesioner SUS pada pengajar lainnya. Pada Gambar berikut ini merupakan sampel dari kuesioner SUS yang diisi responden sebelumnya.



1. Sampel Google Form SUS

Semenjak peluncuran dilakukan dari tanggal 26 Agustus 2023, telah didapatkan 16 responden SUS dengan hasil sebagai berikut.

1. Responden SUS

| **Nama** | **Peran** | **Skor Untuk Pertannyaan** | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| Muhammad Lutfi, S.Pd | Guru | 5 | 1 | 4 | 1 | 5 | 1 | 4 | 2 | 3 | 2 |
| Ajeng Yeni Ratnasari, S.Pd | Guru | 5 | 3 | 4 | 2 | 5 | 3 | 5 | 2 | 4 | 1 |
| Dita Putri Bestari S.Pd | Guru | 5 | 1 | 3 | 1 | 4 | 2 | 5 | 1 | 5 | 2 |
| Ayu Nulantika S.Pd | Guru | 4 | 2 | 4 | 1 | 5 | 3 | 4 | 1 | 4 | 2 |
| Ahmad Arfandi, S.Kom | Guru | 4 | 1 | 5 | 1 | 4 | 2 | 5 | 2 | 5 | 3 |
| Virgawaty Dewy, S.Pd | Guru | 5 | 2 | 5 | 1 | 4 | 3 | 3 | 1 | 4 | 1 |
| Isma Nurfitriah, S.Pd | Guru | 4 | 3 | 5 | 2 | 5 | 2 | 4 | 1 | 3 | 2 |
| Renny Noviani Riandy, S.S | Guru | 5 | 1 | 4 | 1 | 5 | 1 | 5 | 2 | 5 | 1 |
| Sukma Erawan, S.Kom | Guru | 3 | 2 | 3 | 1 | 5 | 2 | 5 | 1 | 4 | 2 |
| Dr. Dian Mustika Dewi Priyandari, S. Sos., M.M | Dosen | 4 | 1 | 5 | 2 | 4 | 3 | 4 | 3 | 4 | 1 |
| Zamroni Riza S.E | Guru | 5 | 2 | 5 | 1 | 5 | 2 | 5 | 2 | 3 | 3 |
| Alan Syahwandi, S.Pd.I | Guru | 5 | 2 | 5 | 1 | 3 | 1 | 4 | 1 | 5 | 2 |
| Sahlul Efendi, S.pd | Guru | 4 | 2 | 5 | 1 | 4 | 1 | 3 | 2 | 4 | 2 |
| Yayah Shulhiyyah, S.Pd | Guru | 5 | 3 | 4 | 3 | 3 | 3 | 4 | 1 | 5 | 3 |
| Iyan Maulana, M.pd.i | Guru | 4 | 3 | 3 | 1 | 4 | 1 | 5 | 2 | 5 | 3 |
| Retno Dewi Kusumaningsih, S.Pd | Guru | 4 | 1 | 4 | 3 | 5 | 1 | 4 | 1 | 4 | 2 |

Berdasarkan hasil responden SUS tersebut, didapatkan rata-rata hasil skor tiap pertanyaan sebagai berikut

1. Rata-Rata Skor Tiap Pertanyaan SUS

| **No** | **Pertanyaan** | **Skor Rata-Rata** |
| --- | --- | --- |
| 1 | Saya berpikir akan menggunakan aplikasi penilaian esai otomatis berbasis web ini lagi | 4.4375 |
| 2 | Saya merasa aplikasi penilaian esai otomatis berbasis web ini sulit digunakan | 1.875 |
| 3 | Saya merasa aplikasi penilaian esai otomatis berbasis web ini mudah digunakan | 4.25 |
| 4 | Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan aplikasi penilaian esai otomatis berbasis web ini | 1.4375 |
| 5 | Saya merasa fitur-fitur aplikasi penilaian esai otomatis berbasis web ini berjalan dengan semestinya | 4.375 |
| 6 | Saya merasa banyak hal yang tidak konsisten / tidak serasi pada aplikasi penilaian esai otomatis berbasis web ini | 1.9375 |
| 7 | Saya merasa orang lain akan memahami cara menggunakan aplikasi penilaian esai otomatis berbasis web ini dengan cepat | 4.3125 |
| 8 | Saya merasa aplikasi penilaian esai otomatis berbasis web ini membingungkan | 1.5625 |
| 9 | Saya merasa tidak ada hambatan dalam menggunakan aplikasi penilaian esai otomatis berbasis web ini | 4.1875 |
| 10 | Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan aplikasi penilaian esai otomatis berbasis web ini | 2 |

Berdasarkan skor rata-rata penilaian SUS diatas, didapatkan bahwa pertanyaan nomor 9 merupakan pertanyaan positif dengan nilai paling rendah. Hal ini kemungkinan dikarenakan batasan aplikasi, khususnya pada fitur penilaian jamak yang hanya bisa digunakan apabila pengguna memiliki jawaban yang ingin diperiksa dalam bentuk berkas CSV.

Sedangkan pada pertanyaan nomor 10 merupakan pertanyaan negatif dengan skor paling tinggi, hal ini dapat terjadi dikarenakan aplikasi yang dikembangkan merupakan sebuah teknologi baru yang belum pernah ada sebelumnya. Dengan ini, pengguna perlu membiasakan diri terlebih dahulu untuk dapat memahami cara kerja aplikasi ini dari perspektif pengguna sepenuhnya.

Diketahui dengan rumus yang telah dijelaskan pada sub bab 2.2.11 tentang System Usability Scale, setiap skor pertanyaan bernomor ganjil merupakan skor dikurang 1, sedangkan setiap skor pertanyaan bernomor genap merupakan 5 dikurangi skor. Maka dari itu, didapat hasil penilaian sebagai berikut.

1. Skor SUS Ganjil dan Genap

| **Nama** | **Peran** | **Skor Untuk Pertannyaan** | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| Muhammad Lutfi, S.Pd | Guru | 4 | 4 | 3 | 4 | 4 | 4 | 3 | 3 | 2 | 3 |
| Ajeng Yeni Ratnasari, S.Pd | Guru | 4 | 2 | 3 | 3 | 4 | 2 | 4 | 3 | 3 | 4 |
| Dita Putri Bestari S.Pd | Guru | 4 | 4 | 2 | 4 | 3 | 3 | 4 | 4 | 4 | 3 |
| Ayu Nulantika S.Pd | Guru | 3 | 3 | 3 | 4 | 4 | 2 | 3 | 4 | 3 | 3 |
| Ahmad Arfandi, S.Kom | Guru | 3 | 4 | 4 | 4 | 3 | 3 | 4 | 3 | 4 | 2 |
| Virgawaty Dewy, S.Pd | Guru | 4 | 3 | 4 | 4 | 3 | 2 | 2 | 4 | 3 | 4 |
| Isma Nurfitriah, S.Pd | Guru | 3 | 2 | 4 | 3 | 4 | 3 | 3 | 4 | 2 | 3 |
| Renny Noviani Riandy, S.S | Guru | 4 | 4 | 3 | 4 | 4 | 4 | 4 | 3 | 4 | 4 |
| Sukma Erawan, S.Kom | Guru | 2 | 3 | 2 | 4 | 4 | 3 | 4 | 4 | 3 | 3 |
| Dr. Dian Mustika Dewi Priyandari, S. Sos., M.M | Dosen | 3 | 4 | 4 | 3 | 3 | 2 | 3 | 2 | 3 | 4 |
| Zamroni Riza S.E | Guru | 4 | 3 | 4 | 4 | 4 | 3 | 4 | 3 | 2 | 2 |
| Alan Syahwandi, S.Pd.I | Guru | 4 | 3 | 4 | 4 | 2 | 4 | 3 | 4 | 4 | 3 |
| Sahlul Efendi, S.pd | Guru | 3 | 3 | 4 | 4 | 3 | 4 | 2 | 3 | 3 | 3 |
| Yayah Shulhiyyah, S.Pd | Guru | 4 | 2 | 3 | 2 | 2 | 2 | 3 | 4 | 4 | 2 |
| Iyan Maulana, M.pd.i | Guru | 3 | 2 | 2 | 4 | 3 | 4 | 4 | 3 | 4 | 2 |
| Retno Dewi Kusumaningsih, S.Pd | Guru | 3 | 4 | 3 | 2 | 4 | 4 | 3 | 4 | 3 | 3 |

Setelah dilakukan hal tersebut, akan dijumlahkan hasil skor setiap responden dan dikalikan dengan 2.5. Dengan ini, didapatkan skor SUS lanjutan sebagai berikut.

1. Total Skor SUS Setiap Responden Dikali 2.5

| **Nama** | **Peran** | **Total Skor x2.5** |
| --- | --- | --- |
| Muhammad Lutfi, S.Pd | Guru | 85 |
| Ajeng Yeni Ratnasari, S.Pd | Guru | 80 |
| Dita Putri Bestari S.Pd | Guru | 87.5 |
| Ayu Nulantika S.Pd | Guru | 80 |
| Ahmad Arfandi, S.Kom | Guru | 85 |
| Virgawaty Dewy, S.Pd | Guru | 82.5 |
| Isma Nurfitriah, S.Pd | Guru | 77.5 |
| Renny Noviani Riandy, S.S | Guru | 95 |
| Sukma Erawan, S.Kom | Guru | 80 |
| Dr. Dian Mustika Dewi Priyandari, S. Sos., M.M | Dosen | 77.5 |
| Zamroni Riza S.E | Guru | 82.5 |
| Alan Syahwandi, S.Pd.I | Guru | 87.5 |
| Sahlul Efendi, S.pd | Guru | 80 |
| Yayah Shulhiyyah, S.Pd | Guru | 70 |
| Iyan Maulana, M.pd.i | Guru | 77.5 |
| Retno Dewi Kusumaningsih, S.Pd | Guru | 82.5 |
| **Rata-Rata** | | **81.875** |

Dengan kalkulasi tersebut, maka skor SUS aplikasi yang dikembangkan dari penelitian ini mendapat skor dengan nilai 81.875 dengan rentang nilai A yaitu sangat baik.

### Hasil Pengujian Black Box

Pada tahap ini, peneliti membandingkan ekspektasi hasil pengujian yang diharapkan pada Tabel 3.9 sebelumnya dengan hasil yang didapat setelah peluncuran aplikasi untuk memastikan bahwa aplikasi bekerja sesuai kebutuhan dan semestinya. Pada tabel dibawah ini dapat diperhatikan hasil pengujian *black box* yang dilakukan.

1. Hasil Pengujian Fungsional *Black Box*

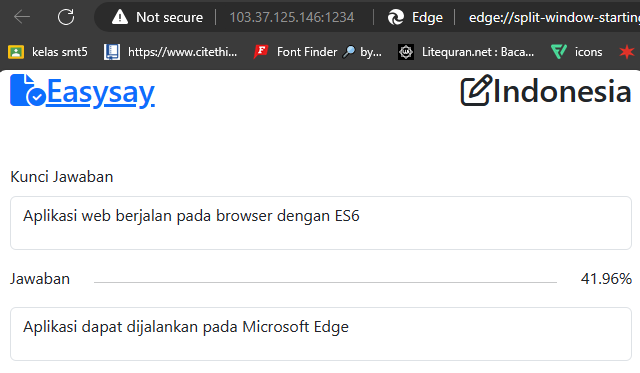
| **PARENT ID** | **ID** | **Pengujian** | **Ekspektasi** | **Hasil** |
| --- | --- | --- | --- | --- |
| F1A | F1A-1 | Aplikasi web dapat menerima masukkan teks sebagai jawaban pengajar | Aplikasi web berhasil menyimpan teks sebagai jawaban pengajar | Aplikasi web telah berhasil menyimpan teks sebagai jawaban pengajar |
| F1A-2 | Aplikasi web dapat menerima masukkan teks sebagai jawaban pelajar | Aplikasi web berhasil menerima teks sebagai jawaban pelajar | Aplikasi web telah 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 | Aplikasi web telah 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 web berhasil menerima masukkan dari fungsi F1A untuk digunakan pada penilaian otomatis tunggal bahasa Indonesia | Aplikasi web telah 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. | Aplikasi web telah berhasil menerima permintaan bahasa yang digunakan untuk penilaian dan menggunakan bahasa tersebut untuk penilaian. |
| F2A | F2A-1 | Aplikasi web dapat menerima masukkan berupa berkas berbentuk CSV sebagai kumpulan jawaban pelajar | Aplikasi web berhasil menyimpan berkas CSV sebagai kumpulan jawaban pelajar | Aplikasi web telah berhasil menyimpan berkas CSV 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 berkas CSV | Aplikasi web telah berhasil menyimpan nama kolom yang berisi nama pelajar pada berkas CSV |
| 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 berkas CSV | Aplikasi web telah berhasil menyimpan nama kolom yang berisi jawaban pelajar pada berkas CSV |
| 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 berkas CSV untuk proses penilaian masal | Aplikasi web telah berhasil menentukan pasangan nama dan jawaban pada berkas CSV 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 syarat penilaian pada algoritma penilaian otomatis bahasa Inggris | Aplikasi web berhasil membuat berkas CSV dari berkas CSV, nama, dan jawaban pelajar untuk diproses pada model penilaian otomatis bahasa Inggris | Aplikasi web telah berhasil membuat berkas CSV dari berkas CSV, nama, dan jawaban pelajar untuk 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 | Algoritma penilaian telah 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 | Aplikasi web telah 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 berkas CSV | Aplikasi web telah berhasil membuat dokumen .json setelah menentukan pasangan nama dan jawaban pada berkas CSV |
| 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 | Algoritma penilaian otomatis bahasa Indonesia telah 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 | Dokumen .json yang merupakan hasil penilaian otomatis dari algoritma bahasa Indonesia maupun bahasa Inggris telah berhasil ditampilkan pada halaman web |

Pada Tabel 4.15 diatas dapat diperhatikan hasil pengujian fungsional *black box* terhadap aplikasi yang dikembangkan pada penelitian ini telah berhasil dilakukan untuk setiap pengujiaannya. Tentunya ini merupakan hasil akhir pengujian dari peluncuran terakhir, karena sebelum ini didapatkan beberapa *error* yang telah diperbaiki oleh peneliti dan diluncurkan kembali pada server untuk kembali digunakan.

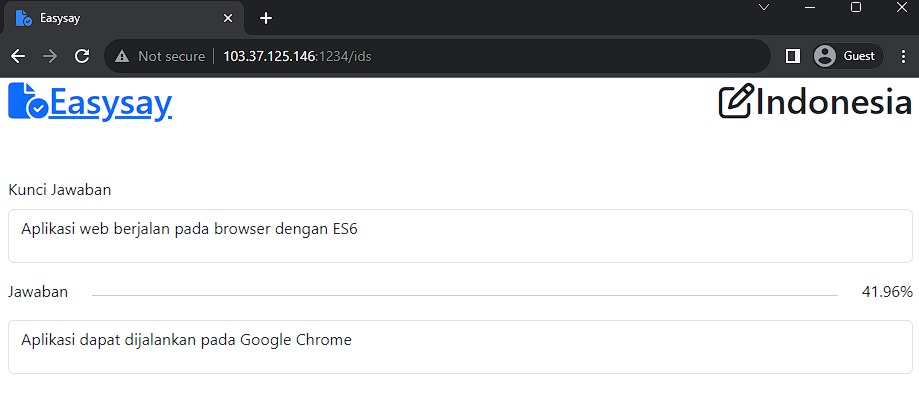
1. Hasil Pengujian Non Fungsional Black Box

| **PARENT ID** | **ID** | **Pengujian** | **Ekspektasi** | **Hasil** |
| --- | --- | --- | --- | --- |
| NF1 | NF1-1 | Aplikasi dapat dijalankan pada ponsel | Aplikasi web berhasil diakses dan digunakan pada ponsel | Aplikasi web telah berhasil diakses dan digunakan pada ponsel |
| NF1-2 | Aplikasi dapat dijalankan pada komputer | Aplikasi web berhasil diakses dan digunakan pada komputer | Aplikasi web telah 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 | Aplikasi web telah berhasil diakses dan dijalankan pada peramban web yang telah mendukung ES6 |
| NF3 | NF3 | Aplikasi tidak memiliki tampilan tumpang-tindih pada resolusi yang berbeda | Aplikasi web berhasil beradaptasi pada setiap resolusi dimana perlu ditampilkan tanpa adanya komponen yang tumpang-tindih | Aplikasi web telah berhasil beradaptasi pada setiap resolusi dimana perlu ditampilkan tanpa adanya komponen yang tumpang-tindih |

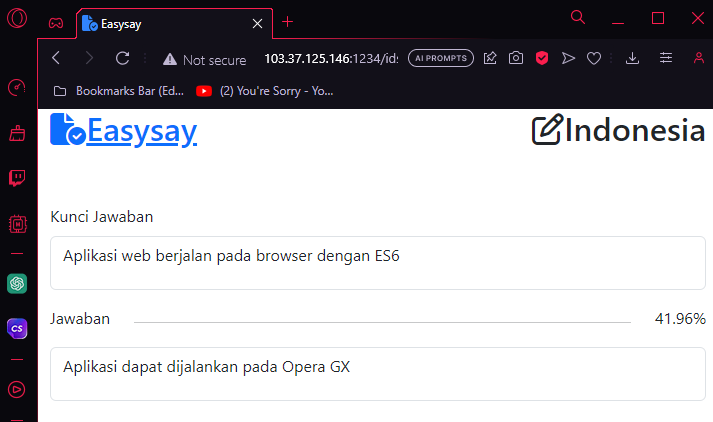
Pada Tabel 4.16, merupakan hasil pengujian *black box* non fungsional. Pada pengujian ini dapat disimpulkan bahwa aplikasi yang dikembangkan penelitian ini memiliki tampilan responsif , tidak memiliki tampilan tumpang-tindih (sebagaimana pada sub bab 4.1.1), dan dapat dijalankan pada setiap peramban web yang mendukung ES6. Gambar dibawah ini merupakan bukti bahwa aplikasi ini dapat digunakan untuk peramban web yang mendukung ES6.



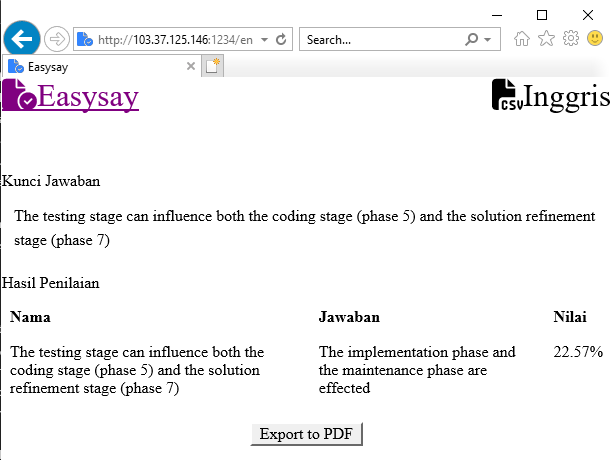
1. Aplikasi pada Microsoft Edge



1. Aplikasi pada Google Chrome

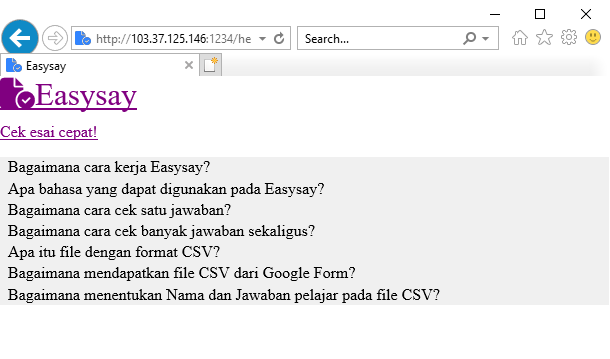


1. Aplikasi pada Opera GX



1. Aplikasi pada Internet Explorer (ekspor ke PDF)

Saat menggunakan Internet Explorer yang tidak mendukung ES6, aplikasi tetap dapat digunakan namun terdapat perbedaan pada tampilan yang ditunjukkan. Selain itu, karena fitur “Export to PDF” dijalankan menggunakan *JavaScript*, Internet Explorer tidak dapat mengekspor hasil penilaian menjadi PDF.



1. Aplikasi pada Internet Explorer (bantuan)

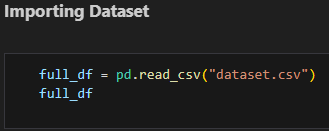
Selain itu, pada halaman bantuan dimana pengguna dapat mengetahui cara menggunakan aplikasi ini juga tidak dapat digunakan. Pengguna tidak dapat melihat isi dari setiap bantuan yang disediakan oleh pengembang. Dengan ini dapat disimpulkan bahwa aplikasi yang dikembangkan hanya layak untuk digunakan pada peramban web yang mendukung ES6.

## Pembahasan

Penelitian ini menghasilkan sebuah produk untuk hasil akhirnya yaitu berupa aplikasi penilaian esai singkat yang dapat melakukan penilaian esai otomatis menggunakan bahasa Indonesia dan bahasa Inggris. Pada tahap pengembangan aplikasi, peneliti mengimplementasikan metode pengembangan perangkat lunak *Test-Driven Development* (TDD) yang merupakan SDLC Agile.

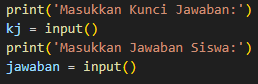
Pengembangan dilakukan dengan mengumpulkan kebutuhan dari 10 narasumber yang merupakan pengajar di Indonesia, tahap ini termasuk dalam tahap *red* pada metode pengembangan TDD. Setelah terkumpulnya kebutuhan ini, peneliti menuliskan pengujian berupa *scenario testing* yang perlu diloloskan mencapai kesuksesan aplikasi yang dibangun. Setelahnya, dibuatlah *mockup* aplikasi yang dapat memenuhi kebutuhan yang terkumpul, seperti masukkan teks, tombol memilih metode penilaian, dan lainnya. Setelahya, peneliti akan masuk ke tahap antara *red* dan *green*, yaitu menuliskan kode untuk setiap kebutuhan fungsional dan non-fungsional yang terdata pada Tabel 3.4 dan 3.5. Apabila suatu fungsi / fitur berhasil memenuhi uji coba skenario yang dituliskan, berarti peneliti akan memasuki tahap *green* dari metode pengembangan TDD, dengan ini pengembang dapat membersihkan / refaktor kode pada aplikasi yang dibangunnya yang merupakan tahap *refactor* pada metode pengembangan TDD. Setelah refaktor kode dilakukan, pengembang dapat langsung mengembangkan fungsi / fitur selanjutnya. Pada siklus hidup ini lah aplikasi penilaian esai singkat otomatis Easysay dibuat.

Ada 2 model *machine learning* yang digunakan pada aplikasi yang dikembangkan, model yang bekerja bahasa Indonesia, dan bahasa Inggris. Masing-masing model memiliki cara kerja yang berbeda.



1. Cara Penilaian Model *Machine Learning* Bahasa Inggris

Pada model *machine learning* bahasa Inggris yang dapat dilihat pada Gambar 4.16 diatas, model bekerja dengan cara menilai berkas CSV yang sudah memiliki pasangan jawaban pengajar dan jawaban pengajar pada berkas bernama “dataset.csv”, hal ini menunjukkan bahwa model bahasa Inggris secara mendasar melakukan penilaian secara jamak. Pada implementasi, untuk melakukan penilaian tunggal, *backend* aplikasi yang dikembangkan akan membuat berkas CSV dengan satu pasang jawaban pengajar dan pelajar. Selanjutnya untuk melakukan penilaian jamak, *backend* aplikasi membuat berkas yang dapat mendeteksi nama dari *header* kolom berisi jawaban pelajar, dan membuat kolom baru berisi jawaban pengajar yang didapat dari masukkan pengguna. Dengan ini, dapat dibuat berkas CSV dengan pasangan jawaban pengajar dan jawaban pelajar yang berbeda. Kemudian dengan teknik *child-process*, *backend* menjalankan model *machine learning* bahasa Inggris ini dan mengirimkan juga berkas CSV yang perlu diperiksa. Setelah selesai, model akan mengirimkan ke *backend* sebuah JSON yang berisi hasil penilaian yang dapat ditampilkan pada *frontend*. Setelah selesai, berkas CSV yang telah selesai digunakan akan dihapus untuk tidak memenuhi penyimpanan server.



1. Cara Penilaian Model *Machine Learning* Bahasa Indonesia

Kemudian pada model *machine learning* bahasa Indonesia yang dapat dilihat pada Gambar 4.17, model bekerja dengan memeriksa satu masukkan jawaban pengajar dan pelajar melalui *input*. Dengan ini, model diketahui untuk bekerja secara tunggal dalam melakukan penilaian esainya. Untuk penilaian esai tunggal, *backend* menjalankan model dengan metode *child-process* dengan argumen bawaan yaitu jawaban pengajar dan jawaban pelajar untuk diperiksa. Selanjutnya model akan membuat berkas teks yang berisi skor penilaian dan mengirimkan argumen berisikan “done” pada *backend* untuk menandakan bahwa proses penilaian pada model telah selesai. Dengan ini, *backend* akan membuka berkas teks berisi skor tersebut sebagai JSON untuk ditampilkan pada *frontend* dan menghapus berkas teks tersebut agar tidak memenuhi penyimpanan pada server. Sedangkan untuk penilaian jamak, *backend* akan mendapatkan pasangan jawaban pengajar dan pelajar dari setiap baris pada berkas CSV yang diunggah oleh pengguna dan menjalankan *child-process* pada model untuk setiap pasangan jawaban tersebut, dalam kata lain, proses penilaian jamak yang dilakukan pada model bahasa Indonesia adalah proses penilaian tunggal yang dilakukan secara berulang. Apabila penilaian selesai, maka *backend* akan menghapus berkas CSV dan teks berisi penilaian setelah menampilkannya pada *frontend*.

Untuk memastikan bahwa selama proses implementasi model *machine learning* pada aplikasi web ini tidak terjadi pengaruh terhadap penilaian yang dilakukan, dilakukanlah pengujian skenario. Pengujian dilakukan dengan melakukan penilaian dengan masukkan yang sama pada aplikasi web dan pada model *machine learning* langsung. Skor yang dihasilkan oleh model bahasa Inggris adalah nilai numerik dari 0.0 hingga 5.0, dan nilai numerik dari 0 hingga 100 untuk model bahasa Indonesia. Pada pengujian yang dilakukan pada *scenario testing* SC1, didapatkan skor dari model bahasa Inggris senilai 1.128299 yang merupakan 22% dari 5.0, yang mana aplikasi web mendapatkan hasil 22%. Kemudian pada model bahasa Indonesia, khususnya pada *scenario testing* SC3, didapatkan skor senilai 55 dari 100, yang mana aplikasi web mendapatkan skor senilai 55%. Dengan ini, dapat dipastikan bahwa tidak ada perbedaan antara hasil penilaian pada model yang telah diimplementasikan pada aplikasi web dan model aslinya.

Aplikasi penilai esai singkat ini dapat digunakan pada peramban web apabila pengguna berhak mengakses laman web yang sudah diluncurkan, atau menjalankannya pada komputer lokal.

Untuk menjalankannya pada komputer lokal, pengguna perlu *runtimer* Node.js dengan minimal versi 16.16.0 dan Python dengan minimal versi 3.11.5. Untuk spesifikasi processor, ram, dan penyimpanan tidak memiliki minimal, namun apabila spesifikasi yang digunakan lebih besar maka akan lebih baik untuk performa. Selanjutnya juga pengguna perlu menginstal seluruh *dependencies* Node.js dan Python yang digunakan pada aplikasi ini.

Kemudian apabila pengguna ingin melakukan penilaian, pengguna dapat membuka aplikasi pada laman web ataupun lokal, pada tahap ini pengguna akan diarahkan ke halaman utama. Pada halaman ini, pengguna dapat memilih bahasa yang ingin digunakan pada penilaian esai. Selanjutnya pengguna dapat memilih metode penilaian apa yang ingin digunakan untuk melakukan penilaian, dengan ini pengguna akan diarahkan ke halaman yang bersangkutan untuk bahasa dan metode penilaian terkait.

Apabila pengguna memilih metode penilaian tunggal, pengguna akan berada di halaman dimana pengguna dapat memasukkan satu teks pada kolom “Jawaban Pelajar” dan satu teks pada kolom “Jawaban Pengajar”. Setelahnya, pengguna dapat menekan “periksa” dan aplikasi web akan melakukan penilaian otomatis pada esai yang disediakan. Setelah selesai, pengguna akan diarahkan ke halaman hasil penilaian tunggal dimana pengguna dapat melihat hasil persentase penilaian esai.

Sedangkan apabila pengguna memilih metode pennilaian jamak, pengguna akan diarahkan ke halaman dimana pengguna dapat memasukkan jawaban pengajar, mengunggah berkas CSV berisi jawaban pelajar yang ingin diperiksa, dan menyediakan nama kolom “nama” dan “jawaban pelajar” pada kolom yang disediakan. Hal ini dilakukan agar aplikasi mengetahui mana kolom yang berisi nama dan jawaban pelajar pada berkas CSV untuk dibandingkan dengan jawaban pengajar. Setelah pengguna menekan tombol “periksa”, pengguna diarahkan ke halaman hasil penilaian jamak dimana pengguna dapat melihat pasangan nama dan jawaban pelajar serta nilai hasil penilaian. Pengguna juga dapat mengekspor hasil penilaian ke PDF apabila diperlukan untuk kepentingan pribadi.

# BAB V KESIMPULAN DAN SARAN

## Kesimpulan

Berdasarkan hasil akhir penelitian yang dilakukan ini, maka dapat disimpulkan bahwa:

1. Implementasi aplikasi penilaian esai singkat dapat dilakukan menggunakan SDLC *Test-Driven Development* (TDD) dengan membangun sebuah fungsi / fitur hingga berhasil terlebih dahulu, kemudian melakukan refaktor agar kode tidak berantakan. Setelahya, pengembang dapat membangun fungsi / fitur selanjutnya, dan proses ini akan berulang hingga tidak ada lagi fungsi / fitur yang hendak ditambahkan.
2. Model *pre-trained machine learning* dapat diterapkan pada aplikasi dengan menerapkan konsep *child-process* dimana sebuah proses menjalankan sebuah proses turunan lainnya. Pada aplikasi yang dibangun, *backend* dari aplikasi akan menjalankan model untuk melakukan penilaian pada esai yang ingin dinilai dan mendapatkan hasilnya apabila proses penilaian telah selesai dilakukan.
3. Hasil penilaian pada aplikasi dan pada model *machine learning* dapat dipastikan sama dengan membandingkan hasil penilaian yang didapat dari aplikasi dan model *machine learning* terkait secara langsung. Skor yang dihasilkan kedua sumber ini kemudian dikonversi menjadi rentang 0 hingga 100 dan dibandingkan hasilnya. Hasil menunjukkan bahwa hasil penilaian pada aplikasi yang dikembangkan dan model *machine learning* sama, yang berarti tidak ada perubahan selama proses implementasi model pada saat mengembangkan aplikasi.
4. Untuk pengguna, aplikasi penilaian esai singkat dapat dilakukan dengan menjalankannya pada peramban web yang mendukung ES6 dan pengguna dapat melihat tahapan cara menggunakan aplikasi pada halaman bantuan. Sedangkan untuk sisi teknikal, aplikasi dapat digunakan pada mesin yang menjalankan *runtimer* Node.js untuk menjalankan *server* aplikasi, dan Python sebagai interpreter untuk menjalankan model *machine learning* yang digunakan pada penelitian ini.

## Saran

Pada akhir penelitian ini, peneliti menemukan beberapa saran yang dapat diterapkan terkait penelitian ini ataupun kelanjutan penelitian ini kedepannya.

1. **Saran untuk Lembaga Edukasi**: Seperti yang telah dijelaskan pada latar belakang terkait tidak efisiennya penilaian esai apabila dilakukan secara manual. Dengan landasan itu, sebaiknya setiap lembaga edukasi (sekolah, universitas, atau institusi) dapat menyediakan aplikasi ini pada server untuk dapat digunakan oleh pihak pengajar yang hendak melakukan penilaian esai agar proses penilaian esai dapat dilakukan lebih cepat.
2. **Saran untuk Peneliti Selanjutnya**: Pada aplikasi yang dikembangkan penelitian ini, hanya memiliki dua bahasa yang dapat digunakan untuk penilaian esai yaitu bahasa Indonesia dan Inggris. Hal ini dikarenakan penelitian ini menggunakan 2 model *machine learning* yang dapat melakukan pemeriksaan esai otomatis menggunakan bahasa Indonesia dan Inggris. Apabila di masa depan ada model *machine learning* yang dapat melakukan hal serupa dengan bahasa lain, maka peneliti selanjutnya dapat mengimplementasi model dengan bahasa baru tersebut kedalam aplikasi ini agar semakin dapat digunakan untuk banyak bahasa.

**DAFTAR PUSTAKA (minimal 20)**

[1] U. Nations, “Population.” Accessed: Dec. 25, 2022. [Online]. Available: https://www.un.org/en/global-issues/population

[2] B. P. Statistik, “Jumlah Penduduk Pertengahan Tahun (Ribu Jiwa), 2020-2022.” Accessed: Dec. 26, 2022. [Online]. Available: https://www.bps.go.id/indicator/12/1975/1/jumlah-penduduk-pertengahan-tahun.html

[3] B. P. Statistik, “Jumlah Penduduk Menurut Kelompok Umur dan Jenis Kelamin, 2021.” Accessed: Dec. 26, 2022. [Online]. Available: https://www.bps.go.id/indikator/indikator/view\_data\_pub/0000/api\_pub/YW40a21pdTU1cnJxOGt6dm43ZEdoZz09/da\_03/1

[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: Association for Computational Linguistics, 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)*, Caceres: IEEE, 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: IEEE, Apr. 2019, pp. 689–693. doi: 10.1109/IEA.2019.8714889.

[10] H. K. N. Leung and L. White, “A study of integration testing and software regression at the integration level,” in *Proceedings. Conference on Software Maintenance 1990*, San Diego, CA, USA: IEEE Comput. Soc. Press, 1990, pp. 290–301. doi: 10.1109/ICSM.1990.131377.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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)*, Y. Lu and C. Cheng, Eds., Beijing, China: SPIE, Dec. 2022, p. 25. doi: 10.1117/12.2661778.

[17] I. Glantz and H. Hurtig, “Express.js and Ktor web server performance A comparative study”.

[18] D. Smilkov *et al.*, “TensorFlow.js: Machine Learning for the Web and Beyond”.

[19] S. L. Ingólfsdóttir, 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

[20] 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.

[21] C. Kaner, “An Introduction to Scenario Testing”.

[22] 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/

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] A. Ismail and K. S. Kuppusamy, “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.

[28] 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: IEEE, May 2019, pp. 304–315. doi: 10.1109/ICSE.2019.00045.

[29] M. Shcherbakov, M. Balliu, and C.-A. Staicu, “Silent Spring: Prototype Pollution Leads to Remote Code Execution in Node.js”.

[30] H. Brar, T. Kaur, and Y. Rajoria, “The Better Comparison between PHP, Python-web & Node.js,” *web ..*.

[31] A. Romanelli, S. Serbout, and C. Pautasso, “ExpressO: From Express.js implementation code to OpenAPI interface descriptions”.

[32] 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.

[33] S. Hamza, M. Sarosa, and P. B. Santoso, “Sistem Koreksi Soal Essay Otomatis Dengan Menggunakan Metode Rabin Karp,” vol. 7, no. 2, 2013.

[34] 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.

[35] 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.

[36] 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

[37] 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.

[38] M. E. Delamaro, J. C. Maidonado, and A. P. Mathur, “Interface Mutation: an approach for integration testing,” *IEEE Trans. Softw. Eng.*, vol. 27, no. 3, pp. 228–247, Mar. 2001, doi: 10.1109/32.910859.

[39] 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: IEEE, May 2018, pp. 117–122. doi: 10.1109/DESSERT.2018.8409111.

[40] 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.

[41] A. Dearle, “Software Deployment, Past, Present and Future,” in *Future of Software Engineering (FOSE ’07)*, Minneapolis, MN: IEEE, May 2007, pp. 269–284. doi: 10.1109/FOSE.2007.20.

[42] R. S. Hall, D. Heimbigner, and A. L. Wolf, “A Cooperative Approach to Support Software Deployment Using the Software Dock”.

**LAMPIRAN**

**LAMPIRAN 1 Isi Lampiran**

Isian lampiran, dapat berupa:

* + 1. Foto pengujian di lapangan
    2. Gambar hasil aplikasi
    3. Tampilan UI / UX
    4. Hasil pengujian, contoh : kuesioner, wawancara, bukti pengujian
    5. Source code