

**IMPLEMENTASI ALGORITMA RATCLIFFE/OBERSHELP PADA
SISTEM PENILAIAN OTOMATIS SOAL URAIAN**

SKRIPSI

M TAUFIK BASKORO

171402086



PROGRAM STUDI S1 TEKNOLOGI INFORMASI

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS SUMATERA UTARA

MEDAN

2024

IMPLEMENTASI ALGORITMA RATCLIFFE/OBERSHELP PADA
SISTEM PENILAIAN OTOMATIS SOAL URAIAN

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah Sarjana
Teknologi Informasi

M TAUFIK BASKORO

171402086



PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS SUMATERA UTARA

MEDAN

2024

PERSETUJUAN

Judul : IMPLEMENTASI ALGORITMA RATCLIFFE /
OBERSHELP SISTEM PENILAIAN OTOMATIS
SOAL URAIAN
Kategori : SKRIPSI
Nama : M TAUFIK BASKORO
Nomor Induk Mahasiswa : 171402086
Program Studi : TEKNOLOGI INFORMASI
Fakultas : FAKULTAS ILMU KOMPUTER DAN
TEKNOLOGI INFORMASI

Medan, 21 Mei 2024

Komisi Pembimbing

Pembimbing 2

Pembimbing 1



Prof. Dr. Drs. Opim Salim Sitompul
M.Sc



Dr. Muhammad Anggia Muchtar
S.T., MM.IT.

Diketahui/disetujui oleh

Program Studi S1 Teknologi Informasi

Ketua


Dedy Arisandi, S.T., M.Kom.
NIP. 197908312009121002

PERNYATAAN**IMPLEMENTASI ALGORITMA RATCLIFFLE/OBERSHELP PADA SISTEM
PENILAIAN OTOMATIS SOAL URAIAN****SKRIPSI**

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, 21 Mei 2024

M TAUFIK BASKORO

171402086

UCAPAN TERIMA KASIH

Penulis merasa berterima kasih kepada Allah SWT atas anugerah kesehatan, rahmat, dan bimbingan-Nya. Karena izin-Nya lah skripsi ini dapat terselesaikan dengan sebaik-baiknya. Shalawat serta salam penulis sampaikan kepada Muhammad SAW agar diberikannya syafaat beliau di hari akhir nanti.

Penulisan skripsi ini ditujukan untuk menyelesaikan pendidikan penulis pada jenjang Strata 1 Program Studi Teknologi Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara sebagai syarat memperoleh gelar Sarjana Komputer. Skripsi ini membawa judul “Implementasi Algoritma Ratcliffe/Obershelp pada Sistem Penilaian Otomatis Soal Uraian”.

Penghargaan dan rasa terima kasih penulis ditujukan kepada kedua orang tua penulis, Ibu Ir. Tri Puji Astuti dan Ayah Khairizal Tanjung A.Md, yang senantiasa menyemangati dan mendoakan penulis dari awal perkuliahan sampai saat ini.

Rasa terima kasih penulis juga ditujukan kepada semua yang telah Memberikan inspirasi dan bimbingan selama perjalanan akademik hingga pembuatan skripsi ini, tanpa dukungan serta doa-doa dari rekan semua, sulit rasanya penulis menyelesaikan skripsi ini.

Pihak lain yang ikut terlibat antara lain :

1. Bapak Dr. Muryanto Amin, S.Sos., M.Si selaku Rektor Universitas Sumatera Utara
2. Ibu Dr. Maya Silvi Lydia B.Sc., M.Sc. sebagai Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara
3. Bapak Dedy Arisandi ST., M.Kom. sebagai Ketua Program Studi S1 Teknologi Informasi Universitas Sumatera Utara
4. Bapak Ivan Jaya, S.T., M.Kom. sebagai Sekretaris Program Studi S1 Teknologi Informasi Universitas Sumatera Utara
5. Bapak Dr. Muhammad Anggia Muchtar S.T., MM.IT. sebagai Dosen Pembimbing I dan Bapak Prof. Dr. Drs. Opim Salim Sitompul M.Sc sebagai Dosen

Pembimbing II, yang telah meluangkan banyak waktu untuk membimbing dan memberikan saran kepada penulis

6. Bapak Dr. Romi Fadillah Rahmat B.Comp.Sc., M.Sc. sebagai Dosen Pembimbing I dan Ibu Fanindia Purnamasari S.TI., M.IT sebagai Dosen Pembimbing II yang telah memberikan masukan serta saran dalam menyempurnakan skripsi ini
7. Semua tenaga pengajar dan staff dari Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara yang telah memberikan penulis ilmu yang bermanfaat
8. Partner, Sahabat, serta teman-teman dekat penulis, Fadhilah Ramadhani, Muhammad Ulwan Azmi, Deo Pranata Silitonga, Muhammad Farras Siraj Polem, dan Ade Rizky yang telah memberikan saran serta bersama-sama dengan penulis selama masa perkuliahan
9. Semua individu yang membantu serta membersamai penulis dalam proses penelitian ini.

Semoga Rahmat Allah Yang Maha Esa turut memberikan berkah kepada semua individu yang telah terlibat, baik dalam bentuk dukungan, doa, atau saran kepada penulis dalam proses penyelesaian skripsi.

Medan, 21 Mei 2023

Penulis

ABSTRAK

Penggunaan sistem pembelajaran elektronik (*e-learning*) sedang diterapkan di setiap jenjang pendidikan. Dari tahap pendidikan dasar hingga pendidikan tinggi. Salah satu penerapan yang dilakukan adalah penerapan sistem ujian. Ujian terdiri dari beberapa tipe seperti pilihan berganda, uraian, pencarian kesamaan, jawaban singkat, benar/salah, dan kalkulasi. Ujian uraian sendiri terdiri atas dua jenis, yaitu ujian uraian yang bersifat subjektif dan ujian uraian bersifat objektif. Ujian uraian yang bersifat subjektif lebih mengarah ke pendapat seseorang terhadap suatu kondisi atau peristiwa, sedangkan ujian uraian yang bersifat objektif merupakan ujian uraian yang memiliki sebuah jawaban pasti. Sistem *e-learning* di Indonesia ini hampir semua belum memiliki kemampuan untuk memeriksa jawaban ujian uraian yang bersifat objektif secara otomatis. Maka dari itu diperlukan sistem untuk melakukan penilaian secara otomatis pada soal uraian yang bersifat objektif. Penilaian otomatis pada penelitian ini menggunakan penerapan algoritma Sastrawi untuk menemukan asal kata dan algoritma *Winnowing* untuk melakukan *synonym recognition* dalam membantu algoritma *Ratcliffe/Obershelp* menentukan nilai akurasi yang didapat. Data penelitian ini berasal dari ujian yang diikuti oleh 25 murid, yang terdiri dari 4 soal juga ulangan yang diikuti oleh 12 murid, yang terdiri dari 5 soal. Sistem ini memiliki rata-rata selisih 7,7 poin dibandingkan dengan penilaian yang dilakukan secara manual.

Kata Kunci : *Ratcliffe/Obershelp*, Sistem pembelajaran elektronik, Ujian uraian

IMPLEMENTATION OF RACLIFFE/OBERSHELP ALGORITHM ON AN AUTOMATIC SCORING SYSTEM FOR ESSAY QUESTION

ABSTRACT

The use of electronic learning systems (e-learning) is being implemented at every level of education. From primary education to higher education. One of the applications carried out is the implementation of an examination system. The exam consists of several types such as multiple choice, description, similarity search, short answer, true/false, and calculation. The description exam itself consists of two types, namely the subjective description exam and the objective description exam. A subjective description test focuses more on a person's opinion regarding a condition or event, while an objective description test is a description test that has a definite answer. Almost all e-learning systems in Indonesia do not have the ability to automatically check answers to objective essay exams. Therefore, a system is needed to carry out automatic assessments on objective description questions. The automatic assessment in this research uses the application of the Sastrawi algorithm to find the origin of words and the Winnowing algorithm to perform synonym recognition to help the Ratcliffe/Obershelp algorithm determine the accuracy value obtained. This research data comes from an exam taken by 25 students, which consists of 4 questions and a test taken by 12 students, which consists of 5 questions. This system has an average difference of 7,7 points compared to the assessment carried out manually.

Keywords: E-Learning, Essay, Ratcliffe/Obershelp

DAFTAR ISI

	Hal.
PERSETUJUAN	Error! Bookmark not defined.
PERNYATAAN	ii
UCAPAN TERIMA KASIH	iii
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
Daftar Tabel	ix
Daftar Gambar	xi
BAB 1	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan Penelitian	2
1.4. Batasan Masalah	2
1.5. Manfaat Penelitian	3
1.6. Metodologi Penelitian	3
BAB 2	4
2.1. Ujian	4
2.2. Kata Sinonim	4
2.3. <i>Text Processing</i>	5
2.4. <i>Algoritma Ratcliffe/Obershelp</i>	5

2.5. <i>Synonym Recognition</i>	6
2.6. <i>Rolling Hash</i>	7
2.7. <i>Algoritma Winnowing</i>	7
2.8. <i>Aplikasi Web</i>	8
2.9. <i>Application Programming Interface (API)</i>	8
2.10. <i>Metode Evaluasi</i>	9
2.11. <i>Penelitian Terdahulu</i>	10
BAB 3	14
3.1. <i>Data Penelitian</i>	14
3.2. <i>Analisis Sistem</i>	15
3.3. <i>Perancangan Sistem</i>	16
3.3.1. <i>Arsitektur Umum</i>	16
3.3.2. <i>Pseudo code algoritma kesamaan teks</i>	23
3.3.3. <i>Perancangan antar muka sistem</i>	29
3.4. <i>Tahapan Evaluasi</i>	36
BAB 4	39
4.1. <i>Implementasi Sistem</i>	39
4.2. <i>Pengujian Sistem</i>	44
BAB 5	52
5.1. <i>Kesimpulan</i>	52
5.2. <i>Saran</i>	52
DAFTAR PUSTAKA	53
LAMPIRAN 1: Data Uji	55

Daftar Tabel

Tabel 2.1. Penelitian Terdahulu	11
Tabel 3.1. Data Soal Ujian	14
Tabel 3.1. Data Soal Ujian (Lanjutan)	15
Tabel 3.2. Data Soal Ulangan	15
Tabel 3.3. Contoh Case Folding	18
Tabel 3.4. Contoh Cleansing	19
Tabel 3.5. Contoh Tokenization	19
Tabel 3.5. Contoh Tokenization (Lanjutan)	20
Tabel 3.6. Contoh Stemming	20
Tabel 3.7. Contoh Synonym Recognition	21
Tabel 3.8. Penilaian Skor untuk Ujian	22
Tabel 3.9. Penilaian Skor untuk Ulangan	22
Tabel 3.10. Contoh Variabel untuk Tahapan Evaluasi	37
Tabel 4.1. Hasil Akurasi Ujian pada Sistem	44
Tabel 4.1. Hasil Akurasi Ujian pada Sistem (Lanjutan)	45
Tabel 4.2. Hasil Akurasi Ulangan pada Sistem	45
Tabel 4.2. Hasil Akurasi Ulangan pada Sistem (Lanjutan)	46

Tabel 4.3. Hasil Konversi Nilai Akurasi Ujian yang Diperoleh	46
Tabel 4.3. Hasil Konversi Nilai Akurasi Ujian yang Diperoleh (Lanjutan)	47
Tabel 4.4. Hasil Konversi Nilai Akurasi Ulangan yang Diperoleh	47
Tabel 4.5. Perbandingan Sistem dengan Cara Manual pada Ujian	48
Tabel 4.5. Perbandingan Sistem dengan Cara Manual pada Ujian (Lanjutan)	49
Tabel 4.6. Perbandingan Sistem dengan Cara Manual pada Ulangan	49
Tabel 4.7. Variabel Proses Pengujian	50

Daftar Gambar

Gambar 3.1. Arsitektur Umum	17
Gambar 3.2. Halaman Masuk	29
Gambar 3.3. Halaman Daftar Ujian	30
Gambar 3.4. Halaman Daftar Mata Pelajaran	31
Gambar 3.5. Halaman Daftar Paket Soal	32
Gambar 3.6. Halaman Daftar Bank Soal	33
Gambar 3.7. Halaman Ujian Murid	34
Gambar 3.8. Halaman Hasil Ujian Murid	35
Gambar 3.9. Halaman Ranking Peserta Ujian	36
Gambar 4.1. Tampilan Halaman Masuk	40
Gambar 4.2. Tampilan Halaman Daftar Ujian	40
Gambar 4.3. Tampilan Halaman Daftar Mata Pelajaran	41
Gambar 4.4. Tampilan Halaman Daftar Paket Soal	41
Gambar 4.5. Tampilan Halaman Daftar Bank Soal	42
Gambar 4.6. Tampilan Halaman Ujian	42
Gambar 4.7. Tampilan Halaman Hasil Ujian Murid	43
Gambar 4.8. Tampilan Halaman Ranking Hasil Ujian	43

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Saat ini, penggunaan sistem pembelajaran elektronik (*e-learning*) sedang diterapkan di setiap jenjang pendidikan nasional. Penerapan ini mencakup tahap pendidikan mulai dari pendidikan dasar hingga pendidikan menengah dan tinggi. Penerapan ini meliputi sistem belajar mengajar, pemberian tugas, dan ujian. Ujian dapat disediakan dengan beberapa tipe, yaitu pilihan berganda (*multiple choice*), uraian (*essay*), pencarian kesamaan (*matching*), jawaban isian (*fill in the blank answer*), benar/salah (*true/false*), dan kalkulasi (*calculated*). Kelemahan dari sistem ini adalah, hampir semua *e-learning* yang diterapkan di Indonesia belum memiliki kemampuan untuk memeriksa secara otomatis jawaban untuk soal uraian.

Pada ujian uraian, soal yang disediakan biasanya bersifat antara subjektif ataupun objektif. Soal yang bersifat objektif merupakan soal-soal yang menuntut sekumpulan jawaban dengan konsep tertentu dan memiliki bobot nilai yang dapat ditentukan. Soal yang bersifat subjektif merupakan soal-soal yang menuntut pendapat seseorang terhadap sesuatu kasus. Penilaian otomatis dapat dilakukan pada soal yang bersifat objektif.

Penilaian otomatis sendiri merupakan penerapan teknologi komputer dalam pengoreksian jawaban murid dengan kunci jawaban. Pengoreksian jawaban murid dapat dilakukan dengan mencocokkan string jawaban murid dengan kunci jawaban soal. Beberapa contoh algoritma pencocokan string adalah algoritma Rabin-Karp, algoritma Winnowing, dan algoritma Ratcliffe/Obershelp.

Peneliti Danny (2018) menerapkan sebuah algoritma *Rabin-Karp* pada sistem pendeteksi plagiarisme. Sistem tersebut memberikan hasil sebesar 14.22% pada 50 kali pengujian dengan 43 sampel keberhasilan. Sistem yang akan digunakan pada skripsi hampir mirip dengan sistem pendeteksi plagiarisme tersebut. Kemudian, Bustami (2019) menguji efektivitas algoritma dalam mengukur kemiripan teks. Hasil pengujian menunjukkan bahwa algoritma *Ratcliffe/Obershelp* berhasil mencapai tingkat kesamaan lebih baik 3% dibandingkan dengan algoritma *Rabin-Karp*. Implementasi algoritma *Winnowing* dilakukan Febrilia (2020) dalam sistem penilaian otomatis jawaban uraian pada ujian berbasis web. Terdapat perbedaan rata-rata sekitar 5.683% antara penilaian yang dilakukan oleh sistem otomatis dengan penilaian manual. Dalam penilaian jawaban otomatis, Titan (2020) melakukan sebuah perbandingan antara penggunaan algoritma *Rabin-Karp* dengan algoritma *Winnowing*. Algoritma *Rabin-Karp* menghasilkan selisih yang lebih rendah dibandingkan dengan algoritma *Winnowing*, yaitu 26% dengan algoritma *Rabin-Karp* dan 38% dengan algoritma *Winnowing*.

Atas latar belakang berikut serta perbandingan dari beberapa algoritma yang telah diuji, penulis mengajukan judul dalam penelitian : **Implementasi Algoritma *Ratcliffe/Obershelp* pada Sistem Penilaian Soal Uraian Secara Otomatis**

1.2. Rumusan Masalah

Penerapan ujian daring pada sistem *e-learning* terutama pada pertanyaan uraian, masih memerlukan penilaian secara manual oleh tenaga pengajar untuk menilai hasil dari pertanyaan uraian yang membuat pemeriksaan setiap jawaban dari murid yang berpartisipasi dalam ujian tersebut memerlukan waktu yang panjang. Oleh sebab itu, dibutuhkan sistem penilaian otomatis yang mampu menentukan nilai jawaban pelajar pada pertanyaan uraian yang bersifat objektif.

1.3. Tujuan Penelitian

Penelitian ini bertujuan untuk mengimplementasikan algoritma *Ratcliffe/Obershelp* dalam proses penilaian otomatis jawaban pada soal-soal uraian.

1.4. Batasan Masalah

Penelitian ini akan dilaksanakan dengan membatasi cakupan permasalahan yang akan diteliti. Berikut adalah batasan-batasan yang akan diterapkan:

1. Penelitian ini berfokus kepada perancangan sistem penilaian ujian uraian
2. Jenis soal yang akan diteliti merupakan soal uraian yang bersifat objektif
3. Nilai diberikan berdasarkan tingkat kesamaan kata pada jawaban murid dengan kunci jawaban
4. Penilaian jawaban uraian dapat digunakan untuk jawaban berbahasa Indonesia.

1.5. Manfaat Penelitian

Penelitian ini dilaksanakan dengan tujuan mendapatkan manfaat, seperti:

1. Meningkatkan sistem penilaian pada ujian uraian secara daring
2. Menjadi referensi untuk pengembangan pendeteksi kesamaan pada teks.

1.6. Metodologi Penelitian

Rangkaian tahapan yang diterapkan dalam penelitian ini mencakup:

1. Studi Literatur
Studi Literatur dilakukan dengan pembelajaran terhadap informasi yang berkaitan dengan penelitian yang diperoleh dari sumber-sumber pembelajaran seperti buku, jurnal dan lain-lain
2. Analisis Permasalahan
Ini merupakan analisis terhadap informasi yang telah dikumpulkan, sehingga didapat metode yang dapat digunakan untuk mempermudah pengajar dalam menilai jawaban dari soal uraian
3. Perencanaan
Tahapan ini dilakukan sebagai proses perencanaan sebuah sistem yang kemudian akan digunakan untuk menyelesaikan masalah
4. Implementasi
Pada tahap ini dilakukan implementasi rancangan menjadi sebuah kode program yang akan menyelesaikan masalah yang dihadapi
5. Pengujian
Tahapan ini ditujukan untuk melakukan sebuah pengujian terhadap sistem yang telah dihasilkan untuk melihat kesesuaian sistem dengan yang diharapkan
6. Hasil dan Kesimpulan
Hasil dari pengujian akan di kumpulkan serta dibuat sebuah dokumentasi dari rancangan sistem yang telah dibuat

BAB 2

LANDASAN TEORI

2.1. Ujian

Ujian merupakan kegiatan untuk menilai tingkatan seseorang terhadap suatu mutu dengan batasan yang telah ditentukan. Ujian dilaksanakan berguna untuk dapat mengetahui tingkatan kemampuan seseorang terhadap hal yang sedang diuji, baik kemampuan, hasil belajar, dan lain-lain. Ujian dapat dilaksanakan secara teori atau praktik.

Ujian teori merupakan penilaian terhadap suatu mutu yang ditentukan oleh beberapa pertanyaan yang telah disediakan oleh pihak yang menguji. Pertanyaan yang disediakan bisa dalam bentuk pilihan ganda, jawaban singkat, dan juga soal uraian.

Pada soal uraian, jawaban dapat bersifat subjektif maupun objektif. Soal subjektif merupakan soal dengan jawaban yang bersifat pendapat ataupun pemikiran pelajar. Soal objektif merupakan soal dengan jawaban yang sudah pasti.

2.2. Kata Sinonim

Secara etimologis, sinonim dapat ditelusuri ke akar kata dalam bahasa Yunani kuno, yaitu "*syn*" yang berarti "dengan" dan "*onoma*" yang berarti "nama". Secara sederhana, 'sinonim' berarti bahwa kata-kata tersebut memiliki arti yang serupa atau identik dalam berbagai konteks (Cruse, 2004). Kata sinonim dapat memberikan variasi penggunaan kata terhadap suatu bentuk bahasa tanpa mengubah makna lesikal atau struktural bentuk bahasa tersebut. Cruse (2004) membagi kata sinonim menjadi 3 bentuk, yaitu sinonim mutlak, sinonim kognitif, dan sinonim lokal.

Sinonim mutlak ketika adanya kemungkinan kata-kata tersebut memiliki makna yang sama persis dan dapat digunakan secara universal tanpa perubahan makna.

Sinonim mutlak jarang dapat ditemui dikarenakan setiap kata memiliki makna mereka tersendiri.

Sinonim kognitif memerlukan kondisi kebenaran yang sama untuk melakukan pergantian istilah. Contohnya adalah kata “perkakas” dalam kalimat “saya menggunakan perkakas” dapat diganti dengan kata “alat”.

Sinonim lokal ketika suatu kata hanya dapat saling berganti pada konteks tertentu. Contoh dari sinonim ini adalah kata “bisa” pada kalimat “saya bisa berjalan” dapat digantikan dengan kata “dapat”, namun pada kalimat “ular itu memiliki bisa yang kuat” kata “dapat” tidak dapat menggantikan kata “bisa” pada kalimat tersebut.

2.3. *Text Processing*

Dalam komputasi, pemrosesan teks (*text processing*) adalah mekanisasi otomatis dari pembuatan atau modifikasi teks. Pemrosesan teks tidak harus bingung dengan pengolahan kata. Perbedaan utama adalah kesepakatan pemrosesan teks dengan utilitas pemrosesan teks daripada utilitas pengeditan teks.

Pemrosesan teks bersifat sekuensial dalam pendekatan ahli-ahli akses acak dan bekerja secara langsung pada lapisan presentasi dan secara tidak langsung pada lapisan aplikasi. Tidak seperti pengolahan kata, pemrosesan teks beroperasi pada data mentah dan lebih independen dari teknik kepemilikan. Pemrosesan teks dilakukan dengan bantuan perintah shell atau teks editor.

Pemrosesan teks lebih digunakan untuk membuat artikel berita, buku, dan majalah. Pemrosesan teks juga tidak menyimpan dokumen sumber dalam format prosesor tertentu, dan membantu membuka pintu *add-on* dan fungsi baru, seperti penerjemahan dan pengurai.

2.4. *Algoritma Ratcliffe/Obershelp*

Algoritma Ratcliffe/Obershelp menerapkan prosedur serupa untuk menentukan tingkat kesamaan antara dua pola satu dimensi (Yudhy, 2017). Dengan mempertimbangkan bahwa string teks adalah representasi satu dimensi, algoritma ini dapat dianggap sebagai indikator keandalan untuk mengevaluasi tingkat kesamaan antara dua rangkaian karakter.

Pertama, algoritma ini akan menerima dua buah *string*, dalam hal ini akan dinamai “str1” dan “str2”. Kemudian algoritma akan membuat sebuah matriks “dp” (*dynamic*

programming) dengan ukuran “(m + 1) x (n + 1)”, dimana “m” adalah panjang dari “str1” dan “n” adalah panjang “str2”.

Dalam pengisian matriks, tahap pertama adalah melakukan *loop* dari satu hingga nilai ‘m’ untuk indeks baris matriks ‘dp’. Di dalam *loop* luar, akan dilakukan *loop* dari 1 hingga nilai ‘n’ untuk indeks kolom matriks ‘dp’. Kemudian karakter pada posisi “str1[i – 1]” akan dibandingkan dengan karakter pada posisi “str2[j – 1]”. Jika karakter sama, nilai 1 akan ditambahkan pada matriks “dp[i][j]”, yang berarti panjang *subsequence* umum terpanjang saat ini bertambah. Jika karakter berbeda, maka nilai matriks “dp[i][j]” diisi dengan nilai maksimum dari “dp[i – 1][j]” (nilai di atas) dan “dp[i][j – 1]” (nilai di sebelah kiri), karena algoritma akan mencari *subsequence* umum terpanjang. Nilai terakhir pada sudut kanan bawah matriks “dp” (“dp[m][n]”) akan memberikan *subsequence* umum terpanjang antara “str1” dengan “str2”.

Setelah *subsequence* umum terpanjang ditemukan, algoritma akan mencari nilai rasio kesamaan dengan persamaan :

$$D_{ro} = \frac{2 \cdot K_m}{|S1| + |S2|} \quad (2.1)$$

Dimana :

D_{ro} = Nilai kesamaan

K_m = Panjang *subsequence*

$|S1|$ = Panjang string 1

$|S2|$ = Panjang string 2

Nilai kesamaan yang dihasilkan merupakan nilai diantara 0 – 1, semakin tinggi nilai yang didapat maka semakin dekat kesamaan terhadap dua *string* tersebut.

2.5. *Synonym Recognition*

Synonym recognition merupakan proses untuk mengidentifikasi kata atau frasa pada sebuah teks. Bidang ini aktif dalam pemrosesan bahasa alami, dan telah dipelajari untuk beberapa bahasa termasuk bahasa Indonesia. Salah satu pendekatan untuk *synonym recognition* adalah dengan penyematan kata, yang merupakan representasi matematis dari kata-kata yang memaknainya.

Penyematan kata dapat dilatih pada kumpulan teks yang besar, atau kata-kata yang memiliki penyematan yang serupa. Pendekatan lain untuk *synonym recognition* adalah dengan menggunakan tesaurus atau kamus sinonim untuk mendeteksi kata-kata sinonim.

2.6. *Rolling Hash*

Rolling hash adalah sebuah algoritma untuk menghitung *hash* dari *substring* berurutan yang ada dalam sebuah string. Algoritma ini sering digunakan dalam aplikasi pencocokan string atau pemrosesan teks.

Proses dimulai dengan menghitung *fingerprint* dari *substring* pertama di dalam *string*. Proses ini dapat dilakukan dengan menggunakan fungsi *hash*, dimana fungsi *hash* akan mengambil *string* sebagai input dan menghasilkan angka dengan ukuran yang tetap (*fingerprint*). Kemudian, untuk setiap *substring* berikutnya, *rolling hash* akan memutar *fingerprint* ke depan dengan cara menghapus karakter pertama di dalam *substring* sebelumnya dan menambahkan karakter terakhir pada *substring* saat ini.

Proses ini terjadi secara berulang untuk semua *substring* dalam *string* dan berakhir dengan satu set *fingerprint* yang secara unik akan mengidentifikasi setiap *substring* (Purwitasari, et al. 2009).

2.7. *Algoritma Winnowing*

Algoritma *Winnowing* merupakan algoritma yang menggunakan dokumentasi *fingerprinting* untuk mendeteksi kesamaan pada kata (Schleimer, Wilkerson, & Aiken, 2003: 4). *Winnowing* juga merupakan ekstensi dari penerapan algoritma *Rabin-Karp fingerprint* dengan menggunakan metode window.

Cara kerja algoritma *winnowing* dimulai dengan inisialisasi daftar *fingerprints*, daftar ini digunakan untuk menyimpan hash terkecil dari setiap jendela geser. Kemudian dilakukan inisialisasi *variable min_hash* dan *min_index* ke nilai yang tidak mungkin. Hal ini dilakukan agar nilai hash pertama yang didapat menjadi nilai minimum awal. Kemudian algoritma memulai iterasi melalui teks atau dokumen yang diinput menggunakan jendela geser. Jendela ini adalah sejumlah karakter dengan panjang '*window_size*' yang bergeser pada setiap iterasi. Selanjutnya pada setiap jendela akan dihitung nilai *hash* dengan menggunakan fungsi hash yang sesuai. Nilai *hash* minimum kemudian didapat dengan membandingkan nilai hash dari jendela saat ini dengan nilai pada *variable min_hash*, jika didapat nilai hash saat ini lebih kecil daripada nilai yang

ada pada *variable min_hash* maka nilai tersebut menjadi nilai pada *variable min_hash* dan nilai *min_index* diatur menjadi indeks jendela saat ini.

Ketika jendela geser mencapai batas *window_size*, nilai *hash* minimum ditambahkan ke daftar *fingerprints* untuk mengidentifikasi *hash* terkecil pada setiap jendela yang bergeser. Setelah semua iterasi selesai, nilai *hash* terakhir juga ikut dimasukkan pada daftar *fingerprints*. Daftar *fingerprints* inilah yang kemudian digunakan untuk membandingkan dokumen dengan dokumen lainnya untuk mendeteksi pengulangan dalam teks.

2.8. Aplikasi Web

Aplikasi web adalah perangkat lunak yang beroperasi di server web dan dapat diakses melalui peramban web. Aplikasi ini dijalankan oleh pengguna melalui jaringan, seperti internet. Beberapa aplikasi web bersifat dinamis, dimana diperlukan sebuah pemrosesan di sisi server. Lainnya bersifat *static*, tanpa diperlukannya pemrosesan pada sisi server. Berbagai Teknologi utama yang digunakan pada pengembangan aplikasi web adalah bahasa pemrograman, HTML (*Hypertext Markup Language*), CSS (*Cascading Style Sheets*), basis data, server web, dan API (*Application Programming Interface*).

Berikut bagaimana aplikasi web terlihat :

1. Pengguna mengirimkan permintaan ke server web melalui internet, yang dapat berasal dari peramban (browser) atau tampilan aplikasi
2. Server web mengirim kepada server aplikasi web yang sesuai
3. Server aplikasi web melakukan permintaan yang telah dikirimkan, kemudian menampilkan hasil dari permintaan tersebut
4. Server aplikasi web mengirimkan hasil tersebut kepada server web
5. Server web merespon kembali kepada klien dengan hasil yang didapat, kemudian menampilkan hasil pada tampilan user.

2.9. *Application Programming Interface* (API)

API secara umum merupakan kumpulan dari beberapa protokol yang memungkinkan dua buah atau lebih perangkat lunak untuk saling berkomunikasi. API sering dianggap sebagai kontrak, ketika satu pihak mengirimkan sebuah permintaan kepada pihak kedua, maka pihak kedua akan merespon dengan cara yang sudah mereka tentukan.

API bekerja dengan memanfaatkan sistem *request* dan *response*. Ketika satu perangkat lunak melakukan sebuah *request* kepada API, maka API tersebut akan

mengembalikan sebuah *response* yang biasa berisi data kepada perangkat lunak yang melakukan *request* tersebut.

API bisa diibaratkan seperti kita belanja pada kedai dekat rumah. Kita melakukan *request* dengan mengatakan kepada penjual bahwa kita ingin membeli sebuah minuman, kemudian penjual tersebut menerima *request* tersebut dan memberikan *response* berupa minuman yang kita pesan tadi. Kita pada perumpamaan tadi merupakan sebuah perangkat lunak dan penjual tersebut adalah API yang merespon permintaan kita.

2.10. Metode Evaluasi

Metode evaluasi merupakan suatu metode perhitungan yang digunakan untuk mengetahui seberapa baik proses penilaian otomatis pada ujian ini. Metode ini menerapkan aturan TP (*True Positive*), TN (*True Negative*), FP (*False Positive*), dan FN (*False Negative*). Variabel-variabel tersebut akan digunakan untuk melakukan perhitungan *accuracy*, *precision*, *recall*, dan *f-score* (Baeza-Yates & Ribiero-Neto, 1999).

Accuracy mengukur sejauh mana sistem dapat memberikan penilaian yang sesuai dengan penilaian guru. *Precision* memberikan informasi seberapa banyak hasil positif yang dihasilkan sistem sesuai dengan hasil secara manual. *Recall* memberikan gambaran seberapa banyak sistem dapat mendeteksi nilai positif yang sebenarnya ada. *F1 score* memberikan pandangan yang seimbang antara *precision* dan *recall*.

Accuracy dapat dihitung dengan persamaan :

$$Accuracy = \frac{\text{Jumlah penilaian sistem yang benar}}{\text{Total penilaian yang dilakukan}} \quad (2.2)$$

Precision dapat dihitung dengan persamaan :

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (2.3)$$

Recall dapat dihitung dengan persamaan :

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (2.4)$$

F-score dapat dihitung dengan persamaan :

$$Fscore = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \quad (2.5)$$

2.11. Penelitian Terdahulu

Ruslan et.al. (2018) menggunakan metode *Generalization Latent Semantic Analysis* (GLSA). GLSA berfokus pada pengurangan dimensi data untuk mengurangi kompleksitas, sehingga dapat meningkatkan efisiensi analisis. Tujuannya adalah untuk mengekstrak kata-kata kunci yang dapat dibandingkan dengan jawaban-jawaban kunci yang ada. Dari proses ini, kemudian dibandingkan tahapan penilaian melalui *cosine similarity*. Hasil akurasi yang diperoleh mencapai 90.39%

Sistem penilaian otomatis memiliki sistem kerja yang mirip dengan sistem pendeteksi plagiarisme. Pada sistem pendeteksi plagiarisme, Danny (2018) menerapkan algoritma *Rabin-Karp* pada sistem tersebut. Dengan pengujian sebanyak 50 kali dengan 43 sampel keberhasilan, Danny (2018) mendapatkan hasil sebesar 14.22%

Pengujian terhadap algoritma *Rabin-Karp* dan algoritma *Ratcliffe/Obershelp* dilakukan dalam menghitung kesamaan teks oleh Bustami (2019). Hasil uji menunjukkan bahwa algoritma *Ratcliffe/Obershelp* memberikan tingkat kesamaan yang lebih tinggi dibandingkan algoritma *Rabin-Karp*, dengan perbedaan sekitar $\pm 3\%$. Selain itu, ditemukan bahwa posisi kata dalam kalimat lebih memengaruhi kinerja algoritma *Rabin-Karp*.

Febrilia (2020) menerapkan algoritma *Winnowing* pada sistem penilaian otomatis jawaban uraian pada ujian daring berbasis web. Beliau menggunakan algoritma *Winnowing* sebagai pencocok antara kunci jawaban dan jawaban pelajar. Penilaian yang didapat sistem dengan yang didapat oleh penilaian manual memiliki selisih rata-rata 5.683%.

Titan (2020) melakukan perbandingan antara algoritma *Rabin-Karp* dan algoritma *Winnowing* pada sistem penilaian otomatis. Kedua hasil kesamaan algoritma yang dibandingkan kemudian diubah menjadi penilaian dalam bentuk penilaian manusia (*human rates*). Masing-masing hasil tersebut dibandingkan dengan hasil penilaian yang dilakukan oleh dosen. Selisih yang didapat algoritma *Rabin-Karp* adalah 26% sedangkan selisih yang didapat algoritma *Winnowing* adalah 38%

Tabel 2.1. Penelitian Terdahulu

No.	Peneliti	Judul	Tahun	Keterangan
1.	Ruslan et.al.	Sistem Penilaian Otomatis Jawaban Esai Menggunakan Metode GLSA	2018	GLSA melakukan pengurangan jumlah dimensi untuk mengurangi kompleksitas data dan meningkatkan efisiensi analisis sehingga dapat mengekstrak kata-kata kunci yang dapat dibandingkan dengan jawaban-jawaban kunci yang tersedia. Dari proses ini, kemudian dibandingkan tahapan penilaian melalui <i>cosine similarity</i> . Hasil akurasi yang diperoleh mencapai 90.39%
2.	Danny	Aplikasi Pendeteksi Plagiarisme Tugas dan Makalah pada Sekolah Menggunakan Algoritma Rabin- Karp	2018	Algoritma Rabin-Karp digunakan untuk mencari pola string dalam teks dengan menerapkan teknik hashing. Pengujian dilakukan dengan melakukan 50 kali uji coba, dan dari hasil tersebut, sebanyak 43 kali berhasil. Hasil uji tersebut menunjukkan tingkat keberhasilan sebesar 14.22%.

Tabel 2.1. Penelitian Terdahulu (lanjutan)

No.	Peneliti	Judul	Tahun	Keterangan
3.	Bustami	Analisis Perbandingan Algoritma Rabin-Karp dan Ratcliffe/Obershelp untuk Menghitung Kesamaan Teks dalam Bahasa Indonesia	2019	<i>Dice's Similarity Coefficient</i> digunakan sebagai alat uji dari perbandingan algoritma <i>Rabin-Karp</i> dan algoritma <i>Ratcliffe/Obershelp</i> . Dengan data uji sebanyak 50 halaman berita dari situs tempo.co didapatkan hasil pengujian dengan algoritma Ratcliffe/Obershelp mendapatkan nilai similarity lebih baik $\pm 3\%$ dibandingkan dengan algoritma <i>Rabin-Karp</i> .
4.	Febrilia	Implementasi Algoritma Winnowing pada Sistem Penilaian Otomatis Jawaban Esai pada Ujian Online Berbasis Web	2020	Algoritma Winnowing mencari document fingerprinting untuk mengidentifikasi kesamaan termasuk bagian-bagian kecil yang ada pada suatu <i>string</i> . Penilaian yang didapat menggunakan sistem dan dengan penilaian manual memiliki selisih rata-rata 5.683%

Tabel 2.1. Penelitian Terdahulu (lanjutan)

No.	Peneliti	Judul	Tahun	Keterangan
5.	Titan	Analisis Perbandingan Algoritma <i>Rabin-Karp</i> dan <i>Winnowing</i> Dalam Penilaian Jawaban Otomatis	2020	Data jawaban ujian melalui proses tahap <i>preprocessing</i> yang kemudian hasil <i>preprocessing</i> tersebut akan diproses dengan kedua algoritma yang dibandingkan. Kedua hasil tersebut kemudian diubah menjadi penilaian dalam bentuk penilaian manusia (<i>human rates</i>). Masing-masing hasil tersebut dibandingkan dengan hasil penilaian yang dilakukan oleh dosen. Selisih yang didapat algoritma <i>Rabin-Karp</i> adalah 26% sedangkan selisih yang didapat algoritma <i>Winnowing</i> adalah 38%

BAB 3

ANALISIS DAN PERANCANGAN SISTEM

3.1. Data Penelitian

Pada penelitian ini digunakan hasil ujian yang dilakukan pada SMA Tamansiswa Binjai dan hasil ulangan pada mata pelajaran biologi. Pada ujian yang dilakukan terdapat 25 murid yang mengerjakan soal ujian tersebut. Ujian tersebut terdiri dari 4 soal uraian yang telah dibuat oleh pengajar biologi pada SMA Tamansiswa Binjai. Pada ulangan yang dilakukan terdapat 12 murid yang mengerjakan ulangan tersebut. Ulangan tersebut terdiri dari 5 soal uraian yang telah dibuat oleh pengajar biologi pada SMA Tamansiswa Binjai.

Pada ujian, pengajar memberikan poin kepada setiap soal sebesar 25 poin dan pada ulangan, pengajar memberikan poin sebesar 20 poin pada setiap soal. Adapun soal-soal yang digunakan sebagai data penelitian dapat dilihat pada tabel 3.1 dan tabel 3.2.

Tabel 3.1. Data Soal Ujian

No.	Soal	Jawaban
1.	Tuliskan 3 tahapan proses filtrasi / penyaringan, reabsorpsi / pembentukan urine pada penyerapan kembali, augmentasi / manusia ! pengumpulan	
2.	Jelaskan fungsi uterus / rahim pada sistem reproduksi wanita !	uterus atau rahim berfungsi untuk memberi makan janin dan sebagai tempat perkembangan embrio menjadi janin
3.	Buatlah skema jalannya impuls rangsang pada gerak refleks !	Rangsangan (impuls) - Reseptor (alat indra) - saraf sensorik - sumsum tulang belakang - saraf motorik - efektor (otot / kelenjar)

Tabel 3.1. Data Soal Ujian (Lanjutan)

No.	Soal	Jawaban
4.	Tuliskan secara berurutan organ - organ yang berperan dalam sistem pernapasan manusia !	hidung - faring - laring - trakea - bronkus - bronkeolus - alveolus - paru paru

Tabel 3.2. Data Soal Ulangan

No.	Soal	Jawaban
1.	Jelaskan teori evolusi menurut J.B Lamarch !	Evolusi terjadi melalui penggunaan organ tubuh yang diturunkan pada keturunannya karena adaptasi terhadap lingkungan
2.	Apa yang dimaksud dengan seleksi alam pada teori evolusi charles darwin ?	Terjadi seleksi terhadap individu dalam mempertahankan hidupnya, individu yang kuat akan berkembang sedangkan yang lemah akan tersisih dan mati
3.	Apa yang dimaksud dengan Albuminuria ?	Penyakit yang terjadi akibat ginjal tidak dapat melakukan proses penyaringan terhadap protein
4.	Apa yang dimaksud dengan rabun jauh (miopi) ?	Suatu kondisi pada mata tidak dapat melihat benda yang berjarak jauh karena fokus bayangan jatuh didepan retina
5.	Jelaskan pengertian organ homolog dalam teori evolusi !	Organ-organ yang berasal dari struktur yang sama namun memiliki fungsi yang berbeda

Tesaurus bahasa Indonesia yang digunakan adalah tesaurus yang diambil dari Pusat Bahasa karya Departemen Pendidikan Nasional tahun 2008.

3.2. Analisis Sistem

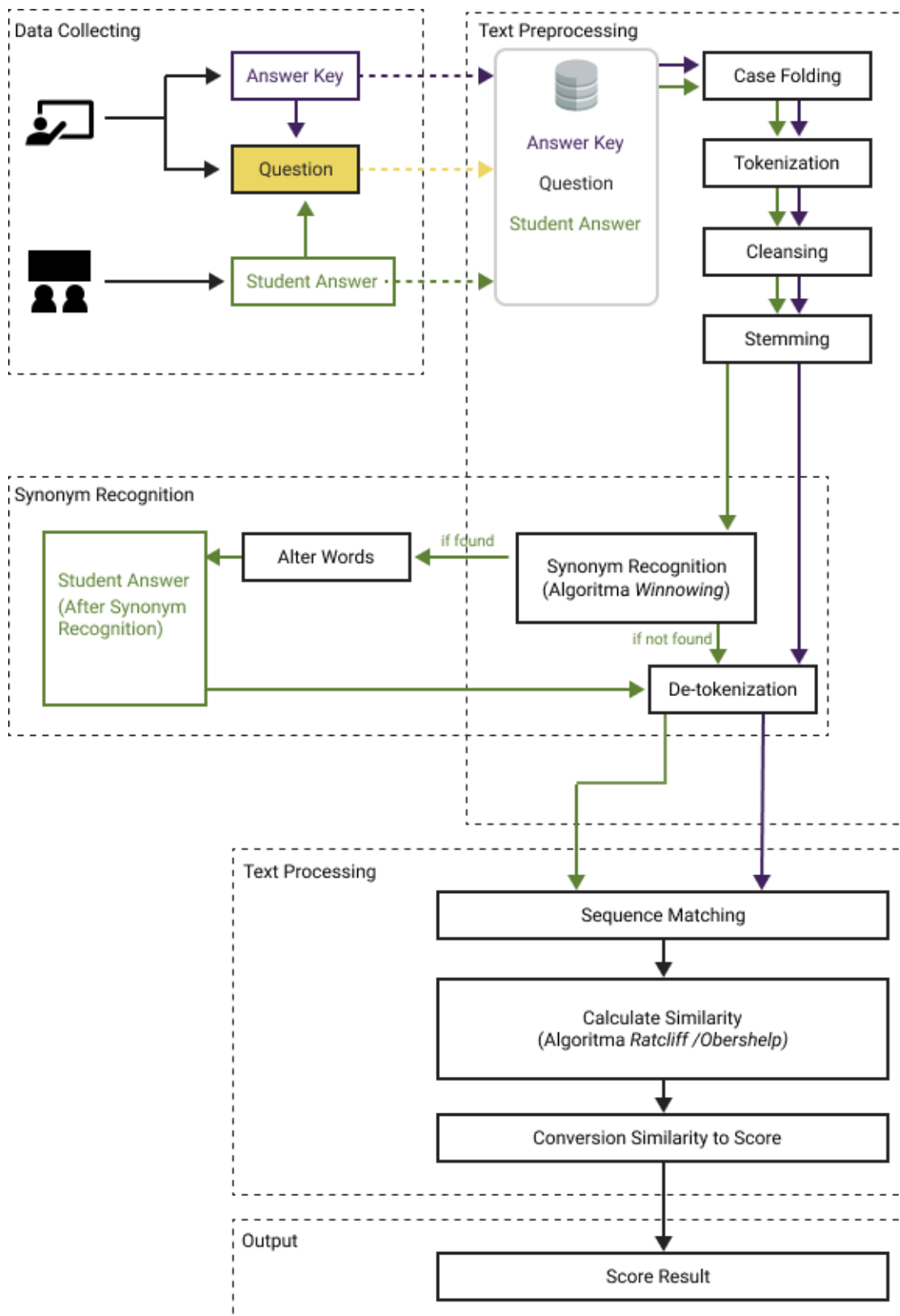
Sistem dibangun untuk dapat melakukan penilaian otomatis terhadap ujian uraian. Sistem dapat menentukan tingkat kesamaan jawaban murid dengan kunci jawaban sehingga nilai terhadap soal tersebut dapat ditentukan.

Sistem menggunakan algoritma *Ratcliffe/Obershelp* dalam penentuan tingkat kesamaan kalimat serta algoritma *Winnowing* dan kamus sinonim dalam menentukan sinonim pada jawaban murid terhadap kunci jawaban.

3.3. Perancangan Sistem

3.3.1. Arsitektur Umum

Sistem yang dirancang terdiri dari susunan tahapan. Tahapan awal merupakan pengumpulan data kunci jawaban dan jawaban murid melalui *website* ujian daring. Kemudian data tersebut akan melalui tahap *pre-processing* seperti *case folding* untuk menyeragamkan format penulisan, *cleansing* untuk membersihkan tanda baca dan simbol, *tokenization* untuk memenggal kunci jawaban dan jawaban murid berdasarkan kata penyusun, dan *stemming* untuk mengembalikan kata-kata tersebut kembali ke bentuk dasar kata tersebut. Setelah itu dilakukan *synonym recognition* untuk mencari kata-kata yang memiliki kesamaan arti diantara kunci jawaban dengan jawaban murid. Setelah itu hasil *tokenization* akan digabungkan kembali menjadi satu string panjang untuk mendapatkan nilai akurasi jawaban murid terhadap kunci jawaban. Gambar 3.1 memvisualisasikan tahapan arsitektur umum yang menjelaskan setiap langkah penelitian.



Gambar 3.1. Arsitektur Umum

1. Data Collecting

Pada tahap ini, data yang diperlukan dikumpulkan dan dimasukkan kedalam *database*. Data dikumpulkan melalui *website* ujian daring. Data tersebut kemudian dibedakan menjadi data jawaban murid dengan data kunci jawaban.

2. Text Pre-processing

Tahap ini terdiri dari :

- a) *Case Folding*, adalah tahapan *pre-processing* yang bertujuan untuk Melakukan penyesuaian format penulisan untuk mencocokkan antara kunci jawaban dan jawaban dari siswa. Pada tahap ini kunci jawaban (*key answer*) dan jawaban murid (*student answer*) akan diubah format setiap hurufnya menjadi huruf kecil. Contoh mengenai *case folding* dapat ditemukan di Tabel 3.3.

Tabel 3.3. Contoh *Case Folding*

Data sebelum di- <i>case folding</i>			Data setelah di- <i>case folding</i>		
Filtrasi	(Penyaringan),	Reabsorsi	filtrasi	(penyaringan),	reabsorsi
(penyerapan kembali), Augmentasi urine			(penyerapan kembali), augmentasi urine		
Memberi makanan dan menampung sel telur yang telah dibuahi sampai menjadi janin atau sampai siap untuk dilahirkan			memberi makanan dan menampung sel telur yang telah dibuahi sampai menjadi janin atau sampai siap untuk dilahirkan		
Rangsangan (impuls) – Reseptor (indra) – Saraf sensorik – Sumsum tulang belakang – Saraf motorik – Efektor (otot)			rangsangan (impuls) – reseptor (indra) – saraf sensorik – sumsum tulang belakang – saraf motorik – efektor (otot)		
Hidung,	Faring,	Laring,	hidung,	faring,	laring,
Bronkiolus,	Paru-Paru	Trakea,	paru-paru	trakea,	bronkiolus,

- b) *Cleansing*, adalah tahapan yang bertujuan untuk membersihkan tanda baca dan simbol yang ada pada kunci jawaban dan jawaban murid. Contoh proses *cleansing* yang tertera dalam Tabel 3.4.

Tabel 3.4. Contoh *Cleansing*

Data sebelum di-cleansing	Data setelah di-cleansing
filtrasi (penyaringan), reabsorpsi (penyerapan kembali), augmentasi urine	filtrasi penyaringan reabsorpsi penyerapan kembali augmentasi urine
memberi makanan dan menampung sel telur yang telah dibuahi sampai menjadi janin atau sampai siap untuk dilahirkan	memberi makanan dan menampung sel telur yang telah dibuahi sampai menjadi janin atau sampai siap untuk dilahirkan
rangsangan (impuls) – reseptor (indra) – saraf sensorik – sumsum tulang belakang – saraf motorik – efektor (otot)	rangsangan impuls reseptor indra saraf sensorik sumsum tulang belakang saraf motorik efektor otot
hidung, faring, laring, trakea, bronkiolus, paru-paru	hidung faring laring trakea bronkiolus paruparu

- c) *Tokenization*, yaitu proses untuk memenggal kunci jawaban dan jawaban murid berdasarkan setiap kata yang menyusun. Dalam tabel 3.5, terdapat contoh mengenai tahapan *tokenization*.

Tabel 3.5. Contoh *Tokenization*

Data sebelum tokenization	Data setelah tokenization
filtrasi penyaringan reabsorpsi penyerapan kembali augmentasi urine	["filtrasi", "penyaringan", "reabsorpsi", "penyerapan", "kembali", "augmentasi", "urine"]
memberi makanan dan menampung sel telur yang telah dibuahi sampai menjadi janin atau sampai siap untuk dilahirkan	["memberi", "makanan", "dan", "menampung", "sel", "telur", "yang", "telah", "dibuahi", "sampai", "menjadi", "janin", "atau", "sampai", "siap", "untuk", "dilahirkan"]
rangsangan impuls reseptor indra saraf sensorik sumsum tulang belakang saraf motorik efektor otot	["rangsangan", "impuls", "reseptor", "indra", "saraf", "sensorik", "sumsum", "tulang", "belakang", "saraf", "motorik", "efektor", "otot"]

Tabel 3.5. Contoh Tokenization (Lanjutan)

Data sebelum <i>tokenization</i>	Data setelah <i>tokenization</i>
hidung faring laring trakea bronkiolus paruparu	["hidung", "faring", "laring", "trakea", "bronkiolus", "paruparu"]

- d) *Stemming*, yaitu tahapan mengembalikan kata menjadi bentuk dasar dari kata tersebut. Tabel 3.6 memperlihatkan contoh proses *stemming*.

Tabel 3.6. Contoh *Stemming*

Data sebelum di-<i>stemming</i>	Data setelah di-<i>stemming</i>
["filtrasi", "penyaringan", "reabsorpsi", "penyerapan", "kembali", "augmentasi", "urine"]	["filtrasi", "nyaring", "reabsorpsi", "serap", "kembali", "augmentasi", "urine"]
["memberi", "makanan", "dan", "menampung", "sel", "telur", "yang", "telah", "dibuahi", "sampai", "menjadi", "janin", "atau", "sampai", "siap", "untuk", "dilahirkan"]	["beri", "makan", "dan", "tampung", "sel", "telur", "yang", "telah", "buah", "sampai", "jadi", "janin", "atau", "sampai", "siap", "untuk", "lahir"]
["rangsangan", "impuls", "reseptor", "indra", "saraf", "sensorik", "sumsum", "tulang", "belakang", "saraf", "motorik", "efektor", "otot"]	["rangsang", "impuls", "reseptor", "indra", "saraf", "sensorik", "sumsum", "tulang", "belakang", "saraf", "motorik", "efektor", "otot"]
["hidung", "faring", "laring", "trakea", "bronkiolus", "paruparu"]	["hidung", "faring", "laring", "trakea", "bronkiolus", "paruparu"]

3. *Synonym Recognition*

Tahapan ini terdiri dari :

- a) *Synonym Recognition*, tahapan ini merupakan pengecekan jawaban pelajar yang telah ditokenisasi dengan kamus sinonim menggunakan algoritma *Winnowing*. Algoritma ini akan memeriksa kata yang ada pada jawaban siswa, dan mencari pada setiap kata di kunci jawaban. Jika ditemukan sinonim pada kata tersebut, maka tahapan *Alter words* akan dilakukan.
- b) *Alter Words*, di tahap ini kata pada jawaban pelajar yang memiliki sinonim pada kata di kunci jawaban akan disesuaikan sesuai dengan kunci jawaban.
- c) *De-tokenization*, tahapan ini untuk mengubah hasil *synonym recognition* menjadi satu string panjang kembali. Terdapat contoh penerapan *synonym recognition* pada Tabel 3.7.

Tabel 3.7. Contoh *Synonym Recognition*

Jawaban pelajar	Kunci jawaban	Jawaban pelajar setelah <i>synonym recognition</i>
heterotrof merupakan bakteri yang mendapatkan makanan dari organisme	heterotrof adalah bakteri yang memperoleh makanan dari organisme lain	heterotrof adalah bakteri yang memperoleh makanan dari organisme
konjugasi adalah perpindahan materi genetika dari suatu bakteri ke bakteri lain	konjugasi adalah perpindahan materi genetika (DNA) dari satu bakteri ke bakteri lain	konjugasi adalah pemindahan materi genetika dari satu bakteri ke bakteri lain

4. *Text Processing*

Tahapan ini terdiri dari :

- a) *Sequence Matching*, yaitu Prosedur mencari subsequence yang ada di string 1 dan string 2 melalui proses pencocokan kedua string tersebut (Yudha, 2017). *Subsequence* terpanjang merupakan *anchor*. Setelah *anchor* ditemukan, langkah

selanjutnya adalah mencari *subsequence* lainnya yang terdapat pada kedua *string*, yaitu *string* 1 dan *string* 2.

- b) *Calculate Similarity*, adalah tahapan setelah semua *subsequence* telah ditemukan dari proses *sequence matching* dan dimana persamaan tersebut akan menentukan poin yang didapat untuk soal tersebut.
- c) *Conversion Similarity to Score*, adalah tahapan dimana nilai *similarity* yang didapat akan diubah menjadi sebuah skor. Skor ini didapat dari pengelompokkan yang telah dibuat oleh pengajar terhadap soal tersebut. Salah satu contoh pengelompokkan ini dapat dilihat pada tabel 3.8 dan tabel 3.9.

Tabel 3.8. Penilaian Skor untuk Ujian

No.	Nilai Similarity	Skor yang didapat
1.	0% - 33%	0
2.	34% - 66%	15
3.	67% - 100%	25

Tabel 3.9. Penilaian Skor untuk Ulangan

No.	Nilai Similarity	Skor yang didapat
1.	0% - 33%	0
2.	34% - 66%	10
3.	67% - 100%	20

Dalam contoh perhitungan ini saya menggunakan “filtrasi nyaring reabsorpsi serap kembali augmentasi urine” sebagai jawaban pelajar dan “filtrasi nyaring reabsorpsi serap kembali augmentasi kumpul” sebagai kunci jawaban.

Terdapat 2 pola yang sama pada kedua string tersebut, yaitu “filtrasi nyaring reabsor” dan “serap kembali augmentasi”. Pada pola pertama terdapat 24 karakter yang sama, dan pada pola kedua terdapat 24 karakter yang sama.

Kemudian dengan persamaan 2.1 dapat kita hitung nilai kesamaan dari kedua string tersebut.

$$D_{ro} = \frac{2 \cdot K_m}{|S1| + |S2|}$$

$$D_{ro} = \frac{2 \cdot (24 + 24)}{57 + 59}$$

$$D_{ro} = \frac{2 \cdot 48}{116}$$

$$D_{ro} = \frac{96}{116}$$

$$D_{ro} = 0.82$$

Dengan nilai kesamaan 0,82 atau 82%, kemudian nilai kesamaan tersebut dikonversikan menjadi poin menggunakan tabel konversi, yaitu pelajar mendapatkan poin sebesar 25 poin pada soal tersebut.

5. *Output*

Setelah kedua data diproses maka dihasilkan output berupa skor yang didapat pelajar untuk soal tersebut.

3.3.2. Pseudo code algoritma kesamaan teks

Berikut pseudo code dari algoritma kesamaan teks yang digunakan pada sistem :

```
app = Flask(__name__)
string_similarities = RatcliffeObershelp()

@app.route("/", methods=["GET"])
def initial_response():
    return {"status_code": 200, "body": "API Start"}
@app.route("/measure", methods=["POST"])
def scan_ocr():
    if request.method == "POST":
        perfect_score = 1.0
        data = request.get_json()
        sentence_1 = data['student_answer']
        sentence_2 = data['essay_answer']
        sentence_1 = string_filtering(sentence_1)
```

```

sentence_2 = string_filtering(sentence_2)
sentence_1, sentence_2 = synonym_normalization(sentence_1,
sentence_2)
sentence_1 = string_stemming(sentence_1)
sentence_2 = string_stemming(sentence_2)
result_score, result_score_1 =
winnow_filtering(sentence_1, sentence_2)
if result_score == result_score_1:
    result_score = perfect_score
else if result_score is None or result_score != 1.0:
    result_score =
    string_similarities.similarity(sentence_1, sentence_2)
result_score = float("{:.2f}".format(result_score * 100))
return {"status_code": 200, "score": result_score}
else:
    return {"status_code": 404, "message": "incorrect method
request"}

```

Pada baris pertama dilakukan inisialisasi aplikasi *flask* dengan nama modul saat ini. Kemudian dilakukan inisialisasi objek ke variable *string_similarities*. Objek tersebut dapat dilihat pada pseudo code berikut :

```

class RatcliffeObershelp(_BaseSimilarity):
/* metode maximum digunakan untuk mengembalikan nilai integer 1
*/
def maximum(self, *sequence: string) -> integer:
    return 1

/* metode _find digunakan untuk menghitung kesamaan antara
beberapa urutan (sequences) string. Fungsi ini menggunakan
algoritma rekursif untuk mencari kesamaan terpanjang. */
def _find(self, *sequence: string) -> integer:
subseq = LongestCommonSubsequence(sequence)
length = length(subseq)
if length == 0:
    return 0
before = [s.substring(0, s.indexOf(subseq)) for s in
sequences]

```

```

        after = [s.substring(s.indexOf(subseq)+length) for s in
sequences]

        return self._find(before...)+length+self._find(after...)

/*  metode  __call__  merupakan  awal  mengukur  kesamaan
Ratcliffe/Obershelp  antara  beberapa  urutan  string.  Jika  quick
answer  yang  telah  dihitung  ditemukan  maka  nilai  dikembalikan.
Jika  tidak,  menghitung  jumlah  urutan  (scount)  dan  jumlah  elemen
(ecount).  Kemudian  memanggil  metode  find  untuk  menghitung
kesamaan  dalam  bentuk  float.  */

def __call__(self, sequences: string) -> float:
    result = self.quick_answer(sequences)

    if result is Not None:
        return result

    scout = length(sequences)
    ecount = sum(map(length, sequences))
    sequences = self._get_sequences(sequences...)
    return scout * self._find(sequences...) / ecount

    length = length(subseq)

    if length == 0:
        return 0

    before = [s.substring(0, s.indexOf(subseq)) for s in
sequences]

    after = [s.substring(s.indexOf(subseq)+length) for s in
sequences]

    return self._find(before...)+length+self._find(after...)

/*  metode  __call__  merupakan  awal  mengukur  kesamaan
Ratcliffe/Obershelp  antara  beberapa  urutan  string.  Jika  quick
answer  yang  telah  dihitung  ditemukan  maka  nilai  dikembalikan.
Jika  tidak,  menghitung  jumlah  urutan  (scount)  dan  jumlah  elemen
(ecount).  Kemudian  memanggil  metode  find  untuk  menghitung
kesamaan  dalam  bentuk  float.  */

def __call__(self, sequences: string) -> float:
    result = self.quick_answer(sequences)

    if result is Not None:
        return result

    scout = length(sequences)
    ecount = sum(map(length, sequences))
    sequences = self._get_sequences(sequences...)

```

```
return scout * self._find(sequences...) / ecount
```

Kemudian dilakukan deklarasi route untuk menentukan akses halaman, yaitu “/” dan “/measure”. Ketika mengakses rute “/measure”, fungsi scan_ocr dipanggil. Fungsi ini melakukan langkah langkah berikut :

1. Memeriksa metode permintaan adalah “POST”
2. Mengambil data JSON dari permintaan
3. Mendeklarasi variable “sentence_1” dengan nilai student answer pada data dan “sentence_2” dengan nilai essay answer pada data
4. Melakukan preprocessing teks dengan fungsi “string_filtering”, “synonym_normalization”, dan “string_stemming”. Berikut pseudo code untuk fungsi – fungsi tersebut :

```
import Re
from ._synonym import load_corpus, get_synonim
from ._ratcliff import _BaseSimilarity, find_ngrams
from difflib import SequenceMatcher as _SequenceMatcher
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
factory = StemmerFactory()
stemmer = factory.create_stemmer()
corpus = load_corpus(corpus_location)

/* fungsi ini melakukan stemming dengan menggunakan algoritma
sastrawi dan mengembalikan kalimat yang telah distemming */
def string_stemming(sentence: string):
    return stemmer.stem(sentence)

/* fungsi ini menggunakan regex untuk menghapus karakter yang
bukan huruf atau angka dari kalimat */
def string_filtering(sentence: string):
    stemmed_sentence = re.sub('[^0-9a-zA-Z]+', ' ', sentence)
    return stemmed_sentence

def synonym_normalization(sentence_1: str, sentence_2: str):
    /* memecah kalimat menjadi list kata-kata */
```

```

array_sentence_1, array_sentence_2 = sentence_1.split(),
sentence_2.split()

/* membandingkan kata-kata pada posisi yang sama pada kedua
kalimat, jika ada perbedaan antara kata yang tidak sinonim,
mencoba mengganti kata dengan sinonim yang cocok dari korpus
sinonim */

if length(array_sentence_1) == length(array_sentence_2)
    for idx, _ in enumerate(array_sentence_1):
        word_1 = array_sentence_1[idx]
        word_2 = array_sentence_2[idx]
        if word_1 == word_2
            continue
        else:
            sinonim_result = get_synonim(word_1, corpus)
            if word_2 in sinonim_result:
                array_sentence_1[idx] = word_2
                array_sentence_2[idx] = word_2
            else:
                sinonim_result = get_synonim(word_2, corpus)
                if word_1 in sinonim_result:
                    array_sentence_1[idx] = word_2
                    array_sentence_2[idx] = word_2

/* mengembalikan kalimat-kalimat yang telah dinormalisasi dengan
sinonim */

array_sentence_1 = " ".join(array_sentence_1)
array_sentence_2 = " ".join(array_sentence_2)
return array_sentence_1, array_sentence_2

```

5. Menggunakan filtering atau pengukuran khusus dengan “winnow_filtering”.

Berikut pseudo code untuk “winnow_filtering” :

```

def winnow_filtering(sentence_essay: string, sentence_answer:
string):
    perfect_score = 1.0

/* memecah kalimat menjadi token */
    token_essay = split_into_token(sentence_essay)
    token_answer = split_into_token(sentence_answer)

```



```

/* mengcopy token ke variable "final" */
final_essay = copy_of(token_essay)
final_answer = copy_of(token_answer)
/* kasus jika kedua kalimat terdiri dari satu token */
if length(token_answer) == 1 and length(token_essay) == 1:
    current_string = join_tokens(token_answer + token_essay)
    result_winnow = winnow(current_string)
/* jika hasil winnow memiliki satu nilai positif*/
if length(result_winnow) == 1 and result_winnow[0][0] > 0:
    return perfect_score, perfect_score
else:
    return None, None
else:
/* menentukan panjang minimum antara kedua kalimat */
    min_distance = min(length(token_essay),
        length(token_answer))
/* iterasi melalui token-token */
    for idx in range(min_distance):
        current_string = join_tokens([token_essay[idx],
            token_answer[idx]])
        result_winnow = winnow(current_string)
/* jika hasil winnow memiliki satu nilai positif */
        if length(result_winnow) == 1 and result_winnow[0][0] >
            0:
            final_essay[idx] = token_essay[idx]
            final_answer[idx] = token_essay[idx]
/* menggabungkan token-token menjadi kalimat */
    final_essay = join_token(final_essay).strip()
    final_answer = join_token(final_answer).strip()
    return final_essay, final_answer

```

6. Jika hasil sama, mengambil nilai pada "perfect_score". Jika hasil filtering tidak sempurna, dilakukan pengukuran kesamaan dengan algoritma Ratcliffe/Obershelp
7. Mengubah nilai skor menjadi persentase dengan dua desimal

8. Mengembalikan hasil dalam format JSON yang berisi skor kesamaan.

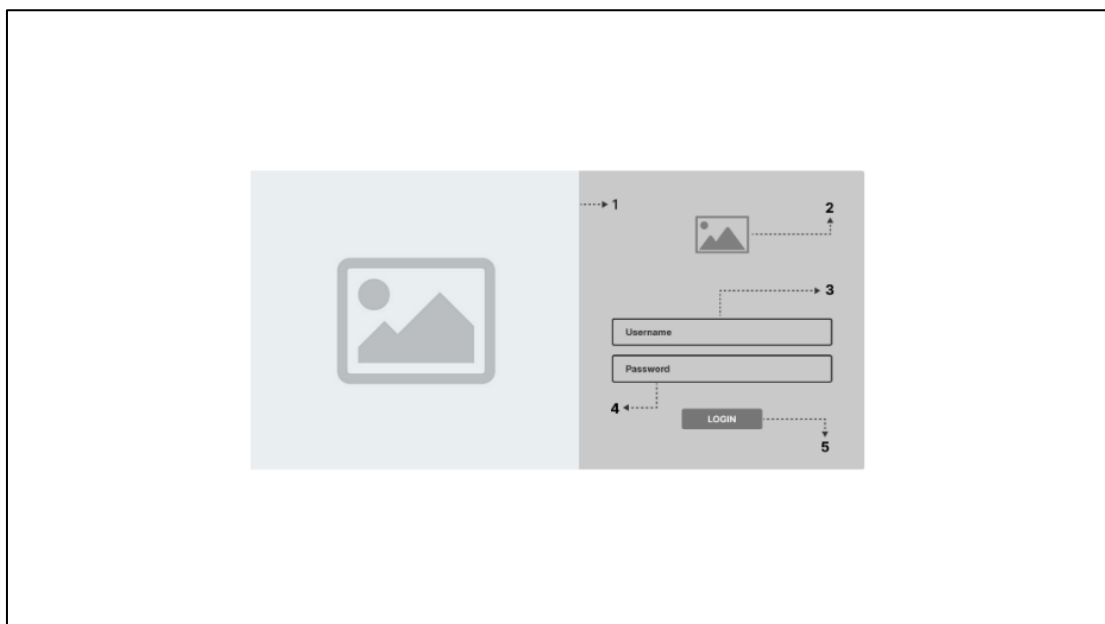
3.3.3. Perancangan antar muka sistem

Rancangan antarmuka sistem dibuat sebagai alat untuk merancang sistem yang akan dibangun untuk memudahkan pengguna untuk menjalankan sistem.

Sistem terdiri dari 8 halaman, yaitu halaman masuk, halaman daftar ujian, halaman daftar mata pelajaran, halaman daftar paket soal, halaman daftar bank soal, halaman ujian murid, halaman hasil ujian murid, dan halaman *ranking* peserta ujian. Adapun penjelasan tampilan sistem dijelaskan sebagai berikut.

1. Halaman Masuk

Pada halaman masuk, terdapat form yang dapat digunakan untuk pengguna yang telah memiliki akun dapat masuk. Rancangan halaman masuk dapat dilihat pada Gambar 3.2.



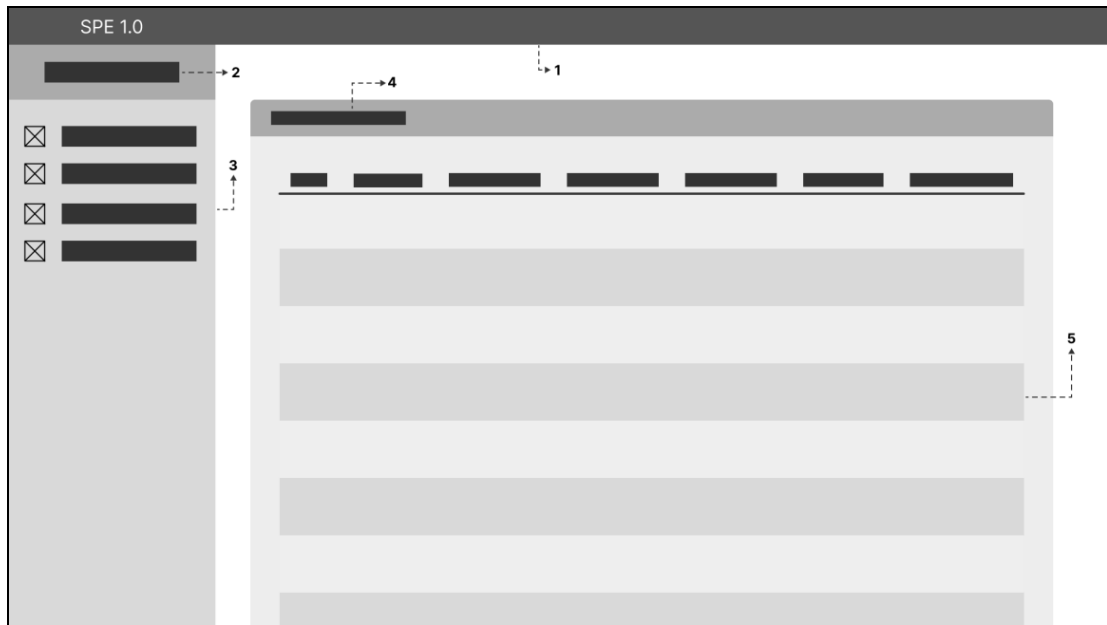
Gambar 3.2. Halaman Masuk

Keterangan :

- 1 : *Background Image*
- 2 : Logo
- 3 : *Form input username*
- 4 : *Form input password*
- 5 : Tombol Masuk

2. Halaman Daftar Ujian

Halaman daftar ujian menampilkan ujian-ujian yang telah dibuat, serta terdapat bagian untuk menambahkan ujian baru, atau mengubah sekaligus menghapus ujian yang telah dibuat. Rancangan halaman daftar ujian dapat dilihat pada Gambar 3.3.



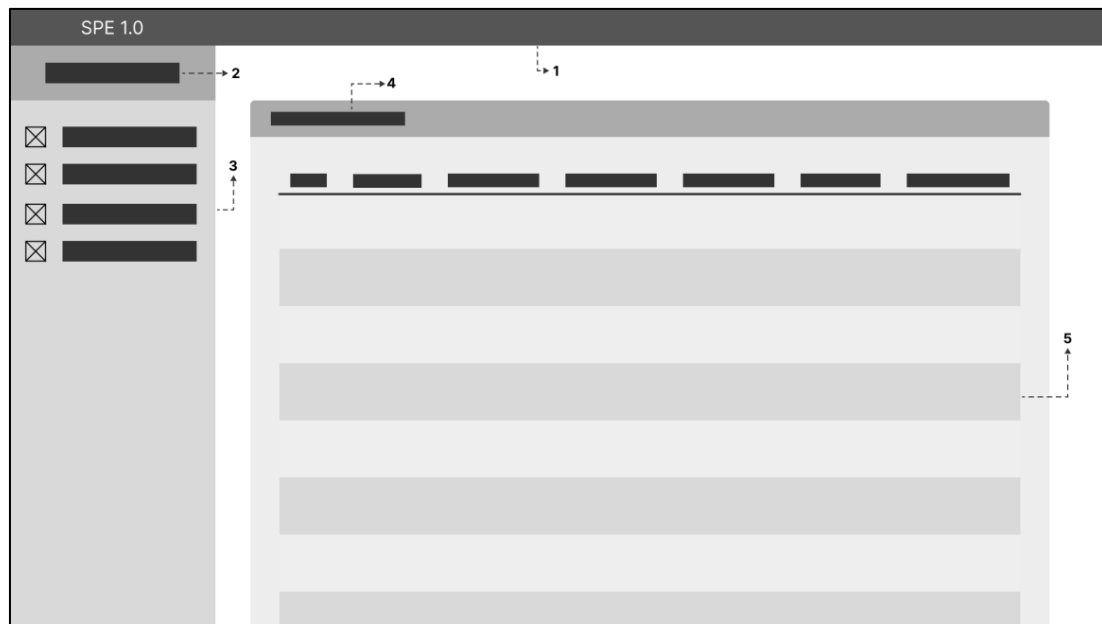
Gambar 3.3. Halaman Daftar Ujian

Keterangan :

- 1 : *Header*
- 2 : Nama *user* yang telah login
- 3 : *Sidebar*
- 4 : Nama halaman
- 5 : Tabel ujian yang telah dibuat

3. Halaman Daftar Mata Pelajaran

Pada halaman daftar mata pelajaran, terdapat tabel yang menampilkan setiap mata pelajaran yang telah dibuat. Halaman ini juga memungkinkan untuk melakukan perbaikan pada detail mata pelajaran, menghapus, juga menambahkan mata pelajaran yang baru. Rancangan halaman daftar mata pelajaran dapat dilihat pada Gambar 3.4.



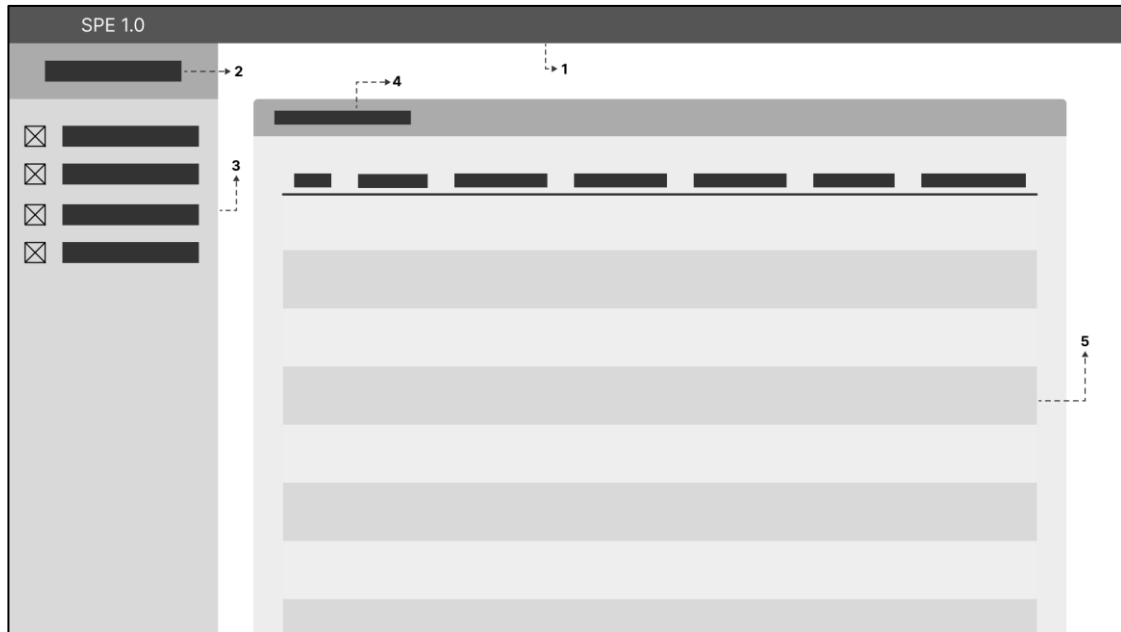
Gambar 3.4. Halaman Daftar Mata Pelajaran

Keterangan :

- 1 : *Header*
- 2 : Nama *user* yang telah login
- 3 : *Sidebar*
- 4 : Nama halaman
- 5 : Tabel mata pelajaran yang telah dibuat

4. Halaman Daftar Paket Soal

Halaman daftar paket soal menampilkan daftar paket soal yang telah dibuat. Terdapat bagian untuk mengubah dan menghapus paket soal yang telah dibuat, serta menambahkan paket soal yang baru. Pada halaman ini juga terdapat bagian untuk menentukan konversi nilai similarity ke skor yang akan digunakan sebagai nilai yang didapat murid.



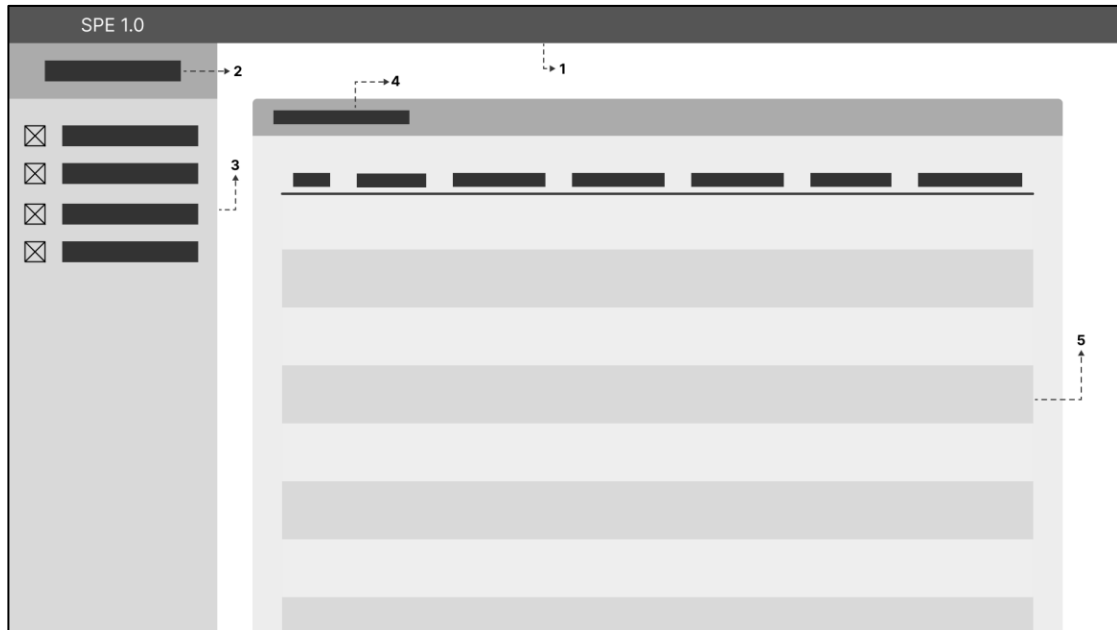
Gambar 3.5. Halaman Daftar Paket Soal

Keterangan :

- 1 : *Header*
- 2 : Nama *user* yang telah login
- 3 : *Sidebar*
- 4 : Nama halaman
- 5 : Tabel paket soal yang telah dibuat

5. Halaman Daftar Bank Soal

Halaman daftar bank soal menampilkan setiap soal yang telah dibuat sebelumnya. Pada halaman ini, terdapat bagian untuk menambahkan, mengubah, serta menghapus soal yang akan digunakan.



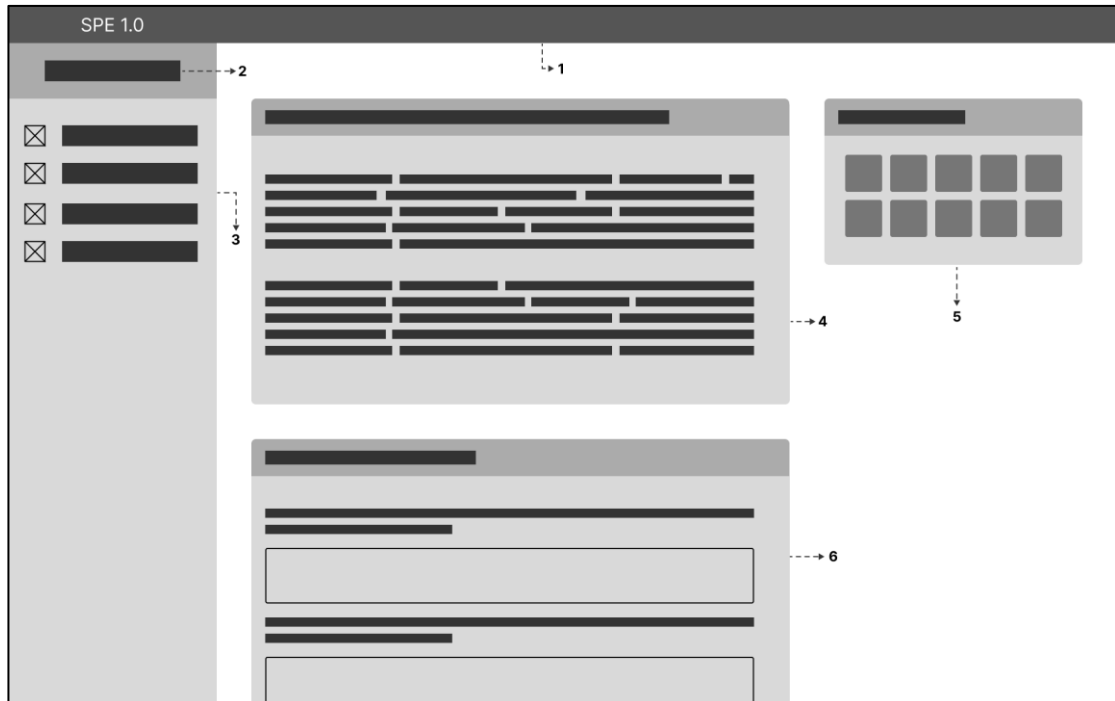
Gambar 3.6. Halaman Daftar Bank Soal

Keterangan :

- 1 : *Header*
- 2 : Nama *user* yang telah login
- 3 : *Sidebar*
- 4 : Nama halaman
- 5 : Tabel soal yang telah dibuat

6. Halaman Ujian Murid

Pada halaman ujian murid, terdapat bagian untuk murid dapat mengerjakan ujian yang diikuti secara soal per soal, serta bagian untuk mengakhiri ujian jika ujian telah selesai.



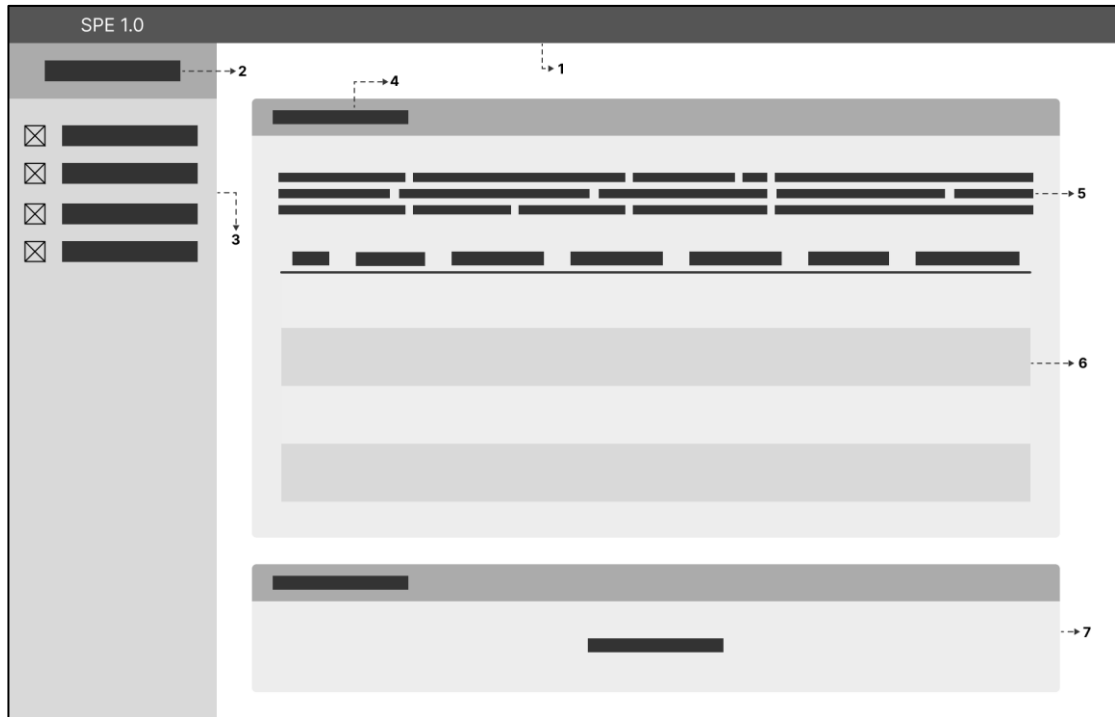
Gambar 3.7. Halaman Ujian Murid

Keterangan :

- 1 : *Header*
- 2 : Nama *user* yang telah login
- 3 : *Sidebar*
- 4 : Deskripsi ujian
- 5 : Nomor soal ujian
- 6 : Soal ujian dan jawaban murid

7. Halaman Hasil Ujian Murid

Halaman hasil ujian murid menampilkan hasil ujian yang didapat setelah murid selesai mengerjakan ujian. Terdapat nilai *similarity*, skor per soal yang didapat, serta skor ujian secara keseluruhan.



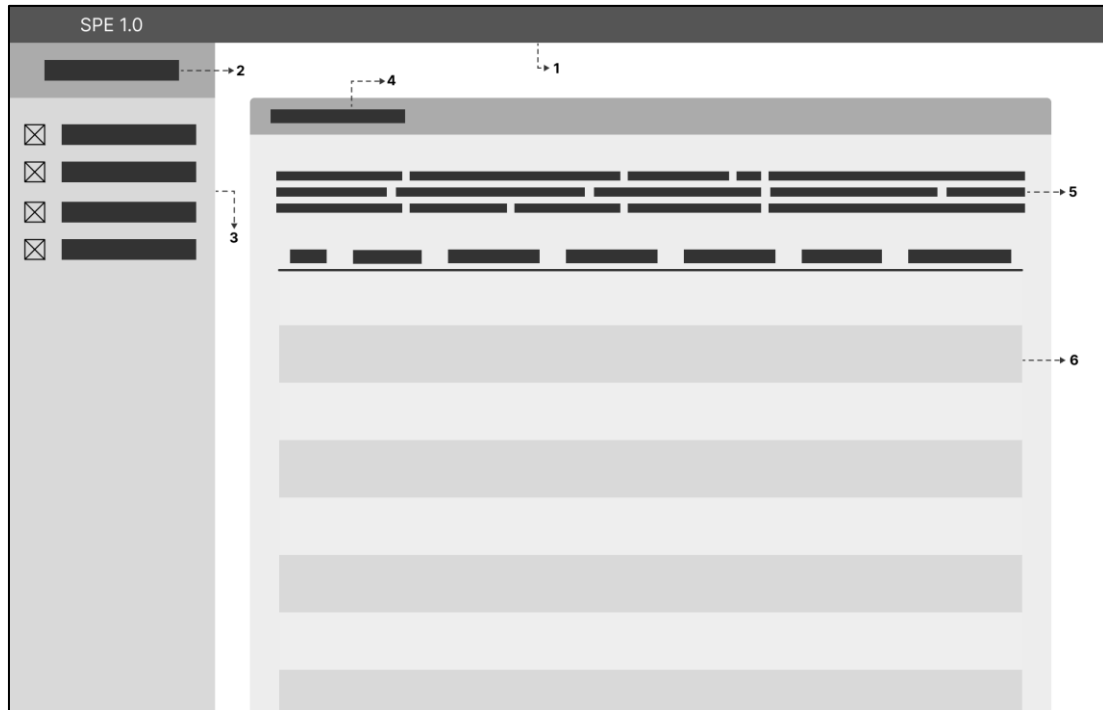
Gambar 3.8. Halaman Hasil Ujian Murid

Keterangan :

- 1 : *Header*
- 2 : Nama *user* yang telah login
- 3 : *Sidebar*
- 4 : Nama halaman
- 5 : Nama dan deskripsi ujian
- 6 : Tabel hasil ujian murid
- 7 : Total nilai

8. Halaman Ranking Peserta Ujian

Halaman ranking peserta ujian menampilkan hasil-hasil yang telah didapat oleh murid. Bagian ini menampilkan skor ujian yang didapat, rata-rata nilai *similarity* yang didapat murid.



Gambar 3.9. Halaman Ranking Peserta Ujian

Keterangan :

- 1 : *Header*
- 2 : Nama *user* yang telah login
- 3 : *Sidebar*
- 4 : Nama halaman
- 5 : Nama dan deskripsi ujian
- 6 : Tabel *ranking* peserta ujian

3.4. Tahapan Evaluasi

Tahapan evaluasi dilakukan untuk mengetahui apakah penilaian yang dilakukan sistem mendekati penilaian yang dilakukan oleh pengajar.

Pengujian dilakukan dengan cara mencari rata-rata selisih dari setiap ujian yang dilakukan pelajar. Rata-rata selisih dapat ditemukan dengan menggunakan persamaan :

$$\bar{x}_{selisih} = \frac{\sum_{i=1}^n |(x_s - x_m)|}{n} \quad (3.1)$$

Dimana :

\bar{x} = rata-rata selisih

x_s = poin dari sistem

x_m = poin secara manual

n = banyak data

Kemudian setiap poin yang didapat akan dibandingkan dengan KKM dari mata pelajaran yang diujikan, dalam kasus data ini KKM yang diambil adalah 80. Maka setiap poin yang lebih rendah daripada 80 akan ditandai sebagai “tidak lulus”, sedangkan yang diatas 80 akan ditandai sebagai “lulus”.

Evaluasi dilakukan dengan menggunakan variabel yang telah dijelaskan pada bagian 2.10 yaitu TP, TN, FP, dan FN.

- *True Positive* (TP) merupakan kondisi dimana penilaian manual menilai “lulus” dan sistem juga menilai “lulus”
- *True Negative* (TN) merupakan kondisi dimana penilaian manual menilai “tidak lulus” dan penilaian sistem juga menilai “tidak lulus”
- *False Positive* (FP) merupakan kondisi dimana penilaian manual menilai “tidak lulus” namun penilaian sistem menilai “lulus”
- *False Negative* (FN) merupakan kondisi dimana penilaian manual menilai “lulus” namun penilaian sistem menilai “tidak lulus”

Variabel-variabel ini akan digunakan untuk menghitung nilai *accuracy*, *precision*, *recall*, dan *f-score* dari sistem. Pada tabel 3.10, ditampilkan 10 data sebagai contoh variabel yang akan digunakan untuk contoh perhitungan evaluasi yang dilakukan pada penelitian ini.

Tabel 3.10. Contoh Variabel untuk Tahapan Evaluasi

Variabel	Jumlah
True Positive (TP)	7
False Positive (FP)	2
True Negative (TN)	0
False Negative (FN)	1
Total	10

Dari Tabel 3.10 dapat dihitung nilai *accuracy*, *precision*, *recall*, dan *f-score* menggunakan persamaan 2.2, 2.3, 2.4 dan 2.5.

$$1. \quad Accuracy = \frac{Jumlah \text{ penilaian sistem yang benar}}{Total \text{ penilaian yang dilakukan}} = \frac{7}{10} = 0,7$$

$$2. \quad Precision = \frac{True \ Positive}{True \ Positive + False \ Positive} = \frac{7}{7+2} = \frac{7}{9} = 0,77$$

$$3. \quad Recall = \frac{True \ Positive}{True \ Positive + False \ Negative} = \frac{7}{7+1} = \frac{7}{8} = 0,87$$

$$4. \quad Fscore = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} = 2 \times \frac{0,77 \times 0,87}{0,77 + 0,87} = 2 \times \frac{0,67}{1,64} = 2 \times 0,41 = 0,82$$

BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1. Implementasi Sistem

4.1.1. Perangkat keras yang digunakan

Rincian perangkat keras yang digunakan dalam pembuatan sistem ini adalah sebagai berikut.

1. Laptop ACER E5-471G
2. Processor Intel Core I5 421U
3. RAM DDR3 Sebesar 16 GB
4. GPU Nvidia GeForce 820m.

4.1.2. Perangkat lunak yang digunakan

Rincian perangkat lunak yang digunakan dalam pembuatan sistem ini adalah sebagai berikut.

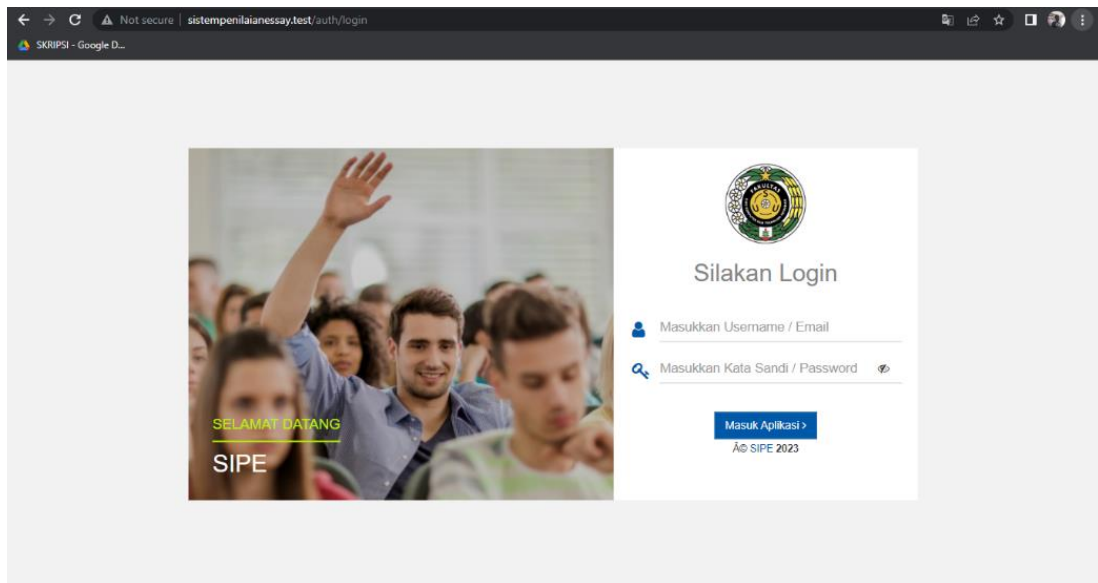
1. Visual Studio Code
2. Anaconda 22.9.0
3. Python 3.9.16
4. PHP 7.4
5. MySQL 8.0.

4.1.3. Implementasi perancangan antar muka

Berikut hasil implementasi rancangan Bab 3 pada sistem

1. Tampilan Halaman Masuk

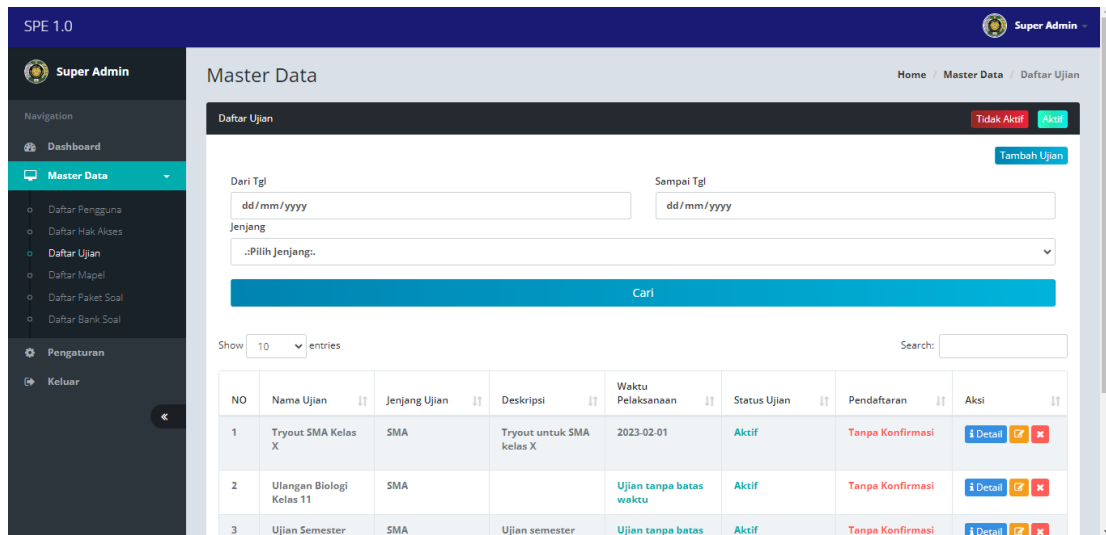
Tampilan ini merupakan tampilan yang pertama kali muncul ketika aplikasi dijalankan. Halaman ini memiliki fitur login, dimana pengguna dengan username dan password dapat masuk. Berikut tampilan untuk halaman masuk sistem dapat dilihat pada gambar 4.1.



Gambar 4.1. Tampilan Halaman Masuk

2. Tampilan Halaman Daftar Ujian

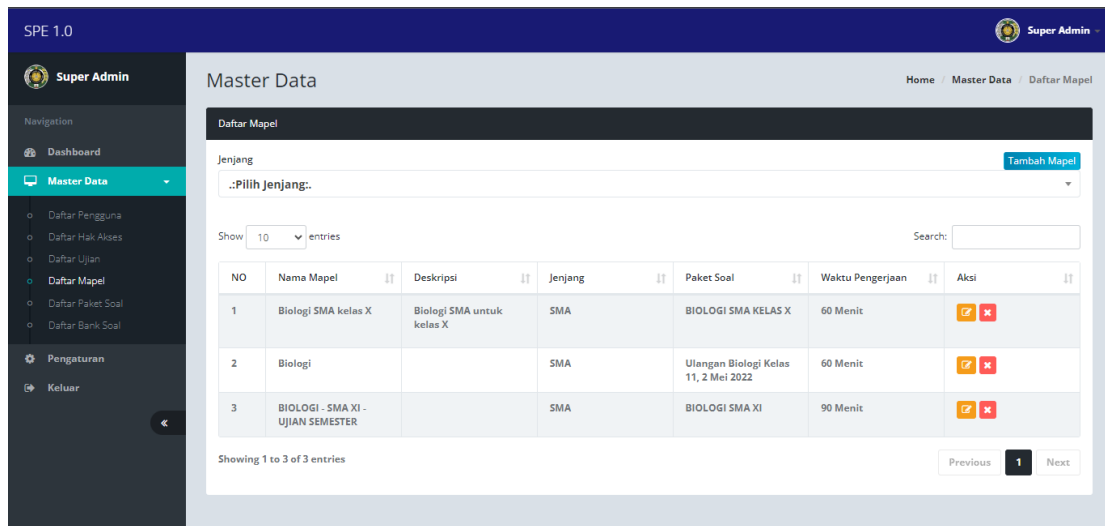
Tampilan ini dimiliki oleh pengguna dengan hak akses admin. Tampilan ini menunjukkan seluruh daftar ujian yang telah dibuat dan mengedit detail mengenai ujian tersebut. Berikut halaman daftar ujian dapat dilihat pada gambar 4.2.



Gambar 4.2. Tampilan Halaman Daftar Ujian

3. Tampilan Halaman Daftar Mata Pelajaran

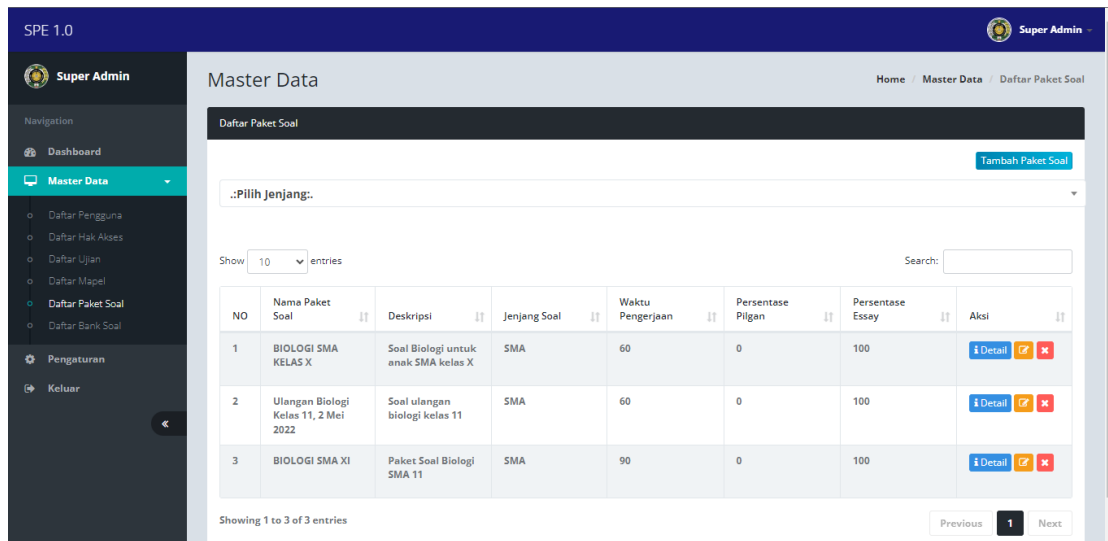
Sama seperti halaman daftar ujian, tampilan ini dapat dilihat dengan hak akses admin. Pada halaman ini ditampilkan seluruh mata pelajaran yang tersedia. Admin juga dapat menambah, mengubah, atau menghapus mata pelajaran yang tersedia. Berikut halaman daftar mata pelajaran dapat dilihat pada gambar 4.3.



Gambar 4.3. Tampilan Halaman Daftar Mata Pelajaran

4. Tampilan Halaman Daftar Paket Soal

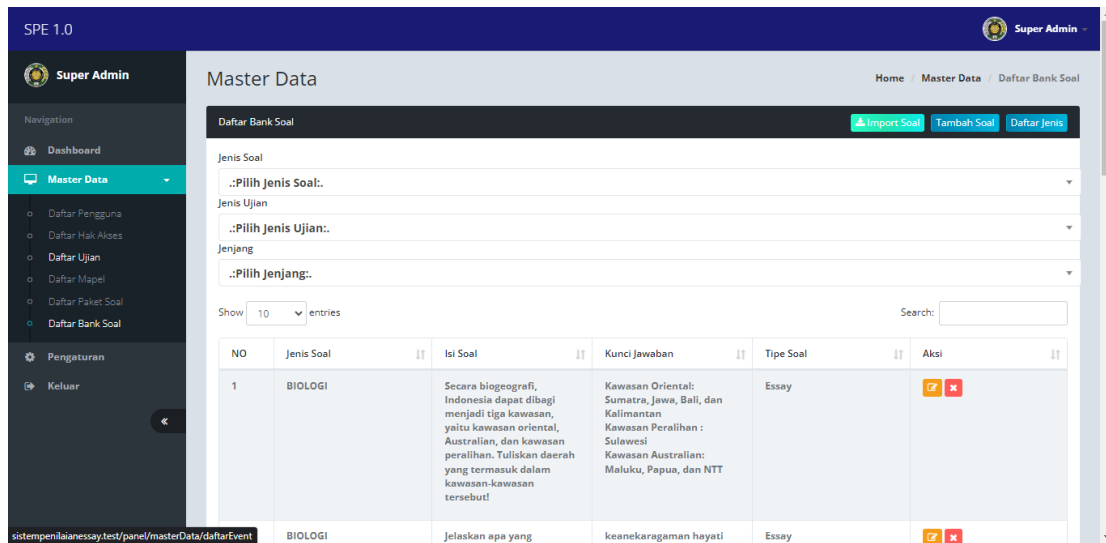
Halaman daftar paket soal dimiliki oleh pengguna yang memiliki hak akses admin. Pada halaman ini ditampilkan seluruh paket soal yang telah dibuat. Admin juga dapat menambahkan, mengubah, atau menghapus paket soal yang ada. Berikut tampilan halaman daftar paket soal dapat dilihat pada gambar 4.4.



Gambar 4.4. Tampilan Halaman Daftar Paket Soal

5. Tampilan Halaman Bank Soal

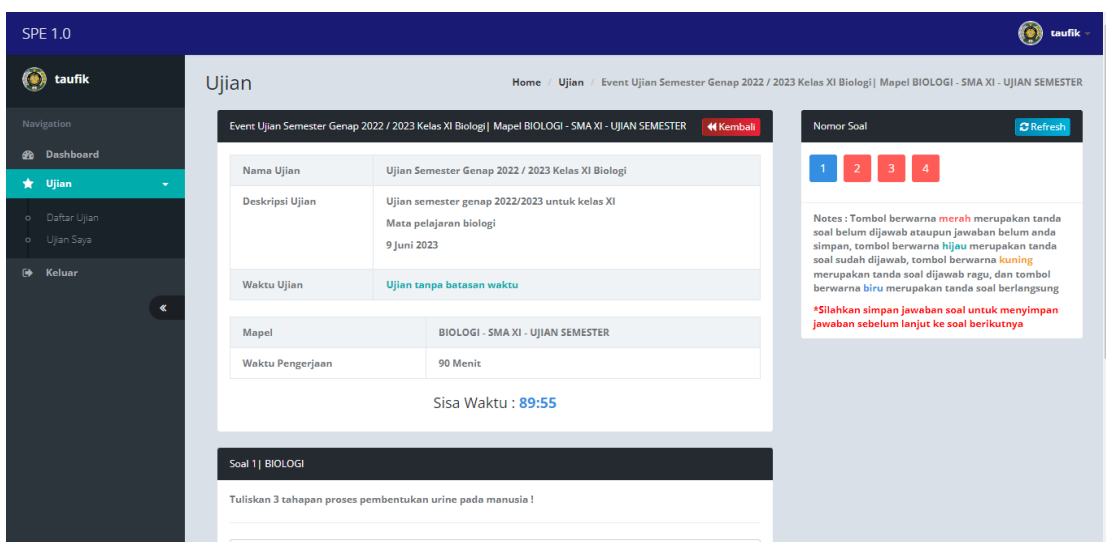
Halaman ini memerlukan pengguna dengan hak akses admin untuk dapat ditampilkan. Pada halaman daftar bank soal, admin dapat melihat, menambahkan, mengubah atau menghapus setiap bank soal yang ada. Tampilan halaman ini dapat dilihat pada gambar 4.5.



Gambar 4.5. Tampilan Halaman Bank Soal

6. Tampilan Halaman Ujian

Halaman ini merupakan halaman dimana murid melakukan ujian. Di halaman ini terdapat deskripsi ujian yang dijalani dan waktu yang dimiliki murid untuk menyelesaikan ujian. Pada bawah deskripsi ujian terdapat soal, dan form untuk menjawab soal. Berikut tampilan dari halaman ujian murid dapat dilihat pada gambar 4.6.



Gambar 4.6. Tampilan Halaman Ujian

7. Tampilan Halaman Hasil Ujian Murid

Halaman ini merupakan halaman setelah murid selesai melakukan ujian. Halaman ini menampilkan hasil kesamaan antara jawaban murid dengan kunci jawaban, serta nilai

keseluruhan yang didapat murid pada ujian tersebut. Tampilan halaman ini dapat dilihat pada gambar 4.7.

No	Jawaban	Koreksi	Point
1	filtrasi (peyaringan), reabsorsi (penyerapan kembali), augmentasi urine	✓	20 (79.28%)
2	memberi makanan dan menampung sel telur yang telah dibuahi sampai menjadi janin atau sampai siap untuk dilahirkan	✓	0 (25%)
3	hidung, faring, laring, trakea, bronkiolus, paru-paru	✓	20 (78.33%)
4	rangsangan (implus) - reseptor (indra) - saraf sensorik - sumsum tulang belakang - saraf motorik - efektor (otot)	✓	20 (78.72%)
Total Nilai			60

Gambar 4.7. Tampilan Halaman Hasil Ujian Murid

8. Halaman Ranking Hasil Ujian

Halaman ini hanya dapat diakses oleh pengguna yang memiliki hak akses admin. Pada halaman ini ditunjukkan seluruh nilai yang didapat peserta beserta nilai rata-rata kesamaannya. Berikut tampilan halaman ini dapat dilihat pada gambar 4.8.

Position	Username	Nama Lengkap	Nama Sekolah	Skor	Skor Akurasi	Durasi Pengerjaan	Aksi
1	03-006-152	Uci Wulan Dari		85	64	00:01:49	Reset Ujian
2	03-006-143	Mika Asty Roniya Nanda M	SMA Tamansiswa	85	63	00:02:12	Reset Ujian
3	03-006-142	Jingga Anggrani	SMA Tamansiswa	80	63	00:02:11	Reset Ujian
4	03-006-144	Nailah Sofiah	SMA Tamansiswa	80	62	00:02:27	Reset Ujian

Gambar 4.8. Tampilan Halaman Ranking Hasil Ujian

4.2. Pengujian Sistem

Untuk dapat melakukan pengujian terhadap sistem, maka sistem perlu mendapatkan nilai akurasi yang didapat murid pada ujian dan ulangan yang telah dilakukan. Ujian dilakukan oleh murid sebanyak 25 orang dengan soal sebanyak 4 soal dan ulangan yang dilakukan oleh murid sebanyak 12 orang dengan soal sebanyak 5 soal. Hasil akurasi ujian yang didapat murid dapat dilihat pada tabel 4.1 dan hasil akurasi ulangan yang didapat murid dapat dilihat pada tabel 4.2.

Tabel 4.1. Hasil Akurasi Ujian pada Sistem

Username	Nilai akurasi			
	Soal 1	Soal 2	Soal 3	Soal 4
03-006-130	87,39%	42,16%	92,16%	83,19%
03-006-131	80,34%	25,93%	92,68%	98,46%
03-006-132	98,31%	57,56%	95,24%	65,82%
03-006-133	89,66%	64,81%	93,14%	98,46%
03-006-134	93,69%	73,24%	93,14%	98,46%
03-006-135	65,91%	42,16%	82,47%	66,07%
03-006-136	92,73%	68,75%	92,61%	98,46%
03-006-137	99,15%	59,30%	89,00%	69,80%
03-006-138	83,50%	50,49%	82,47%	65,82%
03-006-139	93,69%	78,38%	92,16%	98,46%
03-006-140	67,42%	38,46%	82,47%	64,47%
03-006-141	93,69%	49,36%	93,14%	98,46%
03-006-142	93,69%	78,38%	93,14%	98,46%
03-006-143	93,69%	77,55%	90,45%	98,46%
03-006-144	91,89%	75,68%	93,14%	98,46%
03-006-145	72,55%	42,16%	82,47%	74,82%
03-006-146	91,89%	50,73%	70,83%	98,46%

Tabel 4.1. Hasil Akurasi Ujian pada Sistem (Lanjutan)

Username	Nilai akurasi			
	Soal 1	Soal 2	Soal 3	Soal 4
03-006-147	74,51%	42,16%	82,47%	76,26%
03-006-148	92,86%	73,24%	93,14%	98,46%
03-006-149	85,25%	57,56%	88,21%	74,82%
03-006-150	92,86%	73,24%	80,61%	79,62%
03-006-151	92,86%	73,24%	82,47%	79,65%
03-006-152	92,86%	73,74%	93,14%	98,46%
03-006-153	84,62%	59,38%	93,14%	98,46%
03-006-154	88,14%	57,56%	82,47%	74,82%

Tabel 4.2. Hasil Akurasi Ulangan pada Sistem

Username	Nilai akurasi				
	Soal 1	Soal 2	Soal 3	Soal 4	Soal 5
RA-001	49,30%	39,62%	90,57%	48,09%	76,71%
DN-002	73,10%	47,67%	90,55%	48,11%	72,41%
MR-003	35,54%	44,64%	90,40%	35,56%	70,59%
DN-004	42,42%	14,36%	90,57%	55,09%	75,76%
BA-005	35,50%	45,61%	90,50%	86,67%	69,74%
NU-006	49,30%	43,88%	31,58%	85,25%	69,60%
BS-007	35,89%	40,58%	23,14%	85,25%	83,56%
RD-008	49,32%	45,99%	95,83%	70,47%	72,44%
RF-009	47,89%	45,98%	24,10%	48,04%	72,41%
SR-010	49,37%	14,89%	23,14%	85,28%	50,00%
AR-011	60,29%	57,01%	90,53%	85,17%	50,00%

Tabel 4.2. Hasil Akurasi Ulangan pada Sistem (Lanjutan)

Username	Nilai akurasi				
	Soal 1	Soal 2	Soal 3	Soal 4	Soal 5
ND-012	49,24%	14,87%	23,16%	83,12%	50,50%

Nilai akurasi kemudian di konversi menjadi poin. Poin didapat berdasarkan dari pengelompokkan yang telah dibuat oleh pengajar. Tabel konversi nilai akurasi ke poin dapat dilihat pada tabel 3.8 untuk ujian, dan tabel 3.9 untuk ulangan. Hasil konversi dapat dilihat pada tabel 4.3 dan tabel 4.4.

Tabel 4.3. Hasil Konversi Nilai Akurasi Ujian yang Diperoleh

Username	Poin yang diperoleh				Total Poin
	Soal 1	Soal 2	Soal 3	Soal 4	
03-006-130	25	15	25	25	90
03-006-131	25	0	25	25	75
03-006-132	25	15	25	15	80
03-006-133	25	15	25	25	90
03-006-134	25	25	25	25	100
03-006-135	15	15	25	15	70
03-006-136	25	25	25	25	100
03-006-137	25	15	25	25	90
03-006-138	25	15	25	15	80
03-006-139	25	25	25	25	100
03-006-140	25	15	25	15	80
03-006-141	25	15	25	25	90
03-006-142	25	25	25	25	100
03-006-143	25	25	25	25	100
03-006-144	25	25	25	25	100
03-006-145	25	15	25	25	90

Tabel 4.3. Hasil Konversi Nilai Akurasi Ujian yang Diperoleh (Lanjutan)

Username	Poin yang diperoleh				Total Poin
	Soal 1	Soal 2	Soal 3	Soal 4	
03-006-146	25	15	25	25	90
03-006-147	25	15	25	25	90
03-006-148	25	25	25	25	100
03-006-149	25	15	25	25	90
03-006-150	25	25	25	25	100
03-006-151	25	25	25	25	100
03-006-152	25	25	25	25	100
03-006-153	25	15	25	25	90
03-006-154	25	15	25	25	90

Tabel 4.4. Hasil Konversi Nilai Akurasi Ulangan yang Diperoleh

Username	Poin yang diperoleh					Total Poin
	Soal 1	Soal 2	Soal 3	Soal 4	Soal 5	
RA-001	10	10	20	10	20	70
DN-002	20	10	20	10	20	80
MR-003	10	10	20	10	20	70
DN-004	10	0	20	10	20	60
BA-005	10	10	20	20	20	80
NU-006	10	10	0	20	20	60
BS-007	10	10	0	20	20	60
RD-008	10	10	20	20	20	80
RF-009	10	10	0	10	20	50
SR-010	10	0	0	20	10	40
AR-011	10	10	20	20	10	70
ND-012	10	0	0	20	10	40

Untuk setiap total poin yang berada diatas KKM yaitu 80, maka status ujian dan ulangan pelajar tersebut “lulus”. Untuk setiap total poin yang di bawah 80 maka status ujian dan ulangan pelajar tersebut “tidak lulus”.

Poin tersebut kemudian dibandingkan dengan poin yang didapat dengan penilaian secara manual. Perbandingan antara poin yang didapat oleh sistem dengan poin yang didapat secara manual dapat dilihat pada tabel 4.3.

Tabel 4.5. Perbandingan Sistem dengan Cara Manual pada Ujian

Username	Poin yang diperoleh dari sistem	Status Poin dari Sistem	Poin yang diperoleh secara manual	Status Poin dari manual
03-006-130	90	Lulus	95	Lulus
03-006-131	75	Tidak Lulus	95	Lulus
03-006-132	80	Lulus	90	Lulus
03-006-133	90	Lulus	100	Lulus
03-006-134	100	Lulus	95	Lulus
03-006-135	70	Tidak Lulus	80	Lulus
03-006-136	100	Lulus	95	Lulus
03-006-137	90	Lulus	100	Lulus
03-006-138	80	Lulus	85	Lulus
03-006-139	100	Lulus	95	Lulus
03-006-140	80	Lulus	75	Tidak Lulus
03-006-141	90	Lulus	95	Lulus
03-006-142	100	Lulus	95	Lulus
03-006-143	100	Lulus	95	Lulus
03-006-144	100	Lulus	95	Lulus
03-006-145	90	Lulus	95	Lulus
03-006-146	90	Lulus	95	Lulus
03-006-147	90	Lulus	95	Lulus
03-006-148	100	Lulus	95	Lulus
03-006-149	90	Lulus	95	Lulus
03-006-150	100	Lulus	90	Lulus

Tabel 4.5. Perbandingan Sistem dengan Cara Manual pada Ujian (Lanjutan)

Username	Poin yang diperoleh dari sistem	Status Poin dari Sistem	Poin yang diperoleh secara manual	Status Poin dari manual
03-006-151	100	Lulus	90	Lulus
03-006-152	100	Lulus	95	Lulus
03-006-153	90	Lulus	90	Lulus
03-006-154	90	Lulus	95	Lulus

Tabel 4.6. Perbandingan Sistem dengan Cara Manual pada Ulangan

Username	Poin yang diperoleh dari sistem	Status Poin dari Sistem	Poin yang diperoleh secara manual	Status Poin dari manual
RA-001	70	Tidak Lulus	80	Lulus
DN-002	80	Lulus	90	Lulus
MR-003	70	Tidak Lulus	80	Lulus
DN-004	60	Tidak Lulus	60	Tidak Lulus
BA-005	80	Lulus	100	Lulus
NU-006	60	Tidak Lulus	70	Tidak Lulus
BS-007	60	Tidak Lulus	80	Lulus
RD-008	80	Lulus	80	Lulus
RF-009	50	Tidak Lulus	50	Tidak Lulus
SR-010	40	Tidak Lulus	60	Tidak Lulus
AR-011	70	Tidak Lulus	90	Lulus
ND-012	40	Tidak Lulus	40	Tidak Lulus

Nilai rata-rata selisih antara penilaian dari sistem dengan penilaian secara manual dapat dihitung sebagai berikut.

$$\bar{x}_{selisih} = \frac{\sum_{i=1}^n |x_s - x_m|}{n}$$

$$\bar{x}_{selisih} = \frac{285}{37}$$

$$\bar{x}_{selisih} = 7,7$$

Maka rata-rata selisih yang didapat sistem terhadap penilaian secara manual adalah 7,7 poin.

Dari hasil data pada tabel 4.5 dan tabel 4.6, kemudian dilakukan pengujian pada *accuracy*, *precision*, *recall* dan *fscore* yang dijelaskan pada bagian 3.4. Variabel-variabel yang diperlukan untuk melakukan pengujian dapat dilihat pada tabel 4.7.

Tabel 4.7. Variabel Proses Pengujian

Variabel	Jumlah
True Positive (TP)	25
False Positive (FP)	1
True Negative (TN)	5
False Negative (FN)	6
Total	37

Berdasarkan nilai variabel pada tabel x.x, dapat dilakukan pengujian dengan perhitungan *accuracy*, *precision*, *recall*, dan *fscore* yang telah dijelaskan sebelumnya.

a. *Accuracy*

$$Accuracy = \frac{\text{Jumlah penilaian sistem yang benar}}{\text{Total penilaian yang dilakukan}}$$

$$Accuracy = \frac{30}{37} = 0,811$$

b. *Precision*

$$Precision = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

$$Precision = \frac{25}{25 + 1} = \frac{25}{26} = 0,962$$

c. *Recall*

$$Recall = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

$$Recall = \frac{25}{25 + 6} = \frac{25}{31} = 0,806$$

d. *Fscore*

$$Fscore = 2 \times \frac{Presisi \times Recall}{Presisi + Recall}$$

$$Fscore = 2 \times \frac{0,962 \times 0,806}{0,962 + 0,806}$$

$$Fscore = 2 \times \frac{0,775}{1,768} = \frac{1,551}{1,768} = 0,877$$

Berdasarkan hasil pengujian tersebut, sistem mendapatkan nilai *accuracy* sebesar 0,811, *precision* sebesar 0,962, *recall* sebesar 0,806, dan *f1 score* sebesar 0,877.

BAB 5

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Dari hasil pengujian yang telah disajikan dalam Bab 4, beberapa kesimpulan yang dapat diambil oleh penulis adalah sebagai berikut:

1. Algoritma *Ratcliffe/Obershelp* dapat melakukan penilaian secara otomatis untuk jawaban uraian dengan rata-rata selisih yang didapat 7,7 terhadap penilaian secara manual
2. Algoritma *Ratcliffe/Obershelp* mampu menghasilkan akurasi sebesar 81%, presisi sebesar 96%, *recall* sebesar 80%, dan *fscore* sebesar 87% dalam memberikan penilaian terhadap ujian uraian

5.2. Saran

Adapun beberapa saran dari penulis untuk pengembangan penelitian ini kedepannya, yaitu :

1. Menerapkan metode pemeriksaan struktur kalimat pada jawaban murid
2. Memperbanyak jumlah data pada korpus yang digunakan untuk memeriksa sinonim kata pada jawaban murid terhadap kunci jawaban
3. Menerapkan metode *machine learning* agar dapat mengenali kontekstual penggunaan kata pada jawaban murid dan kunci jawaban.

DAFTAR PUSTAKA

- Amini, F. (2020). *Rancang Bangun Aplikasi Deteksi Plagiarisme Menggunakan Algoritma Manber dengan Pendekatan Biword*. Skripsi. Universitas Islam Negeri Maulana Malik Ibrahim.
- Afzalurrahmah. (2019). *Normalisasi Mikroteks Berbentuk Kontekstual Berbahasa Indonesia pada Twitter Menggunakan Dictionary-Based dan Algoritma Longest Common Subsequence (LCS)*. Skripsi. Universitas Sumatera Utara.
- Arifin, E. Z. (2015). *Kesinoniman dalam bahasa Indonesia*. Pujangga: Jurnal Bahasa dan Sastra, 1(1), 1-13.
- Billhaqqi, T. T. I., Wicaksono, G. W., & Aditya, C. S. K. (2021). *Analisis Perbandingan Algoritma Rabin-Karp Dan Winnowing Dalam Penilaian Jawaban Otomatis*. Prosiding SENTRA (Seminar Teknologi dan Rekayasa).
- Cruse, A. (2004). *Meaning in language: An introduction to semantics and pragmatics*. Oxford University Press UK.
- Fahmi, S., Purnamawati, L., Shidik, G. F., Muljono, M. A., & Fanani, A. Z. (2020). *Sentiment Analysis of Student Review in Learning Management System Based on Sastrawi Stemmer and SVM-PSO*. International Seminar on Application for Technology of Information and Communication (iSemantic).
- Fau, A., Mesran., M. & Ginting, G. L. (2017). *Analisa Perbandingan Boyer Moore Dan Knuth Morris Pratt Dalam Pencarian Judul Buku Menerapkan Metode Perbandingan Eksponensial*. Jurnal TIMES, 6(1), 12-22.
- Izzah, N., Yusliani, N. & Roodiah, D. (2022). *Sistem Deteksi Kemiripan Teks Pada Berita Berbahasa Indonesia Menggunakan algoritma Ratcliff/Obershelp*. Jurnal Linguistik Komputasional (JLK). 5(1).
- Joane, Y. L., Sinsuw, A., & Jacobus, A. (2017). *Rancang Bangun Aplikasi Deteksi Kemiripan Dokumen Teks Menggunakan Algoritma Ratcliff/Obershelp*. E-Journal Teknik Informatika, 11(1).

- Metzener, D. E., & Ratclif, J. W. *Pattern matching: The gestalt approach*. Tersedia pada: < Dr. Dobb's The World of Software Development > (diakses 13 Desember 2023)
- Pradnya, W. M. & Kurniawati, F. E. (2020). *Implementasi Algoritma Winnowing Pada Sistem Penilaian Otomatis Jawaban Esai Pada Ujian Online Berbasis Web*. Jurnal Teknik Komputer AMIK BSI, 6(2), 169-175.
- Putra, N. P. & Sularno, S. (2019). *Penerapan Algoritma Rabin-karp dengan Pendekatan Synonym Recognition sebagai Antisipasi Plagiarisme pada Penulisan Skripsi*. Jurnal Teknologi Dan Sistem Informasi Bisnis, 1(2), 130- 140.
- Ruslan, R., Gunawan, G., & Tjandra, S. (2018). *Sistem Penilaian Otomatis Jawaban Esai Menggunakan Metode GLSA*. Seminar Nasional Aplikasi Teknologi Informasi (SNATi).
- Salim, M. A. (2017). *Pengembangan Aplikasi Penilaian Ujian Essay Berbasis Online Menggunakan Algoritma Nazief dan Adriani dengan Metode Cosine Similarity*. IT-Edu: Jurnal Information Technology and Education, 2(1).
- Schleimer, S., Wilkerson, D. S., & Aiken, A. (2003, June). *Winnowing: local algorithms for document fingerprinting*. ACM SIGMOD international conference on Management of data 2003, 76-85.
- Stevesson, D., Agung, H., & Mulia, F. (2018). *Aplikasi Pendeteksi Plagiarisme Tugas dan Makalah Pada Sekolah Menggunakan Algoritma Rabin Karp*. Jurnal Algoritma, Logika dan Komputasi, 1(1). 12-17
- Yusuf, B., Vivianie, S., Marsya, J. M., & Sofyan, Z. (2019). *Analisis Perbandingan Algoritma Rabin-Karp dan Ratcliff/Obershelp untuk Menghitung Kesamaan Teks dalam Bahasa Indonesia*. Seminar Nasional APTIKOM (SEMNASTIK) 2019.

LAMPIRAN 1: Data Uji

Username : 03-006-130

No. Soal	Jawaban Pelajar	Akurasi
1	filtrasi (peyaringan), reabsorsi (penyerapan kembali), augmentasi urine	87,39%
2	memberi makanan dan menampung sel telur yang telah dibuahi sampai menjadi janin atau sampai siap untuk dilahirkan	42,16%
3	rangsangan (implus) - reseptor (indra) - saraf sensorik - sumsum tulang belakang - saraf motorik - efektor (otot)	92,16%
4	hidung, faring, laring, trakea, bronkiolus, paru-paru	83,19%

Username : 03-006-131

No. Soal	Jawaban Pelajar	Akurasi
1	filtrasi, penyaringan, augmentasi, penyerapan kembali, augmentasi urine	80,34%
2	untuk menerima sel telur yang dibuahi yang akan berubah menjadi janin dan menahannya selama perkembangan, dan untuk membantu mendukung janin selama masa kehamilan	25,93%
3	perangsang (impuls) - reseptor (indera) - saraf sensorik - sumsum tulang belakang - saraf motorik - efektor (otot)	92,68%
4	hidung, faring, laring, trakea, bronkus, bronkiolus, alveolus, paru-paru	98,46%

Username : 03-006-132

No. Soal	Jawaban Pelajar	Akurasi
1	filtrasi (penyaringan), rearbsorsi (penyerapan kembali), augmentasi (pengumpulan)	98,31%

2	uterus berfungsi untuk memberi makan dan menampung sel telur yang telah dibuahi sampai menjadi janin atau sampai siap untuk dilahirkan	57,56%
3	perangsang (impuls) - reseptor (alat indera) - saraf sensorik - sumsum tulang belakang - saraf motorik - efektor (otot)	95,24%
4	hidung - faring - epiglottis - laring - trakea - tabung bronkial - bronkiolus - paru - paru - alveolus - diafragma	65,82%

Username : 03-006-133

No. Soal	Jawaban Pelajar	Akurasi
1	filtrasi (penyerapan), reabsorpsi (penyerapan kembali), augmentasi (pengumpulan)	89,66%
2	uterus atau rahim berfungsi untuk memberi makan dan menampung sel telur yang telah dibuahi sampai menjadi janin atau sampai siap untuk dilahirkan	64,81%
3	rangsangan (impuls) - reseptor (indra) - saraf sensorik - sumsum tulang belakang - saraf motorik - efektor (otot)	93,14%
4	hidung, faring, laring, trakea, bronkus, bronkiolus, alveolus, paru-paru	98,46%

Username : 03-006-134

No. Soal	Jawaban Pelajar	Akurasi
1	filtrasi (penyaringan), reabsorpsi (penyerapan kembali), augmentasi	93,69%
2	uterus berfungsi sebagai tempat perkembangan embrio menjadi janin	73,24%
3	rangsangan (impuls), reseptor (indra), saraf sensorik, sumsum tulang belakang, saraf motorik, efektor (otot)	93,14%
4	hidung, faring, laring, trakea, bronkus, bronkiolus, alveolus, paru paru	98,46%