

Aplikasi Penilaian Otomatis Ujian Esai Berbahasa Indonesia Menggunakan Algoritma *K-Nearest Neighbor* (Studi kasus MAN Cimahi)

¹Lucky Ramdani, ²Mufid Jamaluddin, ³Nurindah Yuniarti,
⁴Ani Rahmani, ⁵Jonner Hutahaean

^{1.2.3.4.5}Jurusan Teknik Komputer dan Informatika, Politeknik Negeri Bandung, Bandung 40012
E-mail: {lucky.ramdani.tif16, mufid.jamaluddin.tif16, nurindah.yuniarti.tif16} @polban.ac.id
{ani rahmani, jonnerh} @jtk.polban.ac.id

ABSTRAK

Ujian esai adalah sebuah bentuk evaluasi untuk mengetahui kemampuan siswa dalam penguasaan materi yang telah diberikan. Ujian esai memiliki kekurangan, yaitu guru memerlukan waktu yang lama untuk memeriksa jawaban karena harus membaca satu persatu jawaban siswa. Di samping itu terdapat kendala lain, salah satunya tulisan siswa yang sulit dibaca. Banyaknya jumlah jawaban siswa yang harus dinilai dapat menurunkan kualitas penilaian, karena penilaian yang kurang konsisten. Untuk mengatasi masalah tersebut, Aplikasi Penilaian Otomatis Ujian Esai Berbahasa Indonesia dibuat sebagai alternatif bagi guru dalam menilai ujian esai. Pengembangan aplikasi menggunakan model Waterfall. Penilaian menggunakan *machine learning* dengan metode pembobotan *normalized term frequency - relevance frequency* (TR-RF) dan algoritma *K-Nearest Neighbor* (K-NN). Di samping membantu meningkatkan kecepatan menilai, aplikasi yang dibangun juga menyertakan fitur ujian esai secara *online*, sehingga dapat mengatasi masalah tulisan jawaban siswa yang tidak terbaca. Fitur lainnya adalah laporan hasil ujian, baik laporan satu ujian maupun laporan seluruh ujian yang pernah dilakukan. Terdapat 3 fitur aplikasi yang sudah terimplementasi yaitu pelaksanaan ujian online, penilaian otomatis, dan laporan. Hasil pengujian penilaian otomatis masih menunjukkan akurasi yang belum optimal, sehingga masih harus dikembangkan. Aplikasi ini dibangun berdasarkan pada studi kasus di Madrasah Aliyah Negeri Cimahi dan diharapkan dapat dimanfaatkan di sekolah lainnya.

Kata Kunci:

penilaian otomatis, essay scoring, k-nearest neighbor, normalized term frequency, relevance frequency.

1. PENDAHULUAN

Penilaian sering digunakan untuk mengukur performa peserta didik. Penilaian dianggap sebagai salah satu komponen utama dalam pembelajaran di kelas untuk membantu siswa dalam proses belajar. Salah satu cara penilaian dalam pembelajaran di kelas adalah dengan ujian [1].

Ujian dibagi ke dalam dua bentuk yaitu ujian subjektif dan ujian objektif. Ujian subjektif pada umumnya berbentuk esai (uraian). Ujian bentuk esai adalah sejenis tes kemajuan belajar yang memerlukan jawaban yang bersifat pembatasan atau uraian kata-kata [2].

Ujian esai menjadi salah satu pilihan untuk mengevaluasi pembelajaran siswa. Pelaksanaan ujian

bertipe esai memiliki sejumlah kekurangan. Dalam studi kasus di MAN Cimahi, sejumlah 71% guru mengakui bahwa ujian esai memiliki kekurangan yaitu guru memerlukan waktu yang lama untuk menilai jawaban esai, guru harus membaca satu persatu jawaban siswa dan terkadang tidak sedikit tulisan siswa yang sulit dibaca oleh guru, dan banyaknya jumlah jawaban ujian esai siswa yang harus dinilai dapat menyebabkan kualitas penilaian menurun. Sejumlah 90% guru menyatakan bahwa kelelahan akan mengakibatkan inkonsistensi dalam pemberian skor (*human error*).

Masalah tersebut dapat ditangani dengan membuat perangkat lunak untuk melakukan penilaian otomatis pada ujian esai berbahasa Indonesia. Aplikasi dapat membantu pekerjaan guru dalam melakukan penilaian ujian esai, menggunakan *machine learning*.

Aplikasi dibangun menggunakan pendekatan klasifikasi. Terdapat beberapa algoritma klasifikasi pada *machine learning*, salah satunya adalah algoritma *K-Nearest Neighbor* (K-NN) dengan metode pembobotan *Normalize Term Frequency – Relevance Frequency* (NTF-RF). Metode NTF-RF merupakan pengembangan dari metode *Term Frequency – Relevance Frequency* (TF-RF), yaitu nilai TF-RF satu dokumen dibagi dengan nilai maksimal TF-RF pada keseluruhan dokumen.

Penggunaan algoritma K-NN dengan metode pembobotan TF-RF, dipandang lebih konsisten dalam semua skenario penelitian dibandingkan dengan metode pembobotan TF-Chi dan TF-IDF, dengan rekomendasi nilai $k=3$ dan akurasi yang didapatkan hingga 89% [3].

Selain itu, penilaian esai dengan metode K-NN memiliki akurasi lebih baik dibandingkan dengan metode SVM dan C4.5 [4], metode K-NN memiliki nilai *f-measure* lebih tinggi daripada *Nearest Weighted K-Nearest Neighbor* pada data tidak seimbang pada nilai $k=3$ [5]. Penggunaan K-NN pada penilaian esai dengan nilai $k=3$ lebih baik dibandingkan $k=5$ [3], dan metode pembobotan *supervised term weighting* dengan menggunakan NTF-RF memiliki akurasi yang lebih baik dibandingkan *unsupervised term weighting* [6].

2. METODE PENGEMBANGAN APLIKASI

Metodologi pengembangan aplikasi yang digunakan adalah *software development life cycle* (SDLC) model Waterfall dengan pendekatan objek. Tahapan SDLC tersebut diuraikan pada sub-sub bab bagian ini.

2.1 Analisis Kebutuhan Sistem

Analisis ini dilakukan untuk mengidentifikasi kebutuhan yang harus ada pada aplikasi.

1. Studi Pendahuluan Penilaian Ujian Esai

Pada tahap ini dilakukan identifikasi mengenai proses penilaian ujian esai, seperti studi kasus ujian, pengumpulan soal dan jawaban ujian, proses penilaian ujian esai, serta data master yang perlu dikelola oleh aplikasi yang terdiri dari data kelas, data siswa, data guru dan data mata pelajaran.

2. Menentukan Kebutuhan Fungsional Sistem

Kebutuhan fungsional (fitur) pada aplikasi yaitu:

- Fitur untuk mengelola ujian esai yang dibuat sebagai fasilitas guru, yaitu membuat ujian esai, mengedit ujian esai yang dapat dilakukan jika ujian belum dilaksanakan, menghapus ujian esai yang dapat dilakukan jika ujian belum dilaksanakan, mencari ujian esai, menampilkan ujian esai, memulai ujian esai.
- Fitur untuk mengerjakan ujian esai, yaitu aplikasi menampilkan soal esai dan siswa menjawab soal esai tersebut.
- Fitur untuk menilai ujian esai, yaitu guru menilai sebagian jawaban esai siswa untuk dijadikan dataset latihan jawaban esai siswa, dan guru memulai penilaian secara otomatis oleh aplikasi berdasarkan dataset latihan jawaban esai siswa yang telah dinilai guru.
- Fitur untuk melihat daftar nilai ujian siswa, yaitu aplikasi menampilkan daftar nilai setiap ujian siswa dan nilai seluruh ujian yang telah dilaksanakan.

2.2 Analisis Penilaian Esai Otomatis

Terdapat beberapa hal yang dianalisis dalam pengembangan aplikasi, yaitu:

2.2.1 Penilaian Esai

Dalam melakukan penilaian, terdapat dua teknik yang dapat digunakan oleh guru, pertama adalah *point method*, dan kedua adalah *rating method*. Pada *point method* setiap jawaban dibandingkan dengan jawaban ideal yang telah ditetapkan dalam kunci jawaban dan skor yang diberikan kepada setiap jawaban akan tergantung pada derajat kepedaannya dengan kunci jawaban. Sedangkan dalam *rating method*, Setiap jawaban siswa ditetapkan dalam salah satu kelompok yang sudah dipilah-pilah berdasarkan mutunya selagi jawaban tersebut di baca. Kelompok-kelompok tersebut menyatakan mutu dan menentukan berapa skor yang dapat diberikan kepada setiap jawaban [7].

Selain itu, terdapat dua pendekatan yang digunakan dalam penilaian, yaitu pendekatan semantik dan pendekatan klasifikasi. Pendekatan semantik dilakukan dengan membandingkan antara esai jawaban siswa dengan esai kunci jawaban yang telah disiapkan oleh *human rater*. Pendekatan klasifikasi dilakukan dengan menganggap apabila penilaian esai otomatis merupakan masalah pengelompokan teks secara otomatis yang dapat diselesaikan dengan metode *machine learning* [3].

Pada penilaian esai otomatis, penggunaan klasifikasi dengan metode K-NN memiliki esensi yang sama dengan penilaian esai manual dengan metode *rating method*, yaitu klasifikasi jawaban siswa terhadap jawaban siswa lainnya berdasarkan kata atau frasa penting. Untuk penilaian esai manual pada aplikasi yang dinilai oleh guru sebagai data latih untuk penilaian otomatis, guru dapat menggunakan metode *point method*. Di MAN Cimahi, umumnya guru menggunakan kombinasi antara metode penilaian *point method* dan *rating method*.

Terdapat dua jenis ujian esai, yaitu esai bebas dan esai terbatas. Esai bebas adalah jawaban yang diberikan siswa tidak memiliki batasan dan bertujuan untuk menguji kemampuan analisis, sintesis serta pandangan siswa terhadap suatu masalah sehingga tingkat variansi jawaban siswa akan semakin tinggi [3]. Esai terbatas adalah tes uraian atau esai yang memiliki batasan tertentu, yaitu telah pasti ruang lingkupnya, pasti arah sudut pandang jawabannya, atau pasti indikator-indikator jawabannya [3]. Pada aplikasi yang dibangun, penilaian esai otomatis ditunjukkan untuk menilai esai terbatas karena memiliki variansi kata dan frasa yang rendah.

2.2.2 Konversi Angka ke Huruf

Pada umumnya, skor jawaban siswa yang diberikan oleh guru dalam bentuk angka (0-100), sedangkan kebutuhan input terhadap algoritma K-NN yang akan diolah adalah bentuk kategori huruf (A, B, C, D).

Aplikasi ini akan menerima input skor jawaban ujian esai dalam bentuk angka. Pada saat guru melakukan penilaian jawaban ujian esai secara manual, guru hanya memberikan skor angka pada jawaban dan bukan berupa skor huruf, sehingga harus ada proses konversi skor dari angka ke huruf untuk dapat melanjutkan ke proses penilaian otomatis.

Dalam konversi skor angka menjadi skor huruf, digunakan rumus rentang interval predikat soal pada persamaan 1. Rumus tersebut diperlukan untuk menghitung interval tiap kelas pada konversi dari skor angka ke skor huruf. Ilustrasi penggunaan rumus tersebut dapat dilihat pada Tabel 1. Konversi Skor Huruf ke Skor Angka.

Rentang interval predikat per soal

$$\frac{\text{Skor maksimal} - \text{skor minimal}}{\text{Jumlah kategori huruf}} \quad (1)$$

Karena hasil akhir yang diminta adalah berupa skor angka, maka dilakukan proses konversi skor dari huruf ke angka. Proses ini digunakan untuk menghitung nilai akhir dari ujian esai. Ilustrasi dapat dilihat pada Tabel 2. Ilustrasi Hasil Klasifikasi K-NN.

2.2.3 Pemrosesan Teks (*Text Processing*)

Pemrosesan teks (*text preprocessing*) adalah proses mempersiapkan teks untuk menjadi data untuk dilakukan analisis pada langkah selanjutnya, yaitu tahap *machine learning* [4]. Algoritma *machine learning* yang digunakan dalam pengembangan aplikasi adalah K-NN yang memerlukan representasi jawaban esai dalam bentuk ruang vektor (*vector space*) sehingga dapat dihitung jaraknya.

Dalam pengembangan aplikasi, pemrosesan jawaban esai dilakukan dengan cara menganggap esai sebagai kumpulan kata (*bag of words*) sehingga jawaban esai dapat direpresetasikan dalam bentuk ruang vektor. *Bag of words* merupakan ekstraksi fitur yang tidak memperhatikan urutan dari kata sebuah kalimat [8]. Fitur atau variabel yang dihasilkan diperlukan untuk klasifikasi dengan algoritma K-NN. Langkah pemrosesan teks sebagai berikut:

1. *Case Folding*

Langkah ini adalah proses mengubah semua huruf dalam dokumen menjadi modal kecil dan menghapus semua karakter yang bukan alfabet, yaitu selain a sampai z [4]. Pada aplikasi yang dibangun, selain a sampai z, karakter “-“ yang berada diantara dua kata tidak dihilangkan untuk mengantisipasi kata berulang, yang harus menghasilkan satu kata dasar pada proses *stemming*.

2. *Tokenizing*

Tahap tokenisasi (*tokenizing*) adalah tahap pemotongan teks ke dalam *token* berdasarkan *whitespace* atau tanda baca [9]. Dalam aplikasi, pemotongan teks didasarkan oleh spasi dan karakter selain alfabet a sampai z dan karakter “-“ yang berada diantara dua kata.

3. *Stopword Removal*

Stopword Removal disebut juga *filtering* adalah proses menghilangkan kata-kata *stoplist* yang sering muncul tetapi tidak deskriptif dan tidak terkait

dengan tema tertentu [4]. Daftar *stoplist* yang akan digunakan dalam pengembangan Aplikasi Penilaian Otomatis Ujian Esai Berbahasa Indonesia adalah *stopword* yang diterbitkan oleh Tala pada tahun 2003.

4. Stemming

Stemming adalah proses menemukan akar dari setiap kata sehingga kata-kata itu akan menjadi bentuk dasarnya [4].

Algoritma *stemming* yang digunakan pada aplikasi ini, yaitu algoritma *stemming* Nazief dan Adriani dengan penggunaan *library* PySastrawi versi 1.2, karena akurasi yang diperoleh algoritma Nazief dan Adriani lebih baik daripada algoritma Vega dan Porter, berkisar antara 82,84% sampai dengan 97,93% [10]. Hasil penelitian lain, dari 2.132 kata yang diuji, tingkat kesalahan algoritma *stemming* Nazief-Adriani hanya 5%, sedangkan Porter 21% [10].

5. Perhitungan Term Frequency Unigram dan Bigram

N-gram adalah urutan sekuensial dari n buah kata. Terminologi n-gram dapat merujuk pada dua makna yaitu n-gram yang merupakan kumpulan kata yang berurutan (*word sequence*) dan n-gram yang merupakan model prediktif dengan memanfaatkan probabilitas [3]. Dalam aplikasi ini, karena esai dianggap sebagai kumpulan kata (*bags of word*), n-gram yang dimaksud adalah n-gram yang merupakan kumpulan kata yang berurutan (*word sequence*). N-gram digunakan karena adanya kemungkinan kata kunci dalam suatu esai itu berupa frasa. Nilai n dari n-gram yang digunakan dalam pengembangan aplikasi ini yaitu kombinasi 1-gram (*unigram*) dan 2-gram (*bigram*). Penggunaan kombinasi unigram dan bigram dapat menghasilkan akurasi yang lebih baik [11].

Pada aplikasi yang dibangun, selain proses *unigram* dan *bigram*, dilakukan juga perhitungan jumlah kemunculan *term unigram* dan *bigram* pada setiap jawaban esai siswa.

2.2.4 Term Weighting

Pembobotan *term* merupakan tahap paling penting dalam menentukan efektivitas dari suatu *text classification* karena untuk memberikan bobot yang tepat bagi setiap *term* [3].

Dengan memberikan bobot yang tepat untuk setiap *term*, maka penilaian otomatis ujian esai dapat bekerja lebih baik dalam menentukan apakah suatu

term termasuk kata kunci yang mencirikan suatu kategori nilai atau tidak.

Terdapat dua metode dalam pembobotan *term* yaitu *supervised term weighting* dan *unsupervised term weighting*. Penggunaan metode *supervised term weighting* TF-Chi2 dan TF-RF dapat menghasilkan akurasi yang lebih baik dibandingkan penggunaan metode *unsupervised term weighting* TF-IDF dan TF-RF lebih konsisten dalam semua skenario penelitian [3]. Penelitian Bhamare juga merekomendasikan penggunaan *supervised term weighting* NTF-RF dibandingkan dengan *unsupervised term weighting* [6]. Oleh karena itu, metode yang akan digunakan dalam pengembangan aplikasi ini adalah metode NTF-RF.

Perhitungan *Normalized Term Frequency* (NTF) untuk data latih dapat dilihat pada persamaan 2 sebagai berikut:

$$NTF = \frac{\text{jumlah kemunculan term } i \text{ dalam dokumen } j \text{ (TF)}}{\text{jumlah maksimal kemunculan term } i \text{ dalam semua dokumen}} \quad (2)$$

Untuk menghitung bobot term dengan NTF-RF untuk data yang telah diberikan label nilai oleh guru seperti pada persamaan 3.

$$NTF.RF = NTF \times \log \left(2 + \frac{a}{\max(1,c)} \right) \quad (3)$$

Keterangan:

NTF : nilai NTF.

a : jumlah dokumen, yang mengandung term tersebut, dalam kategori positif.

c : jumlah dokumen, yang mengandung term tersebut, dalam kategori negatif.

Dalam *supervised term weighting*, dokumen yang masuk kategori positif adalah dokumen yang memiliki kategori terpilih, sedangkan sisa dokumen yang tidak masuk kategori terpilih akan masuk ke dokumen negatif.

Esensi NTF-RF data latih adalah memberikan bobot tinggi kepada *term* yang merupakan identitas kategori skor huruf yang diolah.

Selain itu, untuk menghitung bobot *term* terhadap data yang belum memiliki label, yang akan dinilai oleh aplikasi dapat dilihat pada persamaan 4 sebagai berikut:

$$\text{NTF.RF} = \text{TF} \times \text{Max}(\text{RF}) \quad (4)$$

Esensi dari NTF-RF untuk *term* data uji adalah mendekati data uji kepada data latih yang memiliki identitas *term* tersebut melalui pemberian bobot RF maksimal yang terdapat pada data latih kepada data uji yang mengandung *term* tersebut.

2.2.5 Machine Learning dengan K-NN

Machine learning adalah disiplin ilmu yang memberikan komputer kemampuan untuk belajar tanpa secara eksplisit diprogram [12]. Aplikasi Penilaian Otomatis Ujian Esai termasuk ke dalam ranah *machine learning*, di mana aplikasi bertindak sebagai penilai jawaban siswa berdasarkan esai-esai yang telah dilakukan *training* sebelumnya.

K-NN adalah algoritma yang paling sederhana dari semua algoritma *machine learning*. Idenya adalah untuk menyimpan dataset pelatihan dan kemudian melakukan prediksi label terhadap data baru berdasarkan label dari tetangga terdekatnya di dataset pelatihan. Dasar pemikiran di balik metode seperti itu didasarkan pada asumsi bahwa fitur yang digunakan untuk menggambarkan titik domain relevan dengan label dengan titik yang terdekat kemungkinan memiliki label yang sama [13].

Algoritma K-NN ini termasuk pembelajaran berbasis contoh (*instance-based learning*) atau *lazy-learning* dimana fungsinya hanya didekati secara lokal dan semua perhitungan ditangguhkan hingga klasifikasi [14]

Kinerja pengklasifikasi K-NN utamanya ditentukan oleh nilai k serta matrik jarak yang diterapkan. Jika nilai k sangat kecil, perkiraan lokal cenderung menjadi ambigu atau salah label. Nilai k yang besar dengan mudah membuat *over-estimate* sehingga tergolong kedalam kelas lain sehingga lebih berisiko melakukan kesalahan [14].

Karena algoritma klasifikasi K-NN mudah untuk dipahami, dan jika terdapat data baru maka sistem tidak perlu dilatih ulang, maka Aplikasi Penilaian Otomatis Ujian Esai Berbahasa Indonesia yang akan dibangun menggunakan metode K-NN.

2.2.5.1 Cosine Similarity

Cosine similarity adalah matrik yang diterapkan secara luas dan hal yang mendasar untuk menyelesaikan permasalahan dalam bidang *information extraction*, *information retrieval*, dan

text mining [15]. Metode *cosine similarity* memodelkan dokumen teks sebagai vektor *term*. Dengan model ini, kesamaan antara dua dokumen dapat diturunkan dengan menghitung nilai kosinus diantara kedua dokumen tersebut [16]. Nilai *cosine similarity* yang tinggi diantara kedua dokumen menandakan lebih banyak relevansi antara kedua esai tersebut dibandingkan dengan esai lain yang memiliki nilai *cosine similarity* yang lebih rendah. Untuk menghitung *cosine similarity* dapat dilakukan dengan persamaan 5 sebagai berikut:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}, \quad (5)$$

Keterangan:

A: *vector space* esai data latih

B: *vector space* esai data uji

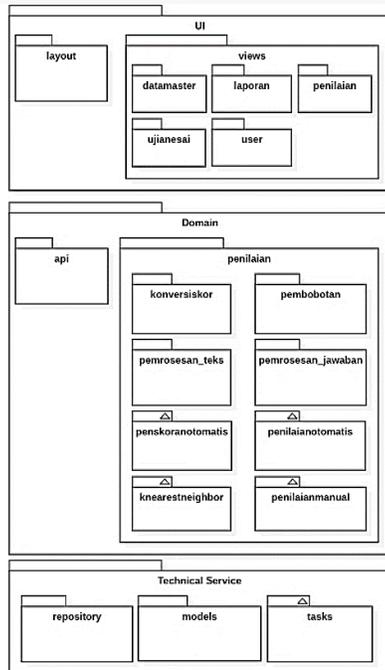
Aplikasi Penilaian Otomatis Ujian Esai Berbahasa Indonesia dikembangkan dengan algoritma K-NN dengan penghitungan kedekatan jawaban esai siswa menggunakan rumus *cosine similarity*. Metode ini digunakan ketika bekerja dengan representasi jawaban esai siswa dalam *vector space model* dengan pendekatan *bag-of-words* yang merupakan hasil dari proses pembobotan *term* terhadap metode *unigram* dan *bigram*, yang terdapat pada tahap *text preprocessing*.

2.2.5.2 Imbalance Dataset

Imbalance dataset adalah kasus khusus untuk masalah klasifikasi di mana distribusi kelas tidak sama di antara kelas-kelas. Hal ini menyebabkan munculnya masalah, yaitu pengambilan keputusan tidak seimbang secara signifikan dalam kinerja sebagian besar algoritma pembelajaran mesin standar [5]. Masalah ini akan menyebabkan sebagian besar algoritma pembelajaran mesin menjadi bias terhadap kelas mayoritas karena dibanjiri oleh kelas mayoritas dan mengabaikan kelas minoritas. Metode yang mudah untuk menyelesaikan masalah tersebut adalah dengan *sampling data* [8].

2.3 Perancangan Aplikasi

Aplikasi dikembangkan dengan model arsitektur pada gambar 1, yang dibuat untuk menyusun struktur aplikasi.



Gambar 1. Arsitektur Software.

Aplikasi dibangun menggunakan framework Flask untuk implementasi domain dan *technical service*, dan dan ReactJS untuk implementasi UI. Bahasa pemrograman yang digunakan adalah Python 3.7 dan Typescript. Dalam proses *stemming* dengan algoritma Nazief-Adriani, digunakan *library* Sastrawi pada modul penilaian. Eksekusi penilaian otomatis dilakukan di *background* aplikasi dengan teknologi Celery dan *message broker* Redis.

Saat ini, aplikasi yang sudah dikembangkan baru fitur mengerjakan ujian esai, menilai ujian esai, dan laporan nilai siswa.

2.4 Alur Kerja Penilaian Otomatis



Gambar 2. Alur Kerja Aplikasi Penilaian Ujian Esai Otomatis

Pada gambar 2, poin 1, 2, 3, 4, 5, 6 merupakan tahapan dari pelatihan atau *training*, sedangkan poin 7, 8, 9, 10, 11 merupakan tahapan dari penilaian esai otomatis.

Tahapan alur kerja sistem penilaian ujian esai otomatis ini dijelaskan secara detail sebagai berikut:

2.4.1. Input Data

Pada tahap pertama, dilakukan input data. Data yang menjadi input untuk aplikasi adalah:

- Keterangan ujian (pertanyaan soal, bobot soal) yang dimasukkan oleh guru.
- Jawaban siswa, diperoleh pada saat siswa selesai melaksanakan ujian esai melalui komputer.

2.4.2 Pelatihan (Training)

Terdapat beberapa proses yang harus dilalui, yaitu proses konversi nilai, *balancing*, dan pemrosesan teks. Setelah jawaban siswa diberi skor angka oleh guru, aplikasi melakukan konversi skor angka tersebut kedalam bentuk label huruf menggunakan rumus 1 dengan ilustrasi pada Tabel 1.

Tabel 1. Konversi Skor Huruf ke Skor Angka

Setelah itu, jika pada kasus khusus seperti pada populasi jawaban (data latih) jumlah label tidak seimbang, yakni perbandingan kelas mayoritas dan minoritas lebih dari 1:1, maka dilakukan *balancing*

dengan metode *undersampling*. Namun, aplikasi yang dibangun saat ini mengasumsikan bahwa tidak akan ada proses *balancing*, dengan asumsi bahwa jawaban ujian esai yang dinilai guru (yang berperan sebagai data latih bagi aplikasi) telah seimbang dalam hal jumlah jawaban yang berlabel A/B/C/D. Setelah itu, dilakukan pemrosesan teks mulai dari langkah *case folding*, *tokenizing*, *stopword removal*, *stemming* dan kombinasi antara *unigram* dan *bigram*. *Output* dari serangkaian proses ini adalah sekumpulan daftar bobot kata (*unigram*) dan frasa (*bigram*) yang disimpan ke dalam database.

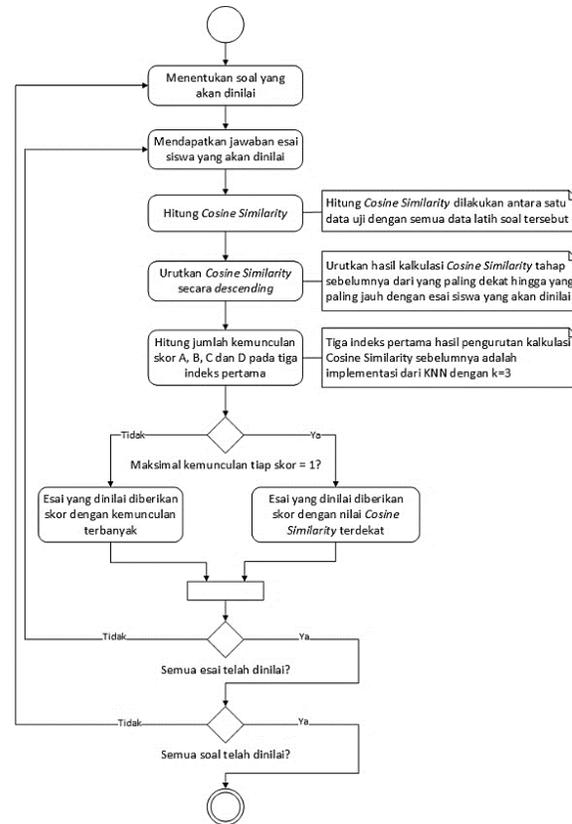
2.4.3 Penilaian Otomatis

Pada saat guru menekan tombol penilaian otomatis, maka jawaban siswa yang belum dinilai akan melalui serangkaian proses seperti pemrosesan teks, pembobotan, dan klasifikasi.

Pemrosesan teks terhadap data uji (jawaban ujian siswa yang belum dinilai) sama seperti yang dilakukan pada data latih (jawaban ujian esai siswa yang telah dinilai oleh guru), yang menghasilkan daftar kata dan frasa (*term*) yang disimpan kedalam database.

Selanjutnya, dilakukan proses pembobotan, yaitu daftar *term* pada data uji dan data latih dilakukan perhitungan bobot dengan rumus yang telah didefinisikan pada persamaan 3 dan persamaan 4. Hasil dari proses ini adalah daftar *term* tersebut memiliki bobot (*value*). Proses pembobotan ini dilakukan untuk menentukan apakah suatu *term* termasuk kata kunci yang mencirikan suatu label kategori skor atau tidak terhadap suatu dokumen. Setelah itu, dilakukan klasifikasi data uji terhadap data latih menggunakan algoritma K-NN dengan nilai $k=3$, dengan alur proses sebagai berikut.

Soal 1			
Skor minimum: 5			
Skor maksimum: 25			
No	Input	Proses	Output
1	Skor perolehan: 13	$(25 - 5) / 4 = 5$	C
2	Skor perolehan: 23	Maka,	A
3	Skor perolehan: 7	A: 25 - 21 B: 20 - 16 C: 15 - 11 D: 10 - 5	D



Gambar 3. Alur Penilaian Otomatis pada Satu Ujian Esai

Proses terakhir adalah menghitung nilai ujian dengan menjumlahkan seluruh skor perolehan dari setiap soal. Karena *output* dari K-NN adalah berupa label skor huruf, diperlukan proses tambahan untuk melakukan konversi skor huruf menjadi skor angka.

Tabel 2. Ilustrasi Hasil Klasifikasi K-NN

Soal 1: Data Uji 1 (Skor Huruf B, Skor Angka 12)			
Nilai Cosine Similarity	Data Latih Soal 1	Skor Huruf	Skor Angka
0,7814	Data Latih 1	B	12
0,6910	Data Latih 2	A	20
0,5712	Data Latih 3	B	13
Soal 2: Data Uji 2 (Skor Huruf A, Skor Angka 19)			
Nilai Cosine Similarity	Data Latih Soal 2	Skor Huruf	Skor Angka
0,614	Data Latih 3	A	19
0,410	Data Latih 7	B	14
0,212	Data Latih 9	C	9
Soal 3: Data Uji 3 (Skor Huruf B, Skor Angka 16)			
Nilai Cosine Similarity	Data Latih Soal 3	Skor Huruf	Skor Angka
0,814	Data Latih 6	A	18
0,310	Data Latih 9	B	16
0,112	Data Latih 1	B	15
Soal 4: Data Uji 4 (Skor Huruf C, Skor Angka 9)			
Nilai Cosine Similarity	Data Latih Soal 4	Skor Huruf	Skor Angka
0,414	Data Latih 5	C	9
0,210	Data Latih 10	B	15
0,012	Data Latih 2	C	8
Soal 5: Data Uji 5 (Skor Huruf A, Skor Angka 20)			
Nilai Cosine Similarity	Data Latih Soal 5	Skor Huruf	Skor Angka
0,614	Data Latih 23	A	20
0,310	Data Latih 6	B	17
0,002	Data Latih 2	D	4

Jika kemunculan maksimal tiap skor adalah satu, maka skor huruf diberikan untuk data uji berdasarkan data latih yang paling dekat. Jika tidak, maka skor huruf yang diberikan untuk data uji berdasarkan voting kemunculan skor huruf pada tiga data latih terdekat.

Selain itu, skor angka diberikan berdasarkan data latih dengan skor huruf yang telah didapatkan. Jika skor huruf yang muncul pada tiga tetangga terdekat lebih dari satu, maka skor angka diberikan berdasarkan data latih dengan skor huruf tersebut dan yang terdekat dengan data uji.

Contohnya, data uji pada ilustrasi soal 1 akan mendapatkan skor huruf B. Data latih dengan label skor huruf B terdekat memiliki skor angka 12, sehingga skor data uji tersebut adalah 12.

Setelah semua data uji pada tiap soal telah diberikan skor angka, dilakukan perhitungan nilai ujian. Pada tabel 2, nilai ujian adalah 76, yang dihitung dengan persamaan 6.

$$\text{nilai ujian} = \sum \text{skor perolehan} \quad (6)$$

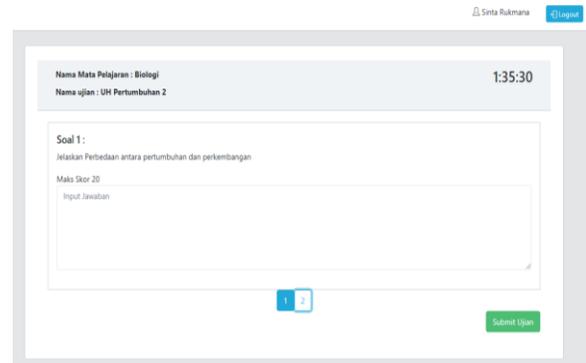
3. HASIL DAN PEMBAHASAN

3.1 Tampilan Aplikasi

Berdasarkan kebutuhan fungsional dari aplikasi yang dikembangkan, sebagai berikut disampaikan pembahasan tentang capaian pengembangan yang telah dilakukan.

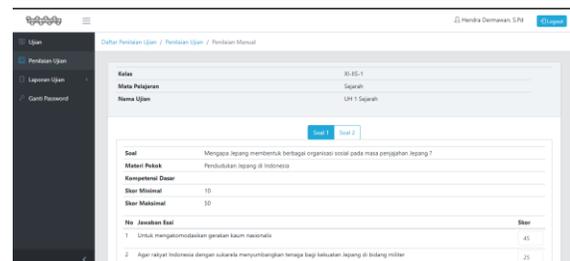
1. Aplikasi mampu menampilkan halaman untuk melakukan ujian esai untuk siswa. Gambar berikut adalah halaman dimana siswa dapat memasukkan jawaban terhadap soal yang tersedia, sebelum siswa dapat melakukan ujian esai, terlebih dahulu guru

diharuskan membuat ujian dan memulai ujian tersebut untuk kelas yang diinginkannya.

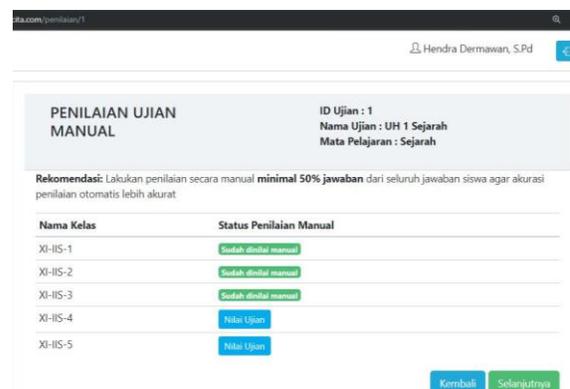


Gambar 4. User Interface Pelaksanaan Ujian Esai

2. Aplikasi mampu menampilkan halaman untuk guru melakukan penilaian terhadap jawaban siswa. Selanjutnya setelah seluruh siswa selesai melakukan ujian esai, guru dapat memberikan skor angka terhadap sebagian jawaban siswa melalui aplikasi.

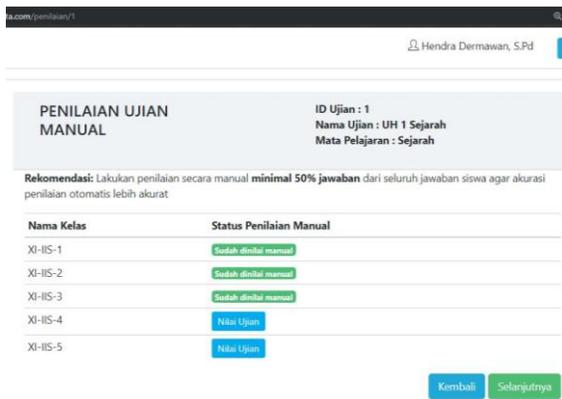


Gambar 5. User Interface Penilaian Manual

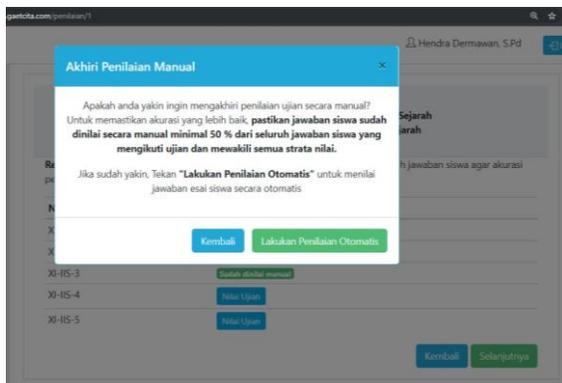


Gambar 6. User Interface Penilaian Manual

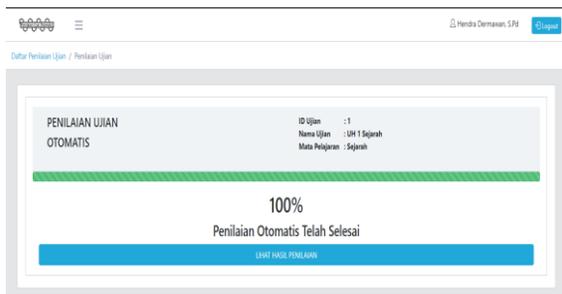
3. Aplikasi mampu menampilkan halaman untuk melakukan penilaian otomatis. Setelah guru selesai menilai sebagian jawaban siswa untuk dijadikan data latih, langkah selanjutnya adalah guru menekan tombol penilaian otomatis untuk jawaban siswa yang belum dinilai dan akan dinilai oleh komputer.



Gambar 7. User Interface Penilaian Manual



Gambar 8. User Interface Akhiri Penilaian Manual



Gambar 9. User Interface Penilaian Otomatis

4. Aplikasi dapat menampilkan laporan ujian siswa. Laporan tersebut menyajikan nilai dalam format angka. Tujuan dari laporan ini adalah menampilkan nilai siswa pada setiap ujian yang telah dikerjakan sehingga guru dapat melihat perkembangan setiap siswa dalam pelaksanaan ujian esai.

NIS	Nama Siswa	Nilai Ujian				Nilai Akhir
		UH 1 Sejarah	UH 2 Sejarah	UH 3 Sejarah	UH 4 Sejarah	
0001412089	Khotijah Guswami	60	85	94	80	79.75
0001607513	Wenny Dewonta	65	56	79	56	64
0002598662	Dani Kumawan	65	77	53	52	61.75
0002715315	Helmy Fauzan Wahyudi	65	91	69	84	77.25
0002874016	Riva Anbarani	60	64	64	82	67.5
0002971144	Siti Fatmahan Zahro	70	81	90	62	75.75
0002971238	Dajni Ariani Ramadhani	60	83	75	89	76.75
0003489303	Putri Melissa	65	76	85	80	76.5
0004340207	Mohammad Iham Ripan	60	50	51	87	62
0004818791	Nuzul Rahmantakim	60	67	71	82	70
0005250574	Muhammad Iran	60	99	85	54	74.5
0005259861	Fira Marlana	75	85	54	80	73.5
0005302810	Alan Fajar Fadilah	85	96	99	73	88.25

Gambar 10. User Interface Hasil Penilaian Ujian Esai

3.2 Pengujian Aplikasi

Pengukuran performa aplikasi digunakan akurasi dan presisi dengan rumus seperti pada persamaan 7. Akurasi atau tingkat pengenalan digunakan untuk mengukur persentase *tuple* yang dilabeli benar [17]. Akurasi yang dimaksud adalah membandingkan hasil penilaian yang dilakukan oleh komputer dengan hasil penilaian yang dilakukan oleh guru. Untuk menghitung akurasi dapat menggunakan persamaan 7 sebagai berikut:

$$\text{akurasi} = \frac{\text{jumlah esai yang diklasifikasikan benar}}{\text{jumlah seluruh esai}} \quad (7)$$

Untuk mengukur konsistensi penilaian, digunakan presisi. Presisi merupakan ukuran kepastian, yaitu berapa persentase *tuple* positif yang dilabeli sebagai positif [17]. Untuk menghitung presisi dapat dilakukan dengan menggunakan persamaan 8 sebagai berikut:

$$\text{presisi (precision)} = \frac{TP}{TP + FP} \quad (8)$$

Keterangan:

TP (*true positive*): jumlah tuple positif yang dilabeli benar oleh klasifikasi

FN (*false positive*): jumlah tuple negatif yang salah dilabeli oleh model klasifikasi

Pengukuran performa penilaian otomatis dengan output angka, digunakan *Root Mean Squared Error* (RMSE). RMSE digunakan untuk mengukur tingkat kesalahan, dengan rumus seperti pada persamaan 9 [18].

$$\text{RMSE Errors} = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}} \quad (9)$$

Keterangan:

\hat{y}_i : skor hasil penilaian otomatis

y_i : skor hasilpenilaian manual

1. Data Pengujian

Dari populasi jawaban yang diperoleh dari hasil ujian online, diambil sebagian jawaban siswa berdasarkan kategori untuk dinilai manual oleh guru. Aplikasi mengasumsikan bahwa guru dapat memilih jawaban siswa berdasarkan kategori jawaban yang bernilai bagus, menengah dan bawah sesuai dengan pengetahuan yang dimiliki oleh guru. Pengujian aplikasi ini menggunakan data latihan dengan komposisi jumlah kategori seperti yang dituliskan pada Tabel 3 berikut:

Tabel 3. Kategori Data Latihan

Kategori Skor Huruf	Jumlah Esai Soal 1	Jumlah Esai Soal 2
A	18	12
B	26	26
C	22	28

Kemudian sejumlah data yang telah dituliskan pada tabel 3 akan dibandingkan dengan sejumlah 82 data uji untuk dinilai otomatis oleh aplikasi.

a. Performa Penilaian Otomatis Klasifikasi Skor Huruf

Hasil pengujian pada tabel 4 menunjukkan bahwa soal nomor satu memiliki akurasi dan presisi yang lebih tinggi daripada soal nomor dua.

Tabel 4. Performa Penilaian Otomatis Skor Huruf

Soal	Akurasi	Presisi
Soal 1	48,39%	48,39%
Soal 2	83,87%	66,67%

b. Performa Penilaian Otomatis Skor Angka

Hasil pengujian menunjukkan terdapat maksimal *error* skor angka yang tinggi, yaitu dua puluh. Karena guru MAN Cimahi membatasi maksimal *error* skor adalah sepuluh, aplikasi ini belum layak untuk digunakan.

Tabel 5. Performa Penilaian Otomatis Skor Angka

Soal	RMSE	Maksimal Error
Soal 1	1,3	20
Soal 2	1,2	20

c. Performa Waktu Penilaian Otomatis

Kebanyakan guru MAN Cimahi menilai esai dalam waktu satu menit per jawaban dan rata-rata tiga menit

per jawaban, sedangkan penilaian otomatis mampu memeriksa dua soal esai dalam waktu satu menit untuk 31 siswa (62 jawaban) data uji dengan 66 siswa (132 jawaban) data latihan menggunakan spesifikasi server pada tabel 7 dan akses satu *user*, sehingga sampai saat ini dapat disimpulkan bahwa aplikasi dapat menyelesaikan masalah kecepatan penilaian esai. Dengan spesifikasi yang lebih tinggi, maka aplikasi dapat menilai otomatis lebih cepat.

Tabel 6. Perangkat yang digunakan dalam pengujian

Operating System	Ubuntu 18.04 x86_64
Virtual Memory	4 GB
CPU	4 Core 2 GB
Storage Space	40 GB SSD

4. KESIMPULAN

Berdasarkan pengembangan aplikasi yang telah dikerjakan, saat ini telah berhasil mengimplementasikan sejumlah fitur yaitu pelaksanaan ujian online, penilaian otomatis, dan laporan nilai ujian.

Fitur yang belum selesai diimplementasi, yaitu implementasi *balancing* dan export laporan penilaian ujian siswa ke format Microsoft Excel.

5. UCAPAN TERIMA KASIH

Dalam rangka menyelesaikan penyusunan *paper* ini penulis telah banyak mendapatkan bantuan serta bimbingan dari berbagai pihak, baik secara langsung maupun tidak langsung. Penulis ingin menyampaikan ucapan terima kasih kepada semua pihak yang turut membantu, khususnya:

1. Allah SWT, karena berkat dan rahmat-Nya kami mendapatkan kemudahan serta kelancaran dalam menyelesaikan *paper* ini.
2. Ibu Ani Rahmani, S.Si., M.T. selaku pembimbing pertama.
3. Bapak Jonner Hutahaean, BSET., M.Info.Sys. selaku pembimbing kedua.
4. Keluarga serta rekan-rekan yang selalu mendukung serta memberikan doan dan dukungan.

5. DAFTAR PUSTAKA

- [1] Sulisty, Meiyanto Eko, Ristu Saptono, Adam Asshidiq, *Penilaian Ujian Bertipe Essay Menggunakan Metode Text Similarity*, vol. 12 No. 02, 2015.
- [2] Arikunto, Prof. Dr. Suharsimi, *Dasar-Dasar Evaluasi Pendidikan*, PT Bumi Aksara, 2018.

- [3] Muttaqin, Harut Digdaya, "Analisis Akurasi Metode Term Weighting untuk Penilaian Otomatis Esai Berbahasa Indonesia dengan K-Nearest Neighbor," Politeknik Negeri Bandung, Bandung, 2017.
- [4] Slamet, A. R. Atmadja, D. S. Maylawati, R. S. Lestari, W. Darmalaksana, dan M. A. Ramdhani., "Automated Text Summarization for Indonesian Article Using Vector Space Model," *Conference Series: Materials Science and Engineering.*, vol. Vol.288, 2018.
- [5] Sonak, Apurva dan R.A Patankar, "A Survey on Methods to Handle Imbalance Dataset,," *International Journal of Computer Science and Mobile Computing.*, vol. 4, no. 11, pp. 338-343, 2015.
- [6] Bhamare, Chandrashekar P dan Dinesh D. Patil, "A Novel Text Classifier Using Unsupervised Term Weighting Method," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 4, no. 6, 2015.
- [7] Y. Sunarya, Pedoman Evaluasi Proses dan Hasil Belajar., Bandung: Universitas Pendidikan Indonesia. , 2011.
- [8] Prasanti, Annisya Aprilia, M. Ali Fauzi, dan M. Tanzil Furqon, "Klasifikasi Teks Pengaduan Pada Sambat Online Menggunakan Metode N-Gram dan Neighbor Weighted K-Nearest Neighbor (NW-KNN)," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. Vol 2 No 2., 2018.
- [9] T. Jo, " Text Mining: Concepts, Implementation, and Big Data Challenge," *Springer International Publishing AG*, vol. 45, 2018.
- [10] Sardjono, Mochamad Wisuda, Margi Cahyanti, dan Maulana Mujahidin. , "Pendeteksi Kesamaan Kata untuk Judul Penulisan Berbahasa Indonesia Menggunakan Algoritma Stemming Nazief-Adriani,," vol. 22, no. 2, p. Sebatik, 2018.
- [11] Tripathy, Abinash, Ankit Agrawal, dan Santanu Kumar Rath , "Classification of Sentiment Reviews using N-gram Machine Learning Approach, Expert Systems With Applications,," *Elsevier*, vol. 57, no. 12, p. 117, 2016.
- [12] Samuel, Arthur, "Some Studies in Machine Learning Using the Game of Checkers.," (3) 3 1959. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.368.2254>. [Accessed Rabu Juni 2016].
- [13] Shwartz, Shai Shalev dan David, Shai Ben., in *Understanding Machine Learning From Theory to Algorithms*, New York, Cambridge University Press., 2014.
- [14] Imandoust, Sadeh Bafandeh dan Mohammad Bolandraftar, "Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events: Theoretical Background,," *International Journal of Engineering Research and Applications.*, vol. 3, no. 3, pp. 605-610, 2013.
- [15] Sohangir, Sahar dan Dingding Wang., "Improved Sqrt-Cosine Similarity Measurement," *Journal of Big Data*, vol. 4, p. 25, 2017.
- [16] Rahutomo, Faisal, Masayoshi Aritsugi, dan Teruaki Kitasuka., *Semantic Cosine Similarity.*, Seoul: The 7th International Student Conference on Advanced Science and Technology ICAST , 2012.
- [17] Suyanto, Machine Learning, Teknik Dasar dan Tingkat Lanjut, Bandung: Informatika, 2018.
- [18] S. Holmes, Stanford University, 11 Desember 2000. [Online]. Available: <http://statweb.stanford.edu/~susan/courses/s60/split/node60.html>. [Accessed 20 Juni 2019].