

EVALUASI ESAI OTOMATIS DENGAN ALGORITMA NAZIEF & ADRIANI DAN WINNOWING

Arham Rahim
Universitas Madako Tolitoli
Email: arham.rahim@umada.ac.id

Abstrak

Pemanfaatan teknologi informasi diharapkan dapat memudahkan aktifitas sehari-hari tidak terkecuali proses evaluasi hasil belajar mahasiswa oleh dosen. Pemeriksaan hasil ujian esai secara manual tentu jauh dari kata efektif dan efisien sehingga penelitian ini memanfaatkan fingerprint yang diperoleh dari nilai hash kumpulan teks menggunakan algoritma *Winnowing* untuk selanjutnya dihitung nilai kesamaannya menggunakan *Jaccard's Similarity Coeficient*. Sebelum proses perhitungan, jawaban dalam bentuk teks bahasa Indonesai melalui pre-prosesing dan menggunakan algoritma Nazief & Andriani sebagai *stemmer* sehingga dapat memperoleh hasil evaluasi 30 jawaban siswa yang dibandingkan dengan kunci jawaban dalam waktu 1,62 detik dengan nilai rata-rata kesamaan 81,20%.

Kata Kunci: Nazief Andriani, winnowing, Jaccard's Similarity Coeficient

Abstract

The use of information technology is expected to facilitate daily activities including the evaluation of student learning outcomes by lecturers. Examination of essay exam results manually is certainly far from effective and efficient words so this study utilizes fingerprints obtained from the hash value of a set of texts using the Winnowing algorithm to then calculate the similarity value using Jaccard's Similarity Coefficient. Before the calculation process, answers in the form of Indonesian text through pre-processing and using the Nazief & Andriani algorithm as stemmer so that they can get the results of the evaluation of 30 student answers compared with the answer key in 1.62 seconds with an average value of similarity 81.20%.

Keywords: Nazief Andriani, winnowing, Jaccard's Similarity Coeficient.

PENDAHULUAN

Salah satu indikator pada instrumen penilaian kompetensi pengajar adalah melakukan evaluasi hasil belajar (Dra.Rosinta, Drs.Asrul, 2014) sehingga hal tersebut perlu untuk dilakukan. Salah satu cara melakukan evaluasi yaitu dengan memberi ujian tulis dalam bentuk soal essay. Jika dibandingkan dengan ujian dalam bentuk pilihan ganda, ujian esai lebih sulit untuk dibandingkan dengan kunci jawaban (Lemantara et al., 2018) tetapi masi banyak yang menggunakan ujian dalam bentuk esai. Secara manual evaluasi ini dilakukan dengan cara memberi sejumlah soal kepada peserta didik kemudian dijawab pada lembar jawaban. Selanjutnya pengajar harus memeriksa jawaban tersebut secara manual. Penilaian hasil ujian yang dilakukan secara manual memiliki kelemahan, yaitu pengajar membutuhkan waktu lama untuk memeriksa jawaban esai. Semakin banyak jumlah ujian dan semakin banyak siswa yang mengikuti ujian, semakin banyak pula ujian yang harus diperiksa oleh guru. Hal ini tentu saja menyebabkan kualitas penilaian menurun dan kualitas penilaian tidak lagi objektif. Studi dalam Ilmu Sosial telah mengungkapkan bahwa ketika orang mengevaluasi orang lain, evaluasi mereka sering

mencerminkan bias mereka. Akibatnya, bias penilai dapat memperkenalkan faktor yang sangat subyektif yang membuat evaluasi mereka tidak akurat (Amorim et al., 2018). Efektif dan efisien seringkali menjadi alasan utama pengembangan teknologi informasi untuk membantu kegiatan sehari-hari, tidak terkecuali proses evaluasi hasil belajar di dunia pendidikan. Untuk mendukung kinerja sistem dalam menilai hasil ujian, diperlukan metode yang dapat mengukur kesamaan teks dalam evaluasi ujian esai online, sehingga pekerjaan pengajar akan lebih efektif dan mempermudah untuk melakukan penilaian yang lebih akurat.

Pada penelitian Fachrurrozi, dkk. tentang peringkasan teks berbasis istilah untuk Bahasa Indonesia melakukan proses *stemming* menggunakan algoritma Nazief & Adriani. Dari hasil penelitian tersebut menghasilkan ringkasan yang dapat dipahami oleh responden dengan nilai rata-rata 83,8 (Fachrurrozi et al., 2013). Selanjutnya penelitian tentang klasifikasi berita Indonesia dengan menggunakan metode naïve bayes dan algoritma Nazief & Adriani dari hasil pengujian 200 sampel yang dibagi menjadi 120 data uji dan 80 data latih, menghasilkan nilai akurasi 94% (Septian et al., 2018). Penelitian yang membandingkan stemmer Porter dan Nazief & Adriani pada kinerja algoritma winnowing untuk mengukur plagiarisme membuktikan algoritma Nazief & Adriani lebih baik dengan hasil uji similaritas 70,5% dibandingkan stemmer Porter dengan hasil uji similaritas 65,7% (Rahmatulloh et al., 2019). Pemrosesan teks berita digital Indonesia juga membandingkan stemmer Porter dan Nazief & Adriani yang telah disempunakan oleh Jelita Asia yang membuktikan algoritma Nazief & Adriani dan Jelita Asia memberikan kinerja yang lebih baik dengan hasil evaluasi 125 teks berita digital menghasilkan nilai ~ 100% untuk Precision, 90% untuk Recall dan 94% untuk F-Measure (Prihatini et al., 2017).

Pada penelitian Ratna dkk. Menyimpulkan algoritma winnowing dapat digunakan untuk menilai jawaban esai bahasa jepang secara otomatis karena nilai akurasi rata-rata diatas 80% (Agung Putri Ratna et al., 2018). Ulinuha dkk. juga mengimplementasikan algoritma winnowing untuk mendeteksi plagiarisme dari nilai persentase kesamaan antara dokumen asli dengan dokumen-dokumen uji dan menghasilkan akurasi hingga 91% (Ulinuha et al., 2019). Implementasi algoritma winnowing untuk mendeteksi plagiarisme juga digunakan oleh Arifin dkk. dan menyimpulkan bahwa deteksi plagiarisme bahasa Indonesia dengan pemrosesan paralel dapat menghemat waktu hingga 74,66% dari 208.7894 menit menjadi 52,9052 menit (Arifin et al., 2018).

METODE PENELITIAN

Teks adalah inputan yang akan diproses oleh algoritma winnowing untuk menghasilkan output berupa nilai-nilai dari hash atau fingerprint yang selanjutnya akan digunakan untuk membandingkan teks dengan teks lainnya. Sebelum dilakukan pemrosesan, teks harus melalui tahap pre-processing yang bertujuan untuk mendapatkan hasilnya lebih akurat (Agung Putri Ratna et al., 2018).

1) *Pre-processing*

Preprocessing Text adalah fase penting sebelum menerapkan algoritma apa pun (Kalra & Aggarwal, 2018). Proses ini dilakukan untuk diperlukan untuk membersihkan, menghilangkan, mengubah sumber data, baik itu berupa karakter non alphabet maupun kata-kata yang tidak diperlukan. Berikut ini adalah proses *preprocessing text* yang terdiri

dari *case folding*, *tokenizing*, *filtering*, dan *stemming* (Slamet et al., 2018) (Watequlis Syaifudin et al., 2018).

a) *Case Folding*

Proses mengubah semua teks menjadi karakter dengan huruf kecil dan membuang semua karakter selain a-z. Jika terdapat tanda baca, bilangan numerik dan simbol akan dihilangkan.

b) *Tokenizing*

Proses untuk mengubah bentuk kalimat menjadi kata-kata tunggal. Pemotongan kalimat berdasarkan delimiter yang menyusunnya, yaitu spasi (" "). proses ini bertujuan agar nantinya dapat melakukan proses *stemming*.

c) *Filtering*

Proses filtering dilakukan pembuangan *stopword*. *Stopword* adalah kata-kata yang tidak memiliki makna atau kata yang kurang berarti dan sering muncul dalam kumpulan kata-kata. Proses pembuangan kata-kata yang tidak penting dengan mengecek pada kamus *stopword*.

d) *Stemming*

Proses yang bertujuan untuk mengembalikan suatu kata bentuk akarnya (root word) dengan aturan-aturan tertentu, sehingga setiap kata memiliki representasi yang sama. *Stemming* dalam penelitian ini menggunakan algoritma Nazief & Andriani.

2) *Stemmer Nazief & Andriani*

Algoritma *stemming* Nazief dan Andriani dikembangkan berdasarkan morfologi bahasa Indonesia dengan mengelompokkan imbuhan menjadi awalan (*prefix*), akhiran (*suffix*) serta gabungan awalan dan akhiran (*confixes*). Algoritma Nazief dan Andriani yang dibuat oleh Bobby Nazief dan Andriani ini memiliki tahap – tahap sebagai berikut:

a) Cari kata yang akan disistem dalam kamus. Jika ditemukan maka diasumsikan bahwa kata tersebut adalah root word. Maka algoritma berhenti.

b) *Inflection Suffixes* ("-lah", "-kah", "-ku", "-mu", atau "-nya") dibuang. Jika berupa particles ("-lah", "-kah" atau "-pun") maka langkah ini diulangi lagi untuk menghapus Possessive Pronoun ("ku", "-mu" atau "-nya"), jika ada.

c) Hapus *Derivation Suffixes* ("-i", "-an" atau "-kan"). Jika kata ditemukan di kamus, maka algoritma berhenti. Jika tidak maka ke langkah c1.

1) Jika "-an" telah dihapus dan huruf terakhir dari kata tersebut adalah "-k", maka "-k" juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah c2.

2) Akhiran yang dihapus ("-i", "-an", atau "-kan") dikembalikan, lanjut ke d

d) Hapus Derivation Prefix. Jika pada langkah 3 ada suffixes yang dihapus maka pergi ke langkah d1, jika tidak pergi ke langkah d2.

1) Periksa tabel kombinasi awalan-akhiran yang tidak diizinkan pada tabel 2.5. jika ditemukan maka algoritma berhenti, jika tidak pergi ke langkah d2.

2) For $i = 1$ to 3, temukan tipe awalan kemudian hapus awalan. Jika root word belum juga di temukan lakukan langkah e, jika sudah maka algoritma berhenti. Catatan: jika awalan kedua sama dengan awalan pertama maka algoritma berhenti.

e) Melakukan *Recording*

- f) Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai *root word*. Proses selesai.

Tipe awalan ditentukan melalui langkah-langkah berikut:

- a) Jika awalannya adalah “-di”, “-ke-” atau “-se-” maka tipe awalannya secara berturut-turut adalah “di-”, “-ke” atau “-se”.
- b) Jika awalannya adalah “-te”, “-me”, “-be” atau “-pe” maka dibutuhkan proses tambahan untuk menentukan tipe awalannya. Jika dua karakter pertama bukan “-di”, “-ke”, “-se”, “-te”, “-be-”, “-me-” atau “-pe-” maka berhenti.
- c) Jika tipe awalan adalah “none” maka berhenti. Jika tipe awalan adalah bukan “none” maka awalan dapat dilihat pada tabel 1.

Tabel 1. Kombinasi awalan akhiran yang tidak diizinkan

Awalan	Akhiran yang tidak diizinkan
be-	-i
di-	-an
ke-	-i, -kan
me-	-an
se-	-i, -kan

Tabel 2. Awalan Berdasarkan Tipe Awalannya

Awalan	Akhiran yang tidak diizinkan
di-	-di
ke-	-ke
se-	Se-
te-	-te
ter-	-ter
Ter-luluh	-ter

Untuk mengatasi keterbatasan pada algoritma diatas, maka ditambahkan aturan-aturan dibawah ini.

- a) Aturan untuk reduplikasi
 - 1) Jika kedua kata yang dihubungkan oleh kata penghubung adalah kata yang sama maka root word adalah bentuk tunggalnya, contoh: “buku-buku” root word-nya adalah “buku”.
 - 2) Kata lain, misalnya “bolak-balik”, “berbalas-balasan, dan”seolah-olah”. Untuk mendapatkan root word-nya, kedua kata diartikan secara terpisah. Jika keduanya memiliki root word yang sama maka diubah menjadi bentuk tunggal, contoh: kata “berbalas-balasan”, “berbalas” dan “balasan” memiliki root word yang sama yaitu “balas”, maka root word “berbalas-balasan” adalah “balas”. Sebaliknya, pada kata “bolak-balik”, “bolak” dan “balik” memiliki root word yang berbeda, maka root word-nya adalah “bolak-balik”.
- b) Tambahan bentuk awalan dan akhiran serta aturannya.
 - 1) Untuk tipe awalan “mem-“, kata yang diawali dengan awalan “memp-” memiliki tipe awalan “mem-”.
 - 2) Tipe awalan “meng-“, kata yang diawali dengan awalan “mengk-” memiliki tipe awalan “meng-”.

3) *Algoritma Winnowing*

Winnowing adalah algoritma yang digunakan untuk memproses dokumen sidik jari (Sutoyo et al., 2017). *Winnowing* menggunakan kinsep window yang dapat ditentukan nilai k-gram, hash dan windownya (Mišić et al., 2018) (Chapuis & Garbinato, 2018). Proses k-gram dilakukan untuk mengelompokkan text menjadi kumpulan string sebelum menghitung nilai hash dari setiap k-gram. Fungsi rolling hash digunakan untuk menemukan nilai hash. Lalu, nilai hash dibentuk dalam windows. Selain itu, memilih nilai hash minimum dengan mengutamakan nilai paling kanan jika ada lebih dari satu hash dengan nilai minimum. Selanjutnya nilai hash yang dipilih disimpan menjadi sidik jari adokumen. Sidik jari ini akan digunakan sebagai dasar perbandingan kesamaan antara teks yang telah diproses menggunakan algoritma Jaccard's Similarity Coefficient.

a) *Rolling hash*

Rolling hash adalah metode yang digunakan untuk mencari nilai hash dari gram yang telah dibentuk. Nilai hash adalah nilai numerik yang terbentuk dari kode ASCII (Yudhana et al., 2018) Formu rolling hash sebagai berikut:

$$c_1 * b^{(k-1)} + c_2 * b^{(k-2)} + \dots + c_{k-2} * b + c_k \quad (1)$$

Keterangan

c: Nilai ASCII Karakter

b: Basis Bilangan Prima

k: Nilai k-gram / banyaknya karakter pada gram

b) *Jaccard's Similarity Coefficient*

Jaccard Similarity adalah algoritma yang digunakan untuk membandingkan antara dua dokumen berdasarkan kesamaan string dari dokumen (Rinartha & Suryasa, 2017) (Diana & Hanana Ulfa, 2019). Untuk menghitung presentase kesamaan dengan menggunakan dengan rumus:

$$D(A, B) = \frac{|A \cap B|}{|A \cup B|} \times 100 \quad (2)$$

Keterangan:

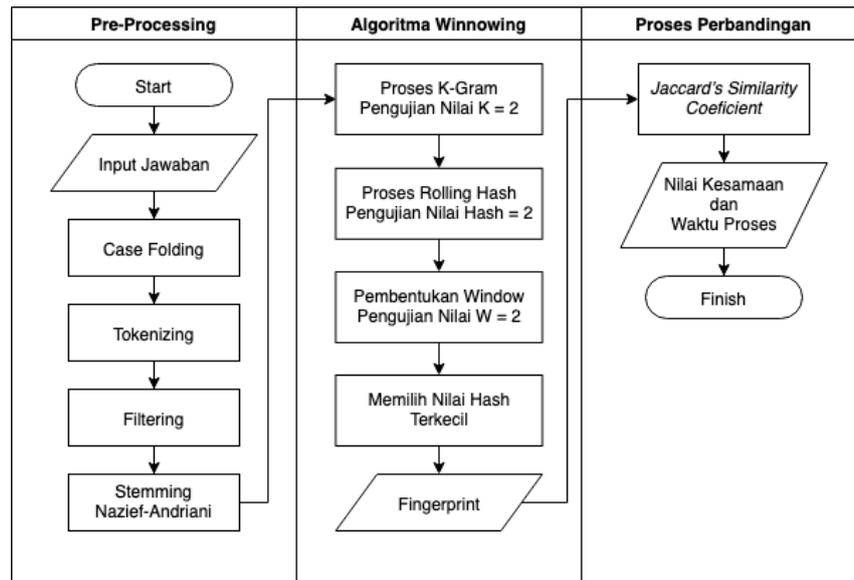
D(A, B): Nilai dari similarity

A: Dokumen 1

B: Dokumen 2

HASIL DAN PEMBAHASAN

Pada penelitian ini, Kami mengimplementasikan algoritma *winnowing* untuk mengevaluasi hasil ujian mahasiswa dalam bentuk soal esai. Input dari proses ini adalah teks yang merupakan kunci jawaban yang disediakan oleh dosen dan jawaban siswa dengan menggunakan bahasa Indonesia. Agar input diproses dengan baik, harus ada beberapa metode untuk menyiapkan input tersebut. Desain secara umum seperti pada Gambar 1.



Gambar 2. Flowchart proses pengolahan data

Sebelum menghasilkan nilai kesamaan dan waktu proses, kunci jawaban dan jawaban mahasiswa terlebih dahulu melalui pre-prosesing dan selanjutnya diproses menggunakan algoritma *winnowing* untuk menghasilkan *fingerprint*. Proses akhir menghitung nilai kesamaan kedua data dengan Jaccard's Similarity.

1) **Data Responden**

Dalam penelitian ini, kami menggunakan jawaban mahasiswa dan soal serta kunci jawaban pada matakuliah Penganter Ilmu Ekonomi di program studi Sosial Ekonomi Perikanan Fakultas Ilmu Kelautan dan Perikanan Universitas Hasanuddin yang terbagi dalam 7 topik seperti yang terlihat pada tabel 3.

Tabel 3. Data Jenis Soal

No	Topik Soal	Label	Jumlah Jawaban Siswa
1	Sistem Ekonomi Kapitalis	Topik 1	4
2	Sistem Ekonomi Sosialis	Topik 2	4
3	Sistem Ekonomi Komunis	Topik 3	3
4	Sistem Ekonomi Kerakyatan / Pancasila	Topik 4	7
5	Sistem Ekonomi Campuran	Topik 5	6
6	Sistem Ekonomi Syariah	Topik 6	4
7	Sistem Ekonomi Fasisme	Topik 7	2
Total Jawaban Siswa			30

Dari 7 topik soal esai, masing-masing mahasiswa menjawab 1 soal secara acak dan menghasilkan 30 data jawaban mahasiswa dan 7 data kunci jawaban.

2) **Pre-prosesing**

Seperti pada gambar 2 flowchart proses pre-prosesing dibagi menjadi empat proses yaitu *case folding*, *tikenizing*, *filtering* dan *stemming* dengan algoritma Nazief-Andriani.

Proses ini dilakukan untuk membersihkan, menghilangkan dan mengubah semua jawaban sebelum diproses menggunakan algoritma *winning*.

a) *Case Folding*

Proses ini bertujuan untuk mengubah semua teks menjadi karakter dengan huruf kecil dan membuang semua karakter selain a-z. Jika terdapat tanda baca, bilangan numerik dan simbol akan dihilangkan. Contoh case folding dapat dilihat pada table 4.

Tabel 4. Proses Case Folding

Text	Hasil Case Folding
Berdasarkan sistem ekonomi campuran masuk kedalam sistem konvensional karena sistem ekonomi campuran memiliki prinsip lakukan pekerjaan sesuai keinginan dalam batas tertentu. Yang memiliki kebebasan tetapi terpimpin. Sumber daya dan sebagainya di bagi dia separuh dilakukan masyarakat dan separuhnya dilakukan oleh pemerintah.	berdasarkan sistem ekonomi campuran masuk kedalam sistem konvensional karena sistem ekonomi campuran memiliki prinsip lakukan pekerjaan sesuai keinginan dalam batas tertentu. yang memiliki kebebasan tetapi terpimpin. sumber daya dan sebagainya di bagi dia separuh dilakukan masyarakat dan separuhnya dilakukan oleh pemerintah.

b) *Tokenizing*

Dari hasil case folding, selanjutnya text diubah menjadi kata-kata tunggal. Pemotongan kalimat berdasarkan delimeter yang menyusunnya, yaitu spasi(" "). Contoh hasil tokenizing kata terlihat pada tabel 5.

Tabel 5. Proses Tokenizing

Hasil Case Folding	Hasil Tokenizing
berdasarkan sistem ekonomi campuran masuk kedalam sistem konvensional karena sistem ekonomi campuran memiliki prinsip lakukan pekerjaan sesuai keinginan dalam batas tertentu. yang memiliki kebebasan tetapi terpimpin. sumber daya dan sebagainya di bagi dia separuh dilakukan masyarakat dan separuhnya dilakukan oleh pemerintah.	- berdasarkan - yang - sistem - memiliki - ekonomi - kebebasan - campuran - tetapi - masuk - terpimpin - kedalam - sumber - sistem - daya - konvensional - dan - karena - sebagainya - sistem - di - ekonomi - bagi - campuran - dia - memiliki - separuh - prinsip - dilakukan - lakukan - masyarakat - pekerjaan - dan - sesuai - separuhnya - keinginan - dilakukan - dalam - oleh

	- batas	- pemerintah
	- tertentu	

c) *Filtering*

Setelah melalui proses tokenizing proses selanjutnya dalah mengubah teks dengan pembuangan stopwords. Contoh hasil filtering kata terlihat pada tabel 6.

Tabel 6. Proses Filtering

Hasil Tokenizing		Hasil Filtering	
- berdasarkan	- yang	- berdasarkan	- terpimpin
- sistem	- memiliki	- sistem	- sumber
- ekonomi	- kebebasan	- ekonomi	- daya
- campuran	- tetapi	- campuran	- sebagainya
- masuk	- terpimpin	- masuk	- bagi
- kedalam	- sumber	- kedalam	- separuh
- sistem	- daya	- sistem	- dilakukan
- konvensional	- dan	- konvensional	- masyarakat
- karena	- sebagainya	- sistem	- separuhnya
- sistem	- di	- ekonomi	- dilakukan
- ekonomi	- bagi	- campuran	- pemerintah
- campuran	- dia	- memiliki	
- memiliki	- separuh	- prinsip	
- prinsip	- dilakukan	- lakukan	
- lakukan	- masyarakat	- pekerjaan	
- pekerjaan	- dan	- sesuai	
- sesuai	- separuhnya	- keinginan	
- keinginan	- dilakukan	- batas	
- dalam	- oleh	- tertentu	
- batas	- pemerintah	- memiliki	
- tertentu		- kebebasan	

d) *Stemming*

Setelah melalui proses filtering, selanjutnya setiap kata dikembalikan dalam bentuk akarnya dengan aturan-aturan pada stemmer Nazief & Andriani. Contoh hasil *stemming* kata terlihat pada tabel 7.

Tabel 7. Proses Stemming

Hasil Filtering		Hasil Stemming	
- berdasarkan	- keinginan	- dasar	- ingin
- sistem	- batas	- sistem	- batas
- ekonomi	- tertentu	- ekonomi	- tentu
- campuran	- memiliki	- campur	- milik
- masuk	- kebebasan	- masuk	- bebas
- kedalam	- terpimpin	- dalam	- pimpin
- sistem	- sumber	- sistem	- sumber
- konvensional	- daya	- konvensional	- daya

- sistem	- sebagainya	- sistem	- bagi
- ekonomi	- bagi	- ekonomi	- bagi
- campuran	- separuh	- campur	- paruh
- memiliki	- dilakukan	- milik	- laku
- prinsip	- masyarakat	- prinsip	- masyarakat
- lakukan	- separuhnya	- laku	- paruh
- pekerjaan	- dilakukan	- kerja	- laku
- sesuai	- pemerintah	- sesuai	- perintah

3) Proses *Winnowing*

Seperti pada gambar 2, setelah membersihkan, menghilangkan dan mengubah text pada pre-prosesing data jawaban siap untuk menggunakan algoritma *winnowing* untuk mencari nilai hash dari text. Algoritma *winnowing* diawali dengan proses *K-Gram* selanjutnya mencari nilai hash dengan rolling hash dan dibentuk dalam window untuk selanjutnya dihitung nilai similaritynya dari hash terkecil.

a) *K-Gram*

Proses ini dilakukan untuk mengelompokkan text menjadi kumpulan string dengan nilai $k = 2$. salah satu contoh jawaban siswa : ['be', 'er', 'rd', 'da', 'as', 'sa', 'ar', 'rk', 'ka', 'an', 'ns', 'si', 'is', 'st', 'te', 'em', 'me', 'ek', 'ko', 'on', 'no', 'om', 'mi', 'ic', 'ca', 'am', ..., 'ka', 'ay', 'ya'] dan kunci jawaban : ['si', 'is', 'st', 'te', 'em', 'me', 'ek', 'ko', 'on', 'no', 'om', 'mi', 'ic', 'ca', 'am', 'mp', 'pu', 'ur', 'rr', 'ru', 'up', 'pa', 'ag', 'ga', 'ab', 'bu', ... 'nt', 'ta', 'ah'].

b) *Rolling Hash*

Untuk mencari nilai hash menggunakan formula (1) dengan nilai basis hash = 2. Contoh pada string “be” sebagai berikut:

$$\begin{aligned} be &= (98 * 2^{((2-1))}) + (101 * 2^{((2-2))}) \\ be &= 196 + 101 \\ be &= 297 \end{aligned}$$

Dari proses tersebut didapatkan nilai hash dari setiap gram dengan contoh salah satu jawaban siswa: [297, 316, 328, 297, 309, 327, 308, 335, 311, 304, 335, 335, 325, 346, 333, 311, ... 311, 315, 339] dan kunci jawaban: [335, 325, 346, 333, 311, 319, 309, 325, 332, 331, 331, 323, 309, 295, ... 336, 329, 298].

c) *Window*

Proses pembentukan window sama seperti proses *K-Gram* dari nilai-nilai hash yang dihasilkan dengan besar window = 2 dengan contoh salah satu jawaban siswa : [[297, 316], [316, 328], [328, 297], [297, 309], [309, 327], [327, 308], [308, 335], [335, 311], [311, 304], [304, 335], [335, 335], [335, 325], [325, 346], [346, 333], ... [313, 311], [311, 315], [315, 339]] dan kunci jawaban : [[335, 325], [325, 346], [346, 333], [333, 311], [311, 319], [319, 309], [309, 325], [325, 332], [332, 331], [331, 331], [331, 323], [323, 309], [309, 295], ... [320, 336], [336, 329], [329, 298]].

d) Memilih Nilai Terkecil

Selanjutnya memilih nilai hash terkecil dari setiap window untuk dijadikan fingerprint dokumen tersebut. Apabila nilai hash terkecil pada window selanjutnya sama dengan nilai hash terkecil pada window sebelumnya maka nilai hash-nya tidak perlu

dituliskan lagi.

Jawaban mahasiswa: [297, 316, 309, 308, 311, 304, 335, 325, 333, 331, 323, 295, 303, 330, 341, 337, 315, 314, 302, 332, 312, 321, 317, 318, 326, 320, 322, 313, 301, 339, 293, 310, 329, 319, 343, 306, 338, 298, 305, 336, 344, 327, 349, 292, 291].

Kunci jawaban: [325, 333, 311, 309, 331, 323, 295, 303, 330, 341, 342, 345, 321, 297, 292, 313, 305, 343, 335, 318, 308, 316, 302, 337, 307, 320, 329, 298, 315, 306, 314, 301, 300, 304, 310, 293, 339, 324, 327, 294, 319, 344, 299, 332, 336, 328].

4) Menghitung Nilai Similarity

Untuk menghitung presentase kemiripan dengan menggunakan Jaccard’s Similarity Coeficient terlebih dahulu mencari jumlah fingerprint yang sama dan keseluruhan fingerprint dari hasil proses diatas.

$$|A \cap B| = 297, 316, 309, 308, 311, 304, 335, 325, 333, 331, 323, 295, 303, 330, 341, 337, 315, 314, 302, 332, 321, 318, 320, 313, 301, 339, 293, 310, 329, 319, 343, 306, 298, 305, 336, 344, 327, 292 = 38$$

$$|A \cup B| = 297, 316, 309, 308, 311, 304, 335, 325, 333, 331, 323, 295, 303, 330, 341, 337, 315, 314, 302, 332, 321, 318, 320, 313, 301, 339, 293, 310, 329, 319, 343, 306, 298, 305, 336, 344, 327, 292, 312, 317, 326, 322, 338, 349, 291, 342, 345, 307, 300, 324, 294, 299, 328 = 53$$

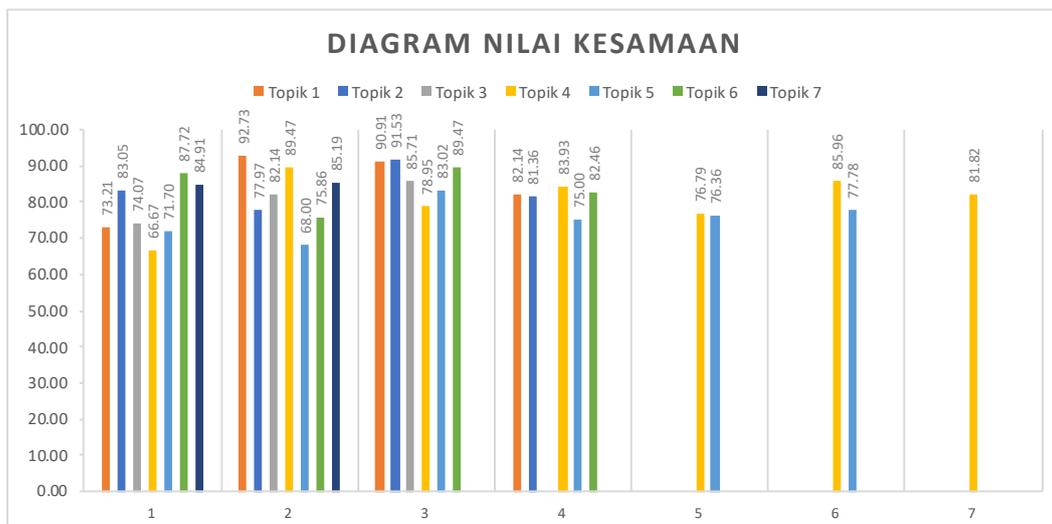
$$D(A, B) = \frac{38}{53} \times 100$$

$$D(A, B) = 0.7169811320754716 \times 100$$

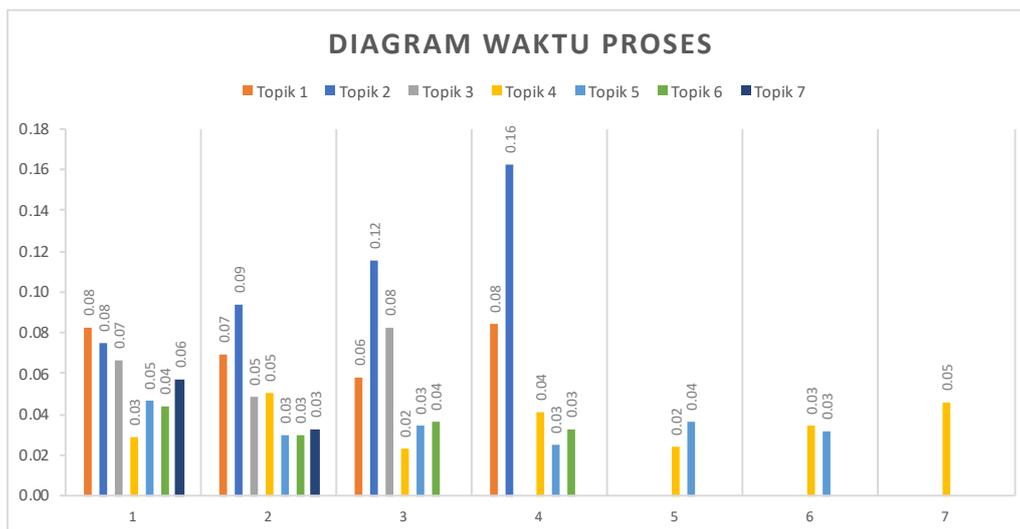
$$D(A, B) = 71.69811320754717$$

5) Hasil Proses

Proses evaluasi dengan algoritma *winnowing* dengan nilai k = 2, basis hash = 2 dan nilai w = 2 yang dikelompokkan berdasarkan topik soal dengan menghitung nilai kesamaan dan waktu proses yang disajikan dalam bentuk grafik seperti pada gambar 3 dan 4.



Gambar 3. Diagram nilai kesamaan jawaban



Gambar 4. Diagram waktu proses perhitungan

Diagram pada gambar 3 dan 4 menyajikan hasil proses topik 1 sebanyak 4 mahasiswa dimana ID 18 menghasilkan nilai tertinggi dengan nilai kesamaan 92,73%, waktu proses 0.07 detik, topik 2 sebanyak 4 mahasiswa dimana ID 17 menghasilkan nilai tertinggi dengan nilai kesamaan 91,53%, waktu proses 0.12 detik, topik 3 sebanyak 3 mahasiswa dimana ID 23 menghasilkan nilai tertinggi dengan nilai kesamaan 85,71%, waktu proses 0.08 detik, topik 4 sebanyak 7 mahasiswa dimana ID 5 menghasilkan nilai tertinggi dengan nilai kesamaan 89,47%, waktu proses 0.05 detik, topik 5 sebanyak 6 mahasiswa dimana ID 15 menghasilkan nilai tertinggi dengan nilai kesamaan 83,02%, waktu proses 0.03 detik, topik 6 sebanyak 4 mahasiswa dimana ID 21 menghasilkan nilai tertinggi dengan nilai kesamaan 89,47%, waktu proses 0.04 detik dan topik 7 sebanyak 7 mahasiswa dimana ID 28 menghasilkan nilai tertinggi dengan nilai kesamaan 85,19%, waktu proses 0.03 detik.

SIMPULAN DAN SARAN

A. Simpulan

Evaluasi otomatis dengan algoritma Nazief & Andriani untuk *stemming* text dari data kunci jawaban dan jawaban siswa dan algoritma *Winnowing* untuk memperoleh *fingerprint* dari data text untuk selanjutnya dihitung kemiripannya dengan Jaccard's *Similarity Coefficient* sehingga diperoleh hasil evaluasi dari 30 jawaban siswa yang dibandingkan dengan kunci jawaban sesuai topik yang ditentukan dapat di proses dalam waktu 1,62 detik dengan nilai rata-rata kesamaan 81,20% dan menentukan nilai tertinggi yang diraih oleh mahasiswa ID 18 dengan nilai kesamaan 92,73%.

B. Saran

Kelemahan dari penelitian ini adalah proses ujian dan pengumpulan data masih dilakukan secara manual menggunakan sistem aplikasi pengolah kata dan dihitung dengan program berbasis *command-line interface* (CLI). Peneliti selanjutnya diharapkan dapat menghasilkan aplikasi berbasis *graphical user interface* (GUI) yang dapat melakukan proses ujian dan penilaian secara langsung sehingga dapat memudahkan dosen untuk melakukan evaluasi terhadap mahasiswa.

DAFTAR RUJUKAN

- Agung Putri Ratna, A., Lalita Luhurkinanti, D., Ibrahim, I., Husna, D., & Dewi Purnamasari, P. (2018). Automatic Essay Grading System for Japanese Language Examination Using Winnowing Algorithm. *Proceedings - 2018 International Seminar on Application for Technology of Information and Communication: Creative Technology for Human Life, ISEmantic 2018*, 565–569. <https://doi.org/10.1109/ISEMANTIC.2018.8549789>
- Amorim, E., Cançado, M., & Veloso, A. (2018). Automated essay scoring in the presence of biased ratings. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, 1*, 229–237. <https://doi.org/10.18653/v1/n18-1021>
- Arifin, Y., Isa, S. M., Wulandhari, L. A., & Abdurachman, E. (2018). Plagiarism Detection for Indonesian Language using Winnowing with Parallel Processing. *Journal of Physics: Conference Series*, 978(1). <https://doi.org/10.1088/1742-6596/978/1/012082>
- Chapuis, B., & Garbinato, B. (2018). Geodabs: Trajectory Indexing Meets Fingerprinting at Scale. *Proceedings - International Conference on Distributed Computing Systems*. <https://doi.org/10.1109/ICDCS.2018.00108>
- Diana, N. E., & Hanana Ulfa, I. (2019). Measuring performance of n-gram and jaccard-similarity metrics in document plagiarism application. *Journal of Physics: Conference Series*, 1196(1). <https://doi.org/10.1088/1742-6596/1196/1/012069>
- Dra.Rosinta, Drs.Asrul, R. A. M. p. (2014). Evaluasi Pembajalaran. In □□□□□□.
- Fachrurrozi, M., Yusliani, N., & Yoanita, R. U. (2013). *Frequent Term based Text Summarization for Bahasa Indonesia*. December. <https://doi.org/10.15242/IIE.E1213550>
- Kalra, V., & Aggarwal, R. (2018). Importance of Text Data Preprocessing & Implementation in Rapid Miner. *Proceedings of the First International Conference on Information Technology and Knowledge Management*, 14(January), 71–75. <https://doi.org/10.15439/2017km46>
- Lemantara, J., Dewiyani Sunarto, M. J., Hariadi, B., Sagirani, T., & Amelia, T. (2018). Prototype of Online Examination on MoLearn Applications Using Text Similarity to Detect Plagiarism. *Proceedings - 2018 5th International Conference on Information Technology, Computer and Electrical Engineering, ICITACEE 2018*, 131–136. <https://doi.org/10.1109/ICITACEE.2018.8576922>
- Mišić, M. J., Protić, J., & Tomašević, M. V. (2018). Improving source code plagiarism detection: Lessons learned. *2017 25th Telecommunications Forum, TELFOR 2017 - Proceedings*. <https://doi.org/10.1109/TELFOR.2017.8249481>
- Prihatini, P. M., Putra, I. K. G. D., Giriantari, I. A. D., & Sudarma, M. (2017). Stemming Algorithm for Indonesian Digital News Text Processing. *International Journal of Engineering and Emerging Technology*, 2(2), 1–7.
- Rahmatulloh, A., Kurniati, N. I., Asyikin, A. Z., Darmawan, I., & Witarsyah, J. D.

- (2019). Comparison between the stemmer porter effect and nazief-adriani on the performance of winnowing algorithms for measuring plagiarism. *International Journal on Advanced Science, Engineering and Information Technology*, 9(4), 1124–1128. <https://doi.org/10.18517/ijaseit.9.4.8844>
- Rinartha, K., & Suryasa, W. (2017). Comparative study for better result on query suggestion of article searching with MySQL pattern matching and Jaccard similarity. *2017 5th International Conference on Cyber and IT Service Management, CITSM 2017*. <https://doi.org/10.1109/CITSM.2017.8089237>
- Septian, G., Susanto, A., & Shidik, G. F. (2018). Indonesian news classification based on NaBaNA. *Proceedings - 2017 International Seminar on Application for Technology of Information and Communication: Empowering Technology for a Better Human Life, ISemantic 2017, 2018-Janua*, 175–180. <https://doi.org/10.1109/ISEMANTIC.2017.8251865>
- Slamet, C., Atmadja, A. R., Maylawati, D. S., Lestari, R. S., Darmalaksana, W., & Ramdhani, M. A. (2018). Automated Text Summarization for Indonesian Article Using Vector Space Model. *IOP Conference Series: Materials Science and Engineering*, 288(1). <https://doi.org/10.1088/1757-899X/288/1/012037>
- Sutoyo, R., Ramadhani, I., & Ardiatma, A. D. (2017). Detecting Documents Plagiarism using Winnowing Algorithm and K-Gram Method. *Cybernetics and Computational Intelligence (CyberneticsCom), 2017 IEEE International Conference*, 67–72.
- Ulinuha, N., Asyhar, A. H., Thohir, M., Arifin, A. Z., Candra, D., & Novitasari, R. (2019). Implementation of Winnowing Algorithm for Document Plagiarism Detection. *Proceeding of the Electrical Engineering Computer Science and Informatics*, 5(5), 631–636.
- Watequlis Syaifudin, Y., Saputra, P. Y., & Puspitasari, D. (2018). The implementation of web service based text preprocessing to measure Indonesian student thesis similarity level. *MATEC Web of Conferences*. <https://doi.org/10.1051/mateconf/201819703019>
- Yudhana, A., Sunardi, & Mukaromah, I. A. (2018). Implementation of winnowing algorithm with dictionary English-Indonesia technique to detect plagiarism. *International Journal of Advanced Computer Science and Applications*, 9(5), 183–189. <https://doi.org/10.14569/IJACSA.2018.090523>