



Desain Website Sebagai Platform Koreksi Essay Menggunakan Metode Rabin Karp

Darso¹, Tarwoto²

^{1,2}Program Studi Informatika, Universitas Amikom Purwokerto,
Jawa Tengah, Indonesia

Email: ¹darso@amikompurwokerto.ac.id, ²tarwoto77@gmail.com

Abstrak

Ujian berdasarkan jenisnya dapat dilakukan dalam 3 bentuk, yaitu ujian pilihan ganda, isian singkat dan *essay*. Ujian dengan sistem *essay* diberikan agar pengajar dapat mengetahui pemahaman dari pelajar akan suatu ilmu. Ujian dengan sistem *essay* ini masih dilakukan secara tertulis dan dikoreksi secara manual, yaitu mengisikan jawaban secara tertulis dan membaca jawaban satu persatu, sehingga pengajar membutuhkan waktu yang lama dalam mengkoreksi dan merekap nilai. Berdasarkan masalah diatas penulis memberikan solusi untuk membuat aplikasi koreksi ujian berbasis website dengan menggunakan metode rabin-karp. Metode ini memiliki beberapa keunggulan dalam penerapannya, karena cocok digunakan untuk string yang panjang. Dengan menggunakan *parsing* k-gram dan hashing dalam menentukan jumlah kecocokan teks atau kalimat yang sama pada jawaban dan kunci jawaban. Metode pengumpulan data yang digunakan adalah observasi, studi literatur, dan wawancara. Metode pengembangan sistem yang digunakan adalah *Extreme Programming* diantaranya *planning*, *design*, *coding* dan *testing*.

Kata Kunci : Essay, Rabin-Karp, Website

1. PENDAHULUAN

Pesatnya perkembangan teknologi saat ini sangat cepat dan hampir menyeluruh disemua kalangan dan semua bidang. Salah satu bidang yang tidak pernah terlepas dari peranan teknologi adalah bidang pendidikan[1][2]. Teknologi informasi telah berkembang dengan pesat dan membawa paradigma perubahan dalam aktifitas manusia dalam belajar dan bekerja[3]. Perkembangan teknologi dan informasi saat ini cenderung mengarah kearah teknologi terdistribusi (*client-server*) dimana fungsi sebuah komputer tidak terbatas hanya pada mesin mandiri, tetapi dapat juga digunakan secara terkoordinasi dengan



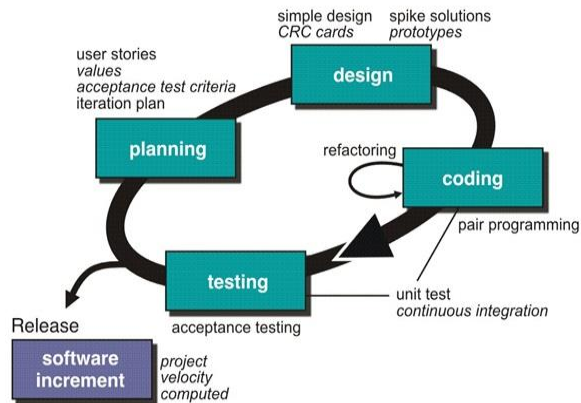
beberapa komputer lainnya[4][5]. Perkembangan tersebut dituntut untuk dapat dioperasionalkan dan dimanfaatkan secara optimal diberbagai bidang.

Dalam bidang pendidikan ujian merupakan bentuk evaluasi untuk mengetahui hasil dari proses pembelajaran[6]. Berdasarkan jenis ujian dapat dilakukan dalam tiga bentuk, yaitu ujian pilihan ganda, isian singkat dan *essay*[7]. Ujian dengan sistem *essay* merupakan bentuk evaluasi dimana pilihan jawaban tidak disediakan, dan siswa harus menjawab dengan kalimat sehingga dapat melatih siswa dalam menyampaikan sesuatu informasi secara verbal[8], selain itu ujian *essay* juga menuntut pemahaman yang lebih baik akan suatu ilmu dan dapat digunakan untuk mengukur tingkat pemahaman manusia akan suatu ilmu secara lebih mendalam[9]. Ujian dengan sistem ini tetap menjadi pilihan pengajar untuk mengevaluasi tingkat kemampuan pemahaman siswa walaupun kenyataannya tidak mudah untuk memberikan penilaian yang objektif pada jawaban siswa[10].

Pada makalah ini metode yang digunakan adalah Rabin-Karp yang menggunakan fungsi *hashing* untuk menemukan *pattern* di dalam *string* teks. Metode ini memiliki beberapa keunggulan dalam penerapannya, salah satunya adalah metode ini sangat cocok digunakan untuk *string* yang panjang[11][12]. Dengan menggunakan *parsing* k-gram dan *hashing* dalam menemukan jumlah kecocokan teks atau kalimat yang sama pada jawaban dan kunci jawaban[13]. Setelah ditemukan jumlah kecocokan teks atau kalimat yang sama maka tahap selanjutnya adalah menghitung nilai *similarity*-nya dengan menggunakan persamaan *Dice's Similarity Coefficient* untuk mengetahui persentase kemiripan kedua teks jawaban siswa dengan kunci jawaban[14]. Dengan mengetahui persentase kemiripan kedua teks jawaban tersebut dapat dijadikan acuan untuk melakukan penilaian. Karena hal-hal tersebut penulis mengusulkan sistem penilaian otomatis pada jawaban ujian berbentuk *essay* menggunakan metode Rabin Karp[15]. Dengan metode ini, diharapkan mampu mengatasi masalah proses pengkoreksian ujian *essay* di institusi pendidikan.

2. METODE

Konsep penelitian yang digunakan adalah model *Extreme Programming* (XP). Pemilihan model *Extreme Programming* dikarenakan sifat dari aplikasi yang dikembangkan dengan cepat melalui tahapan-tahapan yang ada meliputi: *planning*/ perencanaan, *design*/ perancangan, *coding*/ pengkodean dan *testing*/pengujian serta menggunakan pendekatan *object-oriented* sebagai paradigma pengembangan dan mencakup seperangkat aturan. *Extreme Programming* merupakan salah satu metode pengembangan *software* yang termasuk dalam *Agile Software Development*. XP mempunyai 5 (lima) nilai yang menjadi pondasi seperti terlihat pada gambar 1 berikut :



Gambar 1, Model Extreem Programming (XP)

a. Komunikasi (*Communication*)

Komunikasi yang paling efektif dalam XP membutuhkan peran antara *developer* dan *user* dalam menentukan fitur apa yang dibutuhkan serta kegunaan *software* tersebut. XP menekankan adanya kolaborasi yang baik antara *developer* dengan *user*, sehingga pihak *user* dapat terlibat langsung mengenai konsep *software* serta adanya umpan balik terus menerus.

b. Kesederhanaan (*Simplicity*)

Untuk mencapai kesederhanaan, XP membatasi *developer* hanya merancang kebutuhan yang mendesak dibandingkan kebutuhan mendatang, sehingga memudahkan untuk diimplementasikan dalam pemrograman. Jika rancangan harus diubah, dapat dilakukan *refactoring*. *Refactoring* yang dimaksud adalah memungkinkan *software engineer* mengubah struktur internal desain atau *source code* suatu rancangan tanpa mengubah fungsi eksternal serta perilakunya. *Refactoring* dilakukan untuk meningkatkan *efisiensi*, *readability*, dan performa dari suatu rancangan.

c. Umpan balik (*Feedback*)

Umpan balik berasal dari 3 (tiga) sumber yaitu implementasi *software* itu sendiri, *user*, dan anggota tim *software* yang lain. Umpan balik dilihat dari implementasi *output*, fungsi, dan karakteristik *use case*. Akhirnya, sebagai persyaratan baru yang berasal dari bagian *refactoring*, tim memberikan *user* dengan umpan balik yang cepat mengenai biaya.

d. Keberanian (*Courage*)

Praktek yang dilakukan dalam XP lebih menekankan keberanian atau kata yang tepat adalah disiplin. Tim *developer agile XP* harus memiliki kedisiplinan dalam merancang kebutuhan sekarang, mengenali kebutuhan yang akan datang

yang mungkin akan berubah secara drastis, sehingga memungkinkan untuk mengubah rancangan dan implementasi *code*.

e. Menghargai (*Respect*)

Setiap tim *developer* mempunyai nilai yang harus ditanamkan dan dihargai oleh sesama anggota juga antara *stakeholder* dan anggota tersebut.

3. HASIL DAN PEMBAHASAN

a. Planning (Perencanaan)

Kegiatan perencanaan (disebut juga *planning game*) biasanya dimulai dengan mendengarkan suatu kegiatan yang bertujuan untuk mengumpulkan kebutuhan-kebutuhan yang memungkinkan anggota tim xp memahami konteks bisnis untuk perangkat lunak yang akan dikembangkan dan untuk merasakan perlunya keluaran-keluaran (output), fitur-fitur utama, dan fungsionalitas. Aktivitas-aktivitas mendengarkan pada dasarnya mengarah kepada pembuatan serangkaian cerita (juga disebut *user stories*) yang menggambarkan keluaran yang diperlukan, fitur-fitur, dan fungsionalitas-fungsionalitas yang akan dibangun menggunakan perangkat lunak yang akan dikembangkan. Pada tahap ini yang diperoleh berupa data kebutuhan fungsional dan non fungsional.

b. Design (Pemodelan)

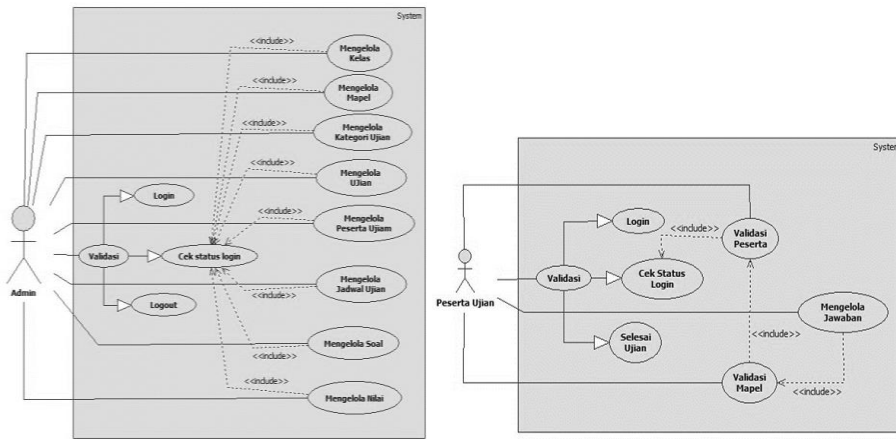
Perancangan system dalam penelitian ini menggunakan Bahasa pemodelan *Unified Modelling Language* (UML). UML adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. Perancangan meliputi :

1) Usecase.

Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri. Berikut adalah deskripsi pendefinisian aktor pada aplikasi koreksi ujian *essay*:

Tabel 1, Deskripsi pengguna.

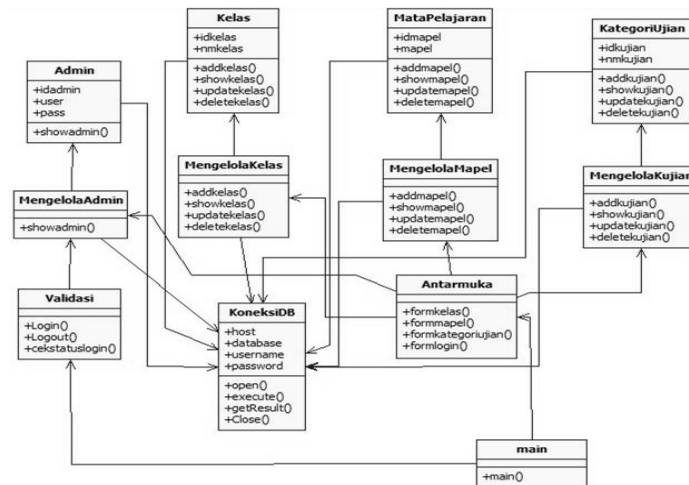
No	Aktor	Deskripsi
1	Admin	Orang yang bertugas dan memiliki hak akses untuk melakukan operasi pengelolaan data peserta ujian, data kelas, data mata pelajaran, data kategori ujian, data ujian, data soal, data jadwal ujian, dan nilai ujian.
2	Peserta Ujian	Peserta ujian adalah user yang bertugas untuk mengisikan jawaban yang akan muncul sesuai ujian yang akan dikerjakan.



Gambar 2, Model usecase admin dan peserta ujian

2) Class Diagram.

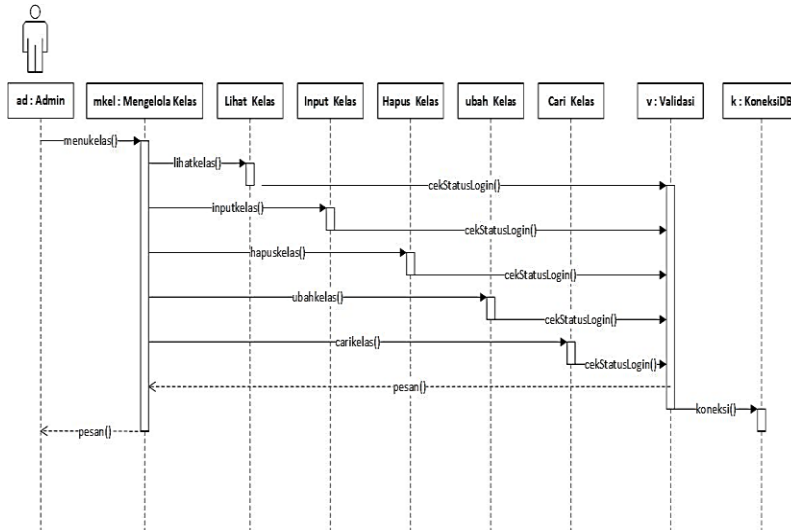
Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas, operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Berikut adalah diagram kelas pada aplikasi ujian essay :



Gambar 3, Class Diagram system essay.

3) Sequence Diagram.

Pada model sequence diagram, disusun langsung pada proses pengelolaan ujian essay yang digabungkan dengan alur pengelolaan system. Berikut gambar 4 adalah proses alur tata kelola system.



Gambar 4, Proses alur pengelolaan system essay.

c. Coding (Pembangunan)

Pemodelan Rabin-Karp akan diujicobakan di tahap ini. Penulis mengambil salah satu contoh soal essay yang akan di uji,

Tabel 2, Sampel pertanyaan essay.

Pertanyaan	Kunci Jawaban	Jawaban Siswa	Skor
Definisikan apa yg dimaksud algoritma	Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis.	Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis.	20

Dari pertanyaan dan jawaban pada tabel 2 diatas tahap pertama pada metode rabin karp adalah *preprocessing*. Berikut adalah hasil setelah melalui tahap *preprocessing*:

Tabel 3, Preprocessing data

Kunci Jawaban	Jawaban Siswa
[algori][lgorit][goritm][oritma][ritma][itma a][tma ad][ma ada][a adal][adala][adalah][dalah][alah u][lah ur][ah uru][h urut][uruta][urutan][rutan][utan l][tan la][an lan][n lang][langk][langka][angkah][ngkah][gkah l][kah la][ah lan][h lang][langk][langka][angkah][ngkah][gkah l][kah lo][ah log][h logi][logis][logis][ogis p][gis pe][is pen][s peny][penye][penyel][enyele][nyeles][yelesa][elesai][lesaia][esaian][saian][aian m][ian ma][an mas][n masa][masal][masala][asalah][salah][alah y][lah ya][ah yan][h yang][yang][yang d][ang di][ng dis][g disu][disus][disusu][isusun][susun][usun s][sun se][un sec][n seca][secar][secara][ecara][cara s][ara si][[algori][lgorit][goritm][oritma][ritma][itma a][tma ad][ma ada][a adal][adala][adalah][dalah][alah u][lah ur][ah uru][h urut][uruta][urutan][rutan][utan l][tan la][an lan][n lang][langk][langka][angkah][ngkah][gkah l][kah la][ah lan][h lang][langk][langka][angkah][ngkah][gkah l][kah lo][ah log][h logi][logis][logis][ogis p][gis pe][is pen][s peny][penye][penyel][enyele][nyeles][yelesa][elesai][lesaia][esaian][saian][aian m][ian ma][an mas][n masa][masal][masala][asalah][salah][alah y][lah ya][ah yan][h yang][yang][yang d][ang di][ng dis][g disu][disus][disusu][isusun][susun][usun s][sun se][un sec][n seca][secar][secara][ecara][cara s][ara si][ra sis][

Setelah melalui tahap *preprocessing* tahap selanjutnya adalah *rolling hash*. Tahapan ini bertujuan untuk mencari nilai dari setiap kata yang telah diparsing. Hasil dari tahapan rolling hash bisa dilihat pada tabel di bawah ini:

Tabel 4, Tahap Rolling Hash

Kunci Jawaban	Jawaban Siswa
[17355058] [19064337] [18379228]	[17355058] [19064337] [18379228]
[19700822] [20065803] [18765976]	[19700822] [20065803] [18765976]
[20411931] [19030262] [16232841]	[20411931] [19030262] [16232841]
[6719931] [17229393] [17681938]	[6719931] [17229393] [17681938]
[17345335] [18957382] [17202731]	[17345335] [18957382] [17202731]
[17388740] [7033893] [20682981]	[17388740] [7033893] [20682981]
[20240186] [20684200] [20253660]	[20240186] [20684200] [20253660]
[17289294] [18340920] [6878517]	[17289294] [18340920] [6878517]
[18973832] [17383668] [19378963]	[18973832] [17383668] [19378963]
[18296991] [18796215] [17201448]	[18296991] [18796215] [17201448]
[17374614] [6878517] [18973832]	[17374614] [6878517] [18973832]

[17383668]	[19378963]	[18296991]	[17383668]	[19378963]	[18296991]
[18796229]	[17201595]	[17376233]	[18796229]	[17201595]	[17376233]
[6896334]	[19169754]	[19538818]	[6896334]	[19169754]	[19538818]
[18283828]	[18651435]	[19152001]	[18283828]	[18651435]	[19152001]
[6942597]	[19678723]	[18051222]	[6942597]	[19678723]	[18051222]
[19635896]	[21123243]	[17996897]	[19635896]	[21123243]	[17996897]
[19038303]	[18092855]	[20093776]	[19038303]	[18092855]	[20093776]
[17302130]	[18482110]	[17289420]	[17302130]	[18482110]	[17289420]
[18342300]	[6893698]	[19140823]	[18342300]	[6893698]	[19140823]
[17449008]	[20097703]	[17345339]	[17449008]	[20097703]	[17345339]
[18957409]	[17203021]	[17391917]	[18957409]	[17203021]	[17391917]
[7068775]	[21066673]	[17374627]	[7068775]	[21066673]	[17374627]
[19279595]	[17203952]	[6772804]	[19279595]	[17203952]	[6772804]
[17811009]	[18765109]	[20402326]	[17811009]	[18765109]	[20402326]
[20696186]	[20385510]	[20511194]	[20696186]	[20385510]	[20511194]

Setelah tahapan rolling hashing pada kunci jawaban selesai, tahapan berikutnya adalah menghitung nilai yang sama antara kedua jawaban tersebut. Perhitungan ini menggunakan rumus Dice Similarity Coefficient.

$$S = 100 * \frac{K \times C}{(A + B)}$$

Dengan Menggunakan data tabel yang ada di atas, proses penilaian dapat diselesaikan sebagai berikut :

A. Jumlah k-gram pada kunci jawaban di tabel tersebut adalah 102

B. Jumlah k-gram pada jawaban siswa di tabel tersebut adalah 102

Jumlah k-gram yang sama pada kunci jawaban dan jawaban siswa di tabel tersebut adalah 102.

$$S = 100 * \frac{2 \times 102}{(102 + 102)}$$

$$S = 100 \%$$

Hasil yang didapat dari tahapan di atas ditemukan nilai kemiripan sebesar 100%. Setelah nilai kemiripan sudah diketahui, tahapan berikutnya adalah menghitung skor yang didapat dari pertanyaan tersebut. Berikut adalah skor yang didapat oleh siswa.

$$\text{Skor} = 20 * \frac{100}{100}$$

$$\text{Skor} = 20$$

Dari hasil pengujian diatas dapat dikatakan bahwa perbedaan nilai rata-rata dari hasil ujian menggunakan aplikasi dengan secara manual oleh guru dengan nilai 2.6 % sehingga sistem bisa mengoreksi hasil ujian dengan baik.

4. KESIMPULAN

Aplikasi dapat melakukan pencocokan kata atau kalimat yang sama pada hasil ujian. Pada hasil uji coba yang dilakukan metode Rabi-Karp dapat mencocokkan kata atau kalimat dengan baik pada jawaban dan kunci jawaban dengan range perbedaan rata-rata nilai dari 1,6% sampai 2,6%. Ujian dengan jawaban yang bersifat opini ini tidak direkomendasikan untuk melaksanakan ujian menggunakan aplikasi ini dikarenakan memiliki beragam jawaban yang bernilai benar. Kesalahan pengetikan saat menginputkan jawaban berpengaruh terhadap hasil akhir dari sebuah ujian.

REFERENCES

- [1] A. Hassanzadeh, F. Kanaani, and S. Elahi, "A model for measuring e-learning systems success in universities," *Expert Syst. Appl.*, vol. 39, no. 12, pp. 10959–10966, 2012, doi: 10.1016/j.eswa.2012.03.028.
- [2] U. Raharja, N. Lutfiani, I. Handayani, and F. M. Suryaman, "Motivasi Belajar Mahasiswa Terhadap Metode Pembelajaran Online iLearning+ Pada Perguruan Tinggi," *Sisfotenika*, vol. 9, no. 2, p. 192, 2019, doi: 10.30700/jst.v9i2.497.
- [3] M. Urh, G. Vukovic, E. Jereb, and R. Pintar, "The Model for Introduction of Gamification into E-learning in Higher Education," *Procedia - Soc. Behav. Sci.*, vol. 197, no. February, pp. 388–397, 2015, doi: 10.1016/j.sbspro.2015.07.154.
- [4] J. Gil-Flores, J. Rodríguez-Santero, and J. J. Torres-Gordillo, "Factors that explain the use of ICT in secondary-education classrooms: The role of teacher characteristics and school infrastructure," *Comput. Human Behav.*, vol. 68, pp. 441–449, 2017, doi: 10.1016/j.chb.2016.11.057.
- [5] R. Scherer, S. K. Howard, J. Tondeur, and F. Siddiq, "Profiling teachers' readiness for online teaching and learning in higher education: Who's ready?," *Comput. Human Behav.*, vol. 118, no. October 2020, p. 106675, 2021, doi: 10.1016/j.chb.2020.106675.
- [6] S. A. Chapman, S. Goodman, J. Jawitz, and A. Deacon, "A strategy for monitoring and evaluating massive open online courses," *Eval. Program Plann.*, vol. 57, pp. 55–63, 2016, doi: 10.1016/j.evalprogplan.2016.04.006.
- [7] P. Sun, "The Application of Cloud Model in Teaching Quality Monitoring and Evaluation," *Appl. Mech. Mater.*, vol. 519–520, pp. 1688–1691, 2014, doi: 10.4028/www.scientific.net/amm.519-520.1688.
- [8] V. N. Lyusev, A. Nazarenko V, N. Osipova V, N. E. Mokievskaya, and E. M. Chertakova, "Monitoring and Evaluation of the Educational Results of

- University Students in a Pandemic," *Eduweb-Revista Tecnol. Inf. Y Comun. En Educ.*, vol. 14, no. 2, pp. 301–312, 2020.
- [9] A. R. Adriyanto, I. Santosa, and A. Syarief, "Evaluasi Heuristik Sistem Pengelolaan Pembelajaran Daring Perguruan Tinggi Di Indonesia," *ANDHARUPA J. Desain Komun. Vis. Multimed.*, vol. 6, no. 02, pp. 215–234, 2020, doi: 10.33633/andharupa.v6i02.3592.
- [10] A. Iglesias Rodríguez, B. García Riaza, and M. C. Sánchez Gómez, "Collaborative learning and mobile devices: An educational experience in Primary Education," *Comput. Human Behav.*, vol. 72, pp. 664–677, 2017, doi: 10.1016/j.chb.2016.07.019.
- [11] A. D. Hartanto, A. Syaputra, and Y. Pristyanto, "Best Parameter Selection Of Rabin-Karp Algorithm In Detecting Document Similarity," in *2019 International Conference on Information and Communications Technology (ICOIACT)*, 2019, pp. 457–461. doi: 10.1109/ICOIACT46704.2019.8938458.
- [12] E. Rasywir, Y. Pratama, Hendrawan, and M. Istoningtyas, "Removal of Modulo as Hashing Modification Process in Essay Scoring System Using Rabin-Karp," in *2018 International Conference on Electrical Engineering and Computer Science (ICECOS)*, 2018, pp. 159–164. doi: 10.1109/ICECOS.2018.8605211.
- [13] J. Priambodo, "Pendeteksian Plagiarisme Menggunakan Algoritma Rabin-Karp," *J. Inform. Univ. Pamulang*, vol. 3, no. 1, pp. 39–45, 2018.
- [14] A. A. B. Ginting and D. P. Utomo, "Perancangan Aplikasi Catalog Wisata Di Sumatera Utara Menggunakan Algoritma Rabin-Karp," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 3, no. 1, pp. 57–63, 2019, doi: 10.30865/komik.v3i1.1568.
- [15] J. Sharma and M. Singh, "CUDA based Rabin-Karp Pattern Matching for Deep Packet Inspection on a Multicore GPU," *Int. J. Comput. Netw. Inf. Secur.*, vol. 7, no. 10, pp. 70–77, 2015, doi: 10.5815/ijcnis.2015.10.08.