

## IMPLEMENTASI STRING MATCHING DENGAN ALGORITMA BOYER-MOORE UNTUK MENENTUKAN TINGKAT KEMIRIPAN PADA PENGAJUAN JUDUL SKRIPSI/TA MAHASISWA (STUDI KASUS: UNIVERSITAS XYZ)

Imam Ahmad<sup>1</sup>, Rohmat Indra Borman<sup>2\*</sup>, Gavan Gorbi Caksana<sup>3</sup>, Jafar Fakhrurozi<sup>4</sup>

<sup>1,4</sup>Sistem Informasi, Universitas Teknokrat Indonesia

<sup>2</sup>Sistem Informasi Akuntansi, Universitas Teknokrat Indonesia

<sup>3</sup>Informatika, Universitas Teknokrat Indonesia

Fakultas Teknik dan Ilmu Komputer, Universitas Teknokrat Indonesia

Jl. ZA. Pagar Alam No.9 -11, Labuhan Ratu, Kec. Kedaton, Kota Bandar Lampung, Indonesia

e-mail: imamahmad@teknokrat.ac.id<sup>1</sup>, rohmat\_indra@teknokrat.ac.id<sup>2</sup>,  
gavangorbicaksana26@gmail.com<sup>3</sup>, jafar.fakhrurozi@teknokrat.ac.id<sup>4</sup>

Received : October, 2020

Accepted : November, 2020

Published : April, 2021

### Abstract

*In determining the research topic as well as the title for the Thesis / TA, it is conceivable that there will be a likeness in the title between one understudy and another, whether intentionally or unintentionally. For this reason, the Study Program must check the titles submitted by students, so as to avoid plagiarism. Endeavors that can be made to make it simpler to check the Thesis title are by performing string matching on the text on the title that will be submitted by students. A string matching algorithm that can be used is Boyer-Moore. Boyer-Moore, who matched starting from the rightmost character then left based on the value of the occurrence heuristic (HO) and match heuristic (MH) in determining shift. This study produced a system that was able to determine the similarity of the proposed thesis title based on the title repository has been there. From the results of testing the processing time by entering 100 and its multiples show that the Boyer-Moore algorithm is able to match quickly. The results also showed that the more data the more time it took, but the average increase in time only increased by 34%.*

**Keywords:** boyer-moore, thesis, string matching

### Abstrak

*Dalam menentukan topik penelitian maupun judul untuk Skripsi/TA akan dimungkinkan terjadi kemiripan judul antara mahasiswa satu dengan yang lain baik secara disengaja maupun tidak sengaja. Untuk itu, Program Studi harus melakukan pengecekan terhadap judul yang diajukan oleh mahasiswa, sehingga dapat menghindari plagiat. Upaya yang dapat dilakukan dalam mempermudah pengecekan judul Skripsi/TA yaitu dengan pencocokan string matching pada teks pada judul yang akan diajukan oleh mahasiswa. Algoritma string matching yang dapat digunakan adalah Boyer-Moore. Boyer-Moore yang melakukan pencocokan dimulai dari karakter paling kanan kemudian ke kiri berdasarkan nilai dari occurrence heuristic (HO) dan match heuristic (MH) dalam menentukan pegeseran.. Penelitian ini menghasilkan sistem yang mampu menentukan kemiripan dari usulan judul Skripsi/TA berdasarkan repository judul yang telah ada. Dari hasil pengujian waktu proses dengan memasukan 100 dan kelipatannya menunjukkan bahwa algoritma Boyer-Moore mampu mencocokkan dengan cepat. Hasil penelitian juga menunjukkan semakin banyak data semakin bertambah waktu yang dibutuhkan akan tetapi rata-rata peningkatan waktu hanya bertambah 34%.*

**Kata Kunci:** boyer-moore, skripsi, string matching

## 1. PENDAHULUAN

Skripsi/Tugas Akhir (TA) merupakan istilah yang digunakan di Indonesia untuk menyebut karya tulis ilmiah mahasiswa jenjang Strata-1 (S1) dan Diploma-3 (D3). Skripsi/TA memuat hasil penelitian guna membahas suatu permasalahan di bidang ilmu tertentu [1]. Tujuan penyusunan Skripsi/TA pada dasarnya adalah untuk membentuk mahasiswa untuk mendapatkan sikap dan mental ilmiah, serta mengasah kemampuan dalam menerapkan ilmunya dalam menyelesaikan permasalahan dalam dunia nyata [2]. Sebelum membuat karya ilmiah dalam bentuk Skripsi/TA, mahasiswa harus membuat usulan judul penelitian untuk diajukan dengan memaparkan topik penelitian. Di Universitas XYZ, untuk pengajuan judul Skripsi/TA diajukan melalui form pengajuan judul dengan mengisi topic, judul dan paparan singkat Skripsi/TA yang akan diajukan.

Dalam menentukan topik penelitian maupun judul untuk Skripsi/TA akan dimungkinkan terjadi kemiripan judul antara mahasiswa satu dengan yang lain baik secara disengaja maupun tidak sengaja. Untuk itu Program Studi harus melakukan pengecekan terhadap judul yang diajukan oleh mahasiswa, sehingga dapat mengambil keputusan untuk menerima ataupun menolak didasari dengan salah satunya adalah judul skripsi apakah sudah ada atau mirip, agar terhindar dari indikasi adanya kesamaan judul dan adanya plagiasi. Judul yang diajukan mahasiswa setiap semesternya sangat banyak, untuk itu perlu ketelitian dan upaya yang lebih untuk melakukan pengecekan kemiripan judul. Pengecekan dengan manual akan membutuhkan waktu dan tenaga yang lebih. Upaya untuk membantu dalam pengecekan judul Skripsi/TA adalah dengan menerapkan *string matching* atau pencocokan kata pada teks pada judul yang akan diajukan oleh mahasiswa.

*String matching* merupakan proses pencarian sebuah *string* yang terdiri dari *pattern* terhadap karakter pada teks yang dicari [3][4]. Dalam pencocokan *string* terdapat dua pendekatan yaitu *exact string matching* dan *inexact string matching*. Pendekatan yang digunakan dalam penelitian ini adalah *exact string matching*, karena pendekatan ini digunakan untuk mencocokkan kata dengan tepat yang melibatkan struktur karakter teks

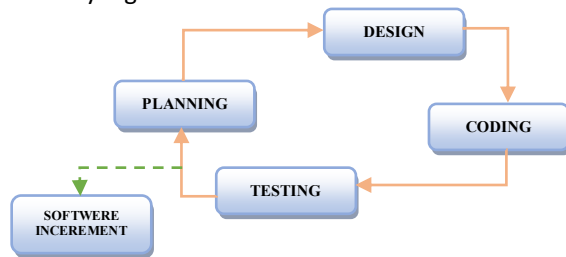
yang dicocokkan memiliki kesamaan dari jumlah urutan dan struktur katanya [5]. Salah satu pendekatan *exact string matching* yang efisien dapat melakukan perbandingan *pattern* adalah Boyer-Moore [6]. Algoritma memiliki ide dasar yaitu mencocokkan dan memindai pola *string* dari kanan ke kiri mulai dari karakter paling kanan dari *string* yang akan dicari [5][7]. Dalam beberapa penelitian algoritma Boyer-Moore menunjukkan kinerja yang baik dalam pencocokan *string*. Pada penelitian analisa dengan membandingkan Boyer-Moore dan Knuth Morris Pratt (KMP) untuk melakukan pencarian judul buku, menunjukkan bahwa algoritma Boyer-Moore lebih cepat dalam melakukan pencarian dan pencocokan *string* dibandingkan dengan algoritma KMP [8]. Selain itu, terdapat penelitian yang membandingkan penerapan Boyer-Moore, Turbo Boyer-Moore, dan Tuned Boyer-Moore untuk mencari kata, menunjukkan pendekatan Boyer-Moore menghasilkan waktu tercepat dibandingkan dengan tiga jenis Boyer-Moore lainnya [9]. Penelitian yang lain, tentang pengembangan *software* kamus untuk istilah kedokteran dengan menerapkan Boyer-Moore, menunjukkan bahwa algoritma Boyer-Moore mampu menampilkan hasil pencarian istilah dengan cepat [10].

Pada penelitian ini akan dikembangkan aplikasi menentukan tingkat kemiripan pada pengajuan judul Skripsi/TA mahasiswa dengan menerapkan *string matching* menggunakan algoritma Boyer-Moore. Aplikasi dilengkapi dengan persentase jumlah kemiripan judul dibandingkan dengan judul sebelumnya. Selain itu, pada penelitian ini data *pattern* yang digunakan untuk pencocokan *string* diambil dari *database repository* Universitas XYZ.

## 2. METODE PENELITIAN

Untuk melakukan pengembangan sistem dalam menentukan kemiripan pada pengajuan Skripsi/TA, pada penelitian ini menggunakan pendekatan metodologi pengembangan sistem *Extreme Programming* (XP). *Extreme programming* (XP) merupakan metodologi yang fokus pada pengkodean yang merupakan aktivitas utama dalam semua tahapan atau siklus pengembangan sistem [11]. XP menawarkan tahapan yang membutuhkan waktu yang cepat dan dapat diulangi pada bagian yang berbeda sesuai dengan fokus

tujuan pengembangan sistem [12]. Berikut ini tahapan-tahapan XP dalam pengembangan sistem yang dilakukan.



Gambar 1. Tahapan Pada *Extreme Programming* (XP)  
(Sumber: Suryantara, 2017 [13])

Berdasarkan gambar 1 diatas, maka tahapan yang dilakukan pada metodologi pengembangan sistem XP adalah sebagai berikut :

#### 1) *Planning* (Perencanaan)

Langkah awal yang dilakukan adalah perencanaan, yaitu dengan memahami alur proses dari aplikasi, mendefinisikan *output*, fitur-fitur yang ada pada aplikasi, fungsi dari aplikasi, dan alur pengembangan dari aplikasi yang akan dikembangkan [14][19]. Selain itu pada tahap perencanaan juga terdapat tahapan dimana pengembang menentukan fungsionalitas keseluruhan dari sistem [15][20][21]. Pada penelitian ini permasalahan yang ada pada Universitas XYZ adalah proses pengecekan judul yang diajukan oleh mahasiswa apakah terdapat kemiripan atau indikasi plagiasi dari judul Skripsi/TA yang sebelumnya. Dari permasalahan tersebut kebutuhan utama dari aplikasi yang akan dibangun adalah sistem memiliki kemampuan menentukan tingkat kemiripan judul berdasarkan *database repository* artikel ilmiah pada Universitas XYZ.

#### 2) *Design* (Perancangan)

Setelah perencanaan telah disusun, berikutnya adalah membuat perancangan sistem. Pada penelitian ini perancangan sistem menggunakan *flowchart*, yang merupakan alat yang digunakan untuk menggambarkan langkah dan urutan prosedur dari suatu program.

#### 3) *Coding* (Pengkodean)

*Coding* atau pengkodean merupakan tahapan dimana, rancangan yang telah dibuat kemudian diterjemahkan dalam bentuk bahasa pemrograman yang dikenali oleh komputer [16]. Sistem dikembangkan dengan

menerapkan algoritma Boyer-Moore dalam pencocokan kemiripan dari judul Skripsi/TA yang diajukan oleh mahasiswa. Sistem diimplementasikan dalam bentuk web untuk *backend* dan android untuk akses *frontend*.

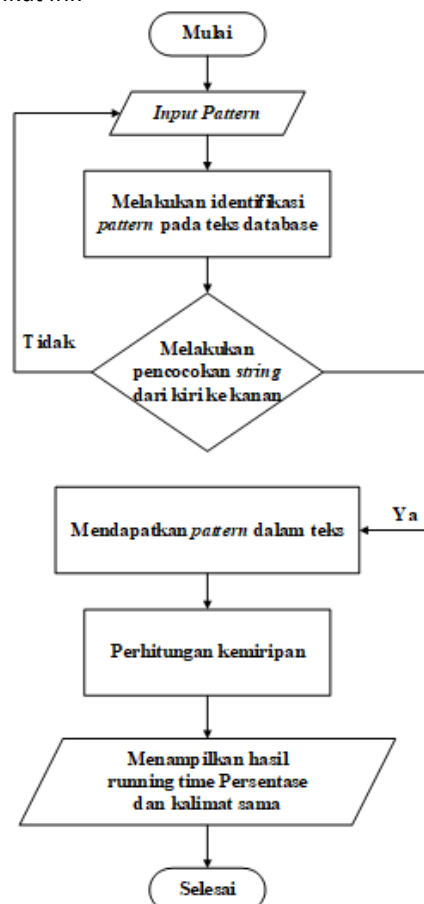
#### 4) *Testing* (Pengujian)

Untuk menegaskan *software* dapat bekerja dengan baik maka sistem harus melalu tahap pengujian untuk menemukan kesalahan – kesalahan [17]. Pengujian atau *testing* yang diterapkan adalah uji terhadap waktu yang dibutuhkan oleh aplikasi untuk memproses dan mendapatkan hasil presentase kemiripan.

### 3. HASIL DAN PEMBAHASAN

#### 3.1. *Flowchart* Alur Sistem

Untuk mempermudah dalam menggambarkan urutan proses yang dilakukan pada implementasi algoritma Boyer-Moore untuk mengetahui tingkat kemiripan judul Skripsi/TA maka alur cara kerja sistem digambarkan dengan *flowchart*. *Flowchart* alur sistem yang akan dikembangkan dapat dilihat pada gambar 2 berikut ini:



Gambar 2. Flowchart Alur Sistem  
(Sumber: Desain Penulis, 2020)

### 3.2. Implementasi Algoritma Boyer-Moore

Algoritma Boyer Moore memiliki cara kerja dengan melakukan *matching* atau pencocokan antara *pattern* dan *text* dengan memperhatikan urutan yang dimulai dari karakter paling kanan terlebih dahulu kemudian kekiri, dimana pencocokan dilakukan dengan dasar nilai dari *occurence heuristic* (HO) dan *match heuristic* (MH) [8][18]. Sebagai contoh penerapan algoritma Boyer-Moore dalam menentukan tingkat kemiripan judul Skripsi/TA dengan menggunakan kata “**ONLINE**” sebagai *string pattern* dan kata “**SISTEM ONLINE**”.

Tabel 1. Nilai *Occurence Heuristic* (OH) dan *Match Heuristic* (MH)  
(Sumber: Penulis, 2020)

Pattern	O	N	L	I	N	E
OH	5	4	3	2	1	0
MH	5	5	5	5	5	1

Pada tahap pertama, untuk pencocokan *string* karakter “E” pada *pattern* tidak ada kecocokan dengan karakter “M”, sehingga akan bergeser nilainya berdasarkan tabel *Occurence Heuristic* (OH) dan *Match Heuristic* (MH) pada tabel 1. Pada tabel OH dan MH karakter “M” tidak ada, maka jumlah pergeseran dilakukan sebanyak 6 langkah, tahapan ini ditunjukkan pada tabel 2.

Tabel 2. Pencocokan *Text* dan *Pattern*  
(Sumber: Penulis, 2020)

Index	0	1	2	3	4	5	6	7	8	9	10	11	12
Text	S	I	S	T	E	M							
Pattern	O	N	L	I	N	E							

Tahap kedua, karakter “E” setelah mengalami pergeseran ternyata tidak sama dengan karakter “N”, maka berikutnya adalah melakukan pergeseran merujuk pada nilai OH. Pada tabel OH untuk karakter “E” memiliki nilai 1, untuk itu selanjutnya akan dilakukan pergeseran sebanyak 1 langkah. Proses ini dijelaskan pada tabel 3.

Tabel 3. Pencocokan *Text* dan *Pattern*  
Dengan 6 Langkah Pergeseran  
(Sumber: Penulis, 2020)

Index	0	1	2	3	4	5	6	7	8	9	10	11	12
Text	S	I	S	T	E	M						N	E
Pattern							O	N	L	I	N	E	

Pada tahap ini, *string* pada *text* paling kanan yaitu karakter “E” telah sama atau cocok dengan *string* pada *pattern* yaitu karakter “E”, begitu pula dengan *string* pada *text* selanjutnya yaitu “N”, “I”, “L”, “N”, “O” telah

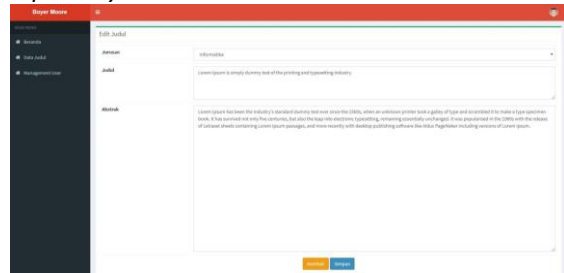
sama dengan *pattern*. Pergeseran berakhir pada tahap ketiga karena telah menemukan kesamaan, dapat dilihat pada tabel 4.

Tabel 4. Pencocokan Dengan 1 Langkah Pergeseran  
(Sumber: Penulis, 2020)

Index	0	1	2	3	4	5	6	7	8	9	10	11	12
Text	S	I	S	T	E	M		O	N	L	I	N	E
Pattern								O	N	L	I	N	E

### 3.3. Penerapan Sistem

Algoritma Boyer-Moore diterapkan dalam bahasa pemrograman untuk pengembangan sistem penentuan tingkat kemiripan Judul Skripsi/TA di Universitas XYZ. Sistem yang akan dikembangkan terdapat dua pengguna, yaitu admin dan pengguna. Admin dapat mengakses melalui *backend* yang dibangun berbasis web. Admin dapat mengelola judul dari Skripsi/TA yang sebelumnya dengan meng-*import* dari *repository* Universitas XYZ.



Gambar 3. Tampilan Pengelolaan Judul  
(Sumber: Aplikasi Yang Dikembangkan Penulis, 2020)

Sedangkan pengguna dapat mencari judul yang telah ada dan melakukan pengecekan dari judul yang akan diusulkan dengan judul yang ada di *repository* Universitas XYZ dengan menghasilkan persentase tingkat kemiripan.

#	Judul	Jurusan	Persentase
1	APRIANSYAH (2017) PENGEMBANGAN MEDIA PEMBELAJARAN SISTEM PENDUKUNG KEPUTUSAN BERBASIS MULTIMEDIA	Informatika	17.39%
2	NURKHOFIYAH (2017) RANCANG BANGUN MEDIA PEMBELAJARAN PEMROGRAMAN WEB MENGGUNAKAN CONSTRUCT 2 (STUDI KASUS TEKNIK KOMPUTER DAN JARINGAN (TKJ) DI SMK N 4 BANDAR LAMPUNG)	Informatika	17.50%
3	ANTONI (2018) SISTEM INFORMASI NILAI SISWA MENGGUNAKAN FRAMEWORK CODEIGNITER (STUDI KASUS : SMK N 1 LIWA LAMPUNG BARAT)	Informatika	15.13%
4	Edy Warsito (2018) GAME PETUALANGAN UNTUK ANAK TUNAGRAHITA MENGGUNAKAN CONSTRUCT 2	Informatika	23.38%
5	NURUL FEBRIYANI UTAMI (2019) PENGEMBANGAN APLIKASI GAME PENGENALAN HURUF DAN ANGKA MENGGUNAKAN CONSTRUCT 2	Informatika	17.58%
6	Muhammad Fadil Akbar (2018) PENGEMBANGAN GAME EDUKASI PENGENALAN HEWAN LANGKA BERBASIS ANDROID MENGGUNAKAN CONSTRUCT 2	Informatika	15.38%

Gambar 4. Hasil Pencocokan Kemiripan Usulan Judul

(Sumber: Aplikasi Yang Dikembangkan Penulis, 2020)

### 3.4. Pengujian Sistem

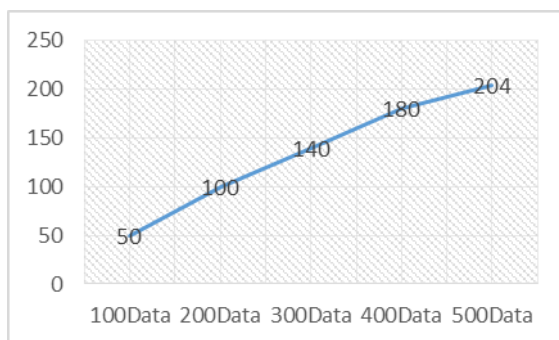
Untuk memperlihatkan kinerja dari algoritma Boyer-Moore maka pada penelitian ini dilakukan pengujian waktu proses pencocokan untuk menunjukkan kecepatan algoritma bekerja. Pengujian waktu proses dilakukan dengan melakukan pencocokan judul dengan memasukan 100 data dan kelipatannya. Berikut ini hasil pengujian terhadap waktu proses.

Tabel 5. Hasil Pengujian Waktu Proses Algoritma Boyer-Moore

(Sumber: Hasil Pengujian Peneliti, 2020)

NO	DATA	WAKTU
1	100	50ms
2	200	100ms
3	300	140ms
4	400	180ms
5	500	204ms

Dari hasil pengujian menunjukkan bahwa Boyer-Moore membutuhkan waktu yang cepat dalam proses *string matching*. Hal ini sesuai dengan hasil penelitian sebelumnya [8][9][10] yang menunjukkan bahwa algoritma Boyer-Moore mampu melakukan pencocokan *string* dengan waktu yang cepat. Dari hasil penelitian untuk masukan data yang lebih banyak maka waktu yang dibutuhkan akan semakin bertambah [18]. Akan tetapi peningkatan waktu yang dibutuhkan algoritma Boyer-Moore dari 100 data hingga 500 data rata-rata adalah 34%. Ini menunjukkan walaupun semakin banyak data semakin bertambah waktu tetapi penambahan waktu yang dibutuhkan masih relatif kecil.



Gambar 5. Grafik Peningkatan Waktu Proses Algoritma Boyer-Moore

(Sumber: Hasil Pengujian Peneliti, 2020)

### 4. KESIMPULAN

Penelitian ini menghasilkan sistem yang dapat menentukan tingkat kemiripan usulan judul

Skripsi/TA mahasiswa dengan *string matching* menggunakan algoritma Boyer-Moore yang menjalankan pencocokan yang dimulai dari karakter sebelah kanan kemudian ke kiri berdasarkan nilai dari *occurrence heuristic* (HO) dan *match heuristic* (MH) dalam menentukan pegeseran. Dari hasil pengujian waktu proses dengan memasukan 100 data sampai dengan 500 data menunjukkan bahwa algoritma Boyer-Moore mampu mencocokkan dengan cepat. Walaupun semakin banyak data semakin bertambah waktu yang dibutuhkan akan tetapi rata-rata peningkatan waktu hanya bertambah 34%. Untuk pengembangan selanjutnya dapat menerapkan beberapa algoritma *string matching* pada sistem, sehingga dapat diperoleh algoritma yang paling efisien.

### PERNYATAAN PENGHARGAAN

Peneliti mengucapkan terima kasih kepada “Yayasan Pendidikan Teknokrat melalui skim penelitian internal dan LPPM Universitas Teknokrat Indonesia” atas dukungan terhadap penelitian ini.

### DAFTAR PUSTAKA

- [1] W. Astuti, “Analisis String Matching Pada Judul Skripsi Dengan Algoritma Knuth-Morris Pratt (KMP),” *Ilk. J. Ilm.*, vol. 9, no. 2, pp. 167–172, 2017.
- [2] S. Hadi, “Pemeriksaan Keabsahan Data Penelitian Kualitatif Pada Skripsi,” *J. Ilmu Pendidik. Univ. Negeri Malang*, vol. 22, no. 1, pp. 74–79, 2016.
- [3] Z. A. Matondang, “Implementasi Algoritma String Matching Pencarian Kata Dari Makna Rambu Lalulintas Berbasis Android,” *J. Sist. Inf. Kaputama*, vol. 2, no. 1, pp. 101–106, 2018.
- [4] F. T. Waruwu and R. Mandala, “Perbandingan Algoritma Knuth Morris Pratt dan Boyer Moore Dalam Pencocokan String Pada Aplikasi Kamus Bahasa Nias,” *J. Ilm. INFOTEK*, vol. 1, no. 1, pp. 36–43, 2016.
- [5] R. I. Borman and A. Pratama, “Penerapan String Matching Dengan Algoritma Boyer Moore Pada Aplikasi Font Italic Untuk Deteksi Kata Asing,” *J. Teknoinfo*, vol. 10, no. 2, pp. 1–5, 2016.
- [6] E. Rahmanita, “Pencarian String Menggunakan Algoritma Boyer Moore Pada Dokumen,” *J. Ilm. NERO*, vol. 1,

- no. 1, pp. 15–26, 2014.
- [7] R. I. Darmawan, A. H. Setianingrum, and A. Arini, “Implementasi Algoritma Boyer Moore Pada Aplikasi Kamus Istilah Kebidanan Berbasis Web,” *QUERY J. Sist. Inf.*, vol. 02, no. 01, pp. 53–62, 2018.
- [8] A. Fau, Mesran, and G. L. Ginting, “Analisa Perbandingan Boyer Moore Dan Knuth Morris Pratt Dalam Pencarian Judul Buku Menerapkan Metode Perbandingan Eksponensial ( Studi Kasus : Perpustakaan STMIK Budi Darma ),” *J. Times (Technology Informatics Comput. Syst.*, vol. 6, no. 1, pp. 12–22, 2017.
- [9] V. Sagita and M. I. Prasetiyowati, “Studi Perbandingan Implementasi Algoritma Boyer-Moore, Turbo Boyer-Moore, dan Tuned Boyer-Moore dalam Pencarian String,” *J. Ultim.*, vol. 5, no. 1, pp. 31–37, 2013.
- [10] Y. Ardi, D. Andreswari, and Y. Setiawan, “Rancang Bangun Aplikasi Kamus Istilah Kedokteran Dengan Menggunakan Algoritma Boyer-Moore Berbasis Android,” *J. Rekursif*, vol. 5, no. 3, pp. 346–359, 2017.
- [11] T. Gumelar, R. Astuti, and A. T. Sunarni, “Sistem Penjualan Online Dengan Metode Extreme Programming,” *J. Telemat.*, vol. 9, no. 2, pp. 87–90, 2017.
- [12] R. I. Borman, A. T. Priandika, and A. R. Edison, “Implementasi Metode Pengembangan Sistem Extreme Programming ( XP ) pada Aplikasi Investasi Peternakan Implementation of Extreme Programming ( XP ) System Development Method in Livestock Investment Application,” *JUSTIN (Jurnal Sist. dan Teknol. Informasi)*, vol. 8, no. 3, pp. 272–277, 2020.
- [13] I. G. N. Suryantara, *Merancang Aplikasi dengan Metodologi Extreme Programming - I Gusti Ngurah Suryantara, S.Kom., M.Kom - Google Buku*. Jakarta: PT. Elex Media Komputindo, 2017.
- [14] I. G. N. Suryantara, “Merancang Aplikasi dengan Metodologi Extreme Programming.” PT. Elex Media Komputindo, Jakarta, 2017.
- [15] L. Rusdiana, “Extreme Programming untuk rancang bangun aplikasi pengelolaan surat keterangan kependudukan,” *Regist. J. Ilm. Teknol. Sist. Inf.*, vol. 4, no. 1, p. 49, 2018.
- [16] M. Melinda, R. I. Borman, and E. R. Susanto, “Rancang Bangun Sistem Informasi Publik Berbasis Web (Studi Kasus : Desa Durian Kecamatan Padang Cermin Kabupaten Pesawaran),” *J. Tekno Kompak*, vol. 11, no. 1, p. 1, 2018.
- [17] A. E. Kumala, R. I. Borman, and P. Prasetyawan, “Sistem Informasi Monitoring Perkembangan Sapi Di Lokasi Uji Performance (Studi Kasus : Dinas Peternakan Dan Kesehatan Hewan Provinsi Lampung),” *J. Tekno Kompak*, vol. 12, no. 1, p. 5, 2018.
- [18] I. J. Sinaga, R. C. G. I. Kembaren, D. M. Br Bangun, and N. Marbun, “Penerapan Algoritma Boyer Moore Pada Aplikasi Filofofer Berbasis Website,” *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 3, no. 1, pp. 436–438, 2019.
- [19] I Ahmad, I Hermadi, Y Arkeman. Financial feasibility study of waste cooking oil utilization for biodiesel production using ANFIS. *TELKOMNIKA Indonesian Journal of Electrical Engineering* 13 (3), 546-554. 2015.
- [20] I Ahmad, H Sulistiani, H Saputra. The Application Of Fuzzy K-Nearest Neighbour Methods for A Student Graduation Rate. *Indonesian Journal of Artificial Intelligence and Data Mining*. Vol 1 No 1. 47-52. 2018.
- [21] I Ahmad, N Shodik, N Neneng. Sistem Rekomendasi Pemilihan Smartphone Snapdragon 636 Menggunakan Metode Simple Multi Attribute Rating Technique (SMART). *Jurnal Nasional Pendidikan Teknik Informatika* 7 (3), 219-227. 2018.