

Implementasi Algoritma Horspool pada Kamus Istilah Internet of Things Berbasis Android

Eferoni Ndruru

Universitas Budi Darma, Medan, Indonesia

Corresponding Email: ronindruru@gmail.com

Abstrak. Perkembangan teknologi *IoT* semakin pesat, namun pemahaman masyarakat umum terhadap istilah-istilah teknis dalam bidang tersebut masih terbatas, termasuk bagi mahasiswa teknik informatika, praktisi *IT*, dan peminat teknologi *IoT*. Untuk mengatasi hal ini, dikembangkan sebuah aplikasi kamus istilah *IoT* berbasis *Android* yang dilengkapi dengan fitur pencarian cepat. Dalam penelitian ini, diimplementasikan algoritma *Horspool* sebagai metode pencarian string untuk meningkatkan efisiensi pencarian istilah dalam kamus. Hasil pengujian menunjukkan bahwa algoritma *Horspool* mampu memberikan hasil pencarian berupa arti istilah *IoT* dengan cepat dan tepat. Selain itu, aplikasi ini dirancang dengan antarmuka yang sederhana dan mudah digunakan, sehingga dapat membantu pengguna, khususnya pemula, dalam memahami arti dari istilah-istilah yang sering digunakan dalam dunia *IoT*. Dengan demikian, aplikasi ini tidak hanya berfungsi sebagai referensi, tetapi juga sebagai media edukasi yang mendukung literasi digital di bidang teknologi.

Kata Kunci: Algoritma Horspool, String Matching, Kamus Istilah *IoT*, *Android*, Literasi Digital.

1. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi yang pesat telah membawa manusia ke era konektivitas tinggi, salah satunya melalui konsep *Internet of Things (IoT)*. *IoT* adalah paradigma teknologi di mana berbagai perangkat fisik seperti sensor, alat rumah tangga, kendaraan, hingga perangkat medis dapat saling terhubung dan bertukar data melalui *internet*. Teknologi ini mendorong otomatisasi dan efisiensi di berbagai sektor, termasuk industri, transportasi, pertanian, dan rumah tangga.

Namun, pemahaman masyarakat umum, termasuk bagi mahasiswa teknik informatika, praktisi *IT*, dan peminat teknologi *IoT*, terhadap istilah dan konsep *IoT* masih banyak yang terbatas. Hal ini disebabkan oleh keterbatasan referensi yang mudah dipahami. Sebagian besar sumber informasi masih berupa dokumentasi teknis atau artikel ilmiah berbahasa asing, yang sulit diakses oleh pengguna awam. Akibatnya, terjadi kesenjangan literasi digital yang membatasi pemanfaatan *IoT* secara optimal.

Untuk menjembatani masalah ini, diperlukan media pembelajaran yang praktis dan mudah diakses. Mengingat tingginya penggunaan perangkat *Android* di Indonesia, pengembangan aplikasi kamus istilah *IoT* berbasis *Android* menjadi solusi potensial. Aplikasi ini diharapkan menjadi referensi edukatif yang membantu pengguna memahami berbagai istilah *IoT*.

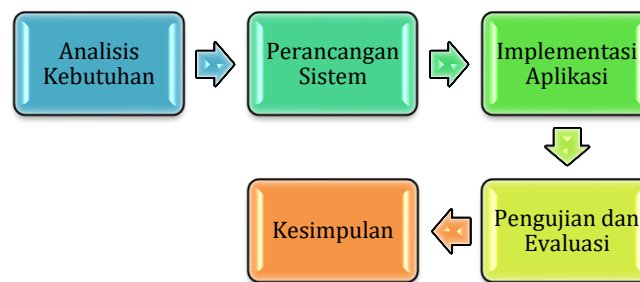
Namun, tantangan utama dalam pengembangan aplikasi kamus adalah efisiensi pencarian istilah, terutama ketika jumlah entri cukup banyak. Untuk itu, diperlukan implementasi algoritma pencarian *string* yang efisien. Salah satu algoritma yang relevan adalah *Horspool*, penyederhanaan dari *Boyer-Moore*, yang dikenal efektif dalam pencarian string tunggal dalam teks besar [1]. Dengan strategi pergeseran karakter, *Horspool* dapat meminimalkan jumlah perbandingan dan mempercepat proses pencarian [2], [3], [4].

Penelitian ini bertujuan mengimplementasikan algoritma *Horspool* dalam aplikasi kamus istilah *IoT* berbasis *Android*. Diharapkan, kombinasi antara kemudahan akses *mobile* dan efisiensi algoritma mampu memberikan pengalaman pengguna yang cepat dan nyaman. Selain itu, aplikasi ini diharapkan turut meningkatkan literasi digital dan pemahaman masyarakat umum terhadap teknologi *IoT* secara lebih luas.

2. METODOLOGI

2.1 Tahap Penelitian

Penelitian ini menggunakan metode *Research and Development* yang difokuskan pada proses perancangan, implementasi, dan pengujian aplikasi kamus istilah *Internet of Things (IoT)* berbasis *Android* dengan integrasi algoritma *Horspool* sebagai mesin pencari string. Metodologi ini dipilih untuk menghasilkan produk berupa aplikasi *mobile* yang tidak hanya fungsional, tetapi juga efisien dalam hal pencarian istilah teknis.



Gambar 1. Alur penelitian

Keterangan:

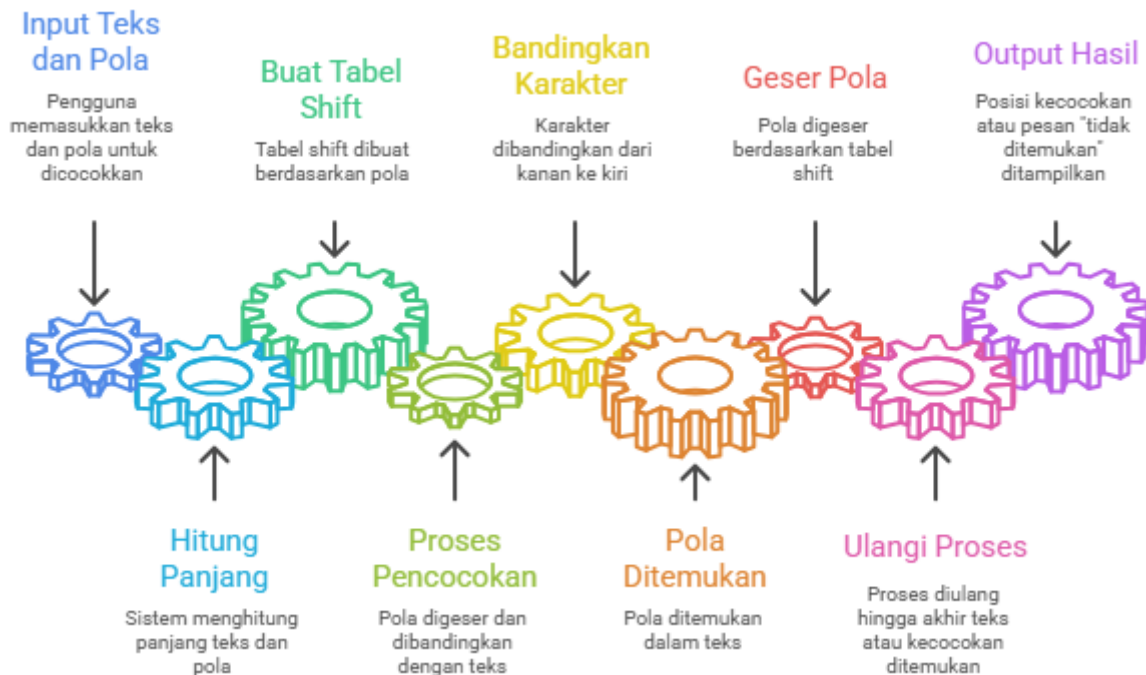
- Analisis Kebutuhan**
Pada tahap awal, dilakukan analisis kebutuhan pengguna terhadap kamus *IoT*, termasuk identifikasi istilah-istilah umum yang sering digunakan dalam pengembangan dan pemanfaatan teknologi *IoT*. Analisis ini dilakukan melalui studi literatur, pengumpulan referensi dari berbagai sumber terpercaya, serta observasi terhadap pengguna sasaran yang memiliki minat atau keterkaitan dengan teknologi *IoT*.
- Perancangan Sistem**
Setelah kebutuhan ditentukan, dilakukan perancangan aplikasi yang mencakup desain antarmuka pengguna, struktur basis data untuk menyimpan istilah dan definisi, serta perancangan algoritma pencarian. Desain antarmuka dirancang agar sederhana, intuitif, dan mudah digunakan. Pada tahap ini juga dilakukan perancangan bagaimana algoritma *Horspool* akan diintegrasikan dalam pencarian arti dari istilah *IoT* berdasarkan kata kunci yang dimasukkan oleh pengguna.
- Implementasi Aplikasi**
Aplikasi dikembangkan menggunakan *Android Studio* dengan bahasa pemrograman Java. *Database* istilah disimpan secara lokal menggunakan *SQLite*. Algoritma *Horspool* diimplementasikan untuk mengoptimalkan proses pencarian string pada daftar istilah. Proses pencarian dilakukan dengan membandingkan input pengguna dengan entri dalam *database* menggunakan mekanisme pergeseran karakter sesuai logika *Horspool*.
- Pengujian dan Evaluasi**
Setelah implementasi selesai, dilakukan proses pengujian aplikasi yang telah diimplementasikan, dengan metode black-box testing guna memastikan semua fitur pada aplikasi tersebut berfungsi sesuai spesifikasi atau tidak.
- Kesimpulan**
Di tahap akhir ini, dilakukan penarikan kesimpulan dari hasil penelitian ini berdasarkan hasil pengujian dan evaluasi yang telah dilakukan pada tahap sebelumnya.

2.2 String Matching

String Matching merupakan teknik untuk menemukan apakah suatu pola *string* tertentu terdapat dalam sebuah *string* yang lebih panjang, serta menentukan letak kemunculannya [5]. Algoritma ini sangat berguna dalam berbagai bidang, seperti pencarian teks dalam dokumen, mesin pencari di *internet*, koreksi otomatis ejaan, analisis urutan DNA, hingga deteksi *plagiarisme* [6], [7].

2.3 Horspool

Horspool merupakan pengembangan dari algoritma *Boyer-Moore* yang melakukan pencocokan *string* dengan memulai perbandingan dari sisi kanan pola terhadap *text* [8]. Algoritma ini diperkenalkan oleh R. Nigel Horspool pada tahun 1980, untuk menangani ketidaksesuaian antara karakter dalam pola dan karakter dalam teks [1]. Algoritma ini menerapkan metode *bad match* secara berulang hingga seluruh karakter dalam pola berhasil dicocokkan dengan teks. Proses pencocokan *string* pada algoritma ini terdiri dari dua tahap utama, yaitu tahap *pre-processing* dan tahap *searching* [9], [10].



Gambar 2. Langkah-langkah Algoritma *Horspool*

3. HASIL DAN DISKUSI

3.1 Hasil

3.1.1 Analisis Kebutuhan

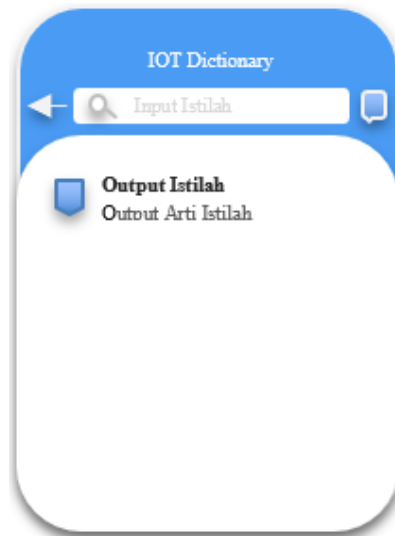
Tahap awal penelitian difokuskan pada identifikasi kebutuhan pengguna terhadap aplikasi kamus istilah *Internet of Things (IoT)*. Melalui studi literatur dari berbagai jurnal, buku, serta dokumentasi teknis, dikumpulkan lebih dari 150 istilah yang umum digunakan dalam pengembangan *IoT*, seperti sensor *node*, *gateway*, *MQTT*, *firmware*, *vulnerability*, dan lain sebagainya. Observasi juga dilakukan terhadap kelompok pengguna sasaran, yang terdiri dari mahasiswa teknik informatika, praktisi IT, dan peminat teknologi *IoT*. Hasil observasi menunjukkan bahwa sebagian besar dari mereka mengalami kesulitan dalam memahami istilah-istilah teknis *IoT*, terutama yang berasal dari literatur berbahasa Inggris. Dari hasil ini disimpulkan bahwa terdapat kebutuhan akan sebuah aplikasi kamus yang ringkas, praktis, dan memiliki fitur pencarian yang cepat.

3.1.2 Perancangan Sistem

Berdasarkan hasil analisis kebutuhan, sistem dirancang dengan tiga komponen utama:

a. Antarmuka Pengguna

Dirancang secara sederhana dan intuitif dengan mempertimbangkan prinsip *user-centered design*. Halaman utama terdiri dari kolom pencarian istilah, *output* istilah, dan *output* arti istilah, dan tombol *back*.



Gambar 3. Desain Antarmuka Pengguna

b. Basis Data

Disusun menggunakan SQLite dengan struktur tabel yang memuat kolom id, istilah, dan arti istilah. Hal ini memungkinkan pencarian cepat tanpa ketergantungan koneksi internet.

Tabel 1. Basis Data

Nama	Type Data	Keterangan
Id	Integer	Primary key, auto-increment
istilah	Text	Istilah yang dicari
arti_istilah	Text	Arti atau penjelasan istilah

c. Algoritma Pencarian

Implementasi algoritma *Horspool* dalam penelitian ini dilakukan berdasarkan langkah-langkah yang telah dipaparkan pada Gambar 2. Studi kasus yang diselesaikan menggunakan implementasi algoritma *Horspool* yaitu pada pemecahan masalah dalam pencarian istilah *IoT*, kata BILITY digunakan sebagai pola yang akan dicocokkan dengan VULNERABILITY sebagai teks.

1. Tahap *Pre-Processing*

Pada tahap *pre-processing*, dilakukan proses penentuan nilai *bad match* berdasarkan data awal pada teks dan pola. Pada data awal diketahui panjang teks adalah 13 dan panjang pola adalah 6. Sehingga berdasarkan data tersebut dapat diketahui nilai pergeseran seperti yang ditunjukkan pada Tabel 2 berikut ini.

Tabel 2. Bad Match

Karakter Pola	Posisi dalam Pola	Pergeseran ($m - 1 - \text{posisi}$)
B	0	$6 - 1 - 0 = 5$
I	1	$6 - 1 - 1 = 4$
L	2	$6 - 1 - 2 = 3$
I	3	$6 - 1 - 3 = 2$ (nilai terbaru untuk I)
T	4	$6 - 1 - 4 = 1$
Y	(karakter terakhir, tidak dihitung)	6 (default untuk semua karakter yang tidak ada di pola)

Tabel 3. Pergeseran Akhir

Karakter	Pergeseran
B	5
I	2
L	3
T	1
(lainnya)	6

2. Tahap *Searching*

Pada tahap *searching*, proses pencocokan *string* dilakukan berdasarkan ketentuan yang telah dibuat pada Tabel 2.

a) Pencocokan *String* Ke-1

Pada tahap pertama, dilakukan pencocokan pola “Y” dengan “R” pada teks. Hasil dari pencocokan menunjukkan bahwa “Y” dengan “R” tidak memiliki kecocokan dan “R” tidak ada di dalam pola, sehingga dilakukan pergeseran pola sebanyak nilai *bad match* “R”, yaitu 6 langkah.

Tabel 4. Pecocokan String Ke-1

Indeks	0	1	2	3	4	5	6	7	8	9	10	11	12
Teks	V	U	L	N	E	R	A	B	I	L	I	T	Y
Pola	B	I	L	I	T	Y							

b) Pencocokan *String* Ke-2

Pada tahap kedua, dilakukan pencocokan pola “Y” dengan “T” pada teks. Hasil dari pencocokan menunjukkan bahwa “Y” dengan “T” tidak memiliki kecocokan dan “T” ada di dalam pola, sehingga dilakukan pergeseran pola sebanyak nilai *bad match* “T”, yaitu 1 langkah.

Tabel 5. Pecocokan String Ke-2

Indeks	0	1	2	3	4	5	6	7	8	9	10	11	12
Teks	V	U	L	N	E	R	A	B	I	L	I	T	Y
Pola							B	I	L	I	T	Y	

c) Pencocokan *String* Ke-3

Pada tahap ketiga, dilakukan pencocokan pola “Y” dengan “Y” pada teks. Hasil dari pencocokan menunjukkan bahwa “Y” dengan “Y” tidak memiliki kecocokan. Selanjutnya dilakukan pencocokan pada karakter selanjutnya dari kanan ke kiri. Pada Tabel 6 ditunjukkan bahwa seluruh karakter pada pola telah memiliki kecocokan atau ditemukan di dalam teks. Sehingga proses pencocokan *string* diberhentikan pada tahap yang ketiga ini.

Tabel 6. Pecocokan String Ke-3

Indeks	0	1	2	3	4	5	6	7	8	9	10	11	12
Teks	V	U	L	N	E	R	A	B	I	L	I	T	Y
Pola								B	I	L	I	T	Y

3.1.3 Implementasi Aplikasi

Hasil pengembangan aplikasi kamus istilah Internet of Things berbasis android dengan penerapan algoritma dalam *Horspool* penelitian ini dapat dilihat pada gambar di bawah ini.



Gambar 4. Antarmuka Aplikasi Kamus Istilah Internet of Things Berbasis Android

Gambar 4 di atas menunjukkan bahwa proses pencarian dilakukan dengan membandingkan *input* data dari pengguna dengan entri dalam *database*, dan pada *output* akhir diberikan penjelasan terkait dengan arti istilah *IoT* yang dicari oleh pengguna.

3.1.4 Pengujian dan Evaluasi

Pengujian aplikasi kamus istilah *IoT* dilakukan menggunakan metode *black-box testing* untuk memastikan seluruh fitur berjalan sesuai dengan spesifikasi fungsional yang telah dirancang. Pengujian difokuskan pada validasi *input* dan *output* tanpa memperhatikan proses internal aplikasi. Berdasarkan hasil pengujian yang telah dilakukan, seluruh fungsi utama aplikasi telah berjalan sesuai dengan spesifikasi yang dirancang. Proses pencarian istilah menunjukkan hasil yang akurat dan cepat, sesuai dengan implementasi algoritma *Horspool* yang terbukti efisien dalam pencocokan *string*. Selain itu, antarmuka aplikasi telah memenuhi prinsip *user-centered design*, dengan tampilan sederhana dan navigasi yang mudah dipahami oleh pengguna. Pengujian juga menunjukkan bahwa sistem dapat menangani berbagai kondisi input, termasuk input kosong dan istilah yang tidak terdaftar, dengan memberikan pesan kesalahan yang informatif.

3.2 Diskusi

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa pengembangan aplikasi kamus istilah *IoT* berhasil memenuhi kebutuhan pengguna dalam memahami terminologi teknis yang sering muncul dalam bidang *IoT*. Hasil analisis kebutuhan menunjukkan adanya kesenjangan pengetahuan pengguna terhadap istilah-istilah *IoT* berbahasa Inggris, sehingga pengembangan aplikasi ini relevan dan tepat sasaran. Dari sisi perancangan, penerapan prinsip *user-centered design* pada antarmuka aplikasi berkontribusi signifikan terhadap kemudahan penggunaan. Tampilan yang sederhana dengan fokus pada kolom pencarian istilah membuat pengguna dapat dengan cepat menemukan arti istilah yang diinginkan. Desain ini terbukti efektif berdasarkan hasil uji coba yang menunjukkan kemudahan navigasi bagi pengguna dari berbagai latar belakang, baik mahasiswa maupun praktisi IT. Dari aspek teknis, implementasi algoritma *Horspool* memberikan dampak positif terhadap efisiensi proses pencarian istilah dalam database. Hasil pencocokan *string* pada studi kasus menunjukkan bahwa algoritma ini mampu menemukan pola dengan pergeseran minimum dan waktu pencarian yang lebih cepat dibandingkan metode pencarian sederhana. Hal ini sesuai dengan teori bahwa *Horspool* merupakan algoritma pencocokan pola yang efisien karena melakukan pergeseran berdasarkan karakter terakhir dari pola. Sementara itu, hasil pengujian menggunakan metode *black-box testing* memperlihatkan bahwa seluruh fungsi utama berjalan sesuai spesifikasi. Fitur pencarian istilah memberikan hasil yang akurat dan cepat, serta mampu menangani berbagai kondisi input dengan baik. Pesan kesalahan yang informatif pada input kosong atau istilah yang tidak terdaftar menunjukkan bahwa aplikasi telah dirancang dengan mempertimbangkan aspek keandalan dan pengalaman pengguna. Secara keseluruhan, aplikasi kamus istilah *IoT* yang dikembangkan tidak hanya memenuhi kebutuhan fungsional pengguna, tetapi juga menunjukkan efisiensi algoritma dalam implementasi nyata. Kombinasi antara desain antarmuka yang ramah pengguna, penggunaan *database SQLite* yang ringan, serta penerapan algoritma *Horspool* yang optimal menjadikan aplikasi ini sebagai solusi efektif untuk membantu pengguna memahami istilah-istilah *IoT* secara cepat dan praktis.

4. KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa aplikasi kamus istilah *IoT* berbasis *Android* berhasil dikembangkan sesuai kebutuhan pengguna dan spesifikasi yang dirancang. Penerapan algoritma *Horspool* terbukti efisien dalam proses pencarian istilah dengan waktu pencocokan yang cepat dan hasil yang akurat. Antarmuka yang sederhana dan berbasis *user-centered design* memudahkan pengguna dalam melakukan pencarian istilah *IoT*, sementara penggunaan *database SQLite* memastikan akses data yang ringan dan stabil tanpa koneksi internet. Hasil pengujian *black-box* juga menunjukkan bahwa seluruh fitur aplikasi berfungsi dengan baik dan mampu menangani berbagai kondisi input dengan *respons* yang sesuai. Dengan demikian, aplikasi ini dapat menjadi solusi efektif untuk membantu mahasiswa, praktisi IT, dan peminat *IoT* dalam memahami istilah-istilah teknis secara cepat dan praktis.

Referensi

- [1] A. Azhar, N. Marbun, S. Aripin, and E. Buulolo, "Implementasi Algoritma Horspool Pada Aplikasi Istilah Fashion," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 3, no. 1, pp. 549–551, 2019, doi: 10.30865/komik.v3i1.1641.
- [2] Z. Abidin, M. S. Mustafa, and H. SY, "Penerapan Horspool Algorithm pada Aplikasi Translate Indonesia ke Bahasa Daerah Berbasis Android: Studi Kasus 29 Bahasa," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 1, pp. 26–36, Dec. 2023, doi: 10.57152/malcom.v4i1.1000.
- [3] M. H. Rifqo and W. A. Kartika, "Android-Based Information And Internet Dictionary Applications Using The Horspool Algorithm," *J. Komputer, Inf. dan Teknol.*, vol. 2, no. 2, Dec. 2022, doi: 10.53697/jkomitek.v2i2.861.
- [4] R. Rohili and E. S. Budi, "Sistem Informasi Penjualan Obat Berbasis Web pada Apotek Diana Menggunakan Algoritma Horspool," *J. Sist. Komput. dan Inform.*, vol. 3, no. 4, p. 536, 2022, doi: 10.30865/json.v3i4.4240.
- [5] D. Permana, K. Komarudin, F. I. Abdurahman, and I. Ramlan, "Implementation Of the Boyer–Moore Model in The Guidance System for Field Work Practice Activities," *Electron. Business, Manag. Technol. J.*, vol. 3, no. 1, pp. 1–11, Aug. 2025, doi: 10.55208/5dpsx553.
- [6] A. Aman Bangsa, B. Pramono, and L. Bahtiar Aksara, "Penerapan String Matching Menggunakan Algoritma Boyer Moore untuk Mencari Data Pada Website UMKM di Konawe Selatan," *Animator*, vol. 2, no. 1, pp. 18–25, 2024, [Online]. Available: <https://animator.uho.ac.id/index.php/journal/article/view/41>
- [7] S. S. Abdul-Jabbar, A. K. Farhan, A. A. Abdelhamid, and M. E. Ghoneim, "Razy: A String Matching Algorithm for Automatic Analysis of Pathological Reports," *Axioms*, vol. 11, no. 10, pp. 1–15, 2022, doi: 10.3390/axioms11100547.
- [8] G. W. Ramadhan, "Implementasi Algoritma Horspool Pada Aplikasi Penelusuran Buku Dipertustakaan Universitas Malikussaleh Berbasis Android," *TECHSI - J. Tek. Inform.*, vol. 15, no. 2, pp. 40–60, Oct. 2024, doi: 10.29103/techsi.v15i2.18914.
- [9] T. Raita, "Tuning the boyer-moore-horspool string searching algorithm," *Softw. Pract. Exp.*, vol. 22, no. 10, pp. 879–884, Oct. 1992, doi: 10.1002/spe.4380221006.
- [10] S. Maesyaroh, G. P. Suwanto, F. I. Komputer, U. Kuningan, P. Kecamatan, and K. Kabupaten, "pencocokan teks; mahasiswa; algoritma horspool; algoritma raita;," vol. 9, pp. 1–9, 2023.