

# Perbandingan Algoritma Brute Force Dengan Boyer-Moore Pada Aplikasi Pencarian Kerja Berbasis Web

Fahmy Fauzi Alvianda\*<sup>1</sup>, Ruuhwan<sup>2</sup>, Yusuf Sumaryana<sup>3</sup>

<sup>1,2,3</sup>Universitas Perjuangan; Jl. Peta No.177, (0265) 326058

<sup>123</sup>Teknik Informatika, Fakultas Teknik, Universitas Perjuangan

e-mail: \*<sup>1</sup>1903010074@unper.ac.id, <sup>2</sup>ruuhwan@unper.ac.id, <sup>3</sup>yusufsumaryana@unper.ac.id

## ABSTRAK

Pencarian lowongan kerja pada situs web pencari kerja berbasis web seperti Didamel.id memerlukan algoritma pencarian yang efektif dan efisien untuk memberikan pengalaman pengguna yang baik. Dalam pengembangan aplikasi pencarian pekerjaan berbasis web, pemilihan algoritma pencarian yang efisien sangat krusial untuk memberikan pengalaman pengguna yang baik. Dua algoritma yang sering digunakan dalam pencarian string adalah algoritma Brute Force dan algoritma Boyer-Moore. Penelitian ini bertujuan untuk membandingkan kinerja algoritma Brute Force dan algoritma Boyer-Moore pada aplikasi pencarian pekerjaan berbasis web. Metode Brute Force melakukan pencocokan karakter per karakter secara berurutan, sementara algoritma Boyer-Moore menggunakan strategi pra-proses untuk mempercepat pencarian dengan memanfaatkan pola pencarian yang terdapat dalam teks. Hasil penelitian menunjukkan bahwa algoritma Boyer-Moore memberikan kinerja yang lebih baik dari pada algoritma Brute Force pada aplikasi pencarian pekerjaan berbasis web. Algoritma Boyer-Moore mengurangi jumlah perbandingan yang perlu dilakukan, sehingga menghemat waktu eksekusi dan sumber daya komputasi. Selain itu, algoritma ini memiliki kompleksitas waktu yang lebih rendah pada pola pencarian yang memiliki kesamaan dan ukuran pola yang besar.

**Kata Kunci:** *algorithm, boyer-moore, bruteforce.*

## ABSTRACT

Searching for job vacancies on a web-based job search website such as Didamel.id requires an effective and efficient search algorithm to provide a good user experience. In the development of web-based job search applications, the selection of efficient search algorithms is crucial to provide a good user experience. Two algorithms that are often used in string search are Brute Force algorithm and Boyer-Moore algorithm. This research aims to compare the performance of Brute Force algorithm and Boyer-Moore algorithm in web-based job search application. The Brute Force method performs character-by-character matching sequentially, while the Boyer-Moore algorithm uses a preprocessing strategy to speed up the search by utilizing search patterns contained in the text. The results show that the Boyer-Moore algorithm provides better performance than the Brute Force algorithm on web-based job search applications. Boyer-Moore algorithm reduces the number of comparisons that need to be performed, thus saving execution time and computational resources. In addition, this algorithm has lower time complexity on search patterns that have similarities and large pattern sizes.

**Keywords:** *algorithm, boyer-moore, brute force*

## 1. PENDAHULUAN

Pencarian informasi lowongan pekerjaan secara online dapat menjaditugas yang sangat kompleks dan membutuhkan algoritma yang efektif dan efisien untuk menghasilkan hasil pencarian yang akurat dan cepat [1]. Dalam hal ini, algoritma brute force dan algoritma boyer-moore merupakan dua algoritma yang sering digunakan dalam pencarian string, termasuk dalam pencarian informasi lowongan pekerjaan pada website pencari kerja.

Algoritma brute force adalah algoritma yang digunakan untuk mencocokkan pattern dengan semua teks antara 0 dan n-m untuk menemukan keberadaan pattern teks. Algoritma brute force memecahkan masalah dengan sangat sederhana, langsung, dan jelas [2]. Meskipun algoritma ini mudah dipahami dan diterapkan, namun kelemahan dari algoritma ini adalah kinerjanya yang sangat lambat saat diterapkan pada data yang besar [3]. Penelitian sebelumnya dilakukan oleh Ramadhoni 2022, tentang penerapan algoritma brute force pada aplikasi sidayko berbasis android, aplikasi ini merupakan sistem informasi penjualan ikan koi yang menerapkan metode brute force dalam mencari produk ikan koi dengan menggunakan bahasa pemrograman java [4].

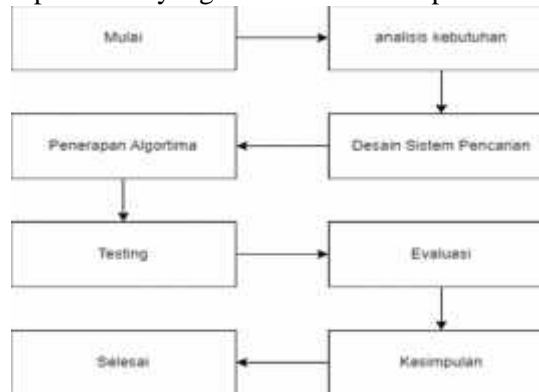
Sementara itu, algoritma boyer-moore merupakan metode pencarian string yang dikembangkan oleh Robert S. Boyer dan J. Stroher Moore pada tahun 1977. Algoritma ini melakukan pencarian dengan pencocokan karakter yang dimulai dari sebelah kanan pattern (inputan kata kunci) ke sebelah kiri sehingga dapat mempercepat pencarian serta mendapatkan informasi lebih banyak, proses pencarian akan lebih cepat dengan menerapkan algoritma boyer moore ke dalam sistem ini dibandingkan proses pencarian lainnya [5]. Hal ini membuat algoritma boyer-moore lebih unggul dari pada algoritma brute force dalam pencarian string pada data yang besar. Penelitian yang dilakukan oleh Muhammad Affif Setiawan 2021, tentang penerapan algoritma boyer-moore dalam pencarian barang hilang pada aplikasi, merupakan aplikasi yang dapat menjadi media informasi barang hilang bagi masyarakat, Aplikasi ini dirancang berbasis android menggunakan Firebase Authentication dan Firebase Storage sebagai database utama, pengujian algoritma dilakukan antara Knuth Morris Pratt dan Boyer Moore, dan hasilnya adalah boyer –moore lebih cepat dalam melakukan pencocokan string [6].

Dalam penelitian ini, penulis menerapkan dua algoritma berbeda yaitu algoritma brute force dan algoritma boyer-moore pada search engine pencarian lowongan kerja yang akan membuat dalam proses pencarian berbeda, sehingga dapat dijadikan acuan algoritma mana yang paling cepat dan tepat dalam pencarian informasi [7].

---

## 2. METODE PENELITIAN

Adapun metode dalam penelitian yang akan dilakukan seperti dibawah ini:



Gambar 1 Metodologi Penelitian

### 2.1 Tahapan Analisi Kebutuhan

Pada tahapan ini dilakukan sebuah pengumpulan data untuk mendapatkan data yang benar-benar valid dengan cara observasi, wawancara, dan studi Pustaka.

### 2.2 Tahapan Desain Sistem Pencarian

Pada tahapan ini merancang desain sistem pencarian lowongan kerja pada fitur pencarian yang akan menerapkan algoritma untuk mempermudah pengguna dalam mencari lowongan pekerjaan dengan cepat dan benar.

### 2.3 Tahapan Penerapan Algoritma

Setelah proses desain sistem dilakukan maka penulis melakukan penulisan kode program pada fitur pencarian lowongan pekerjaan dengan bahasa pemrograman PHP melalui lingkungan Framework Laravel.

### 2.4 Tahapan Testing

Pada tahap ini merupakan tahapan pengujian, perbandingan dari penerapan kedua algoritma yang telah dibangun mana yang lebih baik.

### 2.5 Tahapan Evaluasi

Pada tahap ini hasil yang didapat dari pengujian kemudian dijadikan bahan evaluasi dilaporkan kepada pihak pengelola atau institusi pemilik web.

## 3. HASIL DAN PEMBAHASAN

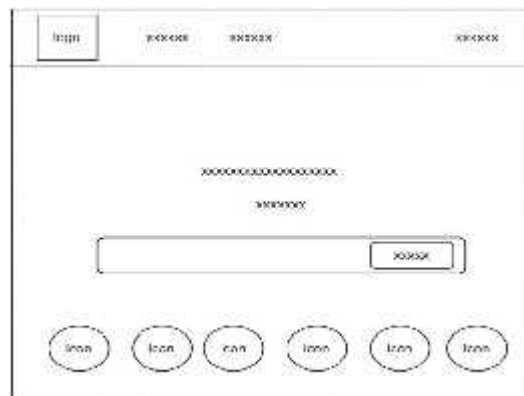
### 3.1 AnalisisKebutuhan

Setelah dilakukannya observasi, wawancara dengan pihak-pihak yang bersangkutan[8] di perusahaan PT. Cantara Digital Innovation dan studi pustaka atau mengumpulkan data daribuku dan sumber lainnya, maka didapatkan informasi berhubungan dengan perbandingan penerapan algoritma brute force denganboyer-moore pada pencarian.

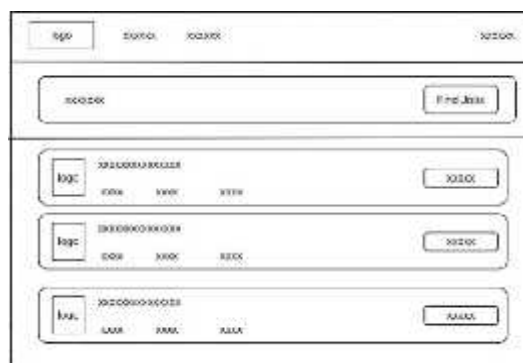
### 3.2Desain SistemPencarian

Merancang desain sistem pencarian lowongan kerja pada fitur pencarian yang akan menerapkan algoritma untuk mempermudah pengguna dalam mencari lowongan pekerjaan dengan cepat dan benar. Berikut perancangan antar muka:

1. Gambar 2 adalah wireframe halaman awal saat aplikasi pertama kali diakses.
2. Gambar 3 adalah wireframe halaman pencarian.
3. Gambar 4 adalah tampilan halaman awal saat aplikasi pertama kali diakses.
4. Gambar 5 adalah halaman pencarian.



Gambar 2 Rancangan tampilan home



Gambar 3Rancangantampilanpencarian



Gambar 4 Halaman tampilan home



Gambar 5 Halaman tampilan pencarian

### 3.3 Penerapan Algoritma

Tahap implementasi merupakan suatu proses dari hasil perancangan yang sudah dimasukkan kode program dan dilakukan implementasi [9] algoritma brute force dan algoritma boyer-moore yang dapat digunakan sebagai pencarian judul pekerjaan. Berikut penerapan algoritma brute force dan algoritma boyer-moore:

#### 3.3.1 Penerapan Algoritma brute force

Berikut potongan kode dalam proses pencarian menggunakan brute force:

```
// Mengambil waktu awal pencarian
$startTime = microtime(true);

// mendapatkan semua lowongan kerja dari database dengan relasi
$jobs = Jobs::with(['category', 'company', 'author'])
->latest()->get();

$filteredJobs = [];

// Loop through each job
foreach ($jobs as $job) {
    // mengubah huruf menjadi kecil
    $title = strtolower($job->title);

    // Lakukan pencarian secara brute force
    for ($i = 0; $i <= strlen($title) - strlen($search);
        $i++) {
        $j = 0;
```

```

        // Cocokkankarakter pada poladengankarakter
        pada teks
        while ($j <strlen($search) && $title[$i + $j]
        == $search[$j]) {
            $j++;
        }

        // Jika seluruhkarakter pada
        polacocokdengankarakter pada teks,
        makaditemukanpola pada teks.
        if ($j == strlen($search)) {
            $filteredJobs[] = $job->id;
            break;
        }
    }
}

// Get jobs based on filtered jobs
$jobs = Jobs::whereIn('id', $filteredJobs)->latest()-
>get();

// Mengambilwaktuakhirpencarian
$endTime = microtime(true);

// Menghitungselisihwaktupencariandalammilidetik
$executionTime = ($endTime - $startTime) * 1000;

```

Gambar 7 merupakan hasil dari pencarian menggunakan algoritma brute force



Gambar 6 Hasil pencarian menggunakan algoritma brute force

### 3.3.2 Penerapan Algoritma Boyer-Moore

Berikut potongan kode dalam proses pencarian menggunakan *boyer-moore*:

```

// Mengambilwaktuawalpencarian
$startTime = microtime(true);

```

---

```

// mendapatkansemualowongankerjadari database denganrelasi

```

```
$jobs = Jobs::with(['category', 'company', 'author'])-
>latest()->get();
$filteredJobs = [];
$n = strlen($search);

// Inisialisasitabel bad character dan good suffix
$badChar = [];
$goodSuffix = [];
for ($i = 0; $i < $n; $i++) {
    $badChar[$search[$i]] = $i;
    $goodSuffix[$i] = -1;
}

// Membuattabel good suffix
$j = 0;
for ($i = $n - 1; $i >= 0; $i--) {
    if ($i + 1 == $n) continue;
    while ($j != -1 && $search[$i] != $search[$j]) {
        $j = $goodSuffix[$j];
    }
    $j++;
    if ($search[$i] == $search[$j - 1]) {
        $goodSuffix[$i] = $j - 1;
    } else {
        $goodSuffix[$i] = $j;
    }
}

// Loop through each job
foreach ($jobs as $job) {
    // mengubahhurufmenjadikecil
    $title = strtolower($job->title);
    $m = strlen($title);

    // Lakukanpencariandarikanankekiri pada teks,
    dimulaidariindekske-n (panjangpola).
    $i = $n - 1;
    while ($i < $m) {
        $k = 0;

        // Cocokkankarakter pada poladengankarakter pada
        teksdarikanankekiri.
        while ($k < $n && $title[$i - $k] == $search[$n - 1
        - $k]) {
            $k++;
        }
    }
}
```

```

// Jika seluruh karakter pada pola cocok dengan karakter
// pada teks, maka ditemukan pola pada teks.
if ($k == $n) {
    $filteredJobs[] = $job->id;
    break;
} else {
    // Menggunakan tabel bad character dan good
    // suffix untuk menghitung pergeseran pola ke kanan
    $charIndex = isset($badChar[$title[$i - $k]])
    ? $badChar[$title[$i - $k]] : -1;
    $i += max(1, $k - $charIndex, $k -
    $goodSuffix[$n - 1 - $k]);
}
}
}

// Get jobs based on filtered jobs
$jobs = Jobs::whereIn('id', $filteredJobs)->latest()-
>get();

// Mengambil waktu akhir pencarian
$endTime = microtime(true);

// Menghitung selisih waktu pencarian dalam milidetik
$executionTime = ($endTime - $startTime) * 1000;

```

Gambar 9 merupakan hasil dari pencarian menggunakan algoritma brute force



Gambar 7 Hasil pencarian menggunakan algoritma *boyer-moore*

### 3.4 Testing

Setelah melalui tahap penerapan algoritma, dilakukan pengujian pada sistem. Pengujian ini dilakukan dengan tujuan membandingkan algoritma [10], cara kegunaan serta hasilnya sesuai dengan yang diharapkan. Berikut pengujian kedua algoritma:



### 3.4.1 Pengujian I

Membandingkan berapa lama waktu pencarian satu karakter sampai lima karakter dan output yang dihasilkan dari metode brute force dan boyer-moore.

Tabel1 Pengujian I

Pattern	Banyak data ditemukan		Lama pencarian	
	Brute force	Boyer-Moore	Brute force	Boyer-Moore
s	1227	1227	165.26103019714	171.57983779907
sa	193	193	138.42511177063	144.28400993347
sal	148	148	108.03508758545	132.65585899353
sale	148	148	134.03081893921	123.49414825439
sales	148	148	112.40792274475	112.6880645752
<b>Jumlah</b>	<b>1864</b>	<b>1864</b>	<b>65.815.997.123.718</b>	<b>58.328.266.143.798</b>

### 3.4.2 Pengujian II

Membandingkan berapa lama waktu pencarian enam karakter sampai 10 karakter dan output yang dihasilkan dari metode brute force dan boyer-moore.

Tabel2 Pengujian II

Pattern	Banyak data ditemukan		Lama pencarian	
	Brute force	Boyer-Moore	Brute force	Boyer-Moore
develo	135	135	137.95304298401	109.75885391235
develop	135	135	137.85409927368	124.21011924744
developm	72	72	120.88704109192	130.1691532135
developme	72	72	114.76397514343	118.75414848328
developmen	72	72	132.61795043945	128.31211090088
<b>Jumlah</b>	<b>486</b>	<b>486</b>	<b>64.407.610.893.249</b>	<b>49.405.214.786.530</b>

### 3.4.3 Pengujian III

Membandingkan berapa lama waktu pencarian 11 karakter sampai 15 karakter dan output yang dihasilkan dari metode brute force dan boyer-moore.

Tabel3 Pengujian III

Pattern	Banyak data ditemukan		Lama pencarian	
	Brute Force	Boyer-Moore	Brute force	Boyer-Moore
creditadm	7	7	116.11294746399	111.34600639343
creditadmi	7	7	106.97102546692	116.76287651062
creditadmin	7	7	133.84413719177	139.56594467163
creditadmini	7	7	113.94309997559	130.2649974823
creditadminis	7	7	121.51694297791	110.759973526
<b>Jumlah</b>	<b>35</b>	<b>35</b>	<b>59.238.815.307.618</b>	<b>38.180.892.705.917</b>

### 3.4.4 Pengujian IV

Membandingkan berapa lama waktu pencarian 16 karaktersampai 20 karakter dan output yang dihasilkan dari metode brute force dan boyer-moore.

Tabel 4P engujian IV

Pattern	Banyak data ditemukan		Lama pencarian	
	Brute Force	Boyer-Moore	Brute force	Boyer-Moore
Business Develop	49	49	122.08795547485	122.84016609192
Business Developm	49	49	132.26413726807	124.14002418518
Business Developme	49	49	126.57284736633	119.47178840637
Business Developmen	49	49	121.09208106995	126.77097320557
Business Development	49	49	134.11998748779	135.14018058777
<b>Jumlah</b>	<b>245</b>	<b>245</b>	<b>63.613.700.866.699</b>	<b>62.836.313.247.681</b>

### 3.4.5 Pengujian V

Membandingkan berapa lama waktu pencarian pattern satu kata dan output yang dihasilkan dari metode brute force dan boyer-moore.

Tabel 5 Pengujian V

Pattern	Banyakdata ditemukan		Lamapencarian	
	Brute force	Boyer-Moore	Brute force	Boyer-Moore
it	234	234	141.28184318542	135.01596450806
finance	41	41	142.44389533997	127.61807441711
marketing	109	109	141.89910888672	132.80987739563
admin	78	78	137.13002204895	138.84592056274
accounting	69	69	149.52111244202	135.69808006287
<b>Jumlah</b>	<b>531</b>	<b>531</b>	<b>71.227.598.190.308</b>	<b>66.998.791.694.641</b>

### 3.4.6 Pengujian VI

Membandingkan berapa lama waktu pencarian pattern dua kata dan output yang dihasilkan dari metode brute force dan boyer-moore.

Tabel 6 Pengujian IV

Pattern	Banyakdata ditemukan		Lamapencarian	
	Brute force	Boyer-Moore	Brute force	Boyer-Moore
account manager	15	15	113.94500732422	118.9649105072
front end	11	11	131.82592391968	107.98692703247
it support	11	11	136.06286048889	128.07202339172
hostlive	16	16	136.64484024048	123.40092658997

content creator	13	13	116.25385284424	115.37504196167
<b>Jumlah</b>	<b>66</b>	<b>66</b>	<b>63.473.248.481.751</b>	<b>48.673.141.002.655</b>

3.4.7 Pengujian VI

Membandingkan berapa lama waktu pencarian pattern tiga kata atau lebih dan output yang dihasilkan dari metode brute force dan boyer-moore.

Tabel 7 Pengujian VII

Pattern	Banyak data ditemukan		Lama pencarian	
	Brute force	Boyer-Moore	Brute force	Boyer-Moore
front end developer	5	5	144.51789855957	122.80488014221
social media specialist	5	5	120.38993835449	136.28888130188
backend engineer	10	10	108.64305496216	128.96704673767
business development manager	18	18	143.47195625305	113.09099197388
credit administration officer	4	4	172.42097854614	123.56805801392
<b>Jumlah</b>	<b>42</b>	<b>42</b>	<b>68.944.382.667.541</b>	<b>62.471.985.816.956</b>

3.4.8 Pengujian VI

Melakukan pencarian pattern yang sama sekali tidak terdapat di dalam database.

Pattern	Banyak data ditemukan		Lama pencarian (second)	
	Brute force	Boyer-Moore	Brute force	Boyer-Moore
psikologi	0	0	124.95088577271	122.10202217102
perpustakaan	0	0	151.44395828247	115.06605148315
teknik informatika	0	0	130.98406791687	124.98688697815
lowongan kerja	0	0	120.00393867493	113.70301246643
universitas perjuang antasikmalaya	0	0	124.48287010193	119.97413635254
<b>Jumlah</b>	<b>0</b>	<b>0</b>	<b>65.186.572.074.891</b>	<b>59.583.210.945.129</b>

### 3.5 Evaluasi

Ada beberapa faktor yang dapat mempengaruhi kinerja algoritma boyer-moore dan mengakibatkan algoritma ini terkadang lebih lambat dari algoritma brute force. Beberapa faktor tersebut antara lain:

1. Jika pola yang dicari relative pendek, maka algoritma boyer-moore tidak dapat mengoptimalkan heuristiknya dengan baik dan justru memerlukan waktu lebih lama untuk memproses pencocokan pola.
2. Panjang string pada setiap data yang diperiksarelatif pendek sehingga algoritma brute force menjadi lebih cepat karena tidak memerlukan banyak operasi perpindahan pointer seperti pada algoritma boyer- moore.
3. Jika teks input sangat kecil, seperti pengujian 1 karakter dimana brute force lebih cepat, maka algoritma Boyer-Moore tidak dapat memanfaatkan fitur-fitur pencocokan heuristiknya dengan baik dan akibatnya memerlukan waktu lebih lama untuk memproses pencocokan pola.
4. Implementasi algoritma boyer-moore yang buruk dapat mengakibatkan kinerja yang buruk juga. Misalnya, jika implementasi tidak efisien dalam membangun tabel bad character atau tabel good suffix, maka algoritma boyer-moore mungkinakan lebih lambat dari pada algoritma brute force.

Namun demikian, algoritma Boyer-Moore tetap diharapkan lebih cepat dari pada algoritma brute force dalam kasus umum karena kompleksitas waktu rata-rata pada kasusterburuk yang lebihrendah. Berdasarkanfaktor- factor tersebut, algoritmaboyer-moore adalah pilihan yang lebih baik untuk aplikasi pencarian kerja berbasis web.

## 4. KESIMPULAN

Dengan adanya Perbandingan Algoritma Brute Force denganAlgoritma Boyer-Moore Pada Aplikasi Pencarian Kerja, dapat diambil kesimpulan sebagai berikut:

1. Telah berhasil menerapkan algoritma brute force dan algoritma boyer-moore pada aplikasi pencarian kerja untuk dibandingkan
2. Telah berhasil membandingkan kecepatanke dua algoritma dalam mencari judul pekerjaan yang sesuai, bahwa algoritma Boyer-Moore lebih efisien dibandingkan algoritma brute force pada aplikasi pencarian kerja. Penggunaan algoritma Boyer-Moore dalam aplikasi pencarian kerja mampu menghasilkan waktu pencarian yang lebihcepat dan efisien dibandingkan dengan algoritma brute force.

## 5. SARAN

Agar memperoleh hasil yang lebih baik kedepannya, maka penulis memberikan saran sebagai berikut:

1. Penerapan ini masih perlu dikembangkan dengan menggunakan algoritma lain seperti binary search, linear search, Exponential Search atau lainnya.

2. Karena dalam penulisan ini membatasi pada pencarian judul pekerjaan, maka untuk perkembangan lebih lanjut sekiranya dapat diperluas kearah yang lebih detail.

#### DAFTAR PUSTAKA

- [1] S. Maisaroh, O. Fajarianto, and M. Nasir, "Sistem Informasi Lowongan Kerja Kota Tangerang Berbasis Android dan Web Service," 2019.
- [2] M. Patasik, N. S. Sambo, and Nirwana, "Penerapan Algoritma Brute Force Pada Aplikasi Penerjemah Bahasa Inggris-Toraja Berbasis Android," 2021.
- [3] B. Erdani, F. DwiAditia, S. Rodiah, and I. Hartami Santi, "Sistem Aplikasi Kamus Istilah Bahasa Pemrograman PHP Menggunakan Algoritma Brute Force Application System Dictionary of PHP Programming Language Terms Using Brute Force Algorithm," 2019.
- [4] A. T. Ramadhoni, I. H. Santi, S. Kirom, U. Islam, and B. Blitar, "PENERAPAN ALGORITMA BRUTE FORCE PADA APLIKASI SIDAYKO BERBASIS ANDROID," 2022.
- [5] V. Friandika, D. Ariyadi, P. Astuti, A. F. Cobantoro, and K. N. Fitri, "PENERAPAN ALGORITMA BOYER MOORE PADA PENGARSIPAN DOKUMEN DI KUA PUNUNG KAB. PACITAN JAWA TIMUR BOYER MOORE ALGORITHM IMPLEMENTATION ON DOCUMENT ARCHIVING IN KUA PUNUNG KAB. PACITAN JAWA TIMUR," 2021.
- [6] M. A. Setiawan, S. Andryana, and A. Gunaryati, "Penerapan Algoritma Boyer Moore Dalam Pencarian Barang Hilang pada Aplikasi Find It Berbasis Android," JURNAL MEDIA INFORMATIKA BUDIDARMA, vol. 5, no. 3, p. 945, Jul. 2021, doi: 10.30865/mib.v5i3.3093.
- [7] A. A. Nababan and M. Jannah, "ALGORITMA STRING MATCHING BRUTE FORCE DAN KNUTH-MORRIS-PRATT SEBAGAI SEARCH ENGINE BERBASIS WEB PADA KAMUS ISTILAH JARINGAN KOMPUTER," Jurnal Mantik Penusa, vol. 3, no. Desember, pp. 87–94, 2019.
- [8] Y. Suharya and A. A. Rohman, "APLIKASI PENCATATAN KEHADIRAN KARYAWAN DI PD.HIKMAH FARM," 2022.
- [9] S. Andryana, A. Andrianingsih, and N. Farhan, "Penerapan Algoritma A-star dan Brute Force pada Aplikasi Jakvel (Jakarta Travel) Berbasis Android," JURNAL MEDIA INFORMATIKA BUDIDARMA, vol. 5, no. 3, p. 879, Jul. 2021, doi: 10.30865/mib.v5i3.3089.
- [10] C. Irawan and M. RiyanPratama, "Perbandingan Algoritma Boyer-Moore dan Brute Force pada Pencarian Kamus Besar Bahasa Indonesia Berbasis Android," BIOS :Jurnal Teknologi Informasi dan Rekayasa Komputer, vol. 1, no. 2, 2020.