

## Perbandingan Algoritma *Berry-Ravindran* dan *Reverse Colussi* pada Aplikasi Kamus Indonesia-Jepang Berbasis Android

Ilka Zufria<sup>1</sup>, Aidil Halim Lubis<sup>2</sup>, Luthfi Fathurahman<sup>3\*</sup>

Ilmu Komputer, Universitas Islam Negeri Sumatera Utara, Medan, Indonesia

\*e-mail *Corresponding Author*: luthfi.fathurahman@uinsu.ac.id

### Abstract

*Japanese is one of the most challenging languages in the world, yet it is one of the most popular languages in Indonesia. As many as 711,732 students in Indonesia are learning Japanese. In the process of learning Japanese, a dictionary is needed as a tool to help users expand their Japanese vocabulary. In this article, an Android-based Indonesian-Japanese dictionary application has been designed using a comparison between two algorithms, namely Berry-Ravindran and Reverse Colussi. In the development of this application, there are two algorithms used, namely the Berry-Ravindran and Reverse Colussi algorithms. Both will be used to compare the average running time of 10 tested words. The running time results for the Berry-Ravindran algorithm are 22.3 ms and for the Reverse Colussi algorithm are 16 ms. Based on these results, it can be concluded that the Reverse Colussi algorithm is faster and more efficient than the Berry-Ravindran algorithm in terms of search time.*

**Keyword:** *Japanese Language; Berry-Ravindran; Reverse Colussi; Android*

### Abstrak

Bahasa Jepang merupakan salah satu bahasa tersulit di dunia, namun merupakan salah satu bahasa yang paling diminati di Indonesia. Sebanyak 711.732 pelajar di Indonesia mempelajari bahasa Jepang. Dalam mempelajari bahasa Jepang, diperlukan sebuah kamus sebagai media untuk membantu pengguna dalam menambah kosakata bahasa Jepang. Pada artikel ini, telah dirancang aplikasi kamus bahasa Indonesia-Jepang berbasis android dengan menggunakan perbandingan antara dua algoritma, yaitu *Berry-Ravindran* dan *Reverse Colussi*. Dalam pengembangan aplikasi ini, terdapat dua algoritma yang digunakan yaitu algoritma *Berry-Ravindran* dan *Reverse Colussi*. Keduanya akan digunakan untuk membandingkan rata-rata *running time* dari 10 sampel kata yang diuji. Hasil *running time* untuk algoritma *Berry Ravindran* adalah 22,3 ms dan pada algoritma *Reverse Colussi* adalah 16 ms. Berdasarkan hasil tersebut, dapat disimpulkan bahwa algoritma *Reverse Colussi* lebih cepat dan efisien dibandingkan algoritma *Berry-Ravindran* dalam hal waktu pencarian.

**Kata kunci:** *Bahasa Jepang; Berry-Ravindran; Reverse Colussi; Android*

### 1. Pendahuluan

Bahasa merupakan sarana untuk berkomunikasi dan menjadi begitu penting dalam keseluruhan hidup manusia [1]. Tanpa bahasa, seseorang kesulitan untuk berkomunikasi dan menyampaikan keinginannya [2]. Secara keseluruhan, terdapat sekitar 7.168 bahasa yang digunakan di seluruh dunia [3]. Dari keberagaman bahasa adakalanya kita dituntut untuk berkomunikasi dengan salah satu bahasa tersebut. Salah satunya adalah bahasa Jepang.

Bahasa Jepang itu sendiri adalah bahasa nasional negara Jepang yang dituturkan sehari-hari oleh masyarakat Jepang [4]. Bahasa Jepang termasuk salah satu bahasa tersulit di dunia dan dikenal sebagai bahasa yang kompleks secara harfiah [5]. Walaupun menjadi bahasa yang sulit dipelajari, namun bahasa Jepang menjadi bahasa asing yang populer di Indonesia. Hal ini disebabkan oleh budaya Jepang yang banyak diminati oleh masyarakat Indonesia [6]. Pada survei yang dilakukan oleh The Japan Foundation di tahun 2021, Indonesia menempati peringkat kedua negara yang mempelajari bahasa Jepang terbanyak di dunia, dengan total 711.732 pelajar [7]. Jumlah ini meningkat dibandingkan tahun 2018, yaitu sebesar 709.479 pelajar [8].

Dengan banyaknya masyarakat Indonesia yang mempelajari bahasa Jepang serta fakta bahwa bahasa Jepang adalah bahasa tersulit di dunia, maka diperlukan alat bantu yang mampu menerjemahkan kata dari bahasa Jepang ke bahasa Indonesia ataupun sebaliknya. Salah satu solusinya ialah menggunakan aplikasi kamus bahasa Indonesia-Jepang berbasis android.

Aplikasi kamus bahasa Indonesia – Jepang berbasis android tersebut dibangun dengan membandingkan algoritma *Berry-Ravindran* dan algoritma *Reverse Colussi* dalam pencarian kata dari bahasa Indonesia – Jepang ataupun sebaliknya. Selain itu, aplikasi ini dapat mencari kata hingga 1000 kata, baik bahasa Indonesia maupun bahasa Jepang. Serta terdapat fitur huruf Jepang untuk membantu *user* dalam mempelajari bahasa Jepang.

Penelitian mengenai algoritma *Berry-Ravindran* dan *Reverse Colussi* sendiri telah dilakukan oleh beberapa peneliti. Penelitian mengenai *Berry Ravindran* dilakukan oleh Sahminul dengan judul “Perancangan Aplikasi Pencarian Undang-Undang Otonomi Khusus Aceh (Qanun) Berbasis Android Menggunakan Metode Algoritma *Berry Ravindran*”. Penelitian tersebut menghasilkan sebuah aplikasi android dengan algoritma *Berry-Ravindran* sebagai algoritma pencarian katanya [9]. Sedangkan untuk penelitian mengenai *Reverse Colussi* dilakukan oleh Annisyah Januarti, Sutardi, dan Mutmainnah Muchtar dengan judul “Aplikasi Istilah Akuntansi Sebagai Media Pembelajaran Berbasis Android Menggunakan Algoritma *Reverse Colussi*”. Penelitian tersebut menghasilkan sebuah aplikasi android yang dapat mencari istilah Akuntansi dengan algoritma *Reverse Colussi* sebagai algoritma pencarian katanya[10].

Tujuan dari artikel ini adalah untuk mengetahui algoritma mana yang lebih efisien dalam hal waktu pencarian kata. Serta memiliki manfaat bagi *developer*, *user*, dan peneliti berikutnya. Bagi *developer*, hasil perbandingan ini dijadikan acuan dalam memilih algoritma terbaik yang dapat digunakan pada aplikasi yang akan dibangun. Bagi *user*, akan dapat menggunakan aplikasi yang memiliki algoritma pencarian kata terbaik dalam segi kecepatan pencarian kata. Dan bagi peneliti, dapat dilakukan penelitian lanjutan mengenai kedua algoritma tersebut.

## 2. Tinjauan Pustaka

Pada penelitian yang dilakukan oleh Saut Dohot Siregar, Mawaddah Harahap, dan Yohana Marbun dengan judul “Aplikasi Penerjemah Kalimat Bahasa Indonesia ke Bahasa Simalungun dengan Algoritma *Berry – Ravindran*” menjelaskan bahwa penelitian ini membangun aplikasi penerjemah kalimat berbasis *desktop* menggunakan algoritma *Berry - Ravindran* dalam proses pencarian katanya. Aplikasi ini dapat bisa digunakan dalam menerjemahkan bahasa Indonesia ke bahasa Simalungun, dan sebaliknya. Namun, masih ada kendala yang ditemukan yaitu tidak semua kata bisa diterjemahkan karena jumlah kata pada *database* masih terbatas [12].

Pada penelitian yang dilakukan oleh Sahminul dengan judul “Perancangan Aplikasi Pencarian Undang-Undang Otonomi Khusus Aceh (Qanun) Berbasis Android Menggunakan Metode Algoritma *Berry Ravindran*”, dijelaskan bahwa tujuan dari penelitian ini agar masyarakat dapat mempelajari atau mencari pasal dalam UU Qanun dengan mudah. Dari penelitian ini, dapat disimpulkan bahwa algoritma *Berry-Ravindran* dapat diterapkan dalam perancangan aplikasi pencarian UU Otonomi Khusus Aceh serta dapat berjalan baik pada *platform* Android [9].

Pada penelitian yang dilakukan oleh Safwani, Suendri, dan Triase dengan judul “Pencarian Arti Ayat Al-Quran dengan *Speech Recognition* Menggunakan Algoritma *Berry Ravindran* Berbasis Android” menjelaskan bahwa aplikasi pencarian arti Al-Quran merupakan aplikasi pencarian berbasis android dan menggunakan algoritma *Berry Ravindran* saat mencari arti ayat Al-Quran. Aplikasi tersebut memiliki fitur *Speech Recognition* yang membantu *user* mencari arti ayat Al-Quran dengan menggunakan suara. Penelitian ini menggunakan sampel 15 kata dan menghasilkan persentase akurasi dengan suara sebesar 82.6%, sedangkan persentase akurasi ketika menggunakan *keyboard* adalah 100% [13].

Pada penelitian yang dilakukan oleh Efelin O. Siburian, Mohammad Andri Budiman, dan Jos Timanta dengan judul “Implementation and Comparison of *Berry-Ravindran* and *Zhu-Takaoka Exact Stringmatching Algorithms* in Indonesian-Batak Toba Dictionary” dijelaskan bahwa penelitian ini bertujuan agar masyarakat umum, dapat mempelajari bahasa Batak dengan mudah. Dari penelitian ini, dapat disimpulkan bahwa algoritma *Berry-Ravindran* dan *Zhu Takaoka* dapat diterapkan dalam aplikasi kamus bahasa Indonesia-Batak Toba berbasis *platform Desktop* dan berdasarkan pengujian sistem, terbukti bahwa waktu yang dibutuhkan oleh algoritma *Zhu-Takaoka* lebih singkat dibandingkan dengan algoritma *Berry-Ravindran* [11].

Pada penelitian yang dilakukan oleh M. Zulhasmi Lubis berjudul “Perancangan Aplikasi *String Matching* Dalam Pencarian Makanan Pantangan Untuk Penderita Penyakit Kanker Dengan

Algoritma *Berry-Ravindran* Berbasis Android” menjelaskan bahwa aplikasi yang dibangun merupakan aplikasi pencarian makanan pantangan untuk penderita kanker. Aplikasi ini berbasis android dan menerapkan algoritma *Berry-Ravindran* saat pencarian makanan pantangan [14].

Pada penelitian yang dilakukan oleh Annisyah Januarti, Sutardi, dan Mutmainnah Muchtar dengan judul “Aplikasi Istilah Akuntansi Sebagai Media Pembelajaran Berbasis Android Menggunakan Algoritma *Reverse Colussi*” menjelaskan bahwa aplikasi yang dibangun adalah sebuah aplikasi yang dibuat untuk membantu pengguna mencari istilah akuntansi. Aplikasi ini berbasis android dan menggunakan Algoritma *Reverse Colussi* dalam pencarian kata. Saat pengujian, aplikasi ini memiliki rata-rata waktu pencarian kurang dari 2 detik, serta memiliki akurasi yang tepat saat melakukan pencarian [10].

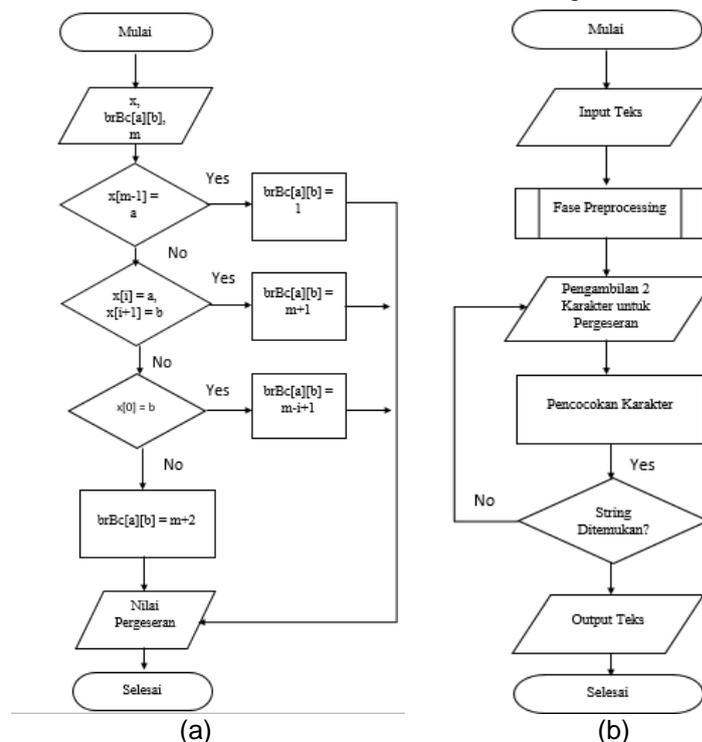
Pada penelitian yang dilakukan oleh Riki Yuspika, Aan Sumiah, dan Tito Sugiarto dengan judul “Rancang Bangun Aplikasi Penggalangan Dana Menggunakan Algoritma *Reverse Colussi* Berbasis Web” menjelaskan bahwa aplikasi penggalangan dana yang dibangun berbasis *website* dan menggunakan algoritma *Reverse Colussi* saat melakukan proses pencarian data penggalangan dana [15].

Penelitian yang kami lakukan memiliki perbedaan dengan penelitian [12], [9], [14], [15], dimana penelitian kami menggunakan 2 algoritma saat melakukan pencarian kata dan menghasilkan waktu pencarian rata-rata setiap algoritma untuk dibandingkan satu sama lain. Untuk penelitian [10] dan [13] bila dibandingkan dengan penelitian kami, memiliki perbedaan. Dimana penelitian tersebut hanya menggunakan 1 algoritma *string matching*, yaitu algoritma *Reverse Colussi* pada penelitian [10] dan algoritma *Berry-Ravindran* pada penelitian [13]. Adapun untuk penelitian [11] bila dibandingkan dengan penelitian kami, memiliki perbedaan. Dimana penelitian tersebut menggunakan algoritma *Berry-Ravindran* dan Zhu-Takaoka sebagai algoritma *string matching*, sedangkan penelitian kami menggunakan algoritma *Berry-Ravindran* dan *Reverse Colussi*. Penelitian tersebut juga tidak memiliki pengujian waktu rata-rata pencarian.

### 3. Metodologi

#### 3.1. Algoritma *Berry-Ravindran*

Algoritma ini terdiri dari dua tahap dalam proses pencocokan string, yaitu tahap preprocessing dan tahap pencarian. Berikut adalah *flowchart* algoritma *Berry-Ravindran*.



Gambar 1. (a) Flowchart Fase Preprocessing *Berry-Ravindran*, (b) Flowchart *Berry-Ravindran*

Pada tahap *preprocessing* akan menghasilkan tabel *brBc* (*Berry-Ravindran Bad Character*). Tabel *brBc* sendiri berisi nilai pergeseran *string*, dimana nilai pergeseran tersebut dihasilkan dari beberapa kondisi yang sudah ditetapkan [14]. Dalam menentukan nilai *brBc*, terdapat empat kondisi, yaitu :

$$brBc[a,b] = \min \begin{cases} 1 & \text{if } x[m - 1] = a, \\ m - i + 1 & \text{if } x[i] x[i + 1] = ab, \\ m + 1 & \text{if } x[0] = b, \\ m + 2 & \text{otherwise} \end{cases}$$

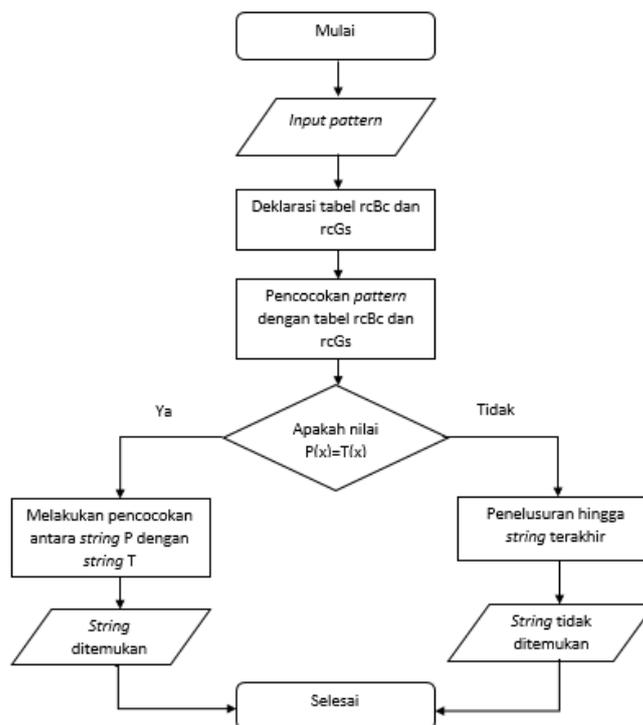
Keterangan :

- brBc : *Berry-Ravindran Bad-character*
- m : panjang *pattern*
- i : index dari karakter ke-n pada teks
- x : *pattern*
- a, b : karakter di dalam teks

Pencarian nilai pergeseran dimulai dua karakter teks berurutan, a dan b, yang berada pada posisi paling kanan setelah *pattern*. Terdapat empat kondisi yang harus dipenuhi untuk mencari nilai pergeseran. Pertama, jika karakter a memiliki posisi [m-1], maka nilai pergeserannya adalah 1. Kedua, jika karakter a dan b sama dengan [i] dan [i+1], maka nilai pergeserannya adalah m-i+1. Ketiga, jika karakter b = [0], maka nilai pergeserannya adalah m+1. Terakhir, jika kedua karakter tersebut tidak memenuhi ketiga kondisi sebelumnya, maka nilai pergeserannya adalah m+2. Selanjutnya, pada tahap pencarian akan dilakukan pencocokan *pattern* berdasarkan nilai *brBc* yang telah ditemukan sebelumnya [9].

### 3.2. Algoritma Reverse Colussi

Saat tahap *preprocessing*, dalam menentukan pergeseran *pattern*, maka akan dilakukan pencarian terhadap sepasang karakter, *special position*, dan *non-special position*. Dan saat tahap pencarian, akan dilakukan operasi pencocokan *pattern* pada teks yang akan dicari [15]. Berikut adalah *flowchart* algoritma *Reverse Colussi*



Gambar 2. Flowchart *Reverse Colussi*

Pada tahap *Preprocessing*, akan ditemukan nilai *rcBc* dan juga ditemukan *special positions* dan *non-special positions*. Nilai *rcBc* akan ditemukan dengan pencarian sepasang karakter pada *patetrn* dengan menggunakan beberapa aturan dan variabel tertentu.

Inisialisasi:

$Y$  merupakan karakter terakhir pada  $T$ .

$s$  merupakan panjang pergeseran.

$k$  merupakan bilangan yang ditemukan pada *pattern* saat karakter ditemukan.

$m$  merupakan panjang *pattern*.

$p$  merupakan posisi *pattern*

Terdapat 3 kondisi dalam pencarian sepasang karakter, yaitu :

- Jika menemukan  $Pm-k-1 = Y$  dan  $Pm-k-s-1 = Pm-s-1$ , maka *input*  $k$  terkecil ke tabel  $rcBc[Y,s]$ .
- Jika menemukan  $Pm-k-1 = Y$  dan  $k > m-s-1$ , maka *input*  $k$  terkecil ke tabel  $rcBc[Y,s]$ .
- Jika kondisi di atas juga tidak terpenuhi, maka *input*  $m$  ke tabel  $rcBc[Y,s]$ .

Langkah selanjutnya adalah mencari *special positions* dan *non-special positions*. *Special position* akan ditemukan jika terdapat satu atau beberapa karakter yang berulang. Setelah ditemukan, maka selanjutnya akan menemukan nilai  $hmin$  dan  $rmin$  untuk diisi pada tabel  $rcGs$ .

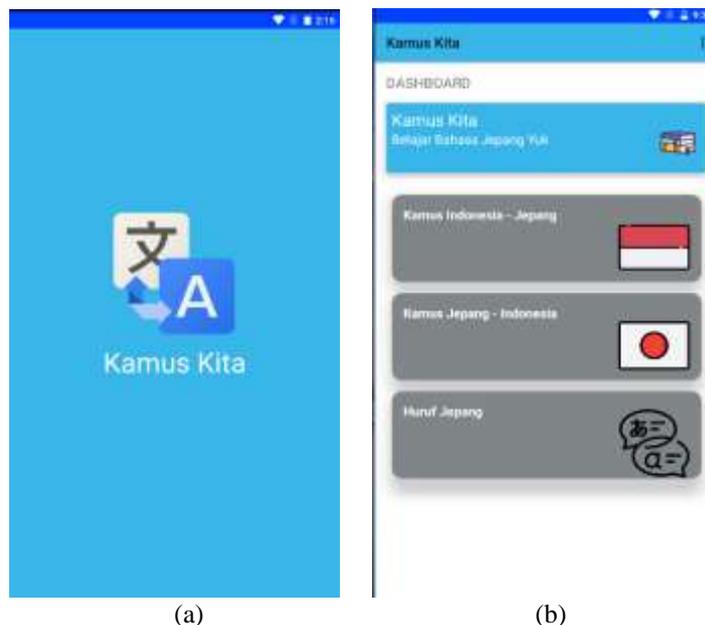
Pada tahap pencarian, akan dilakukan pencocokan *pattern* berdasarkan nilai  $rcBc$  dan  $rcGs$  yang telah ditemukan pada tahap sebelumnya. Terdapat dua syarat pencocokan *pattern* pada tahap ini, yaitu:

- Jika pada tahap pencarian belum ditemukan karakter yang cocok, maka nilai  $rcBc$  dapat digunakan.
- Jika pada tahap pencarian sudah ditemukan karakter yang cocok, maka nilai  $rcGs$  dapat digunakan [10].

#### 4. Hasil dan Pembahasan

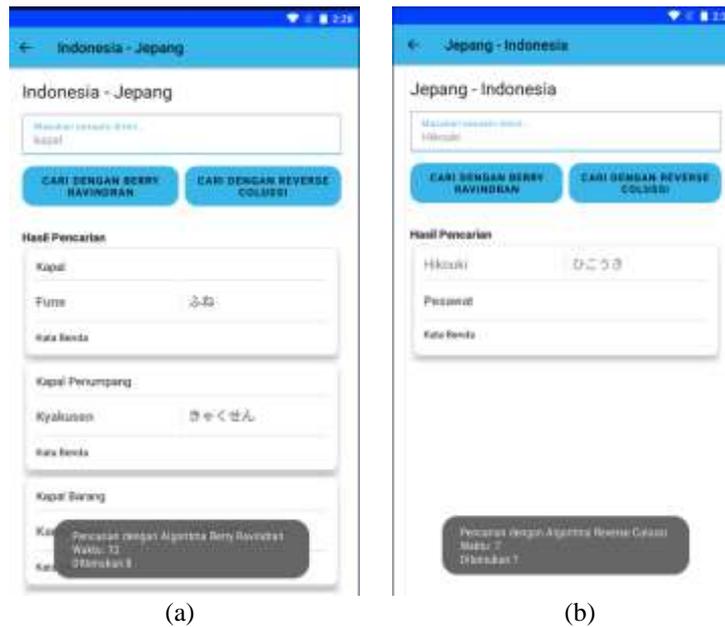
##### 4.1. Tampilan Antarmuka Aplikasi Kamus Bahasa Indonesia-Bahasa Jepang

Gambar 3, 4, 5 dan 6 di bawah adalah tampilan pada aplikasi kamus bahasa Indonesia–bahasa Jepang berbasis android sekaligus fitur-fitur yang ada pada aplikasi tersebut.



Gambar 3. (a) Splash Screen; (b) Menu Utama

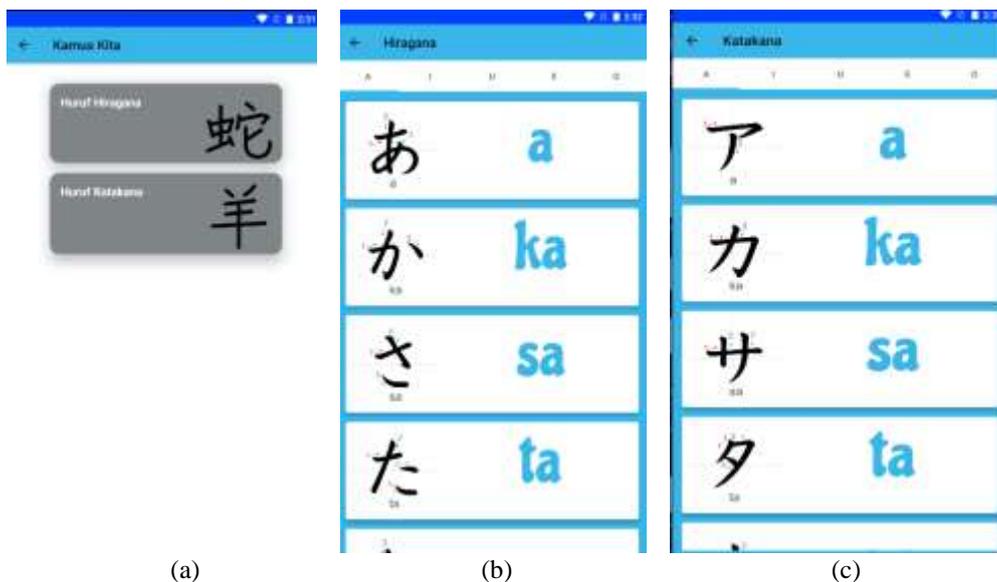
Saat membuka aplikasi, tampilan awal yang muncul ialah tampilan *Splash Screen* (Gambar 3 bagian (a)). *Splash Screen* berisi logo aplikasi “Kamus Kita”. Lalu pengguna akan masuk ke menu utama. Menu Utama berisi *button* “Kamus Indonesia – Jepang”, “Kamus Jepang-Indonesia”, dan, “Huruf Jepang” seperti yang ditunjukkan pada gambar 3 bagian (b).



Gambar 4. (a) Menu Kamus Indonesia-Jepang; (b) Menu Kamus Jepang-Indonesia

Jika pengunjung memilih *button* “Kamus Indonesia-Jepang”, maka pengunjung akan masuk ke menu “Indonesia -Jepang”. Dimana pada menu ini, pengguna dapat memasukkan kata yang akan dicari pada *textbox*. Selanjutnya pengguna dapat memilih akan mencari kata dengan algoritma *Berry-Ravindran* atau *Reverse Colussi*. Selanjutnya muncul *pop-up* yang waktu pencarian dan kata yang ditemukan. Tampilannya ditunjukkan pada gambar 4 bagian (a).

Hal yang sama juga terjadi saat pengguna memilih *button* “Kamus Jepang-Indonesia”. Ketika pengguna masuk ke menu “Kamus Jepang-Indonesia”, mereka dapat memasukkan kata yang ingin dicari ke dalam *textbox*. Selanjutnya pengguna dapat memilih akan mencari kata dengan algoritma *Berry-Ravindran* atau *Reverse Colussi*. Selanjutnya muncul *pop-up* berisi waktu pencarian dan kata yang ditemukan. Tampilannya ditunjukkan pada gambar 4 bagian (b).



Gambar 5. (a) Menu Huruf Jepang; (b) Menu Huruf Hiragana; (c) Menu Huruf Katakana

Jika pengunjung memilih *button* “Huruf Jepang”, maka akan muncul menu “Huruf Jepang”. Dimana terdapat dua pilihan, yaitu *button* “Huruf Hiragana” dan *button* “Huruf Katakana”, seperti pada gambar 5 bagian (a).

Saat pengguna memilih “Huruf Hiragana” maka akan muncul menu “Huruf Hiragana” dimana terdapat tab yang berisi huruf vokal. Saat tab dipilih, maka muncul gambar huruf dan cara baca huruf tersebut. Tampilannya seperti yang ditunjukkan pada gambar 5 bagian (b)

Saat pengguna memilih “Huruf Katakana” maka akan muncul menu “Huruf Katakana” dimana terdapat tab yang berisi huruf vokal. Saat tab dipilih, maka muncul gambar huruf dan cara baca huruf tersebut. Tampilannya seperti yang ditunjukkan pada gambar 5 bagian (c).



Gambar 6. Menu Tentang

Menu terakhir yang ada pada aplikasi ini, adalah menu “Tentang”. Menu ini merupakan halaman yang berisi informasi tentang penulis. Informasi yang ditampilkan pada halaman ini yaitu informasi mengenai judul penelitian yang diteliti, informasi tentang nama dan nim penulis, logo universitas, program studi, fakultas, universitas, kota, dan tahun pembuatan penelitian.

#### 4.2. Pengujian Algoritma *Berry-Ravindran*

Algoritma *Berry-Ravindran* akan mengalami pengujian dengan memasukkan sejumlah kata, kemudian dijalankan menggunakan algoritma tersebut, dan diukur waktu pencariannya. Sebagai contoh, akan diambil 10 kata sebagai sampel, yang terdiri dari 5 kata pada bahasa Indonesia dan 5 kata pada bahasa Jepang.

Tabel 1. Pengujian Rata-rata Waktu Pencarian Algoritma *Berry Ravindran*

Pattern	Kata yang Dicari	Jumlah Data yang Ditemukan	Rata-rata Waktu Pencarian (ms)
<b>Dia</b>	Dia	2	13
<b>Mobil</b>	Mobil	3	20
<b>Pensil</b>	Pensil	2	27
<b>Tanah</b>	Cacing Tanah	2	15
<b>Tempur</b>	Pesawat Tempur	2	14
<b>Aoi</b>	Aoi	1	24
<b>Migi</b>	Migi	1	29
<b>Ringo</b>	Ringo	1	31
<b>Arashi</b>	Arashi	1	32
<b>San</b>	Itokosan	6	18
Total Rata-rata Waktu Pencarian <i>Berry-Ravindran</i> (ms)			<b>22.3</b>

Hasil dari pengujian 10 kata di atas adalah ditemukan rata-rata waktu pencarian dengan Berry Ravindran sebesar 22.3 ms.

### 4.3. Pengujian Algoritma *Reverse Colussi*

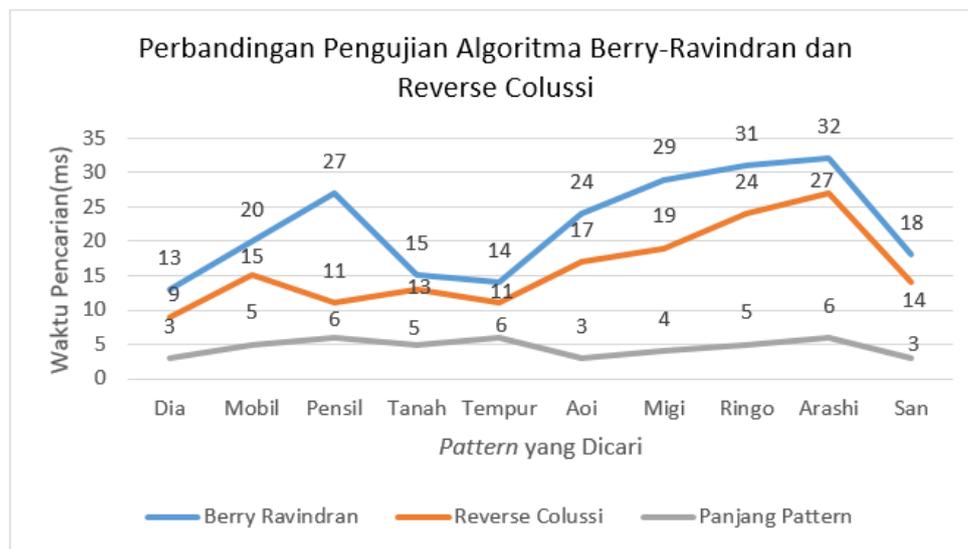
Algoritma *Reverse Colussi* akan mengalami pengujian dengan memasukkan sejumlah kata, kemudian dijalankan menggunakan algoritma tersebut, dan diukur waktu pencariannya. Sebagai contoh, akan diambil 10 kata sebagai sampel, yang terdiri dari 5 kata pada bahasa Indonesia dan 5 kata pada bahasa Jepang.

Tabel 2. Pengujian Rata-rata Waktu Pencarian Algoritma *Reverse Colussi*

Pattern	Kata yang Dicari	Jumlah Data yang Ditemukan	Rata-rata Waktu Pencarian (ms)
Dia	Dia	4	9
Mobil	Mobil	3	15
Pensil	Pensil	3	11
Tanah	Cacing Tanah	3	13
Tempur	Pesawat Tempur	3	11
Aoi	Aoi	1	17
Migi	Migi	1	19
Ringo	Ringo	1	24
Arashi	Arashi	3	27
San	Itokosan	19	14
<b>Total Rata-rata Waktu Pencarian <i>Reverse Colussi</i> (ms)</b>			<b>16</b>

Hasil dari pengujian 10 kata ialah ditemukan rata-rata waktu pencarian dengan *Reverse Colussi* sebesar 16 ms.

### 4.4. Perbandingan Pengujian Algoritma *Berry-Ravindran* dengan *Reverse Colussi*



Gambar 7. Perbandingan *Berry-Ravindran* dan *Reverse Colussi*

Dari hasil pengujian, dapat disimpulkan bahwa waktu *running* (*running time*) pada algoritma Berry Ravindran dan *Reverse Colussi* cenderung meningkat seiring dengan bertambahnya panjang *pattern* yang digunakan dan jumlah data dalam *database*. Rata-rata nilai *running time* dari 10 sampel *pattern* untuk algoritma *Berry-Ravindran* adalah 22.3 ms, sedangkan untuk algoritma *Reverse Colussi* adalah 16 ms.

Hasil tersebut membuat algoritma *Reverse Colussi* menjadi algoritma yang memiliki waktu pencarian tercepat bila dibandingkan dengan algoritma *Berry-Ravindran*. Hasil ini hampir

mirip dengan penelitian [11], dimana *Berry-Ravindran* mencatat waktu yang lebih lambat dibandingkan dengan *Zhu-Takaoka*.

## 5. Simpulan

Pada penelitian di atas, algoritma *Berry-Ravindran* dan algoritma *Reverse Colussi* diuji rata-rata waktu pencarian dengan menggunakan 10 sampel kata dalam bahasa Indonesia dan bahasa Jepang. Hasil pengujian tersebut menghasilkan rata-rata waktu pencarian dengan menggunakan algoritma *Berry-Ravindran* sebesar 22,3 ms, sedangkan rata-rata waktu pencarian dengan menggunakan algoritma *Reverse Colussi* sebesar 16 ms.

Berdasarkan hasil pengujian tersebut, dapat disimpulkan bahwa algoritma yang paling efektif untuk digunakan adalah algoritma *Reverse Colussi*. Algoritma *Reverse Colussi* memiliki waktu pencarian yang lebih cepat dibandingkan algoritma *Berry-Ravindran*. Saat mencari kata dengan sebuah pola, algoritma *Reverse Colussi* juga mampu menemukan semua kata yang memiliki kesamaan dengan pola tersebut, baik yang berada di awal, di tengah, maupun di akhir kata. Sedangkan algoritma *Berry-Ravindran* hanya dapat menemukan sejumlah kata tertentu.

## Daftar Referensi

- [1] O. Mailani, I. Nuraeni, S. A. Syakila, and J. Lazuardi, "Bahasa Sebagai Alat Komunikasi Dalam Kehidupan Manusia," *KAMPRET Journal*, vol. 1, no. 2, pp. 1–10, 2022, [Online]. Available: [www.plus62.isha.or.id/index.php/kampret](http://www.plus62.isha.or.id/index.php/kampret)
- [2] R. Devianty, "Bahasa Sebagai Cermin Kebudayaan," *Jurnal Tarbiyah*, vol. 24, no. 2, pp. 226–245, 2017.
- [3] C. M. Annur, "Indonesia Peringkat Kedua Negara dengan Jumlah Bahasa Terbanyak Dunia." Accessed: Jul. 31, 2023. [Online]. Available: <https://databoks.katadata.co.id/index.php/infografik/2023/03/28/indonesia-peringkat-kedua-negara-dengan-jumlah-bahasa-terbanyak-dunia>
- [4] A. J. B. Dinata, "Media Pembelajaran Interaktif Untuk Materi Bahasa Jepang Level Dasar," *Jurnal Mahasiswa Teknik Informatika*, vol. 2, no. 1, 2018.
- [5] A. Anjani, "Ada 5 Bahasa Tersulit di Dunia, Apa Saja?" Accessed: Jul. 31, 2023. [Online]. Available: <https://www.detik.com/edu/detikpedia/d-5818303/ada-5-bahasa-tersulit-di-dunia-apa-saja>
- [6] L. Perdanawanti and S. Setiajid, "Aplikasi Kamus Dasar Bahasa Jepang Berbasis Android Menggunakan Metode User Centered Design," *Jurnal Telematika*, vol. 10, no. 2, pp. 77–91, 2017.
- [7] The Japan Foundation, "Survey Report on Japanese-Language Education Abroad 2021." Accessed: Sep. 10, 2023. [Online]. Available: [https://www.jpf.go.jp/e/project/japanese/survey/result/dl/survey2021/All\\_contents\\_r2.pdf](https://www.jpf.go.jp/e/project/japanese/survey/result/dl/survey2021/All_contents_r2.pdf)
- [8] The Japan Foundation, "Survey Report on Japanese-Language Education Abroad 2018." Accessed: Sep. 10, 2023. [Online]. Available: [https://www.jpf.go.jp/j/project/japanese/survey/result/dl/survey2018/All\\_contents\\_r2.pdf](https://www.jpf.go.jp/j/project/japanese/survey/result/dl/survey2018/All_contents_r2.pdf)
- [9] Sahminul, "Perancangan Aplikasi Pencarian Undang Undang Otonomi Khusus Aceh (Qanun) Berbasis Android Menggunakan Metode Algoritma Berry Ravindran," *Journal of Information Sistem Research (JOSH)*, vol. 1, no. 4, pp. 229–236, 2020.
- [10] A. Januarti, Sutardi, and M. Muchtar, "Aplikasi Istilah Akuntansi Sebagai Media Pembelajaran Berbasis Android Menggunakan Algoritma Reverse Colussi," *semanTIK*, vol. 3, no. 2, pp. 133–144, 2017.
- [11] E. O. Siburian, M. A. Budiman, and J. T. Tarigan, "Implementation and comparison of Berry-Ravindran and Zhu-Takaoka exact string matching algorithms in Indonesian-Batak Toba dictionary," *Data Science: Journal of Computing and Applied Informatics*, vol. 1, no. 1, pp. 38–44, Jul. 2017, doi: 10.32734/jocai.v1.i1-67.
- [12] S. D. Siregar, M. Harahap, and Y. Marbun, "Aplikasi Penerjemah Kalimat Bahasa Indonesia Ke Bahasa Simalungun Dengan Algoritma Berry-Ravindran," *Jurnal Teknovasi*, vol. 4, no. 2, pp. 40–50, 2017.
- [13] H. Safwani, Suendri, and Triase, "Pencarian Arti Ayat Alquran dengan Speech Recognition Menggunakan Algoritma Berry Ravindran Berbasis Android," *JISTech (Journal of Islamic Science and Technology)*, vol. 6, no. 2, pp. 10–23, 2021.
- [14] M. Z. Lubis, "Perancangan Aplikasi String Matching Dalam Pencarian Makanan Pantanagn Untuk Penderita Penyakit Kanker Dengan Algoritma Berry-Ravindran

- Berbasis Android,” *KLIK: Kajian Ilmiah Informatika & Komputer*, vol. 1, no. 2, pp. 57–64, 2020, [Online]. Available: <https://djournals.com/index.php/klik|Page57>
- [15] R. Yuspika, A. Sumiah, and T. Sugiharto, “Rancang Bangun Aplikasi Penggalangan Dana Menggunakan Algoritma Reverse Colussi Berbasis WEB,” *Jurnal Nuansa Informatika*, vol. 11, no. 2, pp. 26–32, 2017.