

ANALISIS ALGORITMA RABIN-KARP PADA KAMUS UMUM BERBASIS ANDROID

Herriyance¹, Handrizal², Siti Dara Fadilla³

^{1,3}Program Studi S1 Ilmu Komputer,
Fakultas Ilmu Komputer dan Teknologi Informasi
Universitas Sumatera Utara

²Program Studi D3 Manajemen Informatika,
AMIK Tunas Bangsa Pematangsiantar

[1herriyance@yahoo.com](mailto:herriyance@yahoo.com), [2Handrizal_tanjung@yahoo.com](mailto:Handrizal_tanjung@yahoo.com), [3meydin_humairah@yahoo.com](mailto:meydin_humairah@yahoo.com)

Abstract

Development of the era had a considerable impact on the existence of a language. To overcome this there are some efforts to be made, one of which is to create a dictionary, a dictionary that was made to be practical and quick in use. Dictionary in question is a dictionary based on Android. To create a dictionary-based android can use string matching algorithm, one of the string matching algorithm is the Rabin-Karp algorithm, Rabin-Karp algorithm perform string matching hash value based on the text and the pattern hash value. The study produced an android based dictionary application which the base number is used to generate a hash value greatly affects the speed of search words. Average running time of 10 attempts to search for words is 14.9 ms.

Keywords: Android, Dictionary, Rabin-Karp String Matching

Abstrak

Perkembangan jaman memiliki dampak yang cukup besar terhadap keberadaan suatu bahasa. Untuk mengatasi hal itu ada beberapa upaya yang dapat dilakukan, salah satunya adalah dengan membuat kamus, kamus yang di buat haruslah praktis dan cepat dalam penggunaannya. Kamus yang dimaksud adalah kamus berbasis android. Untuk membuat kamus berbasis android dapat menggunakan algoritma string matching, salah satu algoritma string matching adalah algoritma Rabin-Karp, algoritma Rabin-Karp melakukan pencocokan string berdasarkan nilai hash pada teks dan nilai hash pada pattern. Penelitian ini menghasilkan sebuah aplikasi kamus berbasis android yang mana bilangan basis yang digunakan untuk membangkitkan nilai hash sangat mempengaruhi kecepatan pencarian kata. Rata-Rata running time untuk untuk 10 kali percobaan pencarian kata adalah 14.9 ms.

Kata Kunci : Android, Kamus, Rabin-Karp, String Matching

1. PENDAHULUAN

Kemajuan dibidang teknologi seperti media massa, game online, dan media social seperti facebook, twitter, instagram telah menjangkiti kehidupan dalam bermasyarakat, khususnya generasi muda. Salah satu dampak negatif yang ditimbulkan dari kemajuan teknologi ini adalah memudarnya rasa cinta terhadap bahasa negeri sendiri yang mengakibatkan hilangnya nilai-nilai budaya daerah tersebut dalam berbahasa. Menurut Putra[1] jika keadaan seperti ini terus berlanjut maka akan ada dua ancaman serius terhadap perkembangan bahasa,

baik itu bahasa Indonesia maupun bahasa daerah, yaitu ketersingkirian dan kepunahan. Selanjutnya, Putra menjelaskan ketersingkirian bahasa paling tidak ada dua macam. Pertama, tersingkirnya dari arena sosial. Hal ini terjadi karena jarang digunakan ketika berbicara dan digantikan dengan kata-kata dari bahasa lain. Kedua, berkurangnya arena sosial bahasa dalam kehidupan sehari-hari.

Untuk mengatasi dan dapat memahami bahasa yang digunakan oleh setiap daerah, diperlukan suatu alat yang dapat membantu menerjemahkan kosa kata yang ada, alat tersebut adalah kamus, sehingga perbedaan bahasa tidak menjadi suatu kendala dalam melakukan komunikasi. Setiap bentuk kamus memiliki kelemahan dan juga kelebihan. Contohnya kamus dalam bentuk buku memiliki kelebihan dalam jumlah kata yang banyak, tetapi juga memiliki kelemahan dalam hal pencarian kata, yakni membutuhkan waktu yang lama dalam pencarian serta tidak praktis. Media kamus dalam bentuk mobile kamus bisa dijadikan solusi untuk mengatasi masalah dalam pencarian kata dengan cepat. Maka penulis merancang aplikasi kamus umum berbasis android yang diharapkan dapat membantu pengguna untuk lebih mengenal bahasa daerah umum secara praktis. Kamus umum yang akan di rancang menggunakan metode pencocokan *string* (*string matching*).

String matching atau adalah suatu teknik yang digunakan untuk menemukan suatu keakuratan dari satu atau beberapa pola teks yang diberikan [2]. Ada banyak algoritma *string matching* salah satunya adalah algoritma *Rabin Karp*. Algoritma *Rabin-Karp* diciptakan oleh Michael O. Rabin dan Richard Karp pada tahun 1987 yang menggunakan fungsi *hashing* untuk menemukan *pattern* didalam *string* teks [3]. Dengan memanfaatkan fungsi *hash* yang ada pada algoritma *Rabin-Karp*, maka penulis menerapkan algoritma *Rabin-Karp* pada penelitian ini.

2. METODOLOGI PENELITIAN

2.1. Kamus

Menurut Pusat Bahasa [4], kamus adalah buku acuan yang memuat kata dan ungkapan, biasanya disusun menurut abjad berikut keterangan tentang makna, pemakaian, atau terjemahannya, kamus juga disebut buku yang memuat kumpulan istilah atau nama yang disusun menurut abjad beserta penjelasan tentang makna dan pemakaiannya

2.2. *String Matching*

String matching atau pencocokan *string* adalah suatu teknik yang digunakan untuk menemukan suatu keakuratan dari satu atau beberapa pola teks yang diberikan [2].

2.3. Algoritma *Rabin-Karp*

Algoritma *Karp-Rabin* diciptakan oleh Michael O. Rabin dan Richard M. Karp pada tahun 1987 yang menggunakan fungsi *hashing* untuk menemukan *pattern* di dalam *string* teks [3]. Charas and Lecroq [5], didalam bukunya menyatakan bahwa fungsi *hashing* menyediakan metode sederhana untuk menghindari perbandingan jumlah karakter yang kuadratik di dalam banyak kasus atau situasi. Dari pada

melakukan pemeriksaan terhadap setiap posisi dari teks ketika terjadi pencocokan pola, akan lebih baik dan efisien melakukan pemeriksaan hanya jika teks yang sedang proses memiliki kemiripan seperti pada *pattern*. Untuk melakukan pengecekan kemiripan antara dua kata ini digunakan fungsi *hash*.

Ramdhani, [6] dalam penelitiannya menuliskan bahwa fungsi *hash* menyimpan bentuk *string* dalam bentuk lain yaitu enumerasi sehingga suatu *string* tertentu akan memiliki nilai enumerasinya yang unik. Raharja [7] meneliti tentang perancangan dan implementasi sistem penilaian jawaban esai otomatis menggunakan algoritma *Rabin-Karp*, dimana untuk mencari nilai *hash* digunakan persamaan (1):

$$H = C_1 * B^{(m-1)} + C_2 * B^{(m-2)} + \dots + C_{(m-1)} * B^m + C_m \quad (1)$$

Dimana:

H = Nilai Hash

C = ASCII karakter

B = Basis (bilangan prima)

m = Banyak karakter (panjang karakter)

2.4. Android

Android adalah sistem operasi yang berbasis *Linux* untuk telepon seluler seperti *smartphone* dan komputer tablet (Aan, 2015). *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Beberapa keunggulan sistem operasi *Android*, yaitu :

1. *User friendly*, mudah dalam pengoperasiannya. *Interface*, dari segi tampilan yang tidak kalah bagusnya dari para pesaing.
2. *Open Source*, sistem operasi ini memang dibuat *open source*, sehingga banyak *Custom Rom* yang beredar dikembangkan berdasarkan sistem operasi *Android* yang bersifat terbuka.
3. Aplikasi, ketersediaan aplikasi yang dapat diunduh baik yang bersifat gratis ataupun berbayar. Versi-versi dari sistem operasi *Android* dari waktu ke waktu, *Android* terus mengalami pembaruan versi untuk meningkatkan kinerjanya, antara lain:
 - a. *Android* versi 1.0 dan 1.1
Sistem *Android* versi 1.0 ini pertama kali hadir di tahun 2008, tepatnya pada oktober 2008. Ponsel pertama yang menggunakan sistem *Android* adalah *HTC*. Pada bulan februari 2009 rilis *update* pertama *Android* versi 1.1 yang masih belum memiliki nama. Dan untuk *Android* versi selanjutnya *google* memutuskan untuk memberi nama versi *Android* dengan nama makanan ringan, tujuannya adalah agar mudah diingat oleh para pengguna *Android*.
 - b. *Cupcake* (*Android* versi 1.5) Nama *Cupcake* diambil dari sebuah makanan ringan. Dengan penamaan *Cupcake* ini maka penamaan dari versi *Android* dimulai. Di tahun 2009 pada bulan mei, *Android* kembali mengubah versinya dari android 1.1 yang disempurnakan menjadi versi 1.5 *Cupcake*. Adapun perbaikan di versi ini adalah mempunyai kemampuan merekam

dan menonton *video* melalui *camcorder*, *Upload video* ke *youtube* dan *upload* gambar ke *picasa*, aplikasi baru *soft-keyboard* dengan fungsi *text prediction*, *Bluetooth A2DP* dan *AVRCP support*, mempunyai kemampuan otomatis terhubung ke perangkat *Bluetooth* dengan jarak tertentu, *Widget* baru dan *folder* dapat dikumpulkan di layar *home*, perpindahan layar secara otomatis.

- c. *Donut (Android versi 1.6)* *Android* versi ini memiliki fitur yang lebih baik dari versi sebelumnya. Adapun perbaikan di versi ini adalah peningkatan pada *Android Market*, Integrasi kamera, perekam *video* dan tampilan galeri, aplikasi galeri yang baru memungkinkan pengguna memilih banyak foto untuk dihapus. Aplikasi *voice search* yang diperbaharui menjadi lebih cepat merespon dan integrasi dengan aplikasi yang lain termasuk kemampuan mencari kontak, Aplikasi *search* yang ditingkatkan untuk bisa mencari *bookmarks*, *history*, kontak, dan *web* dari layar *home*, Peningkatan dukungan teknologi untuk *CDMA/EVDO.802.1x*, *VPNs*, dan mesin *text to speech* mendukung resolusi layar *WVGA*, *speed improvement* in searching and *camera applications*, perbaikan kecepatan di aplikasi pencarian dan aplikasi kamera.
- d. *Eclair (Android versi 2.0 - 2.1)*
Setelah *Android* meluncurkan 4 versi dalam satu tahun yang sama maka para produsen *gadget* pun mulai tertarik untuk merelakan produknya menggunakan sistem *Android* ini. *Google* selaku pihak utama dalam pengembangan system *Android* ini akan tetap terus mengembangkan *Android* sampai versi yang paling tinggi. Adapun perbaikan di versi ini adalah optimalisasi kecepatan *hardware*, mendukung lebih banyak ukuran layar dan resolusi layar, *Revamped UI*, *User interface* baru pada *browser* dan dukungan *html 5*, daftar kontak baru, Rasio putih-hitam yang lebih baik untuk *backgrounds*, peningkatan aplikasi *Google Maps 3.1.2*, dukungan untuk *Microsoft Exchange*, mendukung *Flash* untuk kamera, *Digital Zoom*, peningkatan pada aplikasi *virtual keyboard*, *Bluetooth 2.1*, *Live Wallpapers*.
- e. *Froyo (Android versi 2.2 - 2.2.3)*
Versi ini dirilis pada bulan mei 2010 dengan *update* memperbaiki segi kecepatan dan pengadopsian *Javascript* dari *browser Google Chrome*. Adapun perbaikan di versi ini adalah optimalisasi kecepatan dan performa *Android OS*, integrasi *chrome v8 Javascript* kedalam aplikasi *browser*, peningkatan dukungan *Microsoft Exchange*, peningkatan aplikasi *luncher* dengan *shortcut* menuju aplikasi *phone* dan *browser*, *USB tethering* dan *WiFi hotspot functionality*, penambahan pilihan untuk menonaktifkan akses data jaringan *mobile*, aplikasi *android market* yang telah diperbaharui dengan fitur *update* otomatis, *Quick switching between multiple keyboard languages and their dictionaries*, *Voice dialing* dan berbagai kontak melalui *Bluetooth*, mendukung *file upload* di aplikasi *browser*, dukungan terhadap aplikasi *Adobe Flash 10.1* terakhir.
- f. *Gingerbread (Android versi 2.3 - 2.3.7)*
Versi ini dirilis pada bulan desember 2010. *Smartphone* pertama yang memakai versi *Android* ini adalah *Nexus S* yang dikeluarkan oleh produsen *Samsung*. Adapun perbaikan di versi *Gingerbread* ini adalah tambahan fitur

- dukungan untuk *SIP internet calling*, kemampuan nirkabel *NFC*, dukungan untuk *dual* kamera, dukungan untuk sensor giroskop dan sensor lainnya, fitur *download manager*, sejumlah *tweak* untuk penggunaan di *Tablet*, dan lainnya.
- g. *Honeycomb* (*Android* versi 3.0 - 3.2)
Untuk versi ini merupakan versi yang ditujukan untuk *gadget Tablet*. *Android Honeycomb* rilis pada Februari 2011. *Android* versi ini mendukung ukuran layar yang lebih besar, *User Interface* pada *Honeycomb* juga berbeda karena sudah didesain untuk *tablet*. *Honeycomb* juga mendukung multi prosesor dan juga akselerasi perangkat keras (*hardware*) untuk grafis.
- h. *Ice Cream Sandwich* (*Android* versi 4.0- 4.0.4)
Diumumkan pada tanggal 19 Oktober 2011, Yang baru dalam *Android* ini adalah perubahan *interface* dari *Android* sebelumnya, antara lain pengoptimalan *multitasking*, variasi layar beranda yang bisa disesuaikan dan interaktivitas mendalam serta cara baru yang ampuh untuk berkomunikasi dan berbagi konten. Penambahan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari *email* secara *offline*, dan berbagi informasi dengan menggunakan *NFC*.
- i. *Jelly Bean* (*Android* versi 4.1 4.3)
Android 4.1 Jelly Bean diumumkan pada 27 Juni 2012 pada konferensi *Google I/O* yang secara resmi dikenalkan ke publik sekitar Oktober 2012. Versi ini adalah yang tercepat dan terhalus dari semua versi *Android*. Fitur baru yang tepat di versi ini adalah meningkatkan kemudahan dan keindahan tampilan dari *Ice Cream Sandwich* dan memperkenalkan pengalaman pencarian *Google* yang baru di *Android*. *Android 4.2 Jelly Bean* juga menawarkan peningkatan kecepatan dan kemudahan *Android 4.1* serta mencakup semua fitur baru seperti *Photo Sphere* dan desain baru aplikasi kamera, *keyboard Gesture Typing*, *Google Now* dan lainnya.
- j. *KitKat* (*Android* versi 4.4)
Awalnya *Android* versi ini di isukan bernama *Key Lime Pie*. Namun pada tanggal oktober 2013 google merilis *KitKat* sebagai generasi *Android* berikutnya. *Android* versi ini memiliki banyak fitur & semakin memanjakan para pengguna *Android*. Diantaranya : *Immersive mode*, Akses kontak langsung dari aplikasi *telepon*, *google now launcher*, dan pastinya memiliki *interface UI* yang baru.
- k. *Lollipop* (*Android* versi 5.0)
Android Lollipop dianggap membawa *update* yang fantastis, banyak perubahan yang disertakan *Google* di dalamnya. Adapun fitur android ini adalah material desain tampilan yang lebih berwarna dan responsive, notifikasi lebih mudah, hemat baterai, dibekalkan fitur keamanan terenkripsi default.
- l. *Marshmallow* (*Android* versi 6.0) Sebagai sebuah sistem operasi terbaru, pastinya *Google* dan tim *Android* telah menambahkan beragam fitur anyar untuk menyempurnakan *Android Lollipop*. Kelebihan *Android Marshmallow*

yang pertama adalah memiliki modus “Doze” yaitu sebuah teknologi terbaru yang mampu memperkecil konsumsi daya baterai ketika *smartphone Android* berada pada kondisi *standby* atau *mode sleep*, dukungan *USB Type C*, *Android Pay*, sensor sidik jari, *App Permission*, *Auto Backup Aplikasi*, *Google Now on Tap* [8].

2.5. Kompleksitas Algoritma

Kebutuhan untuk dapat mengukur masalah kompleksitas, algoritma atau struktur dan dan memperoleh batas serta hubungan kuantitatif untuk kuantitas muncul di beberapa bidang ilmu disamping Ilmu Komputer, cabang tradisional Matematika, Fisika Statistik, Biologi, dan Kedokteran juga dihadapkan lebih banyak dan lebih sering dengan masalah kompleksitas ini. Dalam Ilmu Komputer, kompleksitas diukur oleh kuantitas sumber daya komputasi (waktu, penyimpanan, program, komunikasi) yang digunakan oleh sebuah tugas tertentu [6].

Teori komputasi pada dasarnya dibagi menjadi tiga bagian karakter yang berbeda. Pertama, pengertian yang tepat dari algoritma, waktu, kapasitas penyimpanan, dan-lain-lain harus diperkenalkan. Untuk ini, perbedaan model mesin matematika harus digambarkan, waktu dan penyimpanan kebutuhan perhitungan dilakukan pada kebutuhan ini harus diperjelas (umumnya diukur sebagai fungsi dari ukuran input). Yang paling mendasar pada kompleksitas adalah memberikan kalsifikasi penting dari masalah yang timbul dalam praktek, bahkan timbul di daerah matematika klasik. Kedua, kita harus menentukan kebutuhan sumber daya dari algoritma yang paling penting dalam berbagai bidang matematika, dan memberikan algoritma yang efisien untuk membuktikan bahwa masalah tertentu memiliki kompleksitas tertentu. Ketiga, harus menemukan metode yang membuktikan “hasil negatif” yaitu untuk bukti beberapa masalah sebenarnya terpecahkan di bawah pembatasan sumber daya tertentu [6].

3. HASIL DAN PEMBAHASAN

3.1. Implementasi

Penerapan algoritma *Rabin-Karp* diterapkan pada menu “cari kata” didalam sistem yang mana menu cari digunakan untuk mencari kata terjemahan dari bahasa lain ke bahasa Indonesia maupun dari bahasa Indonesia ke bahasa lain. Selain itu akan diberikan dialog yang berisi informasi tambahan tentang kata yang dicari, jumlah kata yang yang ditemukan dan waktu yang dibutuhkan untuk mencari kata tersebut.

Algoritma *Rabin-Karp* melakukan pencarian berdasarkan kecocokan nilai *hash* pada *pattern* dan teks, dalam hal ini *pattern* merupakan inputan yang diberikan *user* pada kolom edit text dan teks adalah data yang ada didalam database, database yang digunakan adalah *Sqlite*. Nilai *hash* pada *pattern* maupun pada teks dibangkitkan dengan menggunakan persamaan (1), dalam penelitian ini penulis menggunakan bilangan 3 sebagai basis. Contoh perhitungan nilai *hash* menggunakan algoritma *Rabin-Karp* dapat dilihat pada contoh 1.

Contoh 1 :

Diketahui teks tersedia didalam database seperti pada Tabel 1.

Tabel 1. Teks Tersedia Didalam Database

id	Nama bahasa	Bahasa asal	sinonim	indonesia
1	karo	uwai	uwe	iya
2	melayu	awak	-	kamu

Nilai *hash* untuk nama_bahasa = karo, bahasa_asal = uawai, sinonim = uwe, indonesia = iya., perhitungannya dapat dilihat seperti berikut ini:

bahasa_asal = uwai, dengan menggunakan persamaan 1 diketahui $B = 3$, $m = 4$, maka nilai *hash* untuk teks “uwai” adalah :

$$H(\text{uwai}) = (117 \cdot 3^{4-1}) + (119 \cdot 3^{4-2}) + (97 \cdot 3^{4-3}) + (105 \cdot 3^{4-4})$$

$$H(\text{uwai}) = 3159 + 1071 + 291 + 105$$

$$H(\text{uwai}) = 4626$$

Untuk nama_bahasa = melayu, bahasa_asal = awak, sinonim = -, indonesia = kamu, perhitungannya dapat dilihat seperti berikut ini:

bahasa_asal = awak dengan menggunakan persamaan 1 diketahui $B = 3$, $m = 4$, maka nilai *hash* untuk teks “awak” adalah :

$$H(\text{awak}) = (97 \cdot 3^{4-1}) + (119 \cdot 3^{4-2}) + (97 \cdot 3^{4-3}) + (107 \cdot 3^{4-4})$$

$$H(\text{awak}) = 4088$$

Selanjutnya dihitung nilai *hash* untuk pattern yang diinputkan oleh *user*, contoh pattern yang diinputkan oleh *user* : “**uwai**”, selanjutnya sistem menghitung nilai *hash* dengan pattern “uwai” .

Setelah nilai *hash* dari pattern dan teks diperoleh nilainya, selanjutnya sistem membandingkan nilai *hash* keduanya. Berdasarkan pattern inputan “uwai” dengan nilai *hash* 4626 sistem membandingkan dengan nilai *hash* pada kolom bahasa_asal, yang mana masing masing nilai *hash* dapat dilihat pada Tabel 2.

Tabel 2. Hasil Perhitungan Nilai Hash

id	Nama bahasa	Bahasa asal	sinonim	indonesia
1	Karo	uwai = 4626	uwe	iya
2	Melayu	awak = 4088	-	kamu

Nilai *hash* pattern “uwai = ” sama nilainya terhadap nilai *hash* kolom “bahasa_asal pada id = 1, nilai *hash*nya 4626” , maka sistem mengeluarkan hasil “iya” pada kolom indonesia dengan id 1. Sistem membaca dan melakukan perhitungan nilai *hash* perbaris.

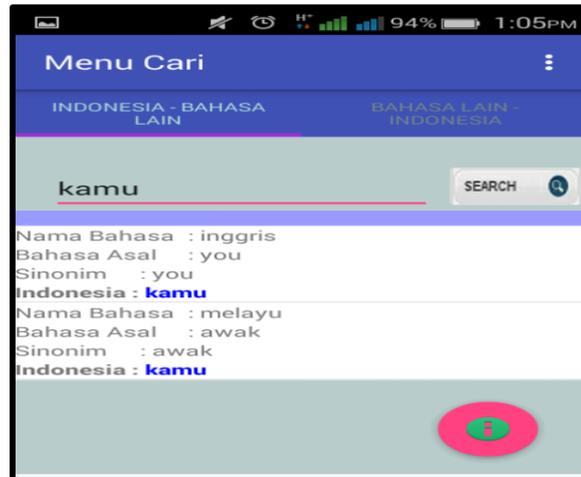
Fungsi pencarian bahasa lain - Indonesia, nilai *hash pattern* akan dibandingkan dengan nilai *hash* yang ada pada kolom bahasa_asal, ketika ada nilai *hash* yang sama maka sistem akan mengeluarkan hasil yang ada pada kolom indonesia. Sebaliknya fungsi pencarian Indonesia - bahasa lain, sistem akan melakukan perbandingan nilai *hash pattern* terhadap nilai *hash* yang ada pada kolom indonesia dan mengeluarkan hasil pada kolom nama_bahasa, bahasa_asal dan sinonim.

3.2. Pengujian

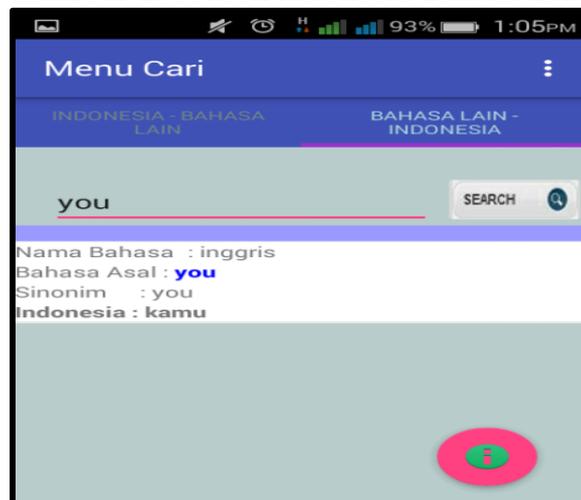
Pengujian sistem dilakukan untuk mengetahui bagaimana kinerja sistem dalam pencarian kata/*string* didalam tabel kamus menggunakan algoritma *Rabin-Karp*. Spesifikasi minimal perangkat yang dibutuhkan untuk menjalankan aplikasi ini adalah semua ponsel pintar dengan sistem operasi android *Jellybean 4.2.*, pengujian dilakukan dengan menggunakan ponsel pintar yang memiliki spesifikasi perangkat seperti berikut ini:

Merk : Evercoss
Type : Winer T/A74A
Operating Sistem : Android Kitkat 4.4.2
Prosesor : Quad Core 1.2GHz
RAM : 1Gb

Gambar 1 menunjukan contoh fungsi pencarian Indonesia-Bahasa Lain dan gambar 2 menunjukan contoh fungsi pencarian Bahasa Lain-Indonesia.

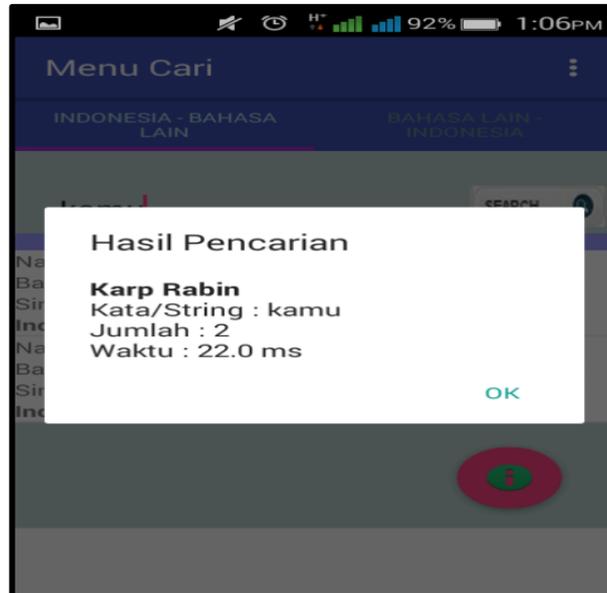


Gambar 1. Contoh Indonesia-Bahasa Lain

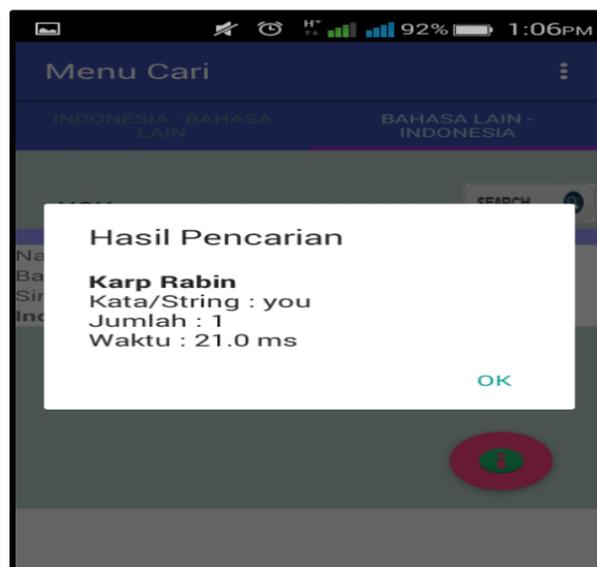


Gambar 2. Contoh Bahasa Lain-Indonesia

Gambar 3 dan 4 masing-masing menunjukkan informasi pencarian untuk kata “kamu” dan kata “you” sesuai dengan fungsi terjemahan yang di jalankan sistem.



Gambar 3. Informasi Pencarian kata “kamu”



Gambar 4. Informasi Pencarian kata “you”

3.3. Hasil Pengujian

Dari pengujian yang sudah dilakukan menggunakan ponsel pintar seperti disebutkan diatas, diperoleh hasil seperti yang terlihat pada tabel 3.

Tabel 3. Hasil Pengujian Pencarian Kata

No	Input	Running time (ms)	Jlh kata	Kejadian
1	nama	13.0	2	Match
2	a	16.0	18	Match
3	b	13.0	2	Match
4	iya	15.0	3	Match
5	mem	16.0	0	Mismatch
6	awak	16.0	1	Match
7	m	16.0	2	Match
8	c	14.0	0	Mismatch
9	n	16.0	8	Match
10	ka	14.0	1	Match
	Jumlah	149		
	Rata-Rata	14.9		

Berdasarkan tabel 3 diketahui bahwa rata-rata *running time* pencarian fungsi pencarian Bahasa Lain-Indonesia maupun fungsi pencarian Indonesia-Bahasa Lain untuk 10 kali percobaan dengan inputan yang berbeda-beda adalah 14.9 ms dengan 8 kejadian *match* dan 2 kejadian *mismatch*.

4. SIMPULAN

Adapun kesimpulan dari penelitian ini adalah:

- a. Berdasarkan penelitian yang telah dilakukan, algoritma *Rabin-Karp* berhasil diterapkan pada aplikasi kamus umum.
- b. Algoritma pada kamus Jawa Indonesia dapat melakukan pencarian kata dari bahasa Indonesia ke bahasa lain maupun dari bahasa lain ke bahasa Indonesia
- c. Bilangan basis yang digunakan mempengaruhi *running time* pencarian kata, semakin besar bilangan basis yang digunakan semakin lama waktu yang dibutuhkan.
- d. *running time* rata rata pencarian kata adalah 14.9 ms untuk 10 kali percobaan pencarian kata.

DAFTAR PUSTAKA

- [1] Putra, H.S.A. (2014). Keragaman budaya dalam keragaman bahasa, ancaman dan tantangan). Prosiding Seminar Tahunan Linguistik Universitas Pendidikan Indonesia. Bandung
- [2] Rassol, A., Tiwari A., Singla, A., Nilay, K. 2012. String Matching Methodologies : A Comparative Analysis. *International Journal of Computer Science and Information Technologies* 3(2):3394-3397

- [3] Hamza, S., Sarosa, M. & Purnomo, B. S. 2013. Sistem Koreksi Soal Essay Otomatis Dengan Menggunakan Metode Rabin Karp. *Jurnal EECCIS* 7(2):153-158
- [4] Pusat Bahasa. 2008. KBBI : Daring. (Online) <http://badanbahasa.kemdikbud.go.id/kbbi/index.php> (15 Desember 2015).
- [5] Charras, C & Lecroq, T. 2004. *Handbook of Exact String Matching Algorithms*. King's College Publication: London Aan Satriya, 2015, Sejarah Perkembangan Sistem Operasi Android, <http://www.capuraca.com/2015/01/sejarah-perkembangan-sistem-operasi-android.html>.
- [6] Ramdhani, P. R . 2012. Analisis Perbandingan Performansi Algoritma Zhu-Takaoka dan Algoritma *Karp-Rabin* Pada Pencarian Kata Di Rumah Buku Baca Sunda. Skripsi. Universitas Komputer Indonesia
- [7] Raharja, I. P. D. A. 2015. Perancangan dan Implementasi Sistem Penilaian Jawaban Esai Otomatis Menggunakan Algoritma Rabin-Karp. Skripsi. Universitas Udayana
- [8] Aan Satriya, 2015, Sejarah Perkembangan Sistem Operasi Android, <http://www.capuraca.com/2015/01/sejarah-perkembangan-sistem-operasi-android.html>.