TIN: Terapan Informatika Nusantara
Vol 1, No 3, Agustus 2020, Hal 126-132
ISSN 2722-7987 (Media Online)

Penerapan Algoritma RAITA pada Kamus Akronim Bahasa Indonesia Berbasis Android

Annisa Fitri Harahap, Guidio Leonarde Ginting*

Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Email: [1]annisa.fitri90@gmail.com, [2,*]guidio.leonarde@stmik-budidarma.ac.id

Abstrak−Bahasa Indonesia adalah bahasa melayu yang dijadikan sebagai bahasa resmi Repulik Indonesia dan bahasa persatuan bangsa Indonesia. Bahasa Indonesia diresmikan penggunaannya setelah Proklamasi Kemerdekaan Indonesia, tepatnya sehari sesudahnya, bersamaan dengan mulai berlakunya konstitusi. Bahasa Indonesia sebagai bahasa persatuan dan bahasa Negara di Negara Kesatuan Republik Indonesia ini memiliki fungsi yang sangat dominan dalam segala aspek di dalam kehidupan bermasyarakat. Akronim merupakan kependekan yang berupa gabungan huruf atau suku kata atau bagian lain yang ditulis dan dilafalkan sebagai kata yang wajar. Akronim disebut juga dengan singkatan merupakan kependekan dari suatu istilah yang dapat berupa gabungan dari awalan kata istilah. Dengan demikian, akronim merupakan singkatan dari suatu istilah atau nama yang dilafalkan sebagai satu kata. Akronim bahasa Indonesia memiliki jumlah kata yang sangat banyak, pada umumnya saat ini Akronim bahasa Indonesia masih dalam bentuk buku cetak. Adapun masalah yang sering dihadapi pengguna kamus akronim berbentuk buku cetak yaitu kesulitan untuk memperoleh hasil pencarian kata akronim bahasa Indonesia dengan cepat dan tepat karena harus dilakukan dengan cara membuka setiap lembaran buku tersebut untuk menemukan kata akronim yang dicari. Untuk mengatasi permasalahan yang telah dijelaskan diatas, pada penelitian ini penulis merancang aplikasi akronim bahasa Indonesia berbasis android. Agar dapat mempermudah untuk mengdapatkan hasil pencarian pada aplikasi akronim bahasa indonesia yang dirancang maka penulis menerapkan algoritma raita di dalam fitur pencarian. Algoritma Raita merupakan bagian dari algoritma exact string matching yaitu pencocokan string secara tepat dengan susunan karakter dalam string yang dicocokkan memiliki jumlah maupun urutan karakter dalam string yang memiliki kesamaan. Raita merancang sebuah algoritma dengan membandingkan karakter yang terakhir dari pola yang diawali dari karakter yang paling kanan yaitu dari “jendela”.

Kata Kunci: Raita, Android, Kata

Abstract−Indonesian is the language of Malay which is used as the official language of the Republic of Indonesia and the language of the unity of the Indonesian nation. The Indonesian language was inaugurated after the Proclamation of Indonesian Independence, to be precise the day after, at the same time as the enactment of the constitution. Indonesian as the language of unity and the language of the State in the Unitary State of the Republic of Indonesia has a very dominant function in all aspects of social life. An acronym is an abbreviation in the form of a combination of letters or syllables or other parts that are written and pronounced as normal words. Acronyms, also called abbreviations, are short for a term which can be a combination of prefixed terms. Thus, an acronym stands for a term or name that is pronounced as a single word. Indonesian acronyms have a very large number of words, in general currently Indonesian acronyms are still in printed book form. The problem that is often faced by acronym dictionary users in the form of printed books is the difficulty in obtaining search results for Indonesian acronym words quickly and precisely because it must be done by opening each sheet of the book to find the acronym word being searched for. To overcome the problems described above, in this study the authors designed an Android-based Indonesian acronym application. In order to make it easier to get search results on the designed Indonesian acronym application, the authors apply the Raita algorithm in the search feature. The Raita algorithm is part of the exact string matching algorithm, which is the matching of strings precisely to the arrangement of characters in the matched string which has the same number and sequence of characters in the string. Raita designed an algorithm by comparing the last character of the pattern starting from the rightmost character from the "window".

Keywords: Raita, Android, Word

1. PENDAHULUAN

Bahasa Indonesia adalah bahasa melayu yang dijadikan sebagai bahasa resmi Repulik Indonesia dan bahasa persatuan bangsa Indonesia. Bahasa Indonesia diresmikan penggunaannya setelah Proklamasi Kemerdekaan Indonesia, tepatnya sehari sesudahnya, bersamaan dengan mulai berlakunya konstitusi. Bahasa Indonesia sebagai bahasa persatuan dan bahasa Negara di Negara Kesatuan Republik Indonesia ini memiliki fungsi yang sangat dominan dalam segala aspek di dalam kehidupan bermasyarakat.

Akronim merupakan kependekan yang berupa gabungan huruf atau suku kata atau bagian lain yang ditulis dan dilafalkan sebagai kata yang wajar. Akronim disebut juga dengan singkatan merupakan kependekan dari suatu istilah yang dapat berupa gabungan dari awalan kata istilah. Dengan demikian, akronim merupakan singkatan dari suatu istilah atau nama yang dilafalkan sebagai satu kata.

Akronim bahasa Indonesia memiliki jumlah kata yang sangat banyak, pada umumnya saat ini Akronim bahasa Indonesia masih dalam bentuk buku cetak. Adapun masalah yang sering dihadapi pengguna kamus akronim berbentuk buku cetak, buku cetak yaitu kesulitan untuk memperoleh hasil pencarian kata akronim bahasa Indonesia dengan cepat dan tepat karena harus dilakukan dengan cara membuka setiap lembaran buku tersebut untuk menemukan kata akronim yang dicari.

Algoritma *Raita* merupakan bagian dari algoritma *exact string matching* yaitu pencocokan *string* secara tepat dengan susunan karakter dalam *string* yang dicocokkan memiliki jumlah maupun urutan karakter dalam *string* yang memiliki

Vol 1, No 3, Agustus 2020, Hal 126-132
ISSN 2722-7987 (Media Online)

kesamaan. Algoritma *Raita* digunakan untuk membandingkan karakter yang terakhir dari pola yang diawali dari karakter yang paling kanan yaitu dari "jendela" [1].

2. METODOLOGI PENELITIAN

2.1 String Matching

String Matching adalah proses pencariansemua kemunculan *query* yang selanjutnya disebut *pattern* ke dalam *string* yang lebih panjang atau teks. *String Matching* dirumuskan sebagai berikut :

$$x = x[0\ldots m-1] \quad (1)$$

$$y = y[0\ldots n-1] \quad (2)$$

Dimana:

x adalah *pattern*
m adalah panjang *pattern*
y adalah teks
n adalah panjang teks

Kedua *string* terdiri dari sekumpulan karakter yang disebut alfabet yang dilambangkan dengan ∑ dan mempunyai ukuran σ (tao). *Stringmatching* dibagi menjadi dua, yakni *exact matching* dan *heuristic*. *Exact Matching* digunakan untuk menemukan *pattern* yang berasal dari satu teks. Contoh pencarian *exact matching* adalah pencarian kata "pelajar" dalam kalimat "saya seorang pelajar" atau "saya seorang siswa". Sistem akan memberikan hasil bahwa kalimat pertama mengandung kata "pelajar" sedangkan kalimat kedua tidak, meskipun kenyataannya pelajar dan siswa adalah kata yang bersinonim. Algoritma *exact matching* diklasifikasi menjadi tiga bagianmenurut arah pencariannya, yaitu :

1. Arah pembacaan dari kiri ke kanan.
   Algoritma yang termasuk kategori ini adalah BruteForce, Kunth Morris Pratt
2. Arah pembacaan dari kanan ke kiri.
   Algoritma yang termasuk kategori ini adalah BoyerMoore, Turbo Boyer-Moore, Tuned Boyer Moore, dan Zhu Takaoka.
3. Arah pencarian yang ditentukan program.
   Algoritma yang termasuk kategori ini adalah algoritma Colussi, Crochemore-Perrin.

Heuristic *matching* adalah digunakan untuk menghubukangkan dua data terpisah ketika *exact matching* tidak mampu mengatasi karena pembatasan pada daat yang tersedia. *Heuristic matching* dilakukan dengan perhitungan *distance* antara *pattern* dengan teks. *Exact* dan *heuristic matching* memiliki kelemahan menemukan kata yang memiliki kemiripan makna tetapi berbeda tulisan[8].

2.2 Raita

Raita merupakan bagian dari algoritma *exact string matching* yaitu pencocokan *string* secara tepat dengan susunan karakter dalam *string* yang dicocokkan memiliki jumlah maupun urutan karakter dalam *string* yang sama. *Raita* merancang sebuah algoritma dengan membandingkan karakter yang terakhir dari pola yang diawali dari karakter paling kanan dari "jendela".Jika mereka cocok, kemudian karakter pertama dari pola teks paling kiri dari jendela juga dibandingkan. Jika mereka cocok, maka akan dibandingkan karakter tengah pola dengan karakter teks tengah jendela. Pada akhirnya, jika mereka benar-benar cocok, maka algoritma membandingkan karakter lain mulai dari pola karakter kedua ke karakter kedua terakhir, dan akan membandingkan dengan karakter tengah lagi[9].

Prosedur *Raita* dalam melakukan pencocokan *string* terdiri dari 5 (lima) tahapan. Berikut tahapan prosedur *Raita* dalam melakukan pencocolan *string*:

1. Buat tabel pergeseran pola yang dicari sebagai kata yang akan dicari pada teks.
2. Jika dalam proses pembandingan terjadi ketidakcocokan antara pasangan karakter pada akhir pola dengan karakter teks, pergeseran dilakukan sesuai nilai karakter teks pada tabel *BmBc*
3. Jika dalam proses pembandingan akhir pola terjadi ketidakcocokan lagi maka karakter akan digeser lagi sesuai tabel *BmBc*
4. Jika karakter akhir pola dengan karakter pada teks yang sedang dibandingkan cocok, maka posisi karakter pada pola dan teks akan memiliki nilai nol (0), dan dilanjutkan pencocokan pada karakter awal pola. Jika cocok maka dilanjutkan pencocokan dengan karakter tengah pola.
5. Jika akhir, awal dan tengah pola telah cocok. Pencocokan dilanjutkan dengan bagian kanan dari awal karakter pada pola, jika cocok maka dicocokkan pada bagian kanan tengah pola[9].

Vol 1, No 3, Agustus 2020, Hal 126-132
ISSN 2722-7987 (Media Online)

3. HASIL DAN PEMBAHASAN

Akronim bahasa Indonesia merupakan kependekan yang berupa gabungan huruf atau suku kata atau bagian lain yang ditulis dan dilafalkan sebagai kata yang wajar dalam bahasa Indonesia dengan kependekan dari suatu istilah yang dapat berupa gabungan dari awalan kata istilah.

Akronim bahasa Indonesia memiliki jumlah kata yang sangat banyak, pada umumnya saat ini Akronim bahasa Indonesia masih dalam bentuk bukucetak. Adapun masalah yang sering dihadapi pengguna kamus akronim berbentuk buku cetak yaitu kesulitan untuk memperoleh hasil pencarian kata akronim bahasa Indonesia dengan cepat dan tepat karena harus dilakukan dengan cara membuka setiap lembaran buku tersebut untuk menemukan kata akronim yang dicari.

Mengatasi permasalahan yang telah dijelaskan diatas mengenai kesulitan dalam mencari akronim bahasa Indonesia pada penelitian ini penulis merancang aplikasi akronim bahasa Indonesia berbasis *android* dengan menerapkan *algoritma raita* menggunakan editor *eclipse juno*. Algoritma *Raita* merupakan pencocokan *string* secara tepat dengan susunan karakter dalam *string* yang dicocokkan memiliki jumlah maupun urutan karakter dalam *string* yang memiliki kesamaan. *Raita* merancang sebuah algoritma dengan membandingkan karakter yang terakhir dari pola yang diawali dari karakter yang paling kanan yaitu dari "jendela"

Aplikasi akronim bahasa Indonesia berbasis *android* dengan menerapkan *algoritma raita* yang dirancang dan dibangun pada penelitian ini dapat mempermudah pengguna aplikasi untuk mendapatkan hasil pencarian berupa akronim bahasa Indonesia pada aplikasi akronim bahasa Indonesia yang dirancang dan dibangun pada penelitian ini.

Tabel 1. Contoh Akronim Bahasa Indonesia

No	Akronim	Kepanjangan
1	TVRI	Televisi Republik Indonesia
2	KPK	Komisi Pemberantasan Korupsi
3	TNI	Tentara Nasional Indonesia
4	KPU	Komisi Pemilihan Umum
5	PSSI	Persatuan Sepakbola Seluruh Indonesia

2.1 Penerapan Algoritma *Raita*

Penerapan algoritma *raita* pada penelitian ini merupakan tahapan yang dilakukan untuk penyelesaian masalah pencarian akronim bahasa Indonesia. Masalah pencarian yang diselesaiakan pada penelitian ini yaitu melakukan pencarian atau pencocokan *pattern* terhadap *text*. Sebagai contoh penyelesaian masalah pencocokan *pattern* dengan *text* dengan algoritma *raita* penulis menggunakan TVRI sebagai *pattern* dan DAFTAR SIARAN TVRI sebagai *text.* Maka diketahui pula perhitungannya sebagai berikut :

Text = DAFTAR SIARAN TVRI
Pattern = TVRI

Diketahui bahwa :
Text = Target Pencocokan *Pattern*
Pattern = Panjang pola

Maka :
Pattern = 4
Dibuatlah table *BmBc* untuk melakukan perhitungan dengan persamaan sebagai berikut:
Pattern – 2 .....(1)
4 – 2 = 2
Mencari nilai *BmBc* (a)
Pattern – 1 – i ..... (2)
Pencarian nilai *BmBc (a)* dengan menggunakan rumus persamaan berikut ini :
Pattern – 1 – i ..... (2)
4 – 1 – 0 = 3 maka nilai diletakkan pada indeks ke-0 dengan krakter T
4 – 1 – 1 = 2 maka nilai diletakkan pada indeks ke-1 dengan karakter V
4 – 1 – 2 = 1 maka nilai diletakkan pada indeks ke-2 dengan karakter R
Untuk nilai karakter I adalah 4 sesuai dengan panjang pola *pattern* yang digunakan pada penelitian ini.

Tabel 2. Perhitungan table *BmBc*

I	0	1	2	3
P	T	V	R	I
BmBc(a)	3	2	1	4

TIN: Terapan Informatika Nusantara
Vol 1, No 3, Agustus 2020, Hal 126-132
ISSN 2722-7987 (Media Online)

Karena abjad yang tidak ada pada tabel maka diinisialisasikan dengan tanda (*) kemudian nilai nya sesuai dengan panjang pola. Maka, untuk perhitungan algoritma *raita* sesuai tabel *BmBc* adalah sebagai berikut :

Tabel 3. Hasil *BmBc* (a)

Text	R	T	V	*
BmBc	1	3	2	4

Perhitungan mencari nilai hasil *BmBc (a)* dengan menggunakan rumus persamaan
Pattern – 2 ..... (1)
4– 2 = 2
Maka, dapatlah hasil tabel *BmBc (a)* menjadi 2 karakter.
Text = DAFTAR SIARAN TVRI
Pattern = TVRI
Langkah selanjutnya adalah melakukan pencocokan karakter menggunakan algoritma *raita* dengan tahap-tahap berikut ini :
1. Tahap pertama, yaitu mencocokan karakter I dengan T.

Tabel 4. Tahap Pertama

Karakter	D	A	F	T	A	R		S	I	A	R	A	N		T	V	R	I
	T	V	R	I														

Karena karakter I dengan T tidak sama, maka dilakukan persegesaran sebesar nilai *BmBc* T yaitu sebanyak 3 (tiga) langkah.

2. Tahap kedua, yaitu mencocokkan karakter I dengan (Spasi)

Tabel 5. Tahap Kedua

Karakter	D	A	F	T	A	R		S	I	A	R	A	N		T	V	R	I
				T	V	R	I											

Karena karakter I dengan (Spasi) tidak sama, maka dilakukan persegesaran sesuai jumlah karakter *pattern*, yaitu sebanyak 4 (empat) langkah.

3. Tahap ketiga, yaitu mencocokkan karakter I dengan R

Tabel 6. Tahap Ketiga

Karakter	D	A	F	T	A	R		S	I	A	R	A	N		T	V	R	I
								T	V	R	I							

Karena karakter I dengan R tidak sama, maka dilakukan persegesaran sebesar nilai *BmBc* R yaitu sebanyak 1 (satu) langkah.
4. Tahap keempat, yaitu mencocokkan karakter I dengan A

Tabel 7. Tahap Keempat

Karakter	D	A	F	T	A	R		S	I	A	R	A	N		T	V	R	I
									T	V	R	I						

Karena karakter I dengan A tidak sama, maka dilakukan persegesaran sesuai jumlah karakter *pattern*, yaitu sebanyak 4 (empat) langkah.
5. Tahap kelima, yaitu mencocokkan karakter I dengan V

Tabel 8. Tahap Kelima

Karakter	D	A	F	T	A	R		S	I	A	R	A	N		T	V	R	I
													T	V	R	I		

Karena karakter I dengan V tidak sama, maka dilakukan persegesaran sebesar nilai *BmBc* V yaitu sebanyak 2 (dua) langkah.

6. Tahap keenam, yaitu mencocokkan karakter I dengan I

Vol 1, No 3, Agustus 2020, Hal 126-132
ISSN 2722-7987 (Media Online)

Tabel 9. Tahap Keenam

Karakter	D	A	F	T	A	R		S	I	A	R	A	N		T	V	R	I
															T	V	R	I

Dapat dilihat pada pencocokan karakter I dengan I, R dengan R, V dengan V, T dengan T adalah sama, Maka pencocokan karakter berhenti.

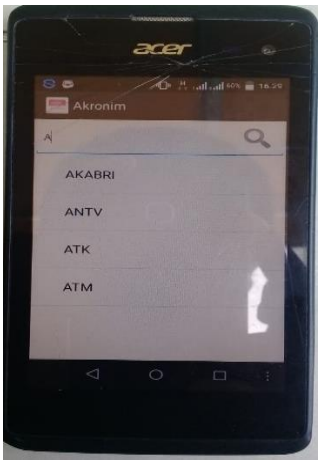
3.2 Implementasi Program

Tampilan program merupakan *interface* atau tampilan antar muka yang berfungsi sebagai media komunikasi antara *user* atau pengguna dengan aplikasi akronim bahasa Indonesia berbasis *android* yang dibangun oleh penulis. Tampilan awal aplikasi akronim bahasa Indonesia berbasis *android* yang dibangun pada penelitian ini terdiri dari halaman menu utama (Akronim, Tentang Aplikasi, dan *Exit),* halaman cari akronim, hasil pencarian, tentang aplikasi.

1. Halaman Menu Utama
   Halaman menu utama berfungsi sebagai media penghubung antara *user* dengan aplikasi akronim bahasa Indonesia berbasis *android* yang dibangun oleh penulis pada penelitian ini. Pada menu utama tersedia 3 (tiga) pilihan menu yaitu Akronim, Tentang Aplikasi, dan *Exit*. Adapun *screenshoot* gambar untuk tampilan menu utama pada aplikasi akronim bahasa Indonesia berbasis *android* yang dibangun oleh penulis pada penelitian ini dapat dilihat pada gambar berikut :

Gambar 1. Halaman Menu Utama

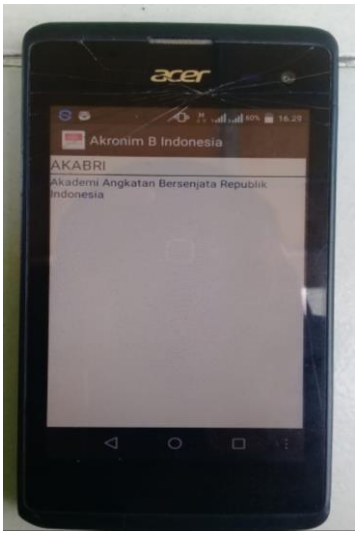
2. Halaman Cari Akronim
   Halaman cari akronim berfungsi untuk melakukan pencarian akronim bahasa Indonesia pada aplikasi akronim bahasa Indonesia berbasis *android* yanga dibangun pada penelitian ini. Adapun *screenshoot* gambar untuk halaman cari akronim pada aplikasi akronim bahasa Indonesia berbasis *android* dapat dilihat pada gambar berikut:



Gambar 2. Halaman Cari Akronim

Vol 1, No 3, Agustus 2020, Hal 126-132
ISSN 2722-7987 (Media Online)

3. Halaman Hasil Pencarian
   Halaman hasil pencarian berfungsi untuk menampilkan hasil pencarian akronim bahasa Indonesia pada aplikasi akronim bahasa Indonesia berbasis *android*. Adapun *screenshoot* gambar untuk hasil pencarian pada penelitian ini dapat dilihat pada gambar berikut :



Gambar 3. Halaman Hasil Pencarian

3.3 Pengujian Program

Hasil pengujian program merupakan tampilan hasil pencarian informasi akronim bahasa indoensia pada aplikasi akronim bahasa Indonesia berbasis *android* yang dibangun oleh penulis pada penelitian ini. Adapun hasil pengujian program ini dapat dilihat pada tabel berikut :

Tabel 10. Hasil Pengujian

No	Cari Akronim	Hasil Pencarian
1	AKABRI	Akademi Angkatan Bersenjata Republik Indonesia
2	ATM	Anjungan Tunai Mandiri
3	ATK	Alat Tulis Kantor
4	ANTV	Andalas Televisi
5	TVRI	Televisi Republik Indonesia
6	SCTV	Surya Citra Televisi
7	SKCK	Surat Keterangan Catatan Kepolisian
8	KRS	Kartu Rencana Studi
9	KTP	Kartu Tanda Penduduk
10	KTM	Kartu Tanda Mahasiswa

Berdasarkan hasil pengujian tabel diatas dapat disimpulkan bahwa pencarian akronim bahasa Indonesia dapat menggunakan *android* dan algoritma *raita.* Aplikasi akronim bahasa Indonesia mampu meminimalisir yang dibutuhkan untuk memperoleh hasil pencarian akronim bahasa Indonesia.

4. KESIMPULAN

Berdasarkan hasil penelitian ini penulis memaparkan beberapa kesimpulan seperti berikut ini:

1. Aplikasi akronim bahasa Indonesia pada penelitian ini dibangun oleh penulis menggunakan *eclipse juno* dan dapat dioperasikan pada *emulator eclipse juno* dan *smartphone android.*
2. Penerapan algoritma *raita* pada aplikasi akronim bahasa Indonesia yang dibangun pada penelitian ini mempermudah *user* untuk mendapatkan hasil pencarian akronim bahasa Indonesia.
3. Aplikasi akronim bahasa Indonesia yang dibangun pada penelitian ini meminimalisir waktu yang dibutuhkan untuk memperoleh hasil pencarian akronim bahasa Indonesia.

Vol 1, No 3, Agustus 2020, Hal 126-132
ISSN 2722-7987 (Media Online)

REFERENCES

Al Bahra Bin Ladjamudin, Analisis dan Desain Sistem Informasi. Tangerang: Graha Ilmu, 2005.

Kusrini, *Konsep dan Aplikasi Sistem Pendukung Keputusan*. Yogyakarta: Andi, 2007.

Riswaya, "Aplikasi Pinjaman Pembayaran Secara Kredit Pada Bank Yudha Bhakti," vol. Vol. 8, 2016.

Saut Dohot Siregar, "Aplikasi penerjemah kalimat bahasa indonesia ke bahasa simalungun dengan algoritma berry – ravindran," *Jurnal Teknovasi*, vol. Volume 04, 2017.

Muhammad Zarlis dan Handrizal, *Algoritma Dan Pemrograman*. Medan: USU Press, 2007.

Mesran, "Implementasi algoritma brute force dalam pencarian data katalog buku perpustakaan," *Majalah Ilmiah*, volume : III, 2014.

Charras C. & Lecroq T, *Handbook of Exact String-Matching Algorithms*. London: King's College Publications, 2004.

Pemodelan Berbasis UML (Unified Modeling Language) dengan Strategi Teknik Orientasi Objek User Centered Design(UCD) dalam Sistem Administrasi Pendidikan,.

R. A. S and M. Shalahuddin, *Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung: Informatika, 2014.

P. P. Widodo, *Menggunakan UML untuk Memodelkan Analisis & Desain Sistem Berorientasi*. Bandung: Informatika, 2011.

Nasruddin Safaat H, *Pemograman Aplikasi Mobile Smartphone Dan Tablet PC Berbasis Android*. Bandung: Informatika Bandung, 2015.

Alfa Satyaputra dan Eva Maulina Aritonang, *Java For Beginners with Eclipse 4.2 Juno*. Jakarta: PT Elex Media Komputindo, 2012.

Uli Fitrianti dan Mutammimul Ula, "Implementasi algoritma levenshtein distance dan algoritma knuth morris pratt pada aplikasi asmaul husna berbasis android," *Jurnal Sistem Informasi*.