

Perancangan Aplikasi Katalog Museum Dengan Menerapkan Algoritma Zhutakaoka(Studi Kasus : Museum Negeri Provinsi Sumatra Utara

Fitri Yani Dalimunthe

Program Studi Teknik Informatika, Universitas Budi Darma, Sumatera Utara, Indonesia

Email: yanip2233@gmail.com

Email Penulis Korespondensi: yanip2233@gmail.com

Abstrak–Katalog museum sangat memudahkan orang untuk melihat daftar beberapa informasi yang disajikan, katalog dapat dikatakan sebagai sebuah media informasi yang berisi tentang daftar koleksi pada suatu museum sehingga masyarakat dapat mengetahui katalog yang akan mereka cari.

String matching menggunakan Algoritma Zhutakaoka untuk melakukan pencocokan dengan pattern yang akan dicari dari makna katalog, sehingga pencocokan dalam pemilihan patern dapat dilihat kecocokkannya. Diharapkan melalui Algoritma Zhutakaoka dapat menyelesaikan pencocokkan dalam pencarian katalog museum, sehingga dapat membantu masyarakat dalam pencarian katalog museum.

Kata Kunci: String Matching; Katalog Museum; Algoritma Zhutakoaka

Abstract–The museum catalog makes it very easy for people to see a list of some of the information presented, the catalog can be said as an information medium that contains a list of collections in a museum so that people can find out the catalog they are looking for. String matching uses the Zhutakaoka Algorithm to match the pattern to be searched from the meaning of the catalog, so that the match in the pattern selection can be seen. It is hoped that the Zhutakaoka Algorithm can complete the matching in the search for the museum catalog, so that it can help the public in searching for the museum catalog.

Keywords: String Matching; Museum Catalog; Zhutakoaka Algorithm

1. PENDAHULUAN

Museum merupakan sebuah lembaga yang bersifat tetap, tidak mencari keuntungan, melayani masyarakat, terbuka untuk umum, pendidikan dan hiburan suatu barang-barang pembuktian manusia dan lingkungannya. Di Indonesia museum terlihat ramai hanya pada saat liburan, studi wisata dan penelitian karya ilmiah yang dilakukan oleh pelajar. Permasalahan tersebut juga terjadi di Museum Negeri Sumatra Utara.

Museum negeri provinsi Sumatera Utara atau biasanya dikenal dengan nama gedung arca, berlokasi di jalan H.M.Joni no. 15, medan yang berjarak sekitar 3 km dari bandara polonia, 25 km dari pelabuhan laut belawan, dan sekitar 3 km dari pusat pemerintahan kantor Gubernur Sumatera Utara. Koleksi yang terdapat pada museum merupakan barang-barang peninggalan sejarah kebudayaan yang terdapat di daerah Sumatera Utara. Adapun peninggalan seperti arca-arca peninggalan lain seperti zaman Hindu-Budha, peninggalan batu nisan, Al-Qur'an, replika masjid azizi ada disini, diorama kehidupan persejarah, dan replika kehidupan kota medan zaman dulu. Jumlah koleksi yang terdapat pada Museum Negeri Provinsi Sumatera Utara terbilang banyak, sehingga sulit bagi para pegawai untuk menghafal ataupun mengingat daftar koleksi tersebut, sehingga para pegawai sering kesulitan menjawab pertanyaan pengunjung. Pengunjung umum kurang begitu tertarik mengunjungi museum karena para pengunjung kurang begitu peduli dengan budaya dan sejarah Indonesia. Masyarakat menilai museum adalah tempat yang membosankan dan kurang menarik untuk dikunjungi karena sebagian orang beranggapan bahwa museum sebagai tempat yang kuno jauh dari sentuhan teknologi. Sebagai upaya untuk mengunjungi museum, perlu diterapkan teknologi tepat guna, sehingga koleksi museum menjadi lebih informatif. Saat ini museum-museum di Indonesia memberikan informasi hanya berupa sejarah singkat mengenai objek benda tersebut, sehingga dibutuhkan seorang pemandu wisata untuk menyampaikan informasi tambahan. Untuk mempermudah para pemandu dalam museum maka diperlukan suatu aplikasi yang berisi daftar koleksi dari museum tersebut dimana aplikasi tersebut menyajikan informasi koleksi barang museum dalam bentuk katalog. Selain itu, untuk mempermudah proses pencarian dalam aplikasi katalog koleksi museum maka diperlukan algoritma untuk melakukan pencocokan string.

Berdasarkan penelitian terdahulu jurnal Perbandingan Dekstop. Algoritma Zhutakaoka hanya bekerja menggunakan dua text karakter untuk menghitung pergeseran karakter berdasarkan bad-character, Untuk mencari ketidakcocokan atau mencari seluruh text menggunakan preprocessing hashing. Hal ini efektif untuk pencarian string dua dimensi dan pencarian pattern di lakukan mulai dari tiap baris mulai dari baris ke 0 sampai berakhir pada baris nI-mI.

Berdasarkan penelitian sebelumnya Algoritma Zhutakaoka merupakan salah satu algoritma pencocokan string (String Matching). Algoritma Boyer moore dan Algoritma Zhutakaoka memiliki ciri yang sama dalam proses pencarian string, yaitu terdapat pada tahap preprocessing Algoritma Horspool dan Algoritma Zhutakaoka dalam Pencarian String berbasis Right-to-left scan, Bad character rule dan Good suffix. Sementara beda dari algoritma tersebut terletak pada tahap Bad character rule. Dalam Algoritma Boyer moore hanya terdiri dari Array satu dimensi sedangkan Algoritma Zhutakaoka melakukan pencocokan karakter dari sebelah kanan pattern menggunakan array dua dimensi untuk menghitung nilai pergeseran sehingga mendapatkan informasi lebih banyak.

2. METODOLOGI PENELITIAN

2.1 String Matching

String matching adalah proses pencarian semua kemunculan query yang selanjutnya di sebut Pattern kedalam string yang lebih panjang (text). Pattern di lambangkan dengan $x=x[0\dots m-1]$ dan panjangnya adalah m . Teks di lambangkan dengan $y=y[0\dots n-1]$ dan panjangnya adalah n . kedua string terdiri dari sekumpulan karakter yang disebut alphabet yang di lambangkan dengan Σ dan mempunyai ukuran $\sigma[3]$. Algoritma string matching mempunyai tiga komponen yaitu:

1. Pattern, yaitu deretan karakter yang di cocokkan dengan teks
2. Teks, yaitu tempat pencocokkan pattern di lakukan
3. Alfabet, yang berisi semua simbol yang di gunakan oleh bahasa pada text pattern

String merupakan bagian penting dari sebuah proses pencarian string (string searching) dalam sebuah dokumen. hasil dari pencarian sebuah string dalam dokumen tergantung dari teknik atau cara pencocokkan string yang di gunakan.

2.2 Teknik String Matching

Secara garis besar teknik string matching dalam pencocokkan string dibedakan menjadi 2 yaitu :

1. Exact string matching

Exact string matching merupakan pencocokkan string secara tepat dengan susunan karakter dalam string yang dicocokkan memiliki jumlah maupun urutan karakter dalam string yang sama. Bagian algoritma ini bermanfaat jika pengguna ingin mencari string dalam dokumen yang sama persis dengan string masukan. Beberapa contoh algoritma exact string matching yang mengemukakan antara lain :

- a. Brute Force

Brute Force membandingkan karakter per karakter sampai ditemukannya pola yang dicari dari awal string sampai dengan akhir string.

- b. Knuth-Morris-Pratt

Knuth-Morris-Pratt membandingkan karakter per karakter sampai ditemukannya pola yang dicari dari awal string sampai dengan akhir string dari arah kiri ke arah kanan

- c. Boyer Moore

Boyer moore melakukan pencocokkan karakter yang dimulai dari kanan ke kiri.

2. Inexact string matching

Inexact string matching merupakan pencocokkan string secara samar, maksudnya pencocokkan string dimana string yang dicocokkan memiliki kemiripan dimana keduanya memiliki susunan karakter yang berbeda tetapi string-string tersebut memiliki kemiripan baik kemiripan tekstual atau penulisan atau kemiripan ucapan (phonetic string matching). Metode Inexact string matching diarahkan untuk mencari nilai dari beberapa string yang mendekati dan tidak hanya menghasilkan cocok atau tidak cocok.

2.3 Algoritma Zhutakaoka

Algoritma Zhutakaoka diciptakan oleh R.F Zhu dan T.Takaoka pada tahun 1989. Algoritma ini menggunakan dua text karakter untuk menghitung pergeseran karakter berdasarkan bad-character. untuk mencari ketidak cocokan atau mencari seluruh text menggunakan processing hashing. algoritma Zhutakaoka menjelaskan bahwa pencarian pattern mulai dari tiap baris mulai dari baris ke 0 dan berakhir pada baris ke $n-1$. Zhu dan takaoka merancang sebuah algoritma yang dapat bekerja menggunakan pergeseran bad-character untuk dua karakter text secara berturut-turut. selama fase pencarian kecocokan karakter di lakukan dari kanan ke kiri dan ketika proses berada di posisi $y[j\dots j+m-1]$ dan ketidak cocokan terjadi di antara $x[m-k]$ dan $y[j+m-k]$ ketika $x[m-k+1\dots m-1]=y[j+m-k+1\dots j+m-1]$ pergeseran dilakukan dengan pergeseran bad-character untuk text karakter $y[j+m-2]$ dan $y[j+m-1]$. tabel good-suffix dari algoritma Boyer-Moore juga digunakan untuk menghitung persamaan [1].

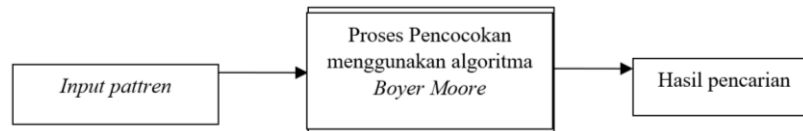
3. HASIL DAN PEMBAHASAN

3.1 Hasil

Katalog museum merupakan sejenis dari gambar atau lembaran buku, katalog perpustakaan ini dibangun dan di kembangkan untuk mengenalkan sejarah buku untuk masyarakat. Tujuan utama dari katalog museum ini untuk memudahkan pengetahuan para pelajar dan masyarakat katalog museum ini memiliki koleksi persejarah. Yang telah di temukan pada sebuah pencarian di *smartphone* berbasis *android*, dan bukanlah hal yang mudah untuk pelajar dalam pencarian sejarah suku jika di lakukan pencarian dari *Smartphone* berbasis *android* yang terbatas maka akan sulit untuk di tampilkan seluruh informasi sejarah suku. Belum lagi beberapa katalog yang memiliki nama yang sama. Akan memerlukan waktu yang lama untuk mencari informasi yang di inginkan. Katalog yang akan di pelajari dan membutuhkan area yang besar dan sulit di tampilkan sehingga perlu mengembangkan sistem dengan merancang Aplikasi katalog museum berbasis *android* yang memiliki fitur pencarian yang baik untuk meminimalkan waktu yang dibutuhkan untuk menemukan hasil pencarian dalam katalog yang akan di pelajari. Maka untuk pelajar penting dalam

pembelajaran dilingkungan sekolah atau diluar sekolah. Para pelajar dan masyarakat perlu mengetahui katalog museum sejarah. Jika pelajar dan masyarakat tidak mengetahui katalog museum maka pelajar akan mengalami kesulitan dalam mencari katalog museum..

Berdasarkan uraian masalah tersebut penulis merancang Aplikasi katalog perpustakaan yang berbasis *Android* menggunakan *Eclipse juno Software*. Dalam penelitian ini penulis menerapkan Algoritma Zhutakaoka untuk menyederhanakan proses pencarian katalog museum dalam Aplikasi katalog museum yang berbasis *android*. Aplikasi katalog museum berbasis *android* yang di rancang dalam penelitian ini dapat dioperasikan pada *smartphone* yang di akhiri oleh sistem operasi berbasis *android* sehingga penggunaan Aplikasi katalog perpustakaan berbasis *android* ini sangat mudah karena tidak memiliki ukurannya terlalu besar sehingga bisa digunakan di mana saja dan kapan saja.



Gambar 1: Tampilan Prosedur

3.2 Penerapan Algoritma Zhu-Takaoka

Penerapan algoritma Zhu-Takaoka pada aplikasi katalog museum dapat dilakukan dengan menentukan teks dan pattern. Pada penelitian ini zhu-takaoka diterapkan untuk melakukan pencarian koleksi museum, maka yang menjadi teks dalam penelitian ini adalah nama koleksi dari museum tersebut. Langkah pencarian diuraikan dibawah ini :

Teks : BERAGAM PUSAKA KERIS
 Pattern : KERIS

Langkah pertama, membuat dan menentukan table *ztBc* (Zhutakaoka *Bad Character*) dan *bmGs* (Boyer moore *Good suffix*) kemudian melakukan pencocokan.

1. Membuat Tabel ZtBc

Langkah awal adalah menentukan table *bmBc*. Untuk memudahkan menentukan table *bmBc* adalah dengan menemukan tabel *ztBc* untuk mengidentifikasi setiap perpindahan.

Pada tahap ini untuk mencari nilai OH: $M - 1 - i$

Keterangan:

i: nilai index

M: pattern /karakter(N)

Tabel 1. bmBc

Index	0	1	2	3	4
Karakter	K	E	R	I	S
Nilai OH	4	3	2	1	0

1. Untuk mendapatkan nilai K yaitu: $5 - 1 - 0$ maka dapat nilai K adalah 4.
2. Untuk mendapatkan nilai E yaitu: $5 - 1 - 1$ maka dapat nilai E adalah 3
3. Untuk mendapatkan nilai R yaitu: $5 - 1 - 2$ maka dapat nilai R adalah 2.
4. Untuk mendapatkan nilai I yaitu: $5 - 1 - 3$ maka dapat nilai I adalah 1.
5. Untuk mendapatkan nilai S yaitu: $5 - 1 - 4$ maka dapat nilai S adalah 0.

Jika ada karakter yang sama muncul maka diambil nilai terkecil pada OH.

Tabel 2: bmBc

Index	0	1	2	3	4
Karakter	K	E	R	I	S
Nilai OH	4	3	2	1	0

Tabel *bmBc* akhir merupakan acuan untuk menentukan tabel *ztBc*

2. Mencari nilai ZtBc (Zhu Takaoka Bad Character)

Langkah selanjutnya adalah masukkan semua karakter yang terdapat pada sumber teks dan buatlah menjadi dua dimensi, karakter yang dimasukkan di urutkan sesuai abjad tak berulang.

Tabel 3 ztBc

	E	I	K	R	S	*
E						
I						
K						
R						
S						
*						

Setelah tabel dua dimensi terbentuk ,langkah selanjutnya adalah mengisi tabel dengan cara melihat karakter pada baris dan karakter pada baris dan karakter pada kolom. Untuk mengisi nilai pada tabel tersebut , ada beberapa aturan sebagai berikut:

1. Jika pola yang dicari ada, maka masukkan nilai OH ke karakter kolom.
2. Jika pola yang dicari tidak ada maka masukkan panjang *pattern* .
3. Jika karakter pada suatu kolom yang di tuju adalah karakter pertama pada *pattern*, maka masukkan nilai OH karakter tersebut ,tanpa melihat karakter baris.

Tabel 4 ztBc

	E	I	K	R	S	*
E	5	5	5	5	5	5
I	5	5	1	5	2	5
K	5	3	5	5	5	5
R	3	5	5	5	5	5
S	5	5	5	5	5	5
*	5	5	5	5	5	5

3. Membuat tabel bmGs

Setelah mendapatkan hasil akhir pada tabel *bmZc*, maka dilanjutkan dengan membuat tabel *suffix* dari *pattern*. Berikut adalah tabel bmGs.

Tabel 5 bmGs

<i>Index</i>	0	1	2	3	4
<i>Karakter</i>	K	E	R	I	S
<i>Nilai MH</i>	5	5	5	5	1

Keterangan:

1. Index adalah index dari *pattern*.
2. Karakter adalah *pattern* yang akan dicari.
3. Nilai Match Heuristic (MH) adalah nilai pergeseran *good suffix*, nilai MH nantinya akan digunakan ketika ditemukan kecocokan pada saat string karakter pertama atau lebih.

Untuk mengisi nilai MH langkah pertama adalah membuat tabel *suffix* dari kanan ke kiri.

4. Proses Pencocokan

Langkah terakhir adalah melakukan pencocokan *pattern* dengan sumber teks berikut:

Teks : BERAGAM PUSAKA KERIS

Pattern : KERIS

Percobaan ke-1

<i>Index</i>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
<i>Teks</i>	B	E	R	A	G	A	M		P	U	S	A	K	A		K	E	R	I	S
<i>Pattern</i>	K	E	R	I	S															

Pada percobaan ke-1 dapat dilihat bahwa karakter akhir pada *pattern* yaitu karakter G sejajar dengan karakter S pada sumber teks. Artinya, pada percobaan ke-1 terjadi ketidakcocokan. Maka dilakukan pergeseran sebanyak 5. Sesuai pada tabel *ZtBc* dengan pola Y A.

<i>Index</i>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
<i>Teks</i>	B	E	R	A	G	A	M		P	U	S	A	K	A		K	E	R	I	S
<i>Pattern</i>						K	E	R	I	S										

Percobaan ke2

Pada percobaan ke-2 dapat dilihat bahwa karakter akhir pada *pattern* yaitu karakter U sejajar dengan karakter S pada sumber teks. Artinya, pada percobaan ke-2 terjadi ketidakcocokan. Maka dilakukan pergeseran sebanyak 6. Sesuai pada tabel *ZtBc* dengan pola E D.

Percobaan ke-3

<i>Index</i>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
<i>Teks</i>	B	E	R	A	G	A	M		P	U	S	A	K	A		K	E	R	I	S
<i>Pattern</i>										K	E	R	I	S						

Pada percobaan ke-3 dapat dilihat bahwa karakter akhir pada *pattern* yaitu karakter sejajar dengan karakter S pada sumber teks. Artinya, pada percobaan ke-3 terjadi ketidakcocokan. Maka dilakukan pergeseran sebanyak 5. Sesuai pada tabel *ZtBc* dengan pola A G.

Percobaan ke-4

<i>Index</i>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
<i>Teks</i>	B	E	R	A	G	A	M		P	U	S	A	K	A		K	E	R	I	S
<i>Pattern</i>															K	E	R	I	S	

Pada percobaan ke-4 dapat dilihat bahwa karakter akhir pada *pattern* yaitu karena jumlah pergeseran lebih besar dibandingkan dengan sisa karakter pada teks, maka pencocokan selesai.

Teks : PAMERAN PATUNG

Pattern : PATUNG

Pada tahap ini untuk mencari nilai OH: M-1-i

Keterangan :

i : Nilai index

M : Pattern/karakter (N)

Tabel 6 bmBc

index	0	1	2	3	4	5
<i>Pattern</i>	P	A	T	U	N	G
<i>Occurrence Heuristic (OH)</i>	5	4	3	2	1	0

1. Untuk mendapatkan nilai P yaitu: 6-1-0 maka dapat nilai P adalah 5
2. Untuk mendapatkan nilai P yaitu: 6-1-1 maka dapat nilai P adalah 4
3. Untuk mendapatkan nilai P yaitu: 6-1-2 maka dapat nilai P adalah 3
4. Untuk mendapatkan nilai P yaitu: 6-1-3 maka dapat nilai P adalah 2
5. Untuk mendapatkan nilai P yaitu: 6-1-4 maka dapat nilai P adalah 1
6. Untuk mendapatkan nilai P yaitu: 6-1-5 maka dapat nilai P adalah 0

Tabel 7 Nilai OH dan MH pada *pattern* penyelesaian :

index	0	1	2	3	4	5
<i>Pattern</i>	P	A	T	U	N	G

<i>Occurrence Heuristic (OH)</i>	5	4	3	2	1	0
<i>MaathHeuristik(MH)</i>	6	6	6	6	6	1

Tahapan pencocokan *string* untuk pencarian karakter *pattern* PATUNG pada *text* PAMERAN PATUNG adalah sebagai berikut ini :

1. Tahap Pertama

Pada tahap pertama, pencocokan *string pattern* dimulai dengan deretan *string text* yang pertama “ERA “.

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Teks	P	A	M	E	R	A	N		P	A	T	U	N	G
Pattren	P	A	T	U	N	G								

Pada tahap ini karakter G dengan A tidak sama dengan karakter *pattern* yang lainnya, sehingga bergeser sebanyak jumlah karakter *pattern* yaitu 4 langkah.

2. Tahap Kedua

Pada tahap kedua yaitu pencocokan *string* PATUNG dengan deretan *text* “PA “.

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Teks	P	A	M	E	R	A	N		P	A	T	U	N	G
Pattren					P	A	T	U	N	G				

Pada tahap ini karakter G dengan A tidak sama, dengan karakter *pattern* yang lainnya, sehingga bergeser sebanyak jumlah karakter *pattern* yaitu 4 langkah.

3. Tahap Ketiga

Pada tahap ketiga dilakukan pencocokan *string pattern* dengan deretan *string text* “PATUNG“.

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Teks	P	A	M	E	R	A	N		P	A	T	U	N	G
Pattren									P	A	T	U	N	G

Pada tahap ini semua karakter *pattern* sesuai dengan karakter *text* yang sejajar dengan karakter *pattern*, sehingga pergeseran karakter *pattern* berhenti.

4. KESIMPULAN

Berdasarkan hasil akhir pemecahan masalah pencarian Katalog museum pada penelitian maka penulis menguraikan beberapa kesimpulan. Adapun kesimpulan-kesimpulan tersebut antara lain sebagai berikut :

1. Implementasi algoritma Zhu Takaoka dapat mempercepat proses pencarian isi pencarian Katalog Museum
2. Aplikasi pencarian Katalog Museum pada penelitian ini dibangun menggunakan editor eclipse .
3. Aplikasi pencarian Katalog Museum yang dibangun pada penelitian ini dapat mempermudah pencarian daftar Katalog Museum.

UCAPAN TERIMAKASIH

Terima kasih disampaikan kepada pihak-pihak yang telah mendukung terlaksananya penelitian ini

REFERENCES

[1] <https://www.kompasiana.com/annisamega/588321f3cc92731105931d89/perkembangan-trend-fashion-di-indonesia?page=all>
 [2] Mesran Aan, Implementasi Algoritma Brute Force Dalam Pencarian Data Katalog Buku Perpustakaan, Informasi dan Teknologi Ilmiah (INTI), 2014.
 [3] Ratih, Implementasi Algoritma Raita Search Pada Aplikasi Filsafat Berbasis Android, KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer), 2018.
 [4] Fince Tinus Waruwu, "Implementasi Algoritma Knuth Morris Pratt Pada Aplikasi Kamus Istilah Latin Flora dan Fauna Berbasis Android," 2014.
 [5] ZARIANI MUTIA SYARA, Implementasi Dan Perbandingan Algoritma Smith Dan Algoritma Raita Pada Pencarian Kata, Skripsi, Universitas Sumatera Utara, 2017.
 [6] Hasan Abdurrahman dan Asep Ririh Riswaya, Aplikasi Pinjaman Pembayaran Secara Kredit Pada Bank, Jurnal Computech, 2014.

- [7] Muhammad Iqbal Suryatama Dan Bahrul Ilmi, Pra Rancangan Pabrik Garment Kemeja Katun Kasual Lengan Panjang Pria Dengan Kapasitas Produksi 2.435.000 PCS/Tahun, Tugas Akhir, Fakultas Teknologi Industri Universitas Islam Indonesia Yogyakarta, 2018.
- [8] Nadia, Fashion bagian terpenting dari Kehidupan Sehari-hari, <http://Elvierameldawati.Blogspot.Com>, Di akses tanggal 13 Juni 2008
- [9] P. P. Widodo, Menggunakan UML untuk Memodelkan Analisis & Desain Sistem Berorientasi, Informatika, Bandung, 2011.
- [10] R. A. S and M. Shalauddin, Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek). Bandung: Informatika, 2014.
- [11] Mohamad arif sudarsono & krisnawati, "analisis dan perancangan aplikasi "fun 2d shapes learning" berbasis mobile android," vol. vol. 15, 2014.
- [12] Rancang Bangun Aplikasi Pembelajaran Budaya Indonesia Untuk Anak Sekolah Dasar Berbasis Android,.
- [13] M. Hilmi Masruri, Buku Pintar Android, PT. Elex Media Komputindo, Jakarta, 2015.
- [14] Nasruddin Safaat H, Pemograman Aplikasi Mobile Smartphone Dan Tablet PC Berbasis Android. Bandung: Informatika Bandung, 2015.
- [15] Fikri Rijul, Pemrograman Java. Yogyakarta: Andi, 2005.
- [16] Uli fitrianti & mutammimul ula, "Implementasi algoritma levenshtein distance dan algoritma knuth morris pratt pada aplikasi asmaul husna berbasis android," Jurnal Sistem Informasi, 2017.