

Penerapan Algoritma Apostolico Giancarlo Pada Pencarian Fasilitas Umum di Provinsi Aceh Berbasis Android

Muasir Pagan, Efori Buulolo

Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia

Email: muasirpagan@gmail.com

Submitted: 14/04/2020; Accepted: 25/09/2021; Published: 29/09/2021

Abstrak—Fasilitas Umum merupakan sarana yang dibangun oleh pemerintah untuk kepentingan masyarakat umum. Adapun permasalahan yang sering dihadapi di Provinsi Aceh adalah sulitnya mengetahui lokasi Fasilitas Umum di Provinsi Aceh dikarenakan sulitnya sinyal, banyak Fasilitas Umum yang belum terdaftar dan dipetakan pada google Maps. Mobile Phone merupakan salah satu teknologi yang berkembang paling pesat. Mobile Phone pada awalnya digunakan untuk mengirim surat dan melakukan panggilan, tetapi sekarang telah berkembang menjadi smartphone yang memiliki fungsi salah satunya pencarian suatu lokasi. Apostolico Giancarlo bekerja dengan cara membandingkan setiap index karakter kata kunci dengan karakter String yang diperiksa. Langkah pencarian string pada algoritma Apostolico Giancarlo dimulai dengan melakukan pre-processing pada kata kunci. Proses pre-processing pada kata kunci. Proses pre-processing ini dilakukan untuk melihat pada index beberapa dari string yang diperiksa munculnya setiap karakter kata kunci.

Kata Kunci: Implementasi; Algoritma Apostolico Giancarlo; Aplikasi; Fasilitas Umum; Aceh

Abstract—Public Facilities are facilities built by the government for the benefit of the general public. The problem that is often faced in Aceh Province is the difficulty of knowing the location of Public Facilities in Aceh Province due to the difficulty of the signal, many Public Facilities are not yet registered and mapped on Google Maps. Mobile Phone is one of the fastest growing technologies. Mobile Phone was originally used to send letters and make calls, but has now evolved into a smartphone that has the function of searching for a location. Apostolico Giancarlo works by comparing each index of keyword characters with the String character being examined. The search for strings in the Apostolico Giancarlo algorithm starts with pre-processing keywords. Pre-processing of keywords. This pre-processing is done to look at the index number of strings that are checked for each keyword character.

Keywords: Implementation; Apostolico Giancarlo Algorithm; Application; Public Facilities; Aceh

1. PENDAHULUAN

Fasilitas Umum merupakan sarana yang dibangun oleh pemerintah untuk kepentingan masyarakat umum. Fasilitas umum terdiri dari 2 kata, Fasilitas dan Umum. Definisi pengertian dari Fasilitas adalah sarana untuk memperlancarkan pelaksanaan fungsi. Definisi dari Umum adalah kebiasaan yang sudah baku dan menjadi milik khalayak berarti untuk orang kebanyakan atau bertujuan untuk masyarakat. Berikut jenis-jenis fasilitas Umum : Fasilitas Jalan, Fasilitas Komunikasi, Fasilitas Transportasi Fasilitas Sosial, Fasilitas Rekreasi dan Olah raga, dan Fasilitas Ekonomi. Tujuan dari membangun Fasilitas Umum untuk mempermudah kegiatan sehari-hari masyarakat mulai dari pagi hingga malam hari. Seperti untuk mengakomodir kebutuhan masyarakat akan transportasi, pemerintah membangun sejumlah sarana publik bagi masyarakat akan transportasi, pemerintah membangun sarana publik bagi masyarakat, baik yang menggunakan kendaraan pribadi maupun kendaraan umum. Untuk memberikan akses listrik bagi masyarakat pemerintah membangun jaringan listrik, untuk masyarakat yang sakit pemerintah menyediakan rumah sakit.

Provinsi Aceh adalah salah satu provinsi yang terletak di pulau Sumatera dan berada di bagian Indonesia Barat. Provinsi Aceh berdiri pada tanggal 7 Desember 1956 dengan dasar hukum UU Nomor 24 tahun 1956. Karena letak dari Aceh yang berada di pegunungan, terdapat beberapa lokasi yang tidak dijangkau sinyal, dan jadi sulit untuk mengetahui dan mencari Fasilitas Umum apa saja yang ada di Provinsi Aceh.

Dimasa sekarang ini peran media dan teknologi semakin penting, *mobile phone* merupakan salah satu teknologi yang berkembang paling pesat. *Mobile phone* pada awalnya digunakan untuk mengirim pesan singkat dan melakukan panggilan, tetapi dimasa ini telah berkembang menjadi *Smartphone* yang memiliki berbagai fungsi, salah satunya pencarian suatu lokasi.

Adapun permasalahan yang sering dihadapi adalah sulitnya mengetahui lokasi Fasilitas Umum di Provinsi Aceh dikarenakan sulitnya sinyal, banyak Fasilitas Umum yang belum terdaftar dan dipetakan pada google Maps. Aplikasi pencarian Fasilitas Umum di provinsi Aceh ini akan dirancang untuk dijalankan pada *Smartphone* dengan yang didukung dengan sistem operasi Android.

Pada penelitian sebelumnya yang dilakukan oleh Maxime Crochemore asal Prancis dari Institute Gaspard-Moge yang berjudul *A Unifying Look at the Apostolico-Giancarlo String-Matching*, mendapatkan hasil bahwa algoritma Apostolico-Giancarlo sederhana dan mudah dari pada Algoritma *Boyer-Moore*[1]. Adapun pada penelitian yang dilakukan oleh Flora Monica Mirabella dari Institute Teknologi Bandung tentang Variasi-Variasi Algoritma *Boyer-Moore* dan perbandingannya dalam Pencarian *String*, mendapatkan hasil bahwa Algoritma *Boyer-Moore* kalah dalam hal penggunaannya yang mudah, sederhana dan akurat oleh Variasi turunannya, yaitu



algoritma *TurboBM* dan algoritma *Apostolico Giancarlo*, sedangkan algoritma *Horspool* menghasilkan perbandingan yang lebih buruk dari Algoritma *Bayer-Moore*[2].

Pada penelitian ini penulis menggunakan Algoritma *Apostolico Giancarlo* untuk memecah masalah tersebut. Algoritma *Apostolico Giancarlo* merupakan varian dari algoritma pencarian string *Boyer-Moore*. *Apostolico Giancarlo* bekerja dengan cara membandingkan setiap *index* karakter kata kunci dengan karakter *string* yang diperiksa. Langkah pencarian *String* pada Algoritma *Apostolico Giancarlo* dimulai dengan melakukan *pre-processing* pada kata kunci. Proses *pre-processing* ini dilakukan untuk melihat pada *index* beberapa dari *string* yang diperiksa munculnya setiap karakter kata kunci. Dalam proses *pre-processing* pada algoritma *Apostolico Giancarlo* adalah mengurutkan setiap karakter *string* yang diperiksa tanpa terjadinya perulangan karakter yang sama. Setelah proses *pre-processing* dilakukan, selanjutnya adalah proses pencocokan karakter pada kata kunci dengan *String* yang diperiksa. Proses pencocokan dimulai dari karakter pertama dari *String* yang diperiksa sepanjang jumlah karakter kata kunci.

2. METODOLOGI PENELITIAN

2.1 String Matching

Pencarian (*Searching*) merupakan tindakan untuk mendapatkan suatu data dalam kumpulan data. Dalam kehidupan sehari-hari, seringkali kita berurusan dengan pencarian, misalnya menemukan nomor telepon seseorang pada buku telepon atau mencari suatu istilah dalam kamus. Pada aplikasi komputer kerap dilakukan, misalnya untuk mendapatkan data dari seseorang, mahasiswa mendapatkan informasi suatu kata dalam kamus digital, mendapatkan nomor telepon berdasarkan suatu alamat atau nama perusahaan. Selain itu, pencarian dapat dilakukan terhadap data yang secara keseluruhan berada dalam memori komputer ataupun terhadap data yang berada dalam memori komputer dikenal dengan sebutan pencarian internal. Sedangkan pencarian yang dilakukan pada media penyimpanan eksternal disebut pencarian Eksternal[5].

2.2 Fasilitas Umum

Fasilitas umum merupakan sarana dan prasarana yang disediakan oleh pemerintah yang digunakan untuk kepentingan bersama dalam melaksanakan kegiatan sehari-hari. Dapat disimpulkan bahwa fasilitas umum merupakan sarana dan prasarana yang disediakan oleh pemerintah dan pihak swasta untuk kepentingan umum. Sedangkan fasilitas umum memiliki pembagian, seperti fasilitas sosial, fasilitas jalan, fasilitas ekonomi, fasilitas rekreasi dan olah raga.

2.3 Algoritma Apostolico-Giancarlo

Algoritma *Apostolico-Giancarlo* menggunakan ide dasar algoritma *Boyer-moore*, algoritma ini memiliki sebuah variabel yang menyimpan akhiran yang sama, akhiran yang sama ini disimpan dalam sebuah *array* yang menyimpan semua karakter sama yang telah ditemukan. Pada prosesnya nanti, tidak semua karakter yang akan dilakukan perbandingan. Karakter yang sama yang pernah dibandingkan dan telah disimpan tidak akan diulangi lagi perbandingannya. *Apostolico Giancarlo* menyimpan *pattern* dari perbandingan yang sama tersebut agar perbandingan tersebut tidak terjadi lagi[2].

3. HASIL DAN PEMBAHASAN

Dalam suatu pembangunan aplikasi, analisis perlu dilakukan sebelum tahap perancangan dilakukan, karena kesalahan pada tahap analisis akan menyebabkan kesalahan pada tahap selanjutnya. Dengan adanya proses ini akan dihasilkan suatu gambaran sistem yang memungkinkan memiliki kesalahan-kesalahan ataupun kelemahan-kelemahan sehingga memungkinkan dilakukan perbaikan. Aplikasi pencarian Fasilitas Umum di Aceh yang akan dirancang ini adalah sebuah aplikasi pencarian berbasis *mobile* dengan sistem operasi android. Perancangan akan dilakukan dengan menggunakan *software* IDE Eclipse sebagai editor kode program dan android SDK (*Software Development Kit*) yang diperlukan untuk memulai pengembangan aplikasi pada *platform* android menggunakan bahasa pemrograman Java serta ADT (*Android Development Tools*) sebagai *plugin* yang didesain untuk IDE Eclipse yang memberikan kemudahan dalam mengembangkan aplikasi android. Setelah dibuat *Source Code* pada Eclipse Juno dan aplikasi bisa dijalankan pada emulator di laptop, maka untuk menjadikannya menjadi suatu aplikasi yang bisa dijalankan pada android Apk yang ada pada tempat penyimpanan aplikasi yang dirancang, diinstal pada *mobile* android. Aplikasi Pencarian Fasilitas Umum di Aceh ini hanya bisa dijalankan pada *smarthphone* dan tablet PC dengan sistem operasi android.

3.1 Penerapan Metode Blind Deconvolution

Dibawah ini adalah data-data Fasilitas Umum yang berada di Kutacane, Kabupaten Aceh Tenggara, Provinsi Aceh, adapun data-data Fasilitas Umum di kutacane Provinsi Aceh yaitu :

1. Masjid



2. Sekolah
3. Tempat Rekreasi
4. Taman Kota, Stadion dan Lapangan
5. Rumah Sakit

Pada Perancangan aplikasi pencarian Fasilitas Umum di Provinsi Aceh dengan menggunakan algoritma *Apostolico Giancarlo* ini, yang menjadi permasalahan adalah bagaimana proses seleksi terhadap kata yang dicari sampai ditemukan berdasarkan ini teks itu sendiri dengan menggunakan kata kunci *pattern* atau berdasarkan kata kunci pencarian yang diinputkan.

Apostolico Giancarlo berkerja dengan cara membandingkan setiap *index* karakter kata kunci dengan karakter *String* yang diperiksa. Langkah pencarian *string* pada algoritma *Apostolico Giancarlo* dimulai dengan melakukan *pre-processing* pada kata kunci. Proses *pre-processing* pada kata kunci. Proses *pre-processing* ini dilakukan untuk melihat pada *index* keberapa dari *string* yang diperiksa munculnya setiap karakter kata kunci, yaitu :

1. Dalam proses *pre-processing* pada algoritma *Apostolico Giancarlo* adalah mengurutkan setiap karakter *string* yang diperiksa tanpa terjadinya perulangan karakter yang sama.
2. Setelah proses *pre-processing* dilakukan, selanjutnya adalah proses pencocokan karakter pada kata kunci dengan *string* yang diperiksa. Proses pencocokan dimulai dari karakter pertama dari *string* yang diperiksa sepanjang jumlah karakter kata kunci.

Sebagai contoh, akan dilakukan pencarian pada *string* "MIN KUTACANE" dengan kata kunci "CANE". Karakter yang tidak ditemukan pada kata kunci seperti ini disebut dengan karakter buruk, dan akan diberi pengkodean sejumlah $n+1$, dimana n adalah nilai maksimum dari *index* kata kunci. Proses *shift* dilakukan dengan terlebih dahulu melihat apakah karakter yang dibandingkan merupakan karakter buruk atau tidak.

Langkah pertama dalam proses *pre-processing* pada algoritma *Apostolico Giancarlo* adalah mengurutkan setiap karakter *string* yang diperiksa tanpa terjadinya perulangan karakter yang sama. Dari contoh *string* "MIN KUTACANE" yang digunakan, hasil pengurutan karakter *string* ini adalah "FGHAKLCNPT".

| | | | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|---|---|
| Karakter | F | G | H | A | K | L | C | N | P | T |
|----------|---|---|---|---|---|---|---|---|---|---|

Selanjutnya, dilakukan pengecekan kemunculan setiap karakter *string* yang diperiksa pada kata kunci pencarian yang diperiksa, dimulai dari *index* karakter pada kanan. Dari contoh penguraian *string* "FGHAKLCPT" yang digunakan, terlihat bahwa karakter "F" dari *string* yang diperiksa tidak ditemukan pada kata kunci.

| | | | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|---|---|
| Karakter | F | G | H | A | K | L | C | N | P | T |
|----------|---|---|---|---|---|---|---|---|---|---|

Karena panjang kata kunci "CANE" berjumlah 4 karakter, maka *index* yang digunakan untuk kata kunci ini adalah 0 sampai 3. Dengan jumlah maksimum *index* kata kunci sebesar 3, maka pengkodean yang digunakan untuk karakter buruk pada contoh kasus ini adalah 4. Untuk karakter kedua dari *string* yang diperiksa, yaitu karakter "G", terlihat bahwa karakter ini juga tidak ditemukan pada kata kunci, sehingga kembali diberikan nilai pengkodean sebesar 4.

| | | | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|---|---|
| Karakter | F | G | H | A | K | L | C | N | P | T |
|----------|---|---|---|---|---|---|---|---|---|---|

Demikian seterusnya dilakukan pengecekan untuk setiap karakter pada *string* yang diperiksa, sehingga diperoleh hasil seperti dibawah ini.

| | | | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|---|---|
| Karakter | F | G | H | A | K | L | C | N | P | T |
| Kode | 4 | 4 | 3 | 1 | 4 | 4 | 0 | 3 | 4 | 4 |

Setelah panjang *pre-processing* dilakukan, selanjutnya adalah proses pencocokan karakter pada kata kunci dengan *string* yang diperiksa. Proses pencocokan dimulai dari karakter pertama dari *string* yang diperiksa sepanjang jumlah karakter kata kunci.

| | | | | | | | | | | | | |
|---|---|---|---|--|---|---|---|---|---|---|---|---|
| T | M | I | N | | K | U | T | A | C | A | N | E |
| P | | | | | | | | | | | | |

Karena panjang kata kunci pada contoh kasus ini adalah 4 karakter, maka pencocokan pertama yang dilakukan adalah antara karakter *string* "MIN " dan kata kunci "CANE". Karakter paling akhir dari kata kunci, yaitu "E" tidak memiliki kesamaan dengan karakter paling akhir dari *string*, yaitu "spasi". Oleh karena tidak ditemukannya kesamaan pada masing-masing akhir karakter dari *string* dan kata kunci, maka akan dilakukan *shift*.

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T | M | I | N | | K | U | T | A | C | A | N | E |
| P | C | A | N | E | | | | | | | | |

Maka, proses *shift* yang dilakukan adalah menggeser posisi *index* pencarian *string* sebanyak 4 karakter ke kanan sehingga terjadi perubahan karakter yang akan dicontohkan dari karakter "MIN " menjadi "KUTA".

| | | | | | | | | | | | | |
|---|---|---|---|--|---|---|---|---|---|---|---|---|
| T | M | I | N | | K | U | T | A | C | A | N | E |
| P | | | | | C | A | N | E | | | | |



Kembali dilakukan pencocokan antara karakter *string* “KUTA’ dengan karakter kata kunci “CANE” karakter paling akhir dari kata kunci, yaitu “E” tidak memiliki kesamaan dengan karakter paling akhir dari *string*, yaitu “A”. karakter “A” termasuk kedalam kategori karakter buruk sehingga dilakukan proses *shift*, dimana terjadi perubahan karakter yang akan dicocokkan dari karakter “KUTA” menjadi “CANE”.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| T | M | I | N | K | U | T | A | C | A | N | E |
| P | | | | | | | | C | A | N | E |

Pada langkah ini, proses pencocokan dihentikan, karena telah menemukan kata yang cocok pada *string* “MIN KUTA CANE” dengan kata kunci “CANE”. Proses pencocokan berhenti setelah melakukan beberapa kali proses pencocokan. *Bad-character shift* adalah pergeseran karakter sebanyak nilai kode dari karakter yang dibandingkan, yang mana pada contoh kasus ini adalah 4. *Good-suffix shift* adalah proses pergeseran dimana karakter kata kunci digeset hingga ditemukan karakter yang sama pada kata kunci dengan karakter pada posisi paling akhir dari *string*.

Pada contoh selanjutnya, akan dilakukan pencarian pada *string* “pesantren DARUL IMAN” dengan kata kunci “IMAN”. Karakter yang tidak ditemukan pada kata kunci disebut dengan karakter buruk, dan akan diberi pengkodean sejumlah $n+1$, dimana n adalah nilai maksimum dari *index* kata kunci. Proses *shift* dilakukan dengan terlebih dahulu melihat apakah karakter yang dibandingkan merupakan karakter buruk atau tidak.

Karena langkah pertama dalam proses *pre-processing* pada algoritma *Apostolico Giancarlo* adalah mengurutkan setiap karakter *string* yang diperiksa tanpa terjadinya perulangan karakter yang sama. Dari contoh *string* “PESANTREN DARUL IMAN” yang digunakan, hasil pengurutan karakter *string* ini adalah “FIKLMNOPRT”.

| | | | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|---|---|
| Karakter | F | I | K | L | M | N | O | P | R | T |
|----------|---|---|---|---|---|---|---|---|---|---|

Selanjutnya, dilakukan pengecekan kemunculan setiap karakter *string* yang diperiksa pada kata kunci pencarian yang diperiksa, dimulai dari *index* karakter pada kanan. Dari contoh penguraian *string* “FIKLMNOPRT” yang digunakan, terlihat bahwa karakter “F” dari *string* yang diperiksa tidak ditemukan pada kata kunci.

| | | | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|---|---|
| Karakter | F | I | K | L | M | N | O | P | R | T |
|----------|---|---|---|---|---|---|---|---|---|---|

Karena panjang kata kunci “IMAN” berjumlah 4 karakter, maka *index* yang digunakan untuk kata kunci ini adalah 0 sampai 3. Dengan jumlah maksimum *index* kata kunci sebesar 3, maka pengkodean yang digunakan untuk karakter buruk pada contoh kasus ini adalah 4. Untuk karakter kedua dari *string* yang diperiksa, yaitu karakter “I”, terlihat bahwa karakter ini terdapat dalam kata kunci “IMAN” pada *index* pertama, maka akan diberi nilai 0.

| | | | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|---|---|
| Karakter | F | I | K | L | M | N | O | P | R | T |
|----------|---|---|---|---|---|---|---|---|---|---|

Demikian seterusnya dilakukan pengecekan untuk setiap karakter pada *string* yang diperiksa, sehingga diperoleh hasil seperti dibawah ini.

| | | | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|---|---|
| Karakter | F | I | K | L | M | N | O | P | R | T |
| Kode | 4 | 0 | 4 | 4 | 1 | 3 | 4 | 4 | 4 | 4 |

Setelah panjang *pre-processing* dilakukan, selanjutnya adalah proses pencocokan karakter pada kata kunci dengan *string* yang diperiksa. Proses pencocokan dimulai dari karakter pertama dari *string* yang diperiksa sepanjang jumlah karakter kata kunci.

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T | P | E | S | A | N | T | R | E | N | D | A | R | U | L | I | M | A | N |
| P | | | | | | | | | | | | | | | | | | |

Karena panjang kata kunci pada contoh kasus ini adalah 4 karakter, maka pencocokan pertama yang dilakukan adalah antara karakter *string* “PESA” dan kata kunci “IMAN”. Karakter paling akhir dari kata kunci, yaitu “N” tidak memiliki kesamaan dengan karakter paling akhir dari *string*, yaitu “A”. Oleh karena tidak ditemukannya kesamaan pada masing-masing akhir karakter dari *string* dan kata kunci, maka akan dilakukan *shift*.

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T | P | E | S | A | N | T | R | E | N | D | A | R | U | L | I | M | A | N |
| P | I | M | A | N | | | | | | | | | | | | | | |

Maka, proses *shift* yang dilakukan adalah menggeser posisi *index* pencarian *string* sebanyak 4 karakter ke kanan sehingga terjadi perubahan karakter yang akan dicontohkan dari karakter “PESA” menjadi “NTRE”.

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T | P | E | S | A | N | T | R | E | N | D | A | R | U | L | I | M | A | N |
| P | | | | | N | T | R | E | | | | | | | | | | |

Kembali dilakukan pencocokan antara karakter *string* “PESA’ dengan karakter kata kunci “IMAN” karakter paling akhir dari kata kunci, yaitu “N” tidak memiliki Kesamaan dengan karakter paling akhir dari *string*,



yaitu "A". karakter "A" termasuk kedalam kategori karakter buruk sehingga proses *shift*, dimana terjadi perubahan karakter yang akan dicocokkan dari karakter "NTRE" menjadi "N DA".

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T | P | E | S | A | N | T | R | E | N | D | A | R | U | L | I | M | A | N |
| P | | | | | | | | | N | D | A | | | | | | | |

Kembali dilakukan pencocokan antara karakter *string* "N DA" dengan karakter kata kunci "IMAN". Karakter paling akhir dari kata kunci, yaitu "N" tidak memiliki kesamaan dengan karakter paling akhir dari *string*, yaitu "A". karakter "A" termasuk kedalam kategori karakter buruk, sehingga proses *shift* yang dilakukan pada langkah ini adalah *bad-character shift*, dimana terjadi perubahan karakter yang dicocokkan dari karakter "N DA" menjadi "RUL".

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T | P | E | S | A | N | T | R | E | N | D | A | R | U | L | I | M | A | N |
| P | | | | | | | | | | | | R | U | L | | | | |

Kembali dilakukan pencocokan antara karakter *string* "RUL " dengan karakter kata kunci "IMAN". Karakter paling akhir dari kata kunci, yaitu "N" tidak memiliki kesamaan dengan karakter paling akhir dari *string*, yaitu "spasi". karakter "spasi" termasuk kedalam kategori karakter buruk, sehingga proses *shift* yang dilakukan pada langkah ini adalah *bad-character shift*, dimana terjadi perubahan karakter yang dicocokkan dari karakter "RUL " menjadi "IMAN".

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T | P | E | S | A | N | T | R | E | N | D | A | R | U | L | I | M | A | N |
| P | | | | | | | | | | | | | | | I | M | A | N |

Kembali dilakukan pencocokan antara karakter *string* "IMAN" dengan karakter kata kunci "IMAN". Karakter paling akhir dari kata kunci, yaitu "N" memiliki kesamaan dengan karakter paling akhir dari *string*, yaitu "N". karakter "N" termasuk kedalam kategori karakter baik, sehingga proses *shift* yang dilakukan pada langkah ini adalah *good-suffix shift*.

Pada langkah ini, proses pencocokan dihentikan, karena telah menemukan kata yang cocok pada *string* "PESANTREN DARUL IMAN" dengan kata kunci "IMAN". Proses pencocokan berhenti setelah melakukan beberapa kali proses pencocokan. *Bad-character shift* adalah pergeseran karakter sebanyak nilai kode dari karakter yang dibandingkan, yang mana pada contoh kasus ini adalah 4. *Good-suffix shift* adalah proses pergeseran dimana karakter kata kunci digeset hingga ditemukan karakter yang sama pada kata kunci dengan karakter pada posisi paling akhir dari *string*.

Contoh selanjutnya merupakan contoh yang salah dan tidak sesuai, dilakukan pencarian pada *string* "Fasilitas Umum di Aceh" dengan kata kunci "Aceh". Karakter yang tidak ditemukan pada kata kunci seperti ini disebut dengan karakter buruk, dan akan diberi pengkodean sejumlah $n+1$, dimana n adalah nilai maksimum dari *index* kata kunci. Proses *shift* dilakukan dengan terlebih dahulu melihat apakah karakter yang dibandingkan merupakan karakter buruk atau tidak.

Langkah pertama dalam proses *pre-processing* pada algoritma *Apostolico Giancarlo* adalah mengurutkan setiap karakter *string* yang diperiksa tanpa terjadinya perulangan karakter yang sama. Dari contoh *string* "Fasilitas Umum di Aceh" yang digunakan, hasil pengurutan karakter *string* ini adalah "FGHBKIONPT".

| | | | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|---|---|
| Karakter | F | G | H | B | K | L | O | N | P | T |
|----------|---|---|---|---|---|---|---|---|---|---|

Selanjutnya dilakukan pengecekan kemunculan setiap karakter *string* yang diperiksa pada kata kunci pencarian yang diperiksa, dimulai dari *index* karakter pada kanan. Dari contoh penguraian *string* "FGHBKIONPT" yang digunakan, terlihat bahwa karakter "F" dari *string* yang diperiksa tidak ditemukan pada kata kunci.

| | | | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|---|---|
| Karakter | F | G | H | B | K | L | C | N | P | T |
|----------|---|---|---|---|---|---|---|---|---|---|

Karena panjang kata kunci "ACEH" berjumlah 4 karakter, maka *index* yang digunakan untuk kata kunci ini adalah 0 sampai 3. Dengan jumlah maksimum *index* kata kunci sebesar 3, maka pengkodean yang digunakan untuk karakter buruk pada contoh kasus ini adalah 4. Untuk karakter kedua dari *string* yang diperiksa, yaitu karakter "G", terlihat bahwa karakter ini juga tidak ditemukan pada kata kunci, sehingga kembali diberikan nilai pengkodean sebesar 4.

| | | | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|---|---|
| Karakter | F | G | H | B | K | L | O | N | P | T |
|----------|---|---|---|---|---|---|---|---|---|---|

Demikian seterusnya dilakukan pengecekan untuk setiap karakter pada *string* yang diperiksa, sehingga diperoleh hasil seperti dibawah ini.

| | | | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|---|---|
| Karakter | F | G | H | B | K | L | O | N | P | T |
| Kode | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |

Setelah panjang *pre-processing* dilakukan, selanjutnya adalah proses pencocokan karakter pada kata kunci dengan *string* yang diperiksa. Setelah *pre-processing* dilakukan dan tidak terdapat karakter pada *string* yang diperiksa, maka proses dihentikan.



4. KESIMPULAN

Berdasarkan hasil penelitian dapat disimpulkan proses Pencarian Fasilitas Umum di Provinsi Aceh dilakukan dengan cara mencari referensi dari Kantor Gubernur Aceh dan *internet*, setelah Fasilitas Umum ditemukan Fasilitas Umum tersebut dikumpulkan dan disimpan pada *database*. Algoritma *Apostolico Giancarlo* dapat diimplementasikan sebagai solusi dalam pencarian Fasilitas Umum di Provinsi Aceh, berhubung Fasilitas Umum di Provinsi Aceh sangat banyak maka sangat sulit untuk merealisasikan penyajian fasilitas yang sesuai dengan urutan abjadnya. Berhubung keterbatasan layar dari masing-masing *smartphone* maka diperlukan *Apostolico Giancarlo* untuk melakukan proses pencarian fasilitas umum. Oleh karena itu, dengan implementasi algoritma *Apostolico Giancarlo* dalam pencarian fasilitas umum, diharapkan dapat menyelesaikan masalah tersebut, dimana dalam algoritma *Apostolico Giancarlo* penyajian fasilitas umum di provinsi Aceh menjadi lebih sedikit karena adanya proses pengecekan berdasarkan *pattern*.

REFERENCES

- [1] Maxime Crochiemore, chirstophe Hancart Livar, Thierry Kecroq, A Unifyng Look at the Apostolico-Giancarlo String-Matching Algoritm. IGM (*Institut Gaspard-Monge*), *universite' de Marne-la-vellee*
- [2] Flora Monica Mirabella, Variasi-variasi Algoritma Boyer-moore dan perbandingannya dalam pencarian String. Program Studi Teknik Informatika, Institut Teknologi Bandung.
- [3] Kantor Gubernur, 20 Mei 2018
- [4] Warisno Syahputra, ISSN : 2301-9425, implementasi Algoritma Blute Force pada perancangan Aplikasi Pengecekan Struktur Kata Serapan Bahasa Indonesia, Volume 9, No 2, maret 2015
- [5] Abdul Kadir, 2012, 332
- [6] Lhakim Sidik, Fasilitas Umum. [http:// www.Slideshare.net](http://www.Slideshare.net)
- [7] A. Nugraha, Rekaya Perangkat Lunak Berorientasi Objek dengan Metode USPD, Yogyakarta: Andi Yogyakarta, 2010.
- [8] Rosa A.S. M Shalahuddin 2011, Rekayasa Perangkat Lunak Penerbit ANDI : Yogyakarta
- [9] Zamrony p. juhara, panduan Lengkap Pemograman ANDROID, Yogyakarta : CV.ANDI OFFSET.2016
- [10] Wina Novita, ST. PENGENALAN ECLIPSE, wina.fatimah@gmail.com
- [11] Alfa Syahputra, Eva Maulana Aritonang, JAVA For Beginners with Eclipse 4.2 juno, penerbit Komputindo, Tangerang, Oktober 2012
- [12] Rachel Kurniawati (dkk), Interaksi Android dengan JSON web sevice berbasis PHP, penerbit Cahaya Atma Pustaka, Yogyakarta, Januari 2015.
- [13] A. A. B. Ginting and D. P. Utomo, "PERANCANGAN APLIKASI CATALOG WISATA DI SUMATERA UTARA MENGGUNAKAN ALGORITMA RABIN-KARP," *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, vol. 3, no. 1, pp. 57-63, 2019.