

## IMPLEMENTASI ALGORITMA STRING MATCHING PENCARIAN KATA DARI MAKNA RAMBU LALULINTAS BERBASIS ANDROID

**Zekson Arizona Matondang**

*STMIK Kristen Neumann Indonesia*

*Jl. Letjen. Jamin Ginting Km.10,5 Medan, Sumatera Utara*

*E-mail :zekson.arizona@yahoo.com*

### ABSTRAK

Rambu lalu lintas merupakan salah satu dari perlengkapan jalan yang dapat berupa lambang, huruf, angka, kalimat atau perpaduan di antaranya yang berfungsi sebagai peringatan, larangan, perintah atau petunjuk bagi pemakai jalan. Indonesia adalah Negara yang padat penduduknya terutama di kota-kota besar. Pada kota-kota besar setiap harinya dipadati dengan kendaraan bermotor yang banyak menyebabkan permasalahan di jalannya itu kemacetan, kecelakaan, terobos rambu lalu lintas, ada banyak kemungkinan faktor penyebab hal itu terjadi seperti minimnya kesadaran dalam mematuhi rambu lalu lintas, kurangnya pemahaman dari arti simbol lalu lintas, dan berbagai macam faktor lain.

**Kata kunci: Rambu Lalu Lintas, Android, String Matching**

### ABSTRACT

*Traffic signs are one of the road tools that can be symbols, letters, numbers, sentences or cleavages that are passed as a destination, prohibition, command or instruction for the road user. Indonesia is a densely populated country especially in big city cities. In big cities every day packed with motor vehicles causing many road conflicts that are traffic jams, accidents, traffic jams, there are many factors that allow to do the same thing as traffic, traffic, traffic, traffic. and various other factors.*

**Keywords: Traffic Sign, Android, String Matching**

### I. PENDAHULUAN

Rambu lalu lintas merupakan salah satu dari perlengkapan jalan yang dapat berupa lambang, huruf, angka, kalimat atau perpaduan di antaranya yang berfungsi sebagai peringatan, larangan, perintah atau petunjuk bagi pemakai jalan. Kasus kecelakaan lalu lintas (laka lantas) di Indonesia, masih relative tinggi. Jumlah kejadian laka lantas di Nusantara, masuk peringkat 10 besar tingkat dunia. Yang sangat memprihatinkan, mayoritas pemicunya karena faktor manusia. Salah

satunya adalah kurangnya kesadaran maupun pengemudi untuk mematuhi rambu-rambu lalu lintas, serta kurangnya pengetahuan para pengendara yang masih awam dalam rambu-rambu lalu lintas yang jumlahnya cukup banyak, disamping itu kurangnya niat membaca para pengemudi tentang arti dan symbol-simbol lalu lintas tersebut. *Smart Phone Android* kini sudah merambah kalangan masyarakat dari kalangan bawah sampai kalangan atas. *and phone* sudah beralih fungsi dari sekedar alat komunikasi biasa menjadi alat

komunikasi yang canggih dan dapat mengakses informasi secara luas dan efisien serta dapat menghubungkan setiap individu yang menggunakannya. Kini *Smart Phone Android* bukan saja merk terkenal dan harga yang mahal, tetapi merk-merk local dengan harga terjangkau juga sudah banyak dipasaran.

Algoritma *string matching* merupakan algoritma pencocokan *string*. Sifat algoritma *string matching* adalah mencari sebuah *string* yang terdiri dari beberapa karakter (yang biasa disebut *pattern*) dalam sejumlah besar text. Pencarian *string* juga bisa digunakan untuk mencari pola bit dalam jumlah besar file binary.

Aplikasi pencarian kata pada makna rambu lalu lintas pada *handphone* belum ada. Oleh karena itu, melihat perkembangan android sekarang ini dan sudah merupakan kebutuhan akan berbagai hal dalam aplikasi pada sistem operasi tersebut maka sangat cocok dan perlu pembuatan aplikasi pencarian kata pada makna simbol lalu lintas.

## 2. METODOLOGI

### 2.1 Rambu Lalu Lintas

Rambu lalu lintas merupakan salah satu dari perlengkapan jalan yang dapat berupa lambang, huruf, angka, kalimat atau perpaduan di antaranya yang berfungsi sebagai peringatan, larangan, perintah atau petunjuk bagi pemakai jalan. Kita sebagai pemakai jalan, apakah seorang pengendara mobil, sepeda motor, atau pejalan kaki sudah seharusnya mematuhi setiap rambu lalu lintas di jalan untuk keamanan bersama. Untuk mematuhi rambu lalu lintas tersebut, terlebih dahulu kita harus memahami dan mengetahui arti dari rambu-rambu lalu lintas.

Rambu lalu lintas diatur menurut Peraturan Menteri Perhubungan Nomor 13 tahun 2014. Agar rambu dapat terlihat baik [siang](#) ataupun [malam](#) atau pada waktu

[hujan](#) maka bahan harus terbuat dari material [retro-reflektif](#).

### 2.2. String Matching

Pengertian *string* menurut *Dictionary of Algorithms and Data Structures, National Institute of Standards and Technology (NIST)* adalah susunan dari karakter-karakter (angka, alfabet atau karakter yang lain) dan biasanya direpresentasikan sebagai struktur dan *array*. *String* dapat berupa kata, frase, atau kalimat. Pencocokan *string* (*string matching*) merupakan bagian penting dari sebuah proses pencarian *string* (*string searching*) dalam sebuah dokumen. Hasil dari sebuah pencarian *string* dalam dokumen tergantung dari teknik dan cara pencocokan *string* yang digunakan. *String Matching* juga tergolong ada dua, yaitu *Heuristic Matching* dan *Exact Matching*

### 2.3 Android

*Android* adalah sistem operasi untuk telepon seluler yang berbasis Linux. *Android* menyediakan platform terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, *Google Inc.* membeli *Android Inc.*, pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan *Android*, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan perantikeras, peranti lunak, dan telekomunikasi, termasuk *Google*, *HTC*, *Intel*, *Motorola*, *Qualcomm*, *T-Mobile*, dan *Nvidia*.

## 3 HASIL DAN PEMBAHASAN

Tahapan analisa terhadap suatu system dilakukan sebelum tahapan perancangan dilakukan. Adapun tujuan dilakukannya analisa terhadap suatu system adalah untuk mengetahui alasan mengapa system tersebut diperlukan, yaitu dengan merumuskan kebutuhan-kebutuhan dari

system tersebut. Perancangan aplikasi penerapan algoritma *String Matching* pada pencarian kata makna gambar rambu-rambu lalu lintas berbasis *android* menggunakan bahasa pemrograman *android* dengan *platform* yang di gunakan adalah *Eclipse Juno*.

*Eclipse* adalah sebuah IDE (*Integrate Development Environment*) untuk mengembangkan perangkat lunak dan dapat di jalankan di semua *platform*. *Eclipse* disini membutuhkan SDK (*Software Development Kit*) windows sebagai alat bantu dan API (*Application Programming Interface*) di perlukan untuk memulai mengem- bangkan aplikasi pada *platform android* menggunakan bahasa pemrograman *java* seperti pad *ahandphone*.

### 3.1 Algoritma String Matching Brute Force dan Knuth-Morrit-Pratt

Algoritma *string matching Brute Force* Cara kerja algoritma *string matchingBrute Force* adalah dimisalkan teks berada dalam array  $T[1..n]$  dan *pattern* berada dalam array  $[1..n]$ .

Contoh:

Teks : dilarang belok kiri

*Pattern* : belok

Dilarang belok kiri

s= 0 belok

s= 1 belok

s= 2 belok

s= 3 belok

s= 4 belok

s= 5 belok

s= 6 belok

s= 7 belok

s= 8 belok

s= 9 belok

*Pattern belok* ditemukan pada posisi indeks ke sepuluh dari awal *text*



Gambar 1 Rambu Larangan

Cara kerja algoritma dengan knuth-Marris-Pratt adalah dengan memelihara informasi yang digunakan untuk melakukan jumlah pergeseran.

*Contoh Kasus :*

*Pattern (X) = Masuk*

Teks (Y) = Wajib Memasuki Jalur Kiri



Wajib memasuki lajur kiri

Gambar 2 Rambu Wajib

Gambar 1. Gambar dan Makna lalu lintas

#### Langkah 1

Teks	W	A	J	i	b	M	a	s	u	k	i	.	J	a	l	a	r	K	i	r	i
Pattern	M	a	s	u	k																
1		M	a	s	u	k															
2			M	a	s	u	k														
3				M	a	s	u	k													
4					M	a	s	u	k												
5						M	a	s	u	k											
6							M	a	s	u	k										
7								M	a	s	u	k									
8									M	a	s	u	k								

Pada langkah pertama tidak ditemukan kecocokan antara karakter pertama pada *pattern* dengan karakter pertama yang ada pada teks, maka dilakukan pergeseran *pattern* satu kali kearah kanan.

#### Langkah 2

Teks	W	A	J	i	b	M	a	s	u	k	i	.	J	a	l	a	r	K	i	r	i
Pattern		M	a	s	u	k															
1			M	a	s	u	k														

Pada langkah ke-2, pencarian belum

selesai karena semua karakter pada *pattern* belum cocok dengan karakter pada teks maka pencarian dilanjutkan dengan menggeser karakter *pattern* kekanan sebanyak satu kali.

#### Langkah 3

Teks	W	A	J	I	B	M	E	M	A	S	U	K	I	J	A	L	U	R	K	I	K	I
Pattern						M	A	S	U	K												
ak=3																						

Pada langkah ke-3 sama dengan langkah kedua, maka dilakukan pergeseran pattern sebanyak satu kali kearah kanan.

#### Langkah 4

Teks	W	A	J	I	B	M	E	M	A	S	U	K	I	J	A	L	U	R	K	I	K	I
Pattern						M	A	S	U	K												
ak=4																						

Pada langkah ke-4 sama dengan langkah kedua, maka dilakukan pergeseran pattern sebanyak satu kali kearah kanan.

#### Langkah 5

Teks	W	A	J	I	B	M	E	M	A	S	U	K	I	J	A	L	U	R	K	I	K	I
Pattern						M	A	S	U	K												
ak=4																						

Pada langkah ke-5 sama dengan langkah kedua, maka dilakukan pergeseran pattern sebanyak satu kali kearah kanan.

#### Langkah 6

Teks	W	A	J	I	B	M	E	M	A	S	U	K	I	J	A	L	U	R	K	I	K	I
Pattern						M	A	S	U	K												
ak=4																						

Pada langkah ke-6 sama dengan langkah kedua, maka dilakukan pergeseran pattern sebanyak satu kali kearah kanan.

#### Langkah 7

Teks	W	A	J	I	B	M	E	M	A	S	U	K	I	J	A	L	U	R	K	I	K	I
Pattern						M	A	S	U	K												
ak=4																						

Pada langkah ke-4 sama dengan langkah kedua, maka dilakukan pergeseran pattern sebanyak satu kali kearah kanan, sampai ketemu di langkah ke 8

#### Langkah 8

Teks	W	A	J	I	B	M	E	M	A	S	U	K	I	J	A	L	U	R	K	I	K	I
Pattern						M	A	S	U	K												
ak=8																						

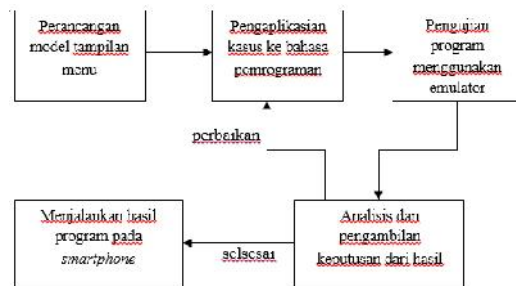
Pattern (X) = 001011

Teks (Y) =

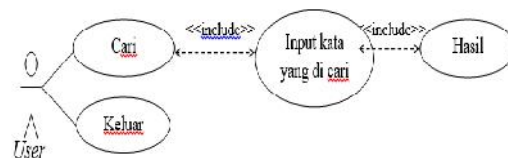
10010101**001011**110101010001

Teks	1	0	0	1	0	1	0	1	0	0	1	0	1	1	1	0	1	0	0	0	1
Pattern	0	0	1	0	1	1															
1		0	0	1	0	1	1														
2			0	0	1	0	1	1													
3				0	0	1	0	1	1												
4					0	0	1	0	1	1											
5						0	0	1	0	1	1										
6							0	0	1	0	1	1									
7								0	0	1	0	1	1								
8									0	0	1	0	1	1							

Uraian tahapan perancangan dalam pembuatan aplikasi *mobile* pencarian kata pada makna gambar rambu-rambu lalulintas:

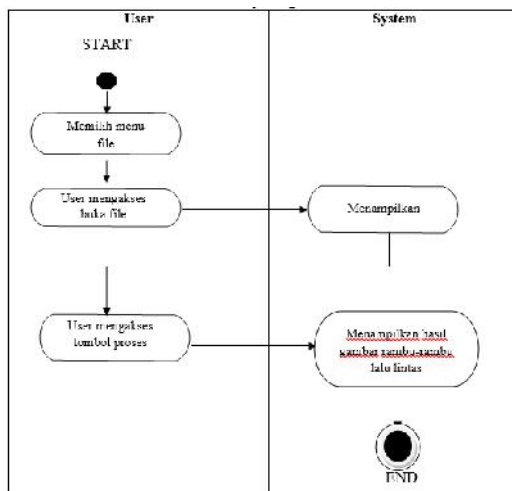


Perancangan system dapat di definisikan sebagai penggambaran, perancangan dan pembuatan sketsa atau pengaturan dari beberapa elemen terpisah menjadi kesatuan yang utuh dan berfungsi. Pemodelan *Aplikasi* ini di susun menggunakan 2 diagram, sebagai berikut: *Use case diagram* yang menggambarkan bagaimana proses yang terjadi pada aplikasi pencarian kata pada makna gambar rambu-rambu lalulintas.

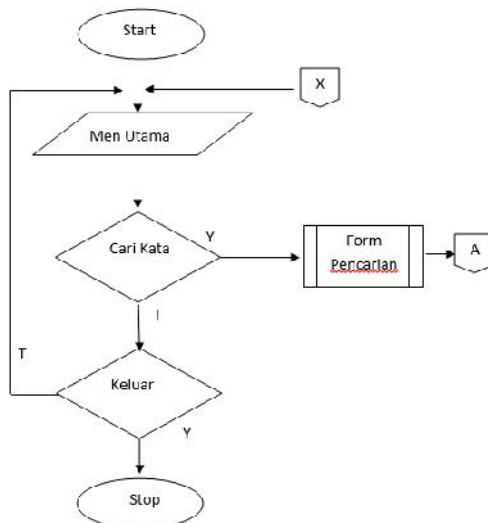


Gambar 3 Use Case

*Activity Diagram* adalah teknik untuk menggambarkan logika *procedural* atau jalur kerja sistem. Berikut ini merupakan *Activity Diagram* dari sistem yang akan dibangun.



Gambar 4 Activity Diagram



Gambar 5 Flowchart

**Prosedur Algoritma**

```

{
  Masukkan m,n sebagai integer
  Masukkan pattern dalam array (0...n-1) sebagai karakter
  Masukkan teks dalam array (0...m-1) sebagai karakter
  Hasil Ketemu dalam array(0...m-1) sebagai ketemu/tidak.
}
Deklarasi
I,j, lanjut sebagai integer
Alg lanjut dalam array[0...n] sebagai integer
Algoritma
Awalalg(n,pattern,alglanjut)
I:=0
Ketika(i<=m-n) lakukan
  
```

```

j:=0
ketika(j<n dan teks[i+j]=pattern[j]) lakukan
  j:=j+1
akhiri
jika(j>=0) maka
  ketemu[i]=benar;
akhiri
lanjut=j-alglanjut[j]
i:=i+lanjut
akhiri
  
```

**3.2 Hasil Pencarian String Matching**

Gambar 6 Tampilan Menu Awal Pencarian



Gambar 7 Tampilan Form Hasil Pencarian

No	Kata yang dicari dari gambar lalu lintas	Gambar Lalu lintas	Hasil Pencarian Teks yang dicari
1	Tanjakan		Peringatan Tanjakan Landai
			Peringatan Tanjakan Curam

#### 4 KESIMPULAN

Berdasarkan hasil analisa dan implementasi aplikasi pencarian kata makna rambu lalu lintas maka dapat beberapa kesimpulan sebagai berikut:

1. Dengan menggunakan algoritma *String Matching* pencarian kata pada makna rambu lalu lintas dapat lebih mudah.
2. Dengan adanya aplikasi ini dapat membantu pengguna smartphone android dalam melakukan pencarian gambar yang memiliki kesamaan kata jika diinput dan dengan kata yang telah di simpan sebelumnya di dalam *database* pencarian rambu lalu lintas.
3. Penerapan aplikasi pencarian kata pada rambu lalu lintas menggunakan *software Eclipse Junosebagaie editor, Software Development Kit (SDK)* sebagai *platform*, *ADT plugin* sebagai *emulator* dan *JDK 6.0.2* sebagai proses atau mesin dari pembuatan aplikasi, serta *SQL ite* sebagai wadah perancangan *data base* sehingga bisa di gunakan pada versi android cupcake android 1.5 ke atas.

#### 5 SARAN

Pada penelitian ini terdapat beberapa saran yang dapat digunakan, diantaranya:

1. Penambahan fitur *sharing* gambar ke *social media* agar pengguna dapat membagi informasi yang dibacanya kepada orang lain.
2. Aplikasi ini masih sederhana, diharapkan dapat dikembangkan lagi menjadi sebuah aplikasi dengan tampilan (GUI) yang lebih menarik.
3. Penambahan fitur *search history* didalam menu pencarian untuk optimalisasi pencarian gambar yang ada pada aplikasi ini.

#### DAFTAR PUSTAKA

- [1]. Syaroni M, Munir R, 2005, Pencocokan String Berdasarkan Kemiripan Ucapan (Phonetic String Matching) Dalam Bahasa Inggris, Institut Teknologi Bandung.
- [2]. Hadiati Desi, 2007, Penerapan Algoritma String Matching Pada Permainan Word Search Puzzle, Institut Teknologi Bandung.
- [3]. Siregar Ivan Michael, 2011, Mengembangkan Aplikasi Enterprise Berbasis Android, Penerbit Gava Media, Yogyakarta, Edisi I.
- [4]. <http://multirambu-ts.blogspot.com/2014/07/pengertian-rambu-lalu-lintas.html> (diakses juni 2018)
- [5]. <https://entripopu.blogspot.com/2014/06/gambar-rambu-lalu-lintas-dan-artinya.html>