

SKRIPSI

**PEMBANGUNAN PERANGKAT LUNAK PENCARIAN
JUDUL BUKU DENGAN MENGGUNAKAN ALGORITMA
MINIMUM EDIT DISTANCE**



MUHAMAD RISFAN SYARID PRATAMA

NPM: 2010730110

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN**

2016

UNDERGRADUATE THESIS

**THE DEVELOPMENT OF BOOK TITLES SEARCHING
SOFTWARE USING MINIMUM EDIT DISTANCE
ALGORITHM**



MUHAMAD RISFAN SYARID PRATAMA

NPM: 2010730110

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND
SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2016**

LEMBAR PENGESAHAN

**PEMBANGUNAN PERANGKAT LUNAK Pencarian
JUDUL BUKU DENGAN MENGGUNAKAN ALGORITMA
MINIMUM EDIT DISTANCE**

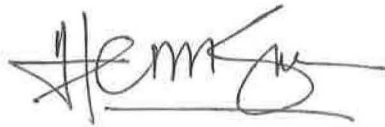
MUHAMAD RISFAN SYARID PRATAMA

NPM: 2010730110

Bandung, 22 Desember 2016

Menyetujui,

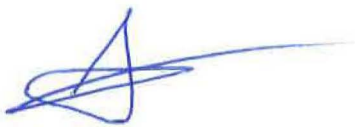
Pembimbing



Dr.rer.nat. Cecilia Esti Nugraheni



Ketua Tim Penguji



Dott. Thomas Anung Basuki

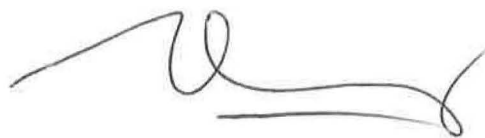
Anggota Tim Penguji



Pascal Alfadian, M.Comp.

Mengetahui,

Ketua Program Studi



Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

PEMBANGUNAN PERANGKAT LUNAK Pencarian Judul Buku DENGAN MENGGUNAKAN ALGORITMA MINIMUM EDIT DISTANCE

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.



Dinyatakan di Bandung,
Tanggal 22 Desember 2016

Risfan



MUHAMAD RISFAN SYARID PRATAMA
NPM: 2010730110

ABSTRAK

Algoritma *minimum edit distance* adalah suatu algoritma untuk mengukur kemiripan antara dua buah teks. Kemiripan antara dua buah teks diukur berdasarkan nilai *minimum edit distance*. Nilai *minimum edit distance* adalah suatu nilai untuk mengukur seberapa mirip dua buah *string*. Nilai tersebut merupakan jumlah dari berapa operasi yang dibutuhkan untuk membuat suatu *string* menjadi *string* lainnya. Operasi yang dimaksud adalah *insertion*, *deletion* dan *substitution*. Semakin kecil nilai tersebut maka semakin mirip kedua teks tersebut.

Pengunjung perpustakaan maupun toko buku sering mengalami kesulitan dalam mencari buku yang diinginkan. Para pengunjung biasanya mencari buku berdasarkan judulnya. Untuk memudahkan para pengunjung perpustakaan maupun toko buku dalam mencari buku sesuai dengan judul buku yang ingin dicari maka diperlukan suatu perangkat lunak untuk melakukan hal tersebut.

Dalam melakukan pencarian judul buku perlu dihilangkan kata yang merupakan *stop word* sebab *stop word* merupakan kata yang dianggap kurang memiliki makna. Hal ini dapat mempercepat hasil pencarian serta membuat hasil pencarian lebih akurat. Setiap judul buku perlu dipetakan menjadi suatu deretan *value* sehingga deretan *value* tersebut yang akan dibandingkan oleh algoritma *minimum edit distance*. Pemetaan ini menggunakan kelas `HashMap` pada *library* Java.

Berdasarkan hasil eksperimen, apabila judul buku yang dicari berupa kalimat panjang atau judul tersebut terdapat pada *database* maka hasil pencarian cukup relevan namun jika judul buku yang dicari berupa kalimat pendek dan tidak terdapat pada *database* maka hasil pencarian menjadi kurang relevan dengan judul buku yang dicari. Kesimpulan dari penelitian ini algoritma *minimum edit distance* kurang baik digunakan sebagai algoritma untuk melakukan pencarian judul buku.

Kata-kata kunci: Algoritma *Minimum Edit Distance*, *Stop Word*, *HashMap*, *Pencarian Judul Buku*.

ABSTRACT

Minimum edit distance algorithm is useful for measuring similarity between two words. The measurement based on minimum edit distance value. The value representing how many operations needed to convert a string into another string. The said operations are insertion, deletion and substitution. The smaller value means that those words are similar.

Visitors of library or book shop usually have a problem to find their desired book. They usually search book by its title. A software needed to help the visitor to find books using its title so the result can be more relevant.

Stop words need to be eliminated in searching process of book titles. Stop words are words with non-significant meaning. Elimination process can make the search process faster and gives more accurate result. Every word of a book title needs to be mapped into a unique value. So a book title is mapped into a sequence of values. That sequence of value will be used in minimum edit distance computation. This mapping uses a HashMap class from Java library.

If a has few words and that use title exist in database, the algorithm can give a quite relevant result but when the title has many words or that title does not exist in database, this algorithm give a poor result. The conclusion is minimum edit distance algorithm is not suitable to be used for searching book titles

Keywords: Minimum Edit Distance Algorithm, Stop Word, HashMap, Searching of Book Titles.

Skripsi ini ditujukan untuk keluarga, teman-teman, kansup dan juga Chelsea Islan.

KATA PENGANTAR

Terima kasih kepada Tuhan Yang Maha Kuasa, pembimbing, ibu Joanna Helga yang membantu saya dan menyemangati saya. Terima kasih juga kepada Bayu Anggara atas bantuan yang amat besar, Arif Bintoro, Ilham Bani Utama, Gilang Faturahman dan juga teman-teman seperjuangan. Skripsi ini hanyalah salah satu pencapaian dalam hidup saya dan tentunya masih banyak halangan dan rintangan kedepannya dan itulah motivasi saya. Terima kasih kepada seluruh teman, sahabat dan keluarga yang tidak bisa disebutkan satu per satu. Salam damai dan bahagia selalu.

Bandung, Desember 2016

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xx
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika Pembahasan	3
2 LANDASAN TEORI	5
2.1 <i>Stop Word</i>	5
2.2 HashMap	5
2.3 Algoritma <i>Minimum Edit Distance</i>	7
2.4 Baca Tulis File	10
3 ANALISIS	13
3.1 <i>Stop Word</i>	13
3.2 HashMap	15
3.3 Algoritma <i>Minimum Edit Distance</i>	16
3.4 Baca Tulis <i>File</i>	20
3.5 Analisis Perangkat Lunak	21
3.5.1 Skenario <i>Use Case</i>	22
3.5.2 Implementasi Algoritma	22
3.5.3 <i>Class Diagram</i>	25
4 PERANCANGAN	27
4.1 Kebutuhan Masukan dan Keluaran	27
4.1.1 Masukan	27
4.1.2 Keluaran	27
4.2 Diagram Kelas Rinci	28
4.3 Rincian Metode	28
4.3.1 Kelas StopWordEngine	29
4.3.2 Kelas HashMapEngine	30
4.3.3 Kelas EditDistanceEngine	30
4.3.4 Kelas Book	31
4.3.5 Kelas BookComparator	32

5	IMPLEMENTASI DAN PENGUJIAN	37
5.1	Implementasi	37
5.1.1	Lingkungan Perangkat Keras	37
5.1.2	Lingkungan Perangkat Lunak	37
5.2	Rancangan Pengujian	37
5.2.1	Pengujian Fungsional	38
5.2.2	Pengujian Kualitatif	39
5.2.3	Hasil Pengujian Kualitatif	40
6	KESIMPULAN DAN SARAN	47
6.1	Kesimpulan	47
6.1.1	Saran	48
	DAFTAR REFERENSI	49
	A KODE PROGRAM	51
	B HASIL PENGUJIAN	59
	C KUMPULAN STOP WORD BAHASA INDONESIA	61
	D KUMPULAN STOP WORD BAHASA INGGRIS	65
	E KUMPULAN JUDUL BUKU	67

DAFTAR GAMBAR

2.1	6
3.1	Aliran Proses.	23
3.2	<i>Class Diagram</i>	25
4.1	<i>Class Diagram</i>	28
5.1	Daftar Judul Buku yang Digunakan Pada Pengujian Fungsional.	38
5.2	Hasil Pencarian Judul Buku yang Terdapat Pada <i>Database</i>	38
5.3	Hasil Pencarian Judul Buku yang Terdapat Pada <i>Database</i> dengan <i>Stop Word</i> yang Dihilangkan.	38
5.4	Hasil Pencarian Judul Buku yang Tidak Terdapat Pada <i>Database</i>	39
5.5	Grafik Penilaian Responden Pada Pengujian 1 Dimana Judul Buku Yang Dicari Terdapat Pada <i>Database</i>	40
5.6	Hasil Pengujian 1.	41
5.7	Grafik Penilaian Responden Pada Pengujian 2 Dimana Posisi Kata-Kata Diacak dan Judul Buku Merupakan Kalimat Panjang	41
5.8	Hasil Pengujian dengan Judul Buku Berupa Kalimat Panjang.	41
5.9	Hasil Pengujian dengan Judul Buku Berupa Kalimat Panjang Namun Kata-katanya Ditukar.	42
5.10	Grafik Penilaian Responden Pada Pengujian 3 Dimana Posisi Kata-Kata Diacak dan Judul Buku Merupakan Kalimat Pendek	42
5.11	Hasil Pengujian dengan Judul Buku Berupa Kalimat Pendek.	43
5.12	Hasil Pengujian dengan Judul Buku Berupa Kalimat Pendek Namun Kata-katanya Ditukar.	43
5.13	Grafik Penilaian Responden Pada Pengujian 4 Dimana Judul Buku yang Dicari Tidak Terdapat Pada <i>Database</i>	44
5.14	Hasil Pengujian dengan Judul Buku yang Dicari Tidak Terdapat Pada <i>Database</i> ..	44
5.15	Grafik Penilaian Responden Pada Pengujian 5 Dimana Judul Buku yang Dicari Tidak Terdapat Pada <i>Database</i> Namun Memiliki Sebagian Kata Pada Judul Buku yang Tersimpan di <i>Database</i>	45

DAFTAR TABEL

2.1	Tabel perhitungan algoritma <i>minimum edit distance</i>	8
2.2	Tabel perhitungan algoritma <i>minimum edit distance</i> untuk mengisi kolom 3 dan baris 3	8
2.3	Tabel perhitungan algoritma <i>minimum edit distance</i> untuk mengisi kolom 4 dan baris 3	9
2.4	Tabel perhitungan algoritma <i>minimum edit distance</i> setelah semua proses dilakukan.	9
3.1	Tabel judul buku yang memiliki <i>stop word</i>	13
3.2	Tabel judul buku yang tidak memiliki <i>stop word</i>	14
3.3	Tabel Judul Buku	14
3.4	Tabel judul buku contoh hashmap	15
3.5	Tabel judul buku yang sudah memiliki deretan <i>value</i>	16
3.6	Tabel Perhitungan Algoritma Minimum Edit Distance	17
3.7	Tabel Hasil Perhitungan Algoritma <i>Minimum Edit Distance</i>	17
3.8	Tabel judul buku yang sudah memiliki nilai <i>minimum edit distance</i>	18
3.9	Tabel judul buku yang sudah memiliki nilai <i>minimum edit distance</i> terurut	18
3.10	Tabel lima hasil pencarian judul buku yang memiliki nilai <i>minimum edit distance</i> terkecil	19
3.11	File judul buku	21
3.12	Skenario <i>Use Case</i>	22
B.1	Tabel Nilai Pengujian Yang Telah Diisi Oleh Responden.	60
C.1	Tabel Kata Yang Merupakan <i>Stop Word</i> Bahasa Indonesia.	63
D.1	Tabel Berisi Kata-kata Yang Merupakan <i>stop word</i> Berbahasa Inggris.	66

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Minimum edit distance adalah suatu algoritma untuk mengukur kemiripan antara dua teks. Algoritma ini menghitung berapa langkah yang dibutuhkan untuk mengubah suatu teks menjadi teks yang lain. Misalkan teks pertama adalah RASA dan teks kedua adalah SARA maka untuk mengubah teks pertama menjadi teks kedua dibutuhkan dua langkah yaitu mengubah huruf 'R' menjadi 'S' pada teks pertama dan mengubah huruf 'S' menjadi 'R' pada teks kedua sehingga teks pertama yaitu RASA berubah menjadi teks kedua yaitu SARA. Selain mengubah satu huruf, menghapus atau menambahkan satu huruf juga dianggap satu langkah.

Pengunjung perpustakaan ataupun toko buku sering mengalami permasalahan dalam hal mencari buku. Pengunjung sulit untuk memastikan bahwa judul buku yang dia inginkan tersedia di toko ataupun perpustakaan tersebut. Oleh karena itu dibutuhkan suatu perangkat lunak untuk mencari judul buku. Namun dalam mencari judul buku bisa saja ada suatu judul buku yang mirip dengan dengan judul yang ingin dicari hanya saja posisi katanya tertukar. Misalkan pengunjung ingin mencari buku berjudul "Bahasa Pemograman Java" namun yang terdapat di perpustakaan atau toko buku itu hanyalah buku berjudul "Pemograman Bahasa Java". Pada kasus tersebut dapat disimpulkan bahwa terdapat judul buku yang mirip dengan judul buku yang dicari namun karena posisi katanya berbeda maka dianggap kedua judul buku itu berbeda. Algoritma *minimum edit distance* dapat digunakan untuk menangani masalah tersebut. Algoritma ini dapat memperhitungkan seberapa mirip suatu kata dengan kata lainnya. Namun demikian, pada kasus ini yang ingin dicari adalah kemiripan suatu judul dengan judul lainnya, karena itu lebih cocok membandingkan kata per kata dibanding huruf per huruf. Untuk itu, setiap kata diberi *id* yang unik dan judul buku dikonversi menjadi deretan *id-id* sesuai kata-kata yang terdapat di dalamnya. Struktur data *Hash Map* digunakan untuk menyimpan *id* kata-kata ini.

Pada penelitian ini dibangun suatu perangkat lunak untuk mencari judul buku dengan menggunakan algoritma *minimum edit distance*. Selanjutnya perangkat lunak ini akan diuji dengan menggunakan survey ke pengguna, untuk mengukur seberapa cocok algoritma ini digunakan dalam kasus pencarian judul buku.

Untuk mempercepat proses pencarian judul buku maka pada penelitian ini dibutuhkan pengetahuan mengenai *stop word*. *Stop word* adalah kata-kata yang dianggap tidak memiliki arti sehingga nantinya harus dihilangkan agar mempermudah dan mempercepat pencarian teks atau kalimat tersebut. Maka dari itu perlu dicari kata apa saja yang termasuk *stop word* agar kata tersebut dihilangkan terlebih dahulu ketika melakukan proses pencarian judul buku.

Perangkat lunak yang dibangun ingin menerima masukan suatu judul buku berupa teks. Teks tersebut adalah judul buku yang akan dicari. Perangkat lunak akan mengolah masukan tersebut dan menampilkan keluaran berupa teks berisi judul-judul buku yang dianggap paling mendekati dengan judul buku yang akan dicari.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, rumusan masalah pada penelitian ini adalah:

- *Stop word* seperti apa yang cocok digunakan untuk kasus pencarian judul buku?
- Bagaimana penggunaan *hash map* untuk memetakan setiap kata menjadi *id*?
- Bagaimana menggunakan algoritma *minimum edit distance* untuk mengetahui seberapa mirip dua judul buku?
- Bagaimana mengimplementasikan algoritma *minimum edit distance* ke dalam perangkat lunak pencarian judul buku.

1.3 Tujuan

Berdasarkan rumusan masalah yang telah dijelaskan, tujuan dari penelitian ini adalah:

- Mencari dan menentukan daftar *stop word* yang cocok untuk kasus pencarian judul buku.
- Mempelajari dan menggunakan HashMap untuk memetakan setiap kata menjadi *value*.
- Mempelajari dan menggunakan algoritma *minimum edit distance* untuk mengetahui seberapa mirip dua judul buku.
- Dapat mengimplementasikan algoritma *minimum edit distance* ke dalam perangkat lunak pencarian judul buku.

1.4 Batasan Masalah

Stop word pada judul buku harus dihilangkan. Jika semua kata pada judul buku merupakan *stop word* maka pencarian tidak dapat dilakukan sehingga terdapat batasan masalah sebagai berikut:

1. Judul buku yang dicari harus memiliki kata yang bukan merupakan *stop word*.
2. Judul buku tidak disimpan dalam *database* melainkan disimpan di *file*.
3. Judul buku yang tersimpan di *file* berjumlah 65.000 judul.

1.5 Metodologi

Metodologi Penelitian yang akan digunakan untuk melakukan penelitian ini adalah:

1. Melakukan studi pustaka untuk memahami teori-teori yang berkaitan dengan *stop word*, *hash map*, baca tulis *file* dalam pemograman Java dan algoritma *minimum edit distance*.
2. Melakukan analisis dengan cara membuat program-program sederhana berdasarkan landasan teori yang didapat dari hasil studi literatur.
3. Mengimplementasikan perangkat lunak pencarian judul buku berdasarkan hasil-hasil yang didapatkan dari analisis yang telah dilakukan.
4. Melakukan pengujian dan survey terhadap pengguna perangkat lunak yang telah dibangun.
5. Membuat kesimpulan berdasarkan penelitian yang telah dilakukan.

1.6 Sistematika Pembahasan

1. Bab 1 Pendahuluan

Bab ini akan menjelaskan latar belakang permasalahan, rumusan masalah, tujuan, batasan masalah, metodologi penelitian dan sistematika penulisan yang digunakan dalam penyusunan penelitian ini.

2. Bab 2 Dasar Teori

Bab ini berisi teori di bidang informatika yang dapat menunjang penelitian yang sedang dilakukan. Teori yang digunakan adalah teori mengenai *stop word*, *hashmap*, *minimum edit distance* dan cara baca tulis *file* pada bahasa pemrograman Java.

3. Bab 3 Analisis

Bab ini berisi analisis tugas akhir yang terdiri atas : analisis *stop word*, HashMap, algoritma *minimum edit distance* baca tulis file dan analisis perangkat lunak.

4. Bab 4 Perancangan

Bab ini berisi mengenai penjelasan perancangan perangkat lunak sederhana pencarian judul buku.

5. Bab 5 Implementasi dan Pengujian

Bab ini akan menjelaskan implementasi perangkat lunak sederhana pencarian judul buku serta cara dan hasil pengujian yang telah dilakukan.

6. Bab 6 Kesimpulan dan Saran

Bab ini akan menjelaskan mengenai kesimpulan dan saran dari penelitian ini.