

**SKRIPSI**

**PERBANDINGAN VECTOR SPACE MODEL DAN MODEL  
TRANSFORMASI FOURIER DISKRET UNTUK TEMU  
KEMBALI INFORMASI**



**Febrian Nathan Mintareja**

**NPM: 2016730020**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS  
UNIVERSITAS KATOLIK PARAHYANGAN  
2020**



**UNDERGRADUATE THESIS**

**COMPARISON OF THE VECTOR SPACE MODEL AND THE  
DISCRETE FOURIER TRANSFORMATION MODEL FOR  
INFORMATION RETRIEVAL**



**Febrian Nathan Mintareja**

**NPM: 2016730020**

**DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES  
PARAHYANGAN CATHOLIC UNIVERSITY  
2020**



## PERNYATAAN

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

### **PERBANDINGAN VECTOR SPACE MODEL DAN MODEL TRANSFORMASI FOURIER DISKRET UNTUK TEMU KEMBALI INFORMASI**

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 27 Juli 2020



Febrian Nathan Mintareja  
NPM: 2016730020

**LEMBAR PENGESAHAN**

**PERBANDINGAN VECTOR SPACE MODEL DAN MODEL  
TRANSFORMASI FOURIER DISKRET UNTUK TEMU  
KEMBALI INFORMASI**

**Febrian Nathan Mintareja**

**NPM: 2016730020**

**Bandung, 27 Juli 2020**

**Menyetujui,**

**Pembimbing**

**Kristopher David Harjono, M.T.**

**Ketua Tim Penguji**

**Anggota Tim Penguji**

**Luciana Abednego, M.T.**

**Lionov, Ph.D.**

**Mengetahui,**

**Ketua Program Studi**

**Mariskha Tri Adithia, P.D.Eng**



## **PERNYATAAN**

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

### **PERBANDINGAN VECTOR SPACE MODEL DAN MODEL TRANSFORMASI FOURIER DISKRET UNTUK TEMU KEMBALI INFORMASI**

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 27 Juli 2020



Febrian Nathan Mintareja  
NPM: 2016730020





## ABSTRAK

Temu kembali informasi sudah menjadi kegiatan sehari-hari yang biasa dilakukan. Untuk mendapatkan informasi menjadi sangat mudah dengan menggunakan mesin pencari di internet. Namun ada faktor-faktor yang mempersulit kegiatan temu kembali informasi. Faktor seperti pengguna yang kurang paham dengan topik yang akan dicari dapat terjadi. Selain faktor pengguna ada juga faktor dari mesin karena hasil pencarian mesin pencari satu dengan yang lain berbeda. Untuk mendapatkan hasil yang tepat tentunya perlu menggunakan metode yang tepat saat diterapkan pada mesin pencari.

Pada skripsi ini membuat mesin pencari untuk teks secara *offline*. Ada berbagai macam metode untuk melakukan temu kembali informasi. Salah satu metode temu kembali informasi dapat menggunakan *Vector Space Model*. *Vector Space Model* akan merepresentasikan dokumen menjadi vektor dan untuk menghitung kemiripan antara dokumen dapat dihitung dengan metode *cosine distance*. Cara lain dalam melakukan temu kembali informasi adalah dengan metode *Fourier Transform*.

Pada *Fourier Transform*, *query* diubah menjadi spektrum dan dokumen-dokumen menjadi sekumpulan *filter*. Untuk melakukan *Fourier Transform*, terdapat dua metode yaitu *Discrete Fourier Transform* dan *Fast Fourier Transform*. *Fast Fourier Transform* merupakan pengembangan dari *Discrete Fourier Transform*. Skripsi ini membahas dan menguji performa dari *Fourier Transform* dan *Vector Space Model* dengan menggunakan *cosine distance*.

Ada beberapa cara untuk mengukur performa dari mesin pencari. Dalam penelitian ini ukuran yang dipakai adalah waktu pemrosesan, *average precision*(AVP), *F-measurement*, *precision*, *recall*, dan *Discounted Cumulative Gain*(DCG). Dari hasil penelitian didapatkan performa *Fast Fourier Transform* bekerja lebih baik daripada *Vector Space Model* dengan *Cosine Distance*. Namun untuk *query* yang pendek *Vector Space Model* bekerja lebih baik daripada *Fast Fourier Transform*.

**Kata-kata kunci:** Temu kembali informasi, *Cosine Distance*, *Discrete Fourier Transform*(DFT), *Fast Fourier Transform*(FFT), *Vector Space Model*(VSM), Vektor, *average precision*(AVP), *F-measurement*, *precision*, *recall*, *Discounted Cumulative Gain*(DCG)



## ABSTRACT

Information retrieval has become common daily activity. Informations become very easy to get by using a search engine on the internet. There are many factors that make information retrieval difficult. Factor like the user don't really understand about the topic that they gonna search can happen. Aside from user factor, there is search engine factor because the result of one search engine can be different to others. To get the right results it is certainly necessary to use the right methods when applied to search engines.

This research makes an offline search engine for text. There are many various methods for information retrieval. Vector Space Model can be used in information retrieval. Vector Space Model will represent documents in vector and cosine distance can be used to calculate the distance between two documents.

Another way to do information retrieval is by using Fourier Transform method. Fourier Transform change the query into spectrum and change documents into a set of filters. There are two methods in Fourier Transform, one is Discrete Fourier Transform and another one is Fast Fourier Transform. This theses will discuss and examine the performance between Fourier Transform and Vector Space Model with cosine distance.

There are some way to measure the performance of search engine. Measurement metric used in this research are processing time, average precision(AVP), F measurement, precision, recall, Discounted Cumulative Gain(DCG). From the research results, the performance of Fast Fourier Transform works better than Vector Space Model with Cosine Distance. But for short queries Vector Space Model works better than Fast Fourier Transform.

**Keywords:** Information retrieval, Cosine Distance, Discrete Fourier Transform(DFT), Fast Fourier Transform(FFT), Vector Space Model(VSM), Vektor, average precision(AVP), F-measurement, precision, recall, Discounted Cumulative Gain(DCG)



*Untuk diri sendiri dan keluarga*



## KATA PENGANTAR

Puji Syukur penulis panjatkan kepada Tuhan Yesus karena berkat-Nya skripsi yang berjudul "Perbandingan Vector Space Model dan Model Transformasi Fourier Diskret untuk Temu Kembali Informasi" berhasil disusun dengan baik. Penulis berharap skripsi ini dapat menambah wawasan serta bermanfaat bagi orang lain. Skripsi ini tentu tidak akan selesai tanpa bantuan dari pihak yang terus mendorong penulis untuk menyelesaikannya. Oleh karena itu penulis menyampaikan terima kasih sebanyak-banyaknya kepada:

- Keluarga penulis, yang senantiasa mendampingi serta memotivasi penulis untuk menyelesaikan skripsi ini
- Ibu Natalia S.Si, M.Si yang selalu membimbing penulis dalam menyusun skripsi ini
- Teman-teman admin khususnya Cahyadi Hartanto, Yehezkiel Rusli, Gunawan Christianto, Ferdian Benyamin, Michael Liondi, Christian Gerald Patrick Saul Nasira yang pernah bekerja dengan penulis serta membantu penulis
- Teman-teman dekat penulis yang selalu memberikan dukungan moral
- Rekan-rekan informatika angkatan 2016 yang tidak dapat disebutkan satu-persatu
- Pihak-pihak lain yang senantiasa mendoakan penulis

Akhir kata, penulis meminta maaf sebesar-besarnya jika ada kesalahan dalam penulisan skripsi ini serta berharap skripsi ini dapat banyak membantu pihak yang membutuhkan.

Bandung, Juli 2020

Penulis





# DAFTAR ISI

<b>KATA PENGANTAR</b>	<b>xv</b>
<b>DAFTAR ISI</b>	<b>xvii</b>
<b>DAFTAR GAMBAR</b>	<b>xxi</b>
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Rumusan Masalah . . . . .	3
1.3 Tujuan . . . . .	3
1.4 Batasan Masalah . . . . .	3
1.5 Metodologi . . . . .	3
1.6 Sistematika Pembahasan . . . . .	4
<b>2 LANDASAN TEORI</b>	<b>5</b>
2.1 Temu Kembali Informasi . . . . .	5
2.2 Text Retrieval . . . . .	5
2.3 Vector Space Model . . . . .	6
2.4 Improved Vector Space Model . . . . .	7
2.5 IDF . . . . .	8
2.6 Term Frequency . . . . .	9
2.7 Normalisasi dan Cosine Distance . . . . .	9
2.8 Transformasi Fourier . . . . .	10
2.8.1 Discrete Fourier Transform . . . . .	11
2.8.2 Fast Fourier Transform . . . . .	11
2.9 Spectral Leakage . . . . .	12
2.10 Least Spectral Power Ranking . . . . .	13
2.10.1 Mentransformasikan <i>query</i> menjadi spektrum . . . . .	14
2.10.2 Mentransformasikan dokumen menjadi filter . . . . .	14
2.10.3 Melakukan perhitungan peringkat pada dokumen . . . . .	16
2.11 Struktur Data Trie . . . . .	17
2.12 Posting List . . . . .	18
2.13 Penilaian Performa . . . . .	18
2.13.1 Precision dan Recall . . . . .	19
2.13.2 F-measurement . . . . .	19
2.13.3 Evaluasi Peringkat . . . . .	20
2.14 Framework dan Tool . . . . .	22
2.14.1 Maven . . . . .	22
2.14.2 Blade Framework . . . . .	22
2.14.3 Bootstrap 4.0 . . . . .	23
<b>3 ANALISIS</b>	<b>25</b>
3.1 Analisis Domain Masalah . . . . .	25

3.2	Penerapan Framework dan Tool . . . . .	25
3.2.1	Penerapan Maven . . . . .	25
3.2.2	Penerapan Framework Blade . . . . .	26
3.3	Penerapan Algoritma . . . . .	27
3.3.1	Penerapan Metode Pemodelan TF-IDF . . . . .	27
3.3.2	Penerapan Cosine Distance . . . . .	29
3.3.3	Penerapan Perhitungan Fourier . . . . .	32
3.4	Rancangan Kelas Awal . . . . .	36
3.5	Mekanisme Program . . . . .	37
3.6	Analisis Masukan . . . . .	39
3.7	Analisis Output . . . . .	40
<b>4</b>	<b>PERANCANGAN</b> . . . . .	<b>41</b>
4.1	Diagram Kelas . . . . .	41
4.2	Penjelasan Kelas . . . . .	42
4.2.1	Kelas App . . . . .	42
4.2.2	Kelas Path . . . . .	42
4.2.3	Kelas Indexer . . . . .	43
4.2.4	Kelas Trie . . . . .	44
4.2.5	Kelas TrieNode . . . . .	46
4.2.6	Kelas PostingList . . . . .	46
4.2.7	Kelas Posting . . . . .	47
4.2.8	Kelas TF-IDF . . . . .	48
4.2.9	Kelas DocLength . . . . .	49
4.2.10	DocScore . . . . .	51
4.2.11	Kelas VSM . . . . .	51
4.2.12	Kelas Fourier . . . . .	52
4.2.13	Kelas Complex . . . . .	56
4.2.14	Kelas IndexController . . . . .	57
4.2.15	Kelas Bridge . . . . .	58
4.2.16	Kelas PerformanceCalculator . . . . .	59
4.3	Sistem Frontend . . . . .	60
4.3.1	Rancangan UI . . . . .	60
<b>5</b>	<b>PENGUJIAN</b> . . . . .	<b>63</b>
5.1	Pengujian Fungsionalitas . . . . .	63
5.2	Pengujian Performa Pencarian . . . . .	71
5.2.1	Query Sinfull Pride Doing Quite Disgrace . . . . .	72
5.2.2	Query From Fairest Creatures . . . . .	77
5.2.3	Query Childe Only Yours . . . . .	78
5.2.4	Query Louers Very Cruell Knife . . . . .	80
5.2.5	Query Beauty Summer . . . . .	82
5.2.6	Query Hidious night . . . . .	83
5.2.7	Query These Beauty . . . . .	85
5.2.8	Query These Beauty Liue . . . . .	86
5.2.9	Query Through Fire . . . . .	88
5.2.10	Query Through Fire Only . . . . .	89
5.2.11	Query Through Fire Only Sleepe . . . . .	91
5.3	Analisis Pengujian . . . . .	93
<b>6</b>	<b>KESIMPULAN DAN SARAN</b> . . . . .	<b>95</b>
6.1	Kesimpulan . . . . .	95

6.2 Saran . . . . .	95
<b>DAFTAR REFERENSI</b>	<b>97</b>
<b>A KODE PROGRAM</b>	<b>99</b>
<b>B PERHITUNGAN PERFORMA</b>	<b>111</b>
<b>C HASIL EKSPERIMEN</b>	<b>145</b>
<b>D DATA SET DOKUMEN</b>	<b>147</b>



## DAFTAR GAMBAR

1.1	Contoh hasil pencarian dengan Yandex . . . . .	1
2.1	Nilai TF dengan log . . . . .	9
2.2	Gelombang sinus . . . . .	13
2.3	Gelombang sinus (2) . . . . .	13
2.4	<i>breadth</i> , <i>ZL</i> dan <i>ZR</i> . . . . .	14
2.5	Spektrum yang belum difilter . . . . .	15
2.6	Spektrum setelah difilter . . . . .	15
2.7	Trie . . . . .	17
2.8	Posting List . . . . .	18
3.1	Contoh Implementasi Mekanisme Perhitungan Peringkat . . . . .	31
3.2	Contoh Implementasi Mekanisme Perhitungan Peringkat(2) . . . . .	31
3.3	Contoh Implementasi Mekanisme Perhitungan Peringkat(3) . . . . .	31
3.4	Ilustrasi Gelombang Spektrum pada <i>query</i> . . . . .	33
3.5	Ilustrasi Gelombang dengan <i>D1</i> Sebagai Filter . . . . .	33
3.6	Ilustrasi Gelombang dengan <i>D2</i> Sebagai Filter . . . . .	34
3.7	Ilustrasi Gelombang dengan <i>D3</i> Sebagai Filter . . . . .	34
3.8	Rancangan Kelas Awal . . . . .	36
3.9	Mekanisme Utama Program . . . . .	37
3.10	Mekanisme VSM . . . . .	38
3.11	Mekanisme FFT . . . . .	39
4.1	Diagram Kelas . . . . .	41
4.2	Diagram Kelas App . . . . .	42
4.3	Diagram Kelas Path . . . . .	42
4.4	Diagram Kelas Indexer . . . . .	43
4.5	Diagram Kelas Trie . . . . .	44
4.6	Diagram Kelas Trienode . . . . .	46
4.7	Diagram Kelas Postinglist . . . . .	46
4.8	Diagram Kelas Posting . . . . .	47
4.9	Diagram kelas TF-IDF . . . . .	48
4.10	Diagram Kelas Doclength . . . . .	49
4.11	Diagram Kelas Docscore . . . . .	51
4.12	Diagram Kelas Vsm . . . . .	51
4.13	Diagram Kelas Fourier . . . . .	52
4.14	Diagram Kelas Complex . . . . .	56
4.15	Diagram Kelas Indexcontroller . . . . .	57
4.16	Diagram Kelas Indexcontroller . . . . .	58
4.17	Diagram Kelas Performance Calculator . . . . .	59
4.18	Output level UI . . . . .	60
4.19	Output level UI(2) . . . . .	61
4.20	Output level UI(3) . . . . .	61

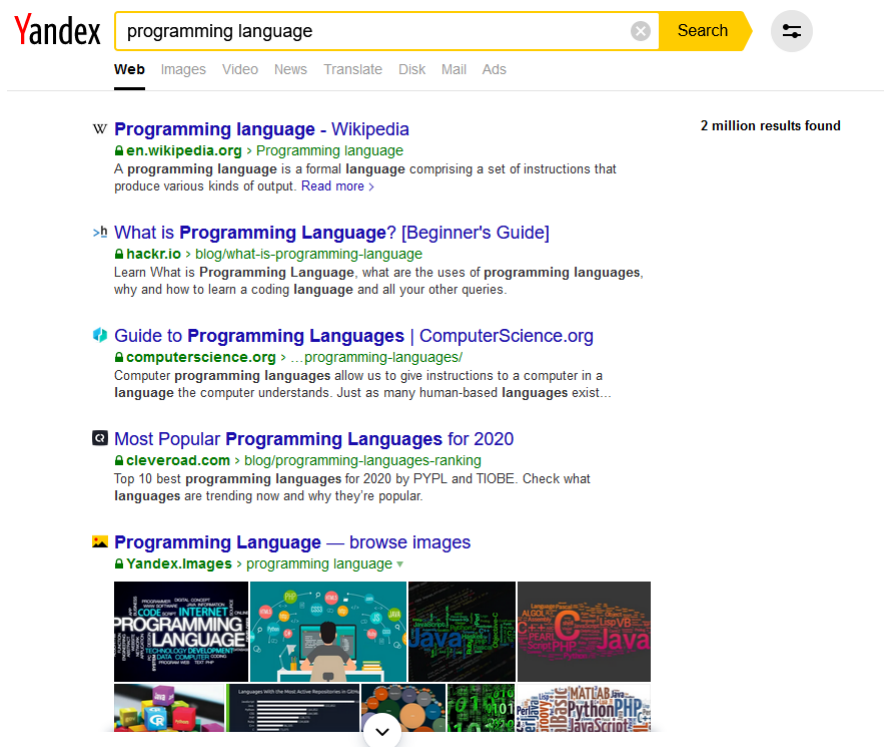
5.1 Uji Fungsionalitas Pertama	63
5.2 Uji Fungsionalitas Pertama (2)	64
5.3 Uji Fungsionalitas Pertama (3)	64
5.4 Uji Fungsionalitas Pertama (4)	65
5.5 Uji Fungsionalitas Pertama (5)	65
5.6 Uji Fungsionalitas Pertama (6)	66
5.7 Uji Fungsionalitas Pertama (7)	66
5.8 Uji Fungsionalitas Pertama (8)	67
5.9 Uji Fungsionalitas Kedua	67
5.10 Uji Fungsionalitas Kedua (2)	68
5.11 Uji Fungsionalitas Kedua (3)	68
5.12 Uji Fungsionalitas Kedua (4)	69
5.13 Uji Fungsionalitas Kedua (5)	69
5.14 Uji Fungsionalitas Kedua (6)	70
5.15 Uji Fungsionalitas Ketiga	70
5.16 Uji Fungsionalitas Ketiga (2)	71
5.17 Uji Fungsionalitas Ketiga (3)	71
C.1 Perbandingan waktu DFT, FFT, VSM	145
C.2 Perbandingan AVP FFT dan VSM	145
C.3 Perbandingan F1 FFT dan VSM	146
C.4 Perbandingan DCG/IDCG FFT dan VSM	146

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang [1]

Temu kembali informasi adalah cabang yang luas dari ilmu komputer terutama dalam memberikan pengguna kemudahan dalam mengakses informasi. Temu kembali informasi dapat berupa pencarian dokumen-dokumen yang terkait dengan *query* kata-kata yang diinginkan oleh pengguna. Setelah itu, hasil dari pencarian akan diurutkan berdasarkan relevansi dokumen. Contohnya pada saat melakukan pencarian dengan *google*. Pengguna memasukan *query* pencarian dan setelahnya akan ditampilkan hasil pencarian dapat berupa judul dokumen yang relevan serta sekumpulan teks yang menandakan bagian dari dokumen yang relevan dengan *query*. Pada zaman sekarang, kegiatan ini tentu sering dilakukan.



Gambar 1.1: Contoh hasil pencarian dengan Yandex

Pada Gambar 1.1 merupakan contoh hasil dari temu kembali informasi secara *online* dengan menggunakan mesin pencari Yandex. Kegiatan mencari informasi belum tentu mudah. Banyak faktor-faktor yang mempersulit pencarian seperti ketika pengguna tidak begitu paham dengan informasi yang ingin pengguna cari sehingga kesulitan dalam mencari *query* yang tepat. Selain

<sup>0</sup>Source: screenshot dari hasil pencarian menggunakan yandex.com



faktor dari manusia, ada juga faktor dari mesin pencariinya sendiri. Performa berbagai mesin pencari dapat berbeda tergantung dari sistem yang diterapkan. Hasil pencarian dari suatu mesin pencari dapat berbeda dengan mesin pencari lain.

Ada beberapa cara untuk mencapai hasil pencarian seperti mencocokkan dokumen satu persatu dengan *query* dan dokumen yang memiliki banyak kesamaan dengan *query* merupakan dokumen yang relevan. Namun hasilnya belum tentu tepat dan efisien. Pada skripsi ini, akan dibuat sebuah perangkat lunak untuk melakukan temu kembali informasi. Ada beberapa teknik yang dapat digunakan sebagai metode pencarian, metode yang digunakan pada skripsi ini adalah *Vector Space Model* (VSM) dan *Fourier Discrete*.

Pada skripsi ini dibuat sebuah mesin pencarian *offline* yang menerima masukan *query* bertipe teks dan objek yang dicari akan berupa *file* bertipe teks. Sebelum melakukan temu kembali informasi, tentunya harus ada himpunan data terlebih dahulu. Untuk data bertipe teks, data dapat dibersihkan dari kata-kata yang kurang penting dengan *stemmer*. *Stemmer* akan mengubah kata berimbuhan menjadi kata dasar dan menghilangkan kata-kata yang kurang penting. Namun pada skripsi ini *stemmer* tidak digunakan. Selain faktor waktu, *stemmer* memiliki kelemahan. Contohnya kata *better* yang seharusnya jika diubah menjadi kata dasar akan menjadi *good* namun *stemmer* gagal untuk mengubahnya. Selain itu, kata *axes* yang merupakan bentuk jamak dari kata *axe* dan *axis*. Namun pada aturan *stemmer* akan menghilangkan huruf "s" saja.

*Posting list* adalah metode penyimpanan kemunculan suatu kata atau term pada dokumen-dokumen atau korpus. Sebelum masuk ke penjelasan *posting list*, korpus adalah koleksi dokumen yang digunakan dalam pencarian. Pada skripsi ini, karakter selain alfabet dihilangkan. Hal ini dikarenakan karakter selain angka tidak berpengaruh banyak terhadap informasi data. Setelah dibersihkan, langkah selanjutnya adalah pengindeksan dokumen-dokumen. Indeks dokumen akan disimpan menggunakan struktur data *trie* dan *posting list*. Secara singkat *trie* adalah sebuah *tree* yang akan menyimpan *posting list*.

Setelah pengindeksan langkah selanjutnya adalah pemodelan metode pencarian. Langkah pertama pemodelan *Vector Space Model* adalah mengubah dokumen dan *query* sebagai vektor. Masing-masing dokumen dan *query* akan direpresentasikan menjadi vektor. Bobot tiap dimensi dari vektor tersebut ditentukan berdasarkan frekuensi kemunculan kata-kata pada dokumen tersebut lalu dikalikan dengan suatu konstanta untuk membuat pembobotan semakin baik. Tiap vektor dokumen akan dihitung kedekatannya dengan vektor *query* menggunakan metode *cosine distance*.

Metode yang digunakan untuk menghitung kedekatan vektor tidak selalu harus menggunakan *cosine distance*. Ada cara lain seperti dengan menggunakan perhitungan jarak antar vektor menggunakan *euclidean distance*. Penggunaan metrik perhitungan jarak yang lain tentu dapat digunakan sesuai dengan kebutuhan. Namun *euclidean distance* kurang cocok digunakan karena jika hasil perhitungan semakin kecil berarti semakin mirip. Karena itu pada skripsi ini, perhitungan jarak yang dipilih adalah *cosine distance* dan metode ini cocok digunakan pada skripsi ini. Perhitungan dengan *cosine distance* akan membagi vektor dengan panjang vektornya sehingga panjang dokumen tidak lagi menjadi masalah. Setelah mendapatkan nilai kedekatannya, dokumen-dokumen tersebut akan diurutkan berdasarkan nilai kedekatannya. Dokumen yang berada pada urutan pertama adalah yang paling relevan dan yang terakhir paling tidak relevan.

Dalam pemodelan *Fourier Diskret*, *query* dipandang sebagai sebuah spektrum dan masing-masing dokumen sebagai sekumpulan filter. Masing-masing kata di dalam *query* akan terkait dengan sebuah sinyal sinusoidal yang kemudian membentuk sebuah fungsi. Hasil spektrum tersebut kemudian akan difilter oleh filter dari masing-masing dokumen. Hasil filter tersebut kemudian diurutkan untuk mendapatkan peringkat relevansi masing-masing dokumen dengan *query*.

Dalam menguji mana yang lebih baik, tentunya harus dilakukan komparasi dari *output* kedua algoritma. Ada beberapa ukuran yang dapat dijadikan patokan dalam menilai performa kedua algoritma seperti *precision* dan *recall*. Secara singkat *precision* merupakan tingkat ketepatan mesin pencari dalam mengeluarkan hasil. Sedangkan *recall* menandakan berapa banyak keluaran yang dihasilkan mesin pencari dari yang seharusnya. Selain *precision* dan *recall*, ukuran yang dilihat juga

adalah peringkat dokumen. Ukuran lain yang dapat digunakan dalam menilai hasil pencarian dapat menggunakan *Discounted Cumulative Gain* (DCG), *Average Precision* (AVP) dan *F-Measurement* yang akan dibahas pada landasan teori. DCG merupakan ukuran untuk melihat apakah peringkat yang diberikan sudah baik atau tidak. AVP merupakan ukuran untuk melihat tingkat presisi dari mesin pencari dalam menaruh dokumen relevan. Sedangkan *F-measurement* merupakan gabungan dari *precision* dan *recall*. Hasil keluaran yang baik menaruh dokumen paling relevan pertama dan yang kurang relevan terakhir atau berada pada posisi akhir.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, rumusan masalah dari penelitian ini adalah sebagai berikut:

- Bagaimana cara memanfaatkan *Vector Space Model* dan *Fourier Diskret* untuk melakukan pencarian pada sistem temu kembali informasi?
- Bagaimana cara mengimplementasikan model *Vector Space Model* dan *Fourier Diskret* dalam temu kembali informasi?
- Bagaimana performa model *Vector Space Model* dibandingkan dengan *Fourier Diskret* dalam temu kembali informasi?

## 1.3 Tujuan

Berdasarkan Rumusan masalah yang telah dipaparkan, tujuan penelitian ini adalah sebagai berikut:

- Memanfaatkan model *Vector Space Model* dan *Fourier Diskret* dalam melakukan pencarian pada sistem temu kembali informasi
- Mengimplementasikan cara kerja model *Vector Space Model* dan *Fourier Diskret* dalam program temu kembali informasi
- Menganalisis performa *Vector Space Model* dan *Fourier Diskret* dalam program temu kembali informasi

## 1.4 Batasan Masalah

Dari fokus yang telah ditetapkan dan keterbatasan waktu, batasan penelitian adalah sebagai berikut:

- *Indexing* tidak akan menggunakan *stemmer*
- Korpus akan berisi teks dan *file* bertipe penyimpanan teks
- Karakter selain alfabet pada teks dokumen akan dihilangkan

## 1.5 Metodologi

Langkah-langkah untuk melakukan penelitian ini adalah :

1. Melakukan studi literatur tentang pencarian temu kembali informasi
2. Mempelajari struktur data *trie* pada sistem
3. Mempelajari model *Vector Space Model*
4. Mempelajari model *Fourier Diskret*
5. Memodelkan *Vector Space Model* dan *Fourier Diskret* pada sitem pencarian temu kembali informasi
6. Merancang perangkat lunak
7. Menerapkan sistem temu kembali informasi dalam bentuk *coding*
8. Menganalisis kinerja pencarian dan mengambil kesimpulan
9. Menulis dokumen skripsi

## 1.6 Sistematika Pembahasan

1. Pada bab pendahuluan membahas tentang latar belakang penelitian ini dilakukan, rumusan masalah, tujuan, batasan masalah, metodologi, dan sistematika pembahasan dari penelitian ini. Pada bab ini memberikan gambaran singkat dan batasan-batasan dari skripsi ini. Pada bab ini juga menjelaskan metode yang digunakan dalam melakukan penelitian.
2. Pada bab landasan teori membahas apa saja teori-teori yang dibutuhkan dalam melakukan penelitian ini. Pada bab ini memuat teori *information* dan *text retrieval*, *Vector Space Model*, *Improved Vector Space Model*, *TF IDF*, *cosine distance*, *Discrete Fourier Transform*, *Fast Fourier Transform*, *Spectral Leakage*, *Least Spectral Power Ranking*, *Trie*, *Posting List*, penilaian performa, *framework* dan *tool*. Pada bagian pembahasan *vector space model* dan algoritma *fourier* membahas tentang teknik-teknik dasar pada pencarian yang digunakan pada skripsi ini. Pada *Spectral Leakage* membahas apa itu *Spectral Leakage* serta mengapa *Spectral Leakage* dibutuhkan. Pada bagian *Least Spectral Power Ranking* membahas definisi *Least Spectral Power Ranking*. Pada bagian *Trie* membahas apa itu *Trie* dan mengapa digunakan pada penelitian ini. Pada bagian *Posting List* membahas apa itu *Posting List*. Pada bagian penilaian performa membahas ukuran apa saja yang digunakan untuk menilai performa dari hasil pencarian. Pada bagian *framework* dan *tool* membahas *framework* apa saja yang digunakan pada penelitian ini serta *tool* atau alat bantuan yang digunakan pada penelitian ini.
3. Pada bab analisis membahas analisis domain masalah, penerapan algoritma, rancangan kelas awal, mekanisme program, analisis *input* dan *output*. Pada bagian analisis domain membahas analisis dari permasalahan pada penelitian ini. Pada bagian penerapan algoritma membahas langkah dalam menerapkan algoritma-algoritma kedalam bentuk kode. Pada bagian kelas awal menampilkan diagram kelas rancangan awal dari program yang dibuat. Pada mekanisme program membahas tentang gambaran kasar dari cara program bekerja. Pada bagian analisis *input* dan *output* membahas tentang data apa yang program terima dan keluaran dari program.
4. Pada bab *software* membahas sistem *backend* program, diagram kelas, penjelasan kelas, dan sistem *frontend*. Pada bagian sistem *backend* hanya berisi pengantar penjelasan dari cara program bekerja pada bagian *backend*. Pada bagian diagram kelas menampilkan gambar diagram kelas dari program. Pada bagian penjelasan kelas membahas semua kelas pada program satu persatu. Setelah itu pada sistem *frontend* akan membahas bagian *frontend* dari program dan menampilkan rancangan dari antar muka program.
5. Pada bab pengujian membahas pengujian fungsionalitas, pengujian performa pencarian dan analisis pengujian. Pada bagian pengujian fungsionalitas membahas apakah program yang dibuat sudah bekerja dengan benar apa belum. Pada bagian pengujian performa pencarian membahas performa pencarian pada tiap *query* dan menampilkan hasil dari perhitungan yang telah dilakukan untuk setiap *query*. Pada bagian analisis pengujian menyatukan hasil pengujian dan menampilkan data secara grafis agar mudah untuk divisualisasikan.
6. Pada bab kesimpulan dan saran membahas tentang kesimpulan yang berhasil didapatkan dari pengujian ini serta membahas saran untuk penelitian ini.