

## ANALISIS *LATENT SEMANTIC INDEXING* MENGGUNAKAN *QR DECOMPOSITION* DENGAN TRANSFORMASI *HOUSEHOLDER* UNTUK Mencari Informasi

### ANALYSIS *LATENT SEMANTIC INDEXING* USING *QR DECOMPOSITION* WITH *HOUSEHOLDER* TRANSFORMATION TO SEARCH FOR INFORMATION

Muhamad Murtha Hamda<sup>1</sup>, Ibnu Asror, ST., M.T.<sup>2</sup>

<sup>1,3</sup>Prodi S1 Teknik Informatika, Fakultas Informatika, Universitas Telkom

<sup>3</sup>Prodi D3 Manajemen Pemasaran, Fakultas Ilmu Terapan, Universitas Telkom

<sup>2</sup>Prodi S1 Desain Komunikasi Visual, Fakultas Industri Kreatif, Universitas Telkom

<sup>1</sup>mhmurthahamda.student@telkomuniversity.ac.id, <sup>2</sup>iasror@telkomuniversity.ac.id

#### Abstrak

Perkembangan *Information Retrieval* telah berkembang dengan banyak metode yang berfungsi menghasilkan tingkat relevansi yang lebih baik. Untuk dapat menghasilkan nilai relevansi yang tinggi, agar maka diperlukan sebuah metode untuk menghasilkan perankingan yang baik dan teruji. Pada Tugas Akhir ini melakukan analisis *Latent Semantic Indexing* menggunakan *QR decomposition* dengan transformasi *householder*, kemudian untuk mengukur kemiripan dokumen terhadap *query* menggunakan *cosine similarity* dan parameter pengujian akurasi sistem menggunakan *recall* dan *precision* supaya dapat membuktikan kemampuan dalam *latent semantic indexing* dapat menemukan dokumen yang diinginkan atau relevan walaupun tidak ada *term* yang ada pada *query* dan melakukan perbandingan waktu proses pencarian dokumen. Hasil pengujian dari tugas akhir ini menunjukkan *latent semantic indexing* menggunakan *QR Decomposition* dengan transformasi *householder* terbukti bisa menemukan dokumen relevan walau tidak mengandung *term* yang terdapat pada *query* kemudian memiliki nilai *recall* dan *precision* nilai akurasi sistem yang baik dan juga mendapatkan proses waktu pencarian dokumen yang relevan yang cepat.

**Kata kunci :** *latent semantic indexing (LSI)*, *qr decomposition*, transformasi *householder*, *recall*, *precision*.

#### Abstract

*The development of Information Retrieval has evolved with many methods that serve to produce a better level of relevance. To be able to produce a high relevance value, so it needs a method to produce good and tested rankings. This Final Project performs Latent Semantic Indexing analysis using QR decomposition with householder transformation, then for measuring document resemblance to query using cosine similarity and parameter of system accuracy using recall and precision in order to prove the ability in latent semantic indexing to find the desired or relevant document Although there is no term in the query and do the time comparison of the document search process. The test result from this final project shows the latent semantic indexing using QR Decomposition with householder transformation proved able to find the relevant document even if it does not contain the term contained in the query then has the value of recall and precision good system accuracy value and also get the process of searching the relevant documents.*

**Keywords:** *latent semantic indexing (LSI)*, *qr decomposition*, *householder transformation*, *recall*, *precision*.

#### 1. Pendahuluan

Kebutuhan manusia akan informasi di masa teknologi semakin meningkat, seiring dengan meningkatnya kecerdasan manusia secara keseluruhan. Semakin meningkatnya kecerdasan manusia, semakin membutuhkan informasi secara cepat dan akurat sesuai dengan kebutuhannya saat itu. Dalam upaya untuk memenuhi kebutuhan tersebut, telah hadir berbagai media untuk mendapatkan informasi dan dapat diakses dimana saja yaitu dengan internet melalui *platform* komputer, laptop, tablet, dan *smartphone*. Namun bagaimana jika terdapat ratusan ribu informasi yang berbeda, sedangkan informasi yang dibutuhkan spesifik. Untuk itu dikembangkanlah cara untuk mengatasi permasalahan ini, yaitu *Information Retrieval*.

*Information Retrieval* adalah proses pencarian informasi dari sekumpulan dokumen yang sesuai dengan informasi yang dibutuhkan. Proses yang terjadi dalam *Information Retrieval* pada dasarnya untuk memenuhi keinginan pengguna akan informasi dengan cara menentukan dokumen dalam sekumpulan dokumen yang harus ditemubalikkan dan bentuk *query* merupakan representasi dari informasi yang diinginkan pengguna[2]. Salah satu metode yang dapat mengatasi ini adalah LSI (*Latent Semantic Indexing*). Metode ini mempresentasikan term dari dokumen menjadi vektor yang berbentuk *term* dokumen matriks sehingga dapat menghubungkan antara *term* dengan dokumen yang dicari walaupun *query* yang dimasukan tidak terdapat dalam dokumen yang dicari. Dari hasil penelitian sebelumnya menyatakan bahwa LSI menggunakan metode *QR Decomposition* dengan teknik *gram schmid* memiliki nilai *recall* dan *precesion* yang baik akan tetapi matriks ortogonalnya yang tidak stabil yang mengakibatkan akurasinya berkurang[13]. Oleh karena itu, untuk

mendekomposisi matriks tersebut penulis akan menggunakan metode *QR decomposition* dengan transformasi *Householder*. Dekomposisi QR (juga disebut QR faktorisasi) dari matriks adalah dekomposisi matriks A menjadi produk  $A = QR$ . Dekomposisi QR adalah salah satu alat dasar dalam komputasi statistika dan digunakan untuk mencari matriks *orthogonal* Q dan matriks segitiga atas R.

## 2. Dasar Teori

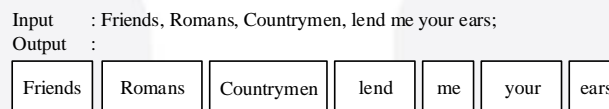
### 2.1 Information Retrieval

*Information Retrieval* adalah studi tentang sistem pengindeksan, pencarian, dan mengingat data, khususnya teks atau bentuk tidak terstruktur lainnya [4]. Baeza-Yates dan Ribiero-Neto menekankan bahwa tujuan utama dari *Information Retrieval* ialah untuk mendapatkan semua dokumen yang relevan terhadap kata kunci (*query*) yang diberikan oleh *user* dengan sesedikit mungkin dokumen yang tidak relevan [1]. *Information Retrieval* merupakan bagian dari *computer science* yang berhubungan dengan pengambilan informasi dari dokumen-dokumen yang didasarkan pada isi dan konteks dari dokumen-dokumen itu sendiri. Untuk pencarian informasi dalam *Information Retrieval* didasarkan pada suatu *query* (inputan *user*) yang diharapkan dapat memenuhi keinginan *user* dari kumpulan dokumen yang ada. Sedangkan, definisi *query* dalam *Information Retrieval* menurut referensi merupakan sebuah formula yang digunakan untuk mencari informasi yang dibutuhkan oleh *user* dalam bentuk yang paling sederhana. Sebuah *query* merupakan *keywords* (kata kunci) dan dokumen yang mengandung *keywords* merupakan dokumen yang dicari dalam *information retrieval*.

### 2.2 Indexing

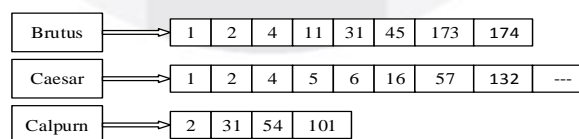
Tahapan dalam *indexing* [9], yaitu :

1. *Tokenization* merupakan tahap memotong suatu dokumen menjadi unit terkecil atau potongan-potongan yang disebut token (gambar 2-1).



Gambar Error! No text of specified style in document.-1 : Proses Tokenization

2. *Stop Words* merupakan proses menghilangkan kata (*term*) yang memiliki nilai kecil atau tidak memiliki makna dan frekuensi kemunculan sangat sering pada setiap dokumen sesuai dengan linguistik bahasa masing-masing.
3. *Case Folding* merupakan proses normalisasi seluruh kata menjadi huruf kecil atau huruf besar semua.
4. *Inverted Index* adalah konsep utama dalam pencarian informasi. *Inverted index* merupakan proses pemetaan *term* berdasarkan kemunculan pada sebuah dokumen (gambar 2-2).



Gambar Error! No text of specified style in document.-2 : Contoh Inverted Index

### 2.3 TF-IDF ( Term Frequency – Inverse Document Frequency )

*Term Frequency* (Tf) [9] merupakan skema pembobotan berdasarkan jumlah frekuensi kemunculan *term* (t) pada dokumen (d). Lalu *document frequency* (df) merupakan jumlah dokumen dalam koleksi yang berisi *term* (t). Sedangkan *inverse document frequency* (idf) dirumuskan sebagai berikut :

$$idf_t = \log \frac{N}{df_t}$$

( Error! No text of specified style in document..1)

Dimana  $N$  adalah jumlah dokumen yang ada. Setelah itu baru menggabungkan definisi istilah  $tf$  dan  $idf$  untuk menghasilkan pembobotan untuk setiap  $term$  ( $t$ ) pada dokumen ( $d$ ). Sehingga skema pembobotan  $Tf-idf$  adalah sebagai berikut :

$$tf \cdot idf_{t,d} = tf_{t,d} \times idf_t$$

( Error! No text of specified style in document..2)

## 2.4 Term by Document Matrix

Dalam *Term by document matrix* (tabel 2-1) setiap kolom mendefinisikan dokumen sedangkan setiap baris mewakili sebuah *term* unik atau kata kaunci dalam koleksi. Nilai – nilai yang disimpan dalam elemen matriks atau sel (*cell*) mendefinisikan frekuensi kemunculan sebuah *term* pada setiap dokumen [2].

Tabel Error! No text of specified style in document.-1 : Small Term by Document Matrix

| Term   | Dokumen 1 | Dokumen 2 | Dokumen 3 | Dokumen 4 |
|--------|-----------|-----------|-----------|-----------|
| Term 1 | 1         | 0         | 1         | 0         |
| Term 2 | 0         | 0         | 1         | 1         |
| Term 3 | 0         | 1         | 1         | 0         |

## 2.5 Latent Semantic Indexing

Latent Semantic Indexing merupakan varian dari *Vector Space Model* (VSM) yang memetakan dimensi yang tinggi ke dimensi yang kecil. Pengurangan dimensi dari data yang berdimensi tinggi bisa meningkatkan akurasi dan efektifitas untuk analisis data [3,8]. LSI memiliki kemampuan untuk menangani kasus sinonim, yaitu kasus dimana kata (*term*) berbeda dapat memiliki makna yang sama. Dalam LSI, konsep dalam *query* maupun dalam semua dokumen yang saling berhubungan akan direpresentasikan oleh nilai yang hampir sama.

Berdasarkan referensi [10], disebutkan bahwa LSI memiliki beberapa keuntungan yaitu:

1. Asumsi dalam LSI yaitu dimensi yang baru memiliki representasi yang lebih baik terhadap dokumen dan *query*.
2. LSI mampu menangani kasus sinonim, yaitu kasus dimana kata berbeda dapat memiliki maksud yang sama. Dalam LSI, konsep dalam *query* maupun dalam semua dokumen yang saling berhubungan akan direpresentasikan oleh nilai yang hampir sama.
3. Polisemi merupakan kasus dimana suatu kata dapat memiliki lebih dari satu makna, dengan banyaknya kata-kata yang bersifat polisemi maka akan menurunkan tingkat ketelitian. Dengan menggunakan representasi yang terbentuk dalam LSI, maka diharapkan untuk menghilangkan beberapa “noise” dari data, yang dapat diterjemahkan sebagai *term* yang jarang digunakan maupun tidak penting.

Vector space model mengasumsikan bahwa *term* bersifat independen. Pada kenyataannya, dalam bahasa terdapat relasi yang kuat antar *term*. Oleh karena itu terdapat kemungkinan untuk meningkatkan performansi dengan menggunakan hubungan *term* dalam proses retrieval.

## 2.6 QR Decomposition

Dalam aljabar linear, QR dekomposisi (juga disebut QR faktorisasi) adalah faktorisasi terhadap suatu matriks sehingga matriks ortogonal ( $Q$ ) dan matriks segitiga atas ( $R$ ). Teknik faktorisasi QR dapat digunakan untuk menemukan nilai karakteristik. Teknik tersebut menggunakan iterasi dari transformasi *Householder* untuk menemukan matriks  $A^{k+1}$  yang ekuivalen dengan matriks  $A$ . Elemen diagonal utama matriks  $A^{k+1}$  adalah nilai karakteristik matriks  $A$ . Faktorisasi QR dapat memberikan nilai karakteristik ( $\lambda$ ) secara simultan[4].

Langkah-langkah menentukan nilai karekteristik menggunakan QR :

1. Faktorisasi  $A$  menjadi matriks ortogonal ( $Q$ ) dan matriks segitiga atas ( $R$ ) sehingga  $A = Q * R$ . Matriks  $Q$  diperoleh menggunakan transformasi *Householder* :  $Q = I - 2ww^T$
2. Tentukan matriks similaritas  $A^{(k+1)}$  sehingga  $A^{(k+1)} = R * Q$

3. Faktorisasi lagi matriks  $A^{(k+1)}$  menjadi matriks  $Q$  dan  $R$  sehingga  $A^{(k+1)} = Q * R$ .
4. Lakukan secara berulang langkah 1 sampai dengan langkah 3 sampai di peroleh matriks  $A^{(k+1)}$  yang konvergen.
5. Elemen diagonal utama pada matriks  $A^{(k+1)}$  adalah pendekatan nilai karakteristik ( $\lambda$ ).

### 2.6.1 Trasformasi Householder

Householder merupakan membentuk matriks similaritas  $A^{(k+1)}$  di mana elemen di bawah diagonal utamanya bernilai 0 atau mendekati 0 [4]. Cara menentukan Matriks Householder adalah :

Tahap 1. Tentukan vektor :  $x_1, x_2, \dots, x_{k-1}$

Tahap 2. Hitung :  $g = \sqrt{a_{k1}^2 + a_{(k+1)1}^2 + a_{n1}^2}$

$$\alpha = \text{sign}(a_{k1})g$$

Tahap 3. Hitung :  $s^2 = (a_{k1} + \alpha)^2 + a_{(k+1)1}^2 + \dots + a_{n1}^2$

$$= g^2 + 2\alpha(a_{k1}) + \alpha^2 = g^2 + 2\alpha(x_{k1}) + g^2$$

$$s = \sqrt{2g^2 + 2\alpha|a_{k1}|} = \sqrt{2g(g + |a_{k1}|)}$$

Tahap 4. Tentukan :  $w_k = \frac{(a_{k1} + \alpha)}{s} = \frac{(a_{k1} + \text{sign}(a_{k1})g)}{s}$

$$w_i = \frac{a_{i1}}{s}, i = k + 1, k + 2, \dots, n$$

$$w = \frac{1}{s} [0 \quad 0 \quad \dots \quad a_{k1} + \alpha \quad a_{(k+1)1} \quad \dots \quad a_{n1}]^T$$

Tahap 5. Tentukan :  $H = I - 2ww^T$

(2,4)

### 2.7 Cosine Similarity

Cara standar untuk menghitung nilai kesamaan atau kecocokkan antara dua dokumen  $d_1$  dan  $d_2$  adalah dengan menghitung *cosine similarity* dari representasi vektor kedua dokumen tersebut. [9].

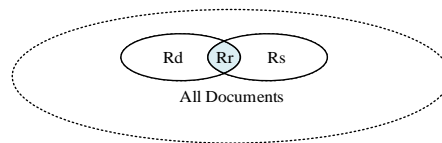
$$\text{sim}(d_1, d_2) = \frac{\vec{v}(d_1) \cdot \vec{v}(d_2)}{|\vec{v}(d_1)| |\vec{v}(d_2)|}$$

( Error! No text of specified style in document.4 )

### 2.7 Recall dan Precision

Sistem temu-kembali informasi yang bertujuan mencari dokumen-dokumen yang relevan terhadap *query User*. Dalam pengukuran relevansi dapat dilihat dengan parameter-parameter keefitfan relevansi seperti, *recall* dan *precision*.

*Recall* adalah proporsi jumlah dokumen yang dapat ditemukan-kembali oleh sebuah proses pencarian di sistem IR. Rumusnya: Jumlah dokumen relevan yang ditemukan / Jumlah semua dokumen relevan di dalam koleksi. Lalu, *precision* adalah proporsi jumlah dokumen yang ditemukan dan dianggap relevan untuk kebutuhan si pencari informasi. Rumusnya: Jumlah dokumen relevan yang ditemukan / Jumlah semua dokumen yang ditemukan. Kedua ukuran di atas biasanya diberi nilai dalam bentuk persentase, 1 sampai 100%. Sebuah sistem informasi akan dianggap baik jika tingkat *recall* maupun *precision*-nya tinggi.



Gambar Error! No text of specified style in document.-3 : Recall dan Precision

Pada gambar 2-3 Rd adalah dokumen yang relevan terhadap query, ketentuan dokumen yang relevan ada pada dataset yang telah ditentukan. Rs adalah dokumen yang relevan terhadap query yang dihasilkan oleh sistem. Rr adalah irisan antara Rd dan Rs.

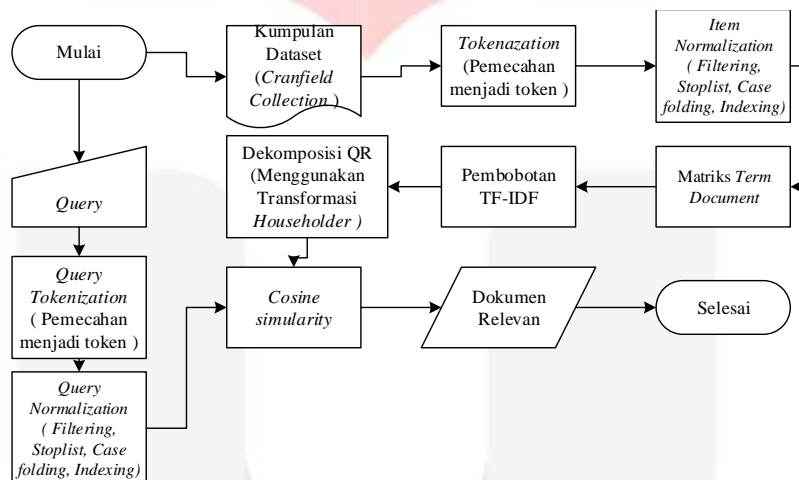
### 3. Perancangan Sistem

#### 3.1. Analisis Kebutuhan Sistem

Beberapa kebutuhan sistem yang ingin dicapai pada tugas akhir ini yaitu :

1. Pengujian untuk mengetahui perbandingan recall dan precision yang dihasilkan oleh QR Decomposition dengan transformasi Householder dalam menentukan dokumen yang relevan berdasarkan query yang diinput oleh pengguna
2. Pengujian untuk mengetahui kemampuan LSI dalam pencarian dokumen yang relevan walaupun tidak ada term pada query yang diproses melalui QR Decomposition dengan transformasi Householder.
3. Pengujian untuk mengetahui perbandingan waktu proses pencarian dokumen relevan terhadap query yang dihasilkan oleh QR Decomposition dengan transformasi Householder.

Gambar sistem yang akan di bangun dapat dilihat dari digaram berikut:



Gambar Error! No text of specified style in document.-4: Flowchart Sistem

### 4. Pembahasan

#### 4.1 Pengujian Indexing dan Pembentukan Matrix dokumen

Proses pada indexing dimulai dari user memasukkan dataset.txt ke dalam sistem. Dataset.txt adalah dokumen yang ada pada dataset. Term dari setiap dokumen akan dibuat menjadi satu kolom dan dilakukan tahap stoplist, setelah itu term yang sudah diproses akan disimpan menjadi Term.m. Ada 3537 term yang terbuat dari dokumen dalam dataset. Setelah melakukan Indexing setiap term akan dicek kembali pada dokumen, berfungsi untuk pembuatan matrix, bila terdapat term yang ada pada dokumen akan diberi nilai sesuai dengan jumlahnya. Pada tugas akhir ini didapat matriks 3537x250, itu disebabkan banyaknya term dan banyaknya dokumen yang dipakai terambil dari dataset.

#### 4.2 Hasil Recall dan Precision dengan QR Decomposition

Tabel Error! No text of specified style in document.-2 :Hasil Recall dan Precision dari QR decomposition

| Parameter/query | Recall | Precision |
|-----------------|--------|-----------|
| Query1          | 0,53   | 0,53      |
| Query2          | 0,62   | 0,26      |

|         |      |      |
|---------|------|------|
| Query3  | 0,8  | 0,21 |
| Query4  | 0,67 | 0,10 |
| Query5  | 0,8  | 0,21 |
| Query6  | 0,8  | 0,21 |
| Query7  | 0,67 | 0,21 |
| Query8  | 1    | 0,26 |
| Query9  | 0,4  | 0,10 |
| Query10 | 1    | 0,16 |

QR *decomposition* mampu mendekomposisikan term matriks dokumen menjadi 2 yaitu matriks Q dan matriks R. pada tabel nilai *recall* dan *precision* terkecil adalah *query* 9 nilai *recall* 40% dan *precision* 10%, pada sistem memunculkan 2 dokumen yang relevan dari 5 dokumen yang relevan yang ada pada dataset dan yang tertinggi adalah pada *query* 8 yang bernilai *recall* 100% dan *precision* 26%.

#### 4.3 Pengujian Parameter

Setelah melakukan terhadap 10 *query* maka didapat *recall* dan *precision* dari hasil QR. Maka dilakukan perbandingan *recall* dan *precision* yang dihasilkan QR. Nilai *recall* tertinggi mengindikasikan banyaknya dokumen yang diambil oleh sistem yang sesuai dengan dataset, sedangkan nilai *precision* yang tinggi mengindikasikan pencarian dokumen yang relevan terhadap *query*. *Cosine similarity* pada QR *decomposition* perhitungan vektor *query* dikalikan dengan matriks R dengan nilai elemen matriksnya tidak terbatas pada nilai tertentu. Pada tabel QR *decomposition* mampu memunculkan rata-rata dokumen yang relevan, sehingga nilai *recall* dan *precision* yang dihasilkan QR *decomposition* lebih baik.

#### 4.4 Pengujian Pencarian dengan Hasil Query

Pengujian ini dilakukan untuk melihat hasil pencarian yang dilakukan oleh sistem yang dibangun. Pengujian dilakukan membuktikan sistem pencarian dokumen yang relevan dari *query* yang diinputkan dan disesuaikan menurut yang ada di dataset:

Query 5 : jet interference with supersonic flow - dash experimental papers. Dokumen yang relevan menurut dataset : 187,174,173,176,177

Tabel Error! No text of specified style in document.-3 : Pembuktian LSI dengan QR

| Metode                                      | Dokumen id hasil pencarian                                               |
|---------------------------------------------|--------------------------------------------------------------------------|
| QR<br>Precision<br>:0,10<br>Recall<br>:0,66 | 176,173,131,245,177,174,243,179,188,46,121,216,21,218,124,103,41,172,137 |

Tabel Error! No text of specified style in document.-4 : Pembuktian kemampuan LSI

| Dokumen        | 174 | 177 | 176 | 173 | 187 |
|----------------|-----|-----|-----|-----|-----|
| 'DASH'         |     |     |     |     |     |
| 'EXPERIMENTAL' |     |     |     |     |     |
| 'FLOW'         | 1   | 1   | 1   | 3   | 4   |
| 'INTERFERENCE' |     |     |     |     |     |
| 'JET'          | 10  | 8   | 7   | 12  |     |



|              |   |   |   |   |   |
|--------------|---|---|---|---|---|
| 'PAPERS'     |   |   |   |   |   |
| 'SUPERSONIC' | 3 | 2 | 1 | 1 | 2 |

Dari tabel 4.3 daftar term pada query 5. pada dokumen 174 hanya ada term flow, jet, supersonic dan dash experimental interference, dan papers tidak ada tetapi mampu memberikan nilai relevansi terhadap query 5. Dari pengujian query 5 dapat disimpulkan bahwa hasil pencarian dokumen untuk menerapkan LSI untuk metode QR Decomposition terbukti mampu menemukan dokumen yang relevan walau tidak mengandung term dari query yang diinputkan akan tetap terambil.

#### 4.5 Pengujian Waktu Pencarian Dokmuen Relevan

Pada table 4.4 hasil waktu pencarian dokumen yang relevan terhadap *query* menggunakan QR.

*Tabel Error! No text of specified style in document.-5* :hasil waktu pencarian dokumen relevan

| Mertode/ <i>Query</i> | QR      |
|-----------------------|---------|
| <i>Query1</i>         | 5,79068 |
| <i>Query2</i>         | 5,80644 |
| <i>Query3</i>         | 5,81169 |
| <i>Query4</i>         | 5,82107 |
| <i>Query5</i>         | 5,81368 |
| <i>Query6</i>         | 5,80166 |
| <i>Query7</i>         | 5,79137 |
| <i>Query8</i>         | 5,80726 |
| <i>Query9</i>         | 5,79404 |
| <i>Query10</i>        | 5,84315 |

Pada tabel 4.4 dan terlihat perbandingan waktu proses pencarian dokumen relevan QR *decomposition*.

## 5. Kesimpulan dan Saran

### 5.1 Kesimpulan

QR Decomposition dengan menggunakan transformasi Householder dapat diimplementasikan pada Latent Semantic Indexing (LSI). Hal ini dibuktikan pada uji coba query yang dipakai, dan kemampuan LSI dalam pencarian dokumen yang relevan terhadap query yang diinputkan walaupun tidak ada term query yang ada dalam query. QR decomposition menggunakan transformasi Householder memiliki nilai recall dan precision lebih baik disebabkan pada perhitungan cosine similarity QR decomposition menggunakan transformasi Householder perhitungan vektor query dikalikan dengan matriks R dengan nilai elemen matriksnya tidak terbatas pada nilai tertentu yang menyebabkan pada QR decomposition menggunakan transformasi Householder mampu memunculkan banyak dokumen yang relevan sesuai dengan sistem. Dalam waktu pencarian dokumen yang relevan pada query QR Decomposition lebih cepat dibandingkan dengan metode QR lainnya.

### 5.2 Saran

Berdasarkan hasil pengujian dan analisis, ada pun saran untuk perbaikan pada Tugas Akhir ini untuk penelitian lebih lanjut, yaitu :

1. Dalam proses indexing pada suatu dokumen disarankan dilakukan pada dokumen yang dokumen.html
2. Untuk membuktikan LSI adalah pencarian informasi retrieval yang baik, sebaiknya dilakukan kombinasi atau penggunaan 2 atau lebih algoritma, sehingga didapatkan pencarian yang optimal.
3. Dalam proses Indexing pada dokumen menggunakan tahap Stemming

**Daftar Pustaka:**

1. Baeza-Yates, R. and Ribiero-Neto, B., 1999, *Modern Information Retrieval*, Pearson Education.
2. Berry, M.W., Browne, Murray. 2005. *Understanding Search Engines, Mathematical Modeling and Text Retrieval, second edition*. SIAM, Society for Industrial and Applied Mathematics.
3. CH. Aswanikumar, A. Gupta, S. Trehan, M. Batool, *Latent Semantic Indexing-based Intelligent Information Retrieval System for Digital Libraries. Journal of Computing and Information Technology*,14, 191–196, 2006.
4. Ruminta. (2014) *Matriks Persamaan Linier dan Pemrograman Linier*. Penerbit Rekayasa Sains, Bandung.
5. Chu W. Liu Z, Mao W. (2002) *Textual Document Indexing and Retrieval via Knowledge Sources and Data Mining*.
6. Cios, Krzysztof J. Etc. (2007) *Data Mining A Knowledge Discovery Approach*, Springer
7. Deerwester, Scott et al, 1990, *Indexing by Latent Semantic Indexing*, Journal of The American Society for Information Science.
8. Gander Walter. 1980. *Algorithms for the QR-Decompositions*. Research Report No. 80-021. CH-8092 Zuerich
9. Golub, Gene H. and Van Loan, Charles F., 1996, *Matrix Computations*, Baltimore and London, John Hopkins University Press.
10. L. XIAO, J. SUN, S. BOYD, A Duality View of Spectral Methods for Dimensionality Reduction. *Proceedings of 23rd International Conference on Machine Learning*, 1041–1048, 2006.
11. Manning, Christopher D., Ragnavan Prabhakar., Schutze, Hinrich. 2009. *Introduction to Information Retrieval*. Cambridge University Press.
12. Rosario, Barbara., 2000, *Latent Semantic Indexing: An overview*. Infosys 240 spring 2000, Final Paper.
13. Sudaraman, Fatwa N., (2014), *Analisa Latent Semantic Indexing dengan Menggunakan QR Decomposition dan Singular Value Decomposition untuk Temu Kembali Informasi*, Tugas Akhir. Universitas Telkom.