

PENCARIAN *FULL TEXT* PADA KOLEKSI SKRIPSI FAKULTAS TEKNIK UHAMKA MENGGUNAKAN METODE *VECTOR SPACEMODEL*

Miftahul Ari Kusuma^{1*}, Mia Kamayani², Arry Avorizano³

Program Studi Teknik Informatika, Fakultas Teknik
Universitas Muhammadiyah Prof. Dr. Hamka, Jakarta.
Jl. Tanah Merdeka No. 6 Kampung Rambutan, Jakarta Timur
E-mail: miftahulari25@gmail.com¹, mia.kamayani@uhamka.ac.id², arry.avorizano@uhamka.ac.id³

Abstrak – Memiliki banyak dokumen terkadang sangat merepotkan, terutama disaat ingin mencari dokumen yang dibutuhkan dengan data yang tepat sesuai dengan keinginan. Banyaknya kumpulan skripsi di fakultas teknik UHAMKA membuat mahasiswa kesulitan dalam mencari skripsi yang relevan atau tepat sesuai dengan keinginan yang dicari, sebenarnya sudah ada mesin pencarian pada Perpustakaan Fakultas Teknik UHAMKA tetapi hanya mencari berdasarkan judul, dan pengarangnya saja. Maka, dibuatlah pengembangan mesin pencarian yaitu dengan membuat mesin pencarian *fulltext*, agar mahasiswa dapat mencari skripsi sesuai yang dibutuhkan dari mesin pencarian. Salah satu metode untuk membuat mesin pencarian adalah dengan metode *vector space model (VSM)*. *Vector Space Model (VSM)* adalah metode untuk melihat tingkat kedekatan atau kesamaan (*similarity*) term dengan cara pembobotan term. Perancangan mesin pencarian *full text* ini menggunakan *PHP* dan ditampilkan berupa halaman *web*. Pengujian relevansi mesin pencarian dilakukan dengan memasukan beberapa *query* dan mesin pencarian akan menampilkan dokumen relevan, dan relevan itu diuji dengan cara menghitung *precision*, *recall* dan *f-measure* dari masing-masing *query* yang ditampilkan di mesin pencarian dengan nilai rata-rata *precision* 0.63, *recall* 0.92 dan *f-measure* 0.71.

Kata kunci: *Full Text, Mesin Pencarian, Vector Space Model, Kumpulan Skripsi*

1 Pendahuluan

Memiliki banyak dokumen terkadang sangat merepotkan, terutama disaat ingin mencari dokumen yang dibutuhkan dengan data yang tepat sesuai dengan keinginan. Banyaknya kumpulan skripsi di Fakultas Teknik UHAMKA membuat mahasiswa kesulitan dalam mencari skripsi yang relevan atau tepat sesuai dengan keinginan yang dicari, maka dari itu, dokumen yang berbentuk fisik harus didigitalisasi agar dapat menghemat ruang dan tempat. Dokumen yang digitalisasi pun masih ada masalah yaitu sulit untuk mencari dokumen sesuai yang diinginkan dari banyaknya kumpulan dokumen yang ada, maka dari itu harus adanya mesin pencarian untuk mencari dokumen.

Perpustakaan Fakultas Teknik UHAMKA sebenarnya sudah mempunyai mesin pencarian, akan tetapi mesin pencarian di perpustakaan masih mencari berdasarkan judul dan pengarangnya saja. Hal ini membuat mahasiswa masih kesulitan dalam mencari skripsi yang relevan atau sesuai

dengan keinginan yang dicari dikarenakan mesin pencarian perpustakaan masih belum *fulltext*.

Maka, dibuatlah pengembangan mesin pencarian dengan membuat mesin pencarian *fulltext*, agar mahasiswa dapat mencari skripsi sesuai yang dibutuhkan dari mesin pencarian. Mesin pencarian pun harus dapat menampilkan kumpulan skripsi yang relevan, diurutkan dari relevansi yang tertinggi ke terendah. Metode dalam rancang bangun mesin pencarian ada beberapa metode yaitu *vector space model*, *boolean model*, *statistic language model*.

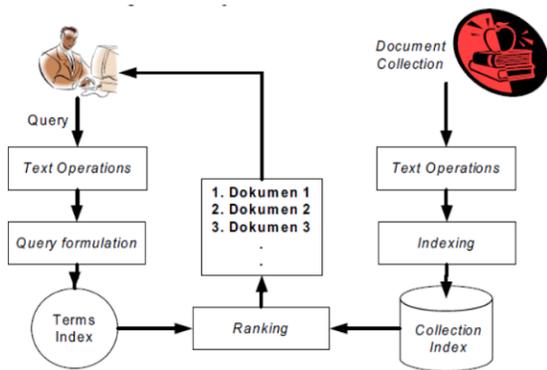
Mesin pencarian *full text* ini dibuat dengan metode *Vector Space Model (VSM)* karena metode ini tidak terlalu sulit, banyaknya pengembangan mesin pencarian dengan metode ini sehingga banyaknya referensi yang didapat dan nilai ketepatannya tinggi. *VSM* menggunakan konsep yang terdapat pada aljabar linear yaitu ruang *vector*. Hasil dari dokumen yang dicari akan diurutkan dari relevansi tertinggi ke relevansi terendah melalui kemiripan *query* yang diinput.

Berdasarkan masalah diatas, maka penulis mengajukan penelitian dan pembahasan dalam bentuk tugas akhir dengan judul “Pencarian *Full Text* Pada Koleksi Skripsi Fakultas Teknik UHAMKA Menggunakan Metode *Vector Space Model*”.

2 Landasan Teori

2.1. Information Retrieval

Information Retrieval (IR) adalah studi untuk membantu pengguna menemukan informasi yang sesuai dengan kebutuhan informasinya. Secara teknis, IR mempelajari akuisisi, organisasi, penyimpanan, pengambilan dan pendistribusian informasi. Secara historis, IR adalah tentang pengambilan dokumen, menegaskan bahwa dokumen itu adalah sebagai unit dasar [9]. Tujuan IR adalah untuk memenuhi kebutuhan informasi pengguna dengan *retrieve* semua dokumen yang relevan, dan pada waktu yang sama *re-retrieve* sesedikit dokumen yang tak relevan [5]. Gambaran bagian-bagian yang terdapat pada suatu sistem IR digambarkan pada Gambar .1.



Gambar 1 Bagian-Bagian Information System Retrieval [3]

2.2. Preprocessing Text

Sebelum koleksi dokumen digunakan untuk *retrieval*, beberapa tugas preprocessing biasanya dilakukan. Untuk dokumen teks tradisional (tidak ada tag HTML), tugasnya adalah penghentian kata kunci (stopword removal), stemming, dan penanganan digit, tanda hubung, tanda baca, dan kasus huruf [9].

2.3. Stopword Removal

Proses *stopword removal* merupakan proses penghapusan *term* yang tidak memiliki arti atau tidak relevan. Proses ini dilakukan pada saat proses tokenisasi. Proses *Filtering* menggunakan daftar *stopword* yang digunakan merupakan mekanisme untuk pengindeksan kata dari koleksi teks yang digunakan untuk mempercepat proses pencarian. Elemen penting dalam struktur *inverted file* ada dua, yaitu: kata (*vocabulary*) dan kemunculan (*occurrences*) [2].

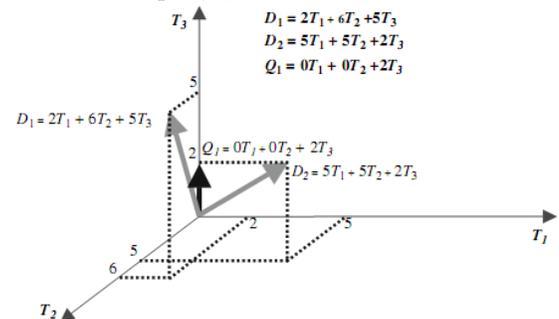
2.3. Stemming

Proses *Stemming* digunakan untuk mengubah *term* yang masih melekat dalam term tersebut awalan, sisipan, dan akhiran. Proses stemming dilakukan dengan cara menghilangkan semua imbuhan (*affixes*) baik yang terdiri dari awalan (*prefixes*), sisipan (*infixes*), akhiran (*suffixes*) dan *confixes* (kombinasi dari awalan dan akhiran) pada kata turunan. *Stemming* digunakan untuk mengganti bentuk dari suatu kata menjadi kata dasar dari kata tersebut yang sesuai dengan struktur morfologi bahasa Indonesia yang benar [2].

2.4. Vector Space Model

Vector Space Model (VSM) mungkin model ini paling banyak orang yang mengetahui dan menggunakannya. Sebuah dokumen dalam model ruang vektor direpresentasikan sebagai bobot vektor, di mana setiap komponen bobot dihitung berdasarkan beberapa variasi skema *TF* atau *TF-IDF* [9].

Vector Space Model (VSM) adalah metode untuk melihat tingkat kedekatan atau kesamaan (*similarity*) term dengan cara pembobotan term. Dokumen dipandang sebagai sebuah vektor yang memiliki *magnitude* (jarak) dan *direction* (arah). Pada *Vector Space Model*, sebuah istilah direpresentasikan dengan sebuah dimensi dari ruang vektor. Relevansi sebuah dokumen ke sebuah *query* didasarkan pada similaritas diantara vektor dokumen dan vektor *query* [1].



Gambar 2. VSM dengan dua dokumen D1 dan D2, dan query Q1 [4].

Selain itu pada VSM, database dari semua dokumen direpresentasikan oleh matriks *term document* (atau matriks *term-frequency*). Dimana setiap sel pada matriks berkorespondensi dengan bobot yang diberikan dari suatu *term* dalam dokumen yang ditentukan. [5]. Bobot lokal suatu *term* i di dalam dokumen j (*tfij*) dapat didefinisikan sebagai [4]

$$tf_{ij} = \frac{f_{ij}}{\max \{f_1, f_2, \dots, f_{|V|}\}} \quad (1)$$

Dimana *fij* adalah jumlah berapa kali term i muncul di dalam dokumen j. Frekuensi tersebut dinormalisasi dengan frekuensi dari *most common term* di dalam dokumen tersebut. Bobot global dari suatu term i pada pendekatan inverse document frequency (*idfi*) dapat didefinisikan sebagai [4].

$$idf_i = \log\left(\frac{n}{df_i}\right) \tag{2}$$

Dimana df_i adalah frekuensi dokumen dari term i dan sama dengan jumlah dokumen yang mengandung term i . \log_2 digunakan untuk memperkecil pengaruhnya relative terhadap tf_{ij} . Bobot dari term i di dalam sistem IR (w_{ij}) dihitung menggunakan ukuran $tf-idf$ yang didefinisikan sebagai berikut [4]

$$w_{ij} = tf_{ij} \times idf_i \tag{3}$$

Bobot tertinggi diberikan kepada term yang muncul sering kali dalam dokumen j tetapi jarang dalam dokumen lain. Salah satu ukuran kemiripan teks yang populer adalah *cosine similarity*. Ukuran ini menghitung nilai cosinus sudut antara dua vektor. Jika terdapat dua vektor dokumen d_j dan query q , serta t term diekstrak dari koleksi dokumen maka nilai cosinus antara d_j dan q didefinisikan sebagai [4]

$$similarity(d_j, q) = \frac{d_j \cdot q}{|d_j| \cdot |q|} \tag{4}$$

2.5. Evaluasi Mesin Pencarian

Recall dan *precision* mengukur kemampuan sistem dalam menemukembalikan dokumen-dokumen yang relevan dan menahan dokumen-dokumen yang tidak relevan. *Recall* merupakan rasio jumlah dokumen relevan yang ditemukembalikan terhadap jumlah seluruh dokumen relevan di dalam koleksi. *Precision* merupakan rasio jumlah dokumen relevan yang ditemukembalikan terhadap jumlah seluruh dokumen yang ditemukembalikan [10].

Precision

$$= \frac{\text{Jumlah dokumen relevan yang terpenggil}(a)}{\text{Jumlah dokumen yang terpenggil dalam pencarian}(a + b)} \times 100\% \tag{5}$$

Keterangan : (a) merupakan banyaknya dokumen relevan yang ditemu kembalikan dan (b) adalah jumlah semua dokumen dari hasil pencarian. sedangkan pada rumus *recall* yaitu sebagai berikut :

Recall

$$= \frac{\text{Jumlah dokumen relevan yang terpenggil}(a)}{\text{Jumlah dokumen relevan yang ada didalam database}(a + c)} \times 100\% \tag{6}$$

Keterangan (c) merupakan banyaknya dokumen relevan yang terdapat pada data *corpus*. Sedangkan pada (d) adalah jumlah dokumen yang tidak relevan yang tidak ditemukan pada *corpus* dan sistem [7]. Dokumen yang relevan merupakan dokumen yang benar dan yang sesuai hasil pencarian.

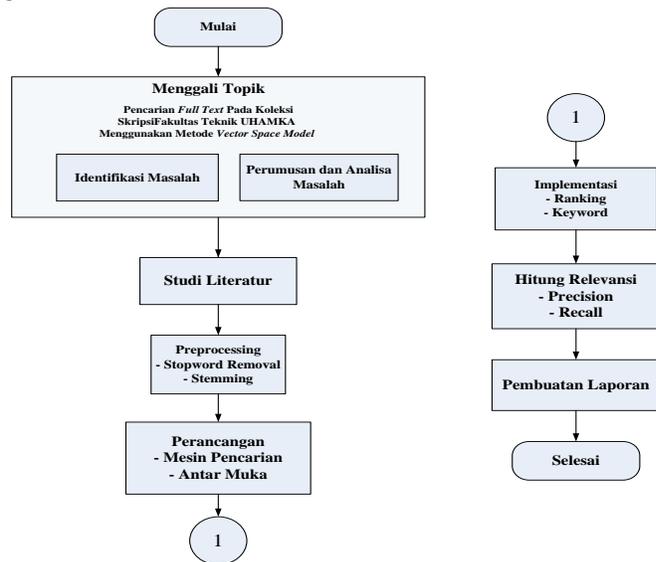
Selain penilaian *precision* dan *recall* yang dilakukan pada mesin pencarian ini maka juga dilakukan penilaian rata-rata

F-Measure. *F-Measure* adalah harmonic mean atau nilai rata-rata dari *precision* dan *recall* [8]. Rumus *F-Measure* adalah sebagai berikut :

$$F_1 = 2 * \frac{Precision * recall}{Precision + recall} \tag{7}$$

3. Metodologi Penelitian

Mulai dari tahapan pertama yaitu menggali topik penelitian, kemudian tahapan kedua yaitu mengumpulkan data dari berbagai jurnal dan buku, tahapan ketiga membuat daftar stopword dan stemming, selanjtunya tahapan keempat masuk tahapan perancangan mesin pencarian lalu masuk ke tahapan implementasi, terakhir, tahapan hitung relevansi yaitu tahapan untuk menilai seberapa relevan hasil pencarian dari mesin pencarian ini Adapun desain penelitian dapat dilihat pada gambar 3.



Gambar 3.1 : Diagram Alur Metodologi

4. Implementasi

Pada implementasi ini akan menampilkan dokumen hasil pencarian dari 10 *query* yang berbeda. Untuk setiap *query* dilakukan pencarian dokumen dengan mesin pencarian menggunakan metode *vector space model*. Tabel 1 menunjukkan daftar *query* yang digunakan dalam implementasi ini.

Tabel 1 Daftar Query

Query ID	Detail Query
Q1	Aplikasi bisnis
Q2	Aplikasi mobile
Q3	Teori UML
Q4	Penelitian di UHAMKA
Q5	Sistem akademik
Q6	Jaringan vpn

Q7	Aplikasi pembelajaran
Q8	Animasi
Q9	Jaringan
Q10	Server

Pada tabel 2 sampai 11 akan menunjukkan dokumen hasil pencarian dengan 10 *query* diatas:

Tabel 2 Hasil Pencarian Dokumen Menggunakan *Query* "aplikasi bisnis"

Rank	Docid	Sim
1	D2	0.0487949
2	D3	0.0326845
3	D9	0.0160074
4	D23	0.0129797
5	D38	0.00855016
6	D34	0.00704329
7	D11	0.00704116
8	D15	0.00602972

Tabel 3 Hasil Pencarian Dokumen Menggunakan *Query* "aplikasi mobile"

Rank	Docid	Sim
1	D7	0.113708
2	D13	0.0827463
3	D11	0.0786736
4	D35	0.0403226
5	D21	0.0330631
6	D23	0.0271925
7	D34	0.0271925
8	D8	0.0217154
9	D9	0.0167678
10	D37	0.00576032
11	D32	0.00558572
12	D14	0.00513806
13	D36	0.00474719

Tabel 4 Hasil Pencarian Dokumen Menggunakan *Query* "teori UML"

Rank	Docid	Sim
1	D7	0.0651898
2	D3	0.0457996
3	D12	0.0293397
4	D1	0.0200344
5	D36	0.0190512
6	D5	0.0168543
7	D8	0.0165995
8	D9	0.0160219
9	D15	0.0144844
10	D37	0.0132097
11	D40	0.0125516
12	D11	0.011276
13	D38	0.0102695
14	D16	0.00967101
15	D6	0.00954145
16	D27	0.0088778
17	D34	0.00845957
18	D2	0.0048839
19	D21	0.00473882
20	D13	0.00431265
21	D39	0.00111638

Tabel 5 Hasil Pencarian Dokumen Menggunakan *Query* "penelitian di UHAMKA"

Rank	Docid	Sim
1	D23	0.167701
2	D36	0.0878301
3	D15	0.0701152
4	D5	0.0679893
5	D22	0.0365621

Tabel 6 Hasil Pencarian Dokumen Menggunakan Query "sistem akademik"

Rank	Docid	Sim
1	D12	0.125813
2	D13	0.0154111
3	D11	0.0150571
4	D5	0.0100736

Tabel 7 Hasil Pencarian Dokumen Menggunakan Query "jaringan vpn"

Rank	Docid	Sim
1	D24	0.575608
2	D29	0.392728
3	D33	0.068316
4	D25	0.0409165
5	D32	0.0380504
6	D14	0.0335723
7	D4	0.0288067
8	D17	0.021112
9	D30	0.0205302
10	D23	0.0189018
11	D28	0.0104288
12	D26	0.0100476
13	D21	0.00574563
14	D12	0.00284584
15	D40	0.00152183
16	D22	0.00109892
17	D10	0.00106517
18	D35	0.000989248
19	D37	0.000800811
20	D9	0.000777032

Tabel 8 Hasil Pencarian Dokumen Menggunakan Query "aplikasi pembelajaran"

Rank	Docid	Sim
1	D7	0.317405
2	D13	0.198315

3	D1	0.162578
4	D20	0.146231
5	D16	0.0418558
6	D11	0.022876
7	D12	0.0158726

Tabel 9 Hasil Pencarian Dokumen Menggunakan Query "animasi"

Rank	Docid	Sim
1	D20	0.48587
2	D19	0.334483
3	D16	0.0512522
4	D22	0.0200139
5	D12	0.0172764
6	D38	0.0100785

Tabel 10 Hasil Pencarian Dokumen Menggunakan Query "jaringan"

Rank	Docid	Sim
1	D33	0.303057
2	D25	0.18151
3	D32	0.168796
4	D24	0.151454
5	D14	0.14893
6	D4	0.12779
7	D17	0.0936549
8	D30	0.091074
9	D23	0.0838504
10	D29	0.0513756
11	D28	0.0462631
12	D26	0.0445722
13	D21	0.0254883
14	D12	0.0126244
15	D40	0.006751
16	D22	0.00487493
17	D10	0.00472522
18	D35	0.00438841
19	D37	0.00355249
20	D9	0.003447

Tabel 11 Hasil Pencarian Dokumen Menggunakan Query "server"

Rank	Docid	Sim
1	D14	0.212205
2	D4	0.0806588
3	D12	0.0377818
4	D9	0.0320568
5	D6	0.0287554

5 Hitung Relevansi

Hasil implementasi mesin dari pencarian dengan memasukan 10 query sudah didapatkan proses selanjutnya adalah pengujian relevansi, dengan cara menghitung precision dan recall dari masing-masing query. Adapun rumus precision dan recall telah dijelaskan di Bab II maka menghasilkan penilaian dari kedua tersebut yaitu, jumlah dokumen yang ter-retrieve maupun dokumen non-relevan dan dokumen non-relevan tidak ter-retrieve maka didapatkan penilaian precision dan recall serta nilai tengah dari kedua evaluasi tersebut (f-measure) yaitu, pada tabel 12 :

Tabel 12 Hasil Perhitungan Precision dan Recall Berdasarkan Keseluruhan Isi Dokumen

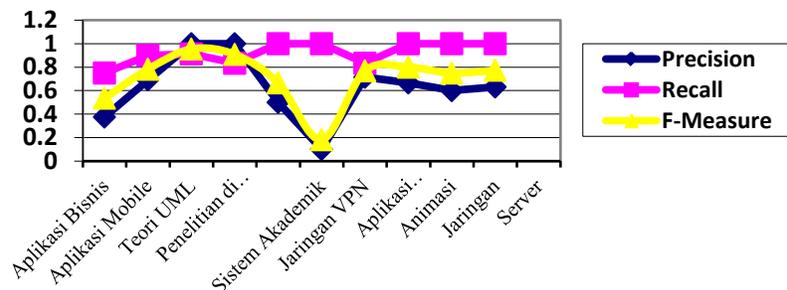
Nilai	Query									
	Aplikasi bisnis	Aplikasi mobile	Teori UML	Penelitian di UHAMKA	Sistem akademik	Jaringan VPN	Aplikasi pembelajaran	Animasi	Jaringan	Server
Precision	0.375	0.692	1.0	1.0	0.5	0.1	0.714	0.666	0.6	0.6315
Recall	0.75	0.9	0.913	0.8333	1.0	1.0	0.8333	1.0	1.0	1.0
F-Measure	0.5357	0.7824	0.9545	0.9091	0.6666	0.1818	0.7689	0.7995	0.75	0.7741

Pada tabel 12 tersebut menunjukkan hasil perhitungan precision dan recall berdasarkan keseluruhan isi dokumen terhadap 10 kriteria query tersebut. Hasil pada tabel tersebut dapat disimpulkan bahwa akan dihitung nilai precision dan recall pada setiap nilai recall standar yaitu, 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, dan 1.0. Adapun nilai precision dan recall dan F-Measure adalah sebagai berikut :

1. Nilai precision yang didapatkan pada query "aplikasi bisnis" yaitu, 0.375 atau 37.5%, pada query "aplikasi mobile" yaitu, 0.692 atau 69.2%, pada query "teori UML" yaitu, 1.0 atau 100%, pada query "penelitian di uhamka" yaitu, 1.0 atau 100%, pada query "sistem akademik" yaitu, 0.5 atau 50%, pada query "jaringan vpn yaitu, 0.1 atau 10%, pada query "aplikasi pembelajaran" yaitu, 0.714 atau 71.4%, pada

query "animasi" yaitu, 0.666 atau 66.6%, pada query "jaringan" yaitu, 0.6 atau 60%, pada query "server" yaitu, 0.6135 atau 61.35%.

2. Sedangkan nilai recall yang didapatkan pada query "aplikasi bisnis" yaitu, 0.75 atau 75%, pada query "aplikasi mobile" yaitu, 0.9 atau 90%, pada query "teori UML" yaitu, 0.913 atau 91.3%, pada query "penelitian di uhamka" yaitu, 0.8333 atau 83.33%, pada query "sistem akademik" yaitu, 1.0 atau 100%, pada query "jaringan vpn yaitu, 1.0 atau 100%, pada query "aplikasi pembelajaran" yaitu, 0.8333 atau 83.33%, pada query "animasi" yaitu, 1.0 atau 100%, pada query "jaringan" yaitu, 1.0 atau 100%, pada query "server" yaitu, 1.0 atau 100%.
3. Pada nilai rata-rata (F-Measure) dari precision dan recall yaitu, didapatkan pada query "aplikasi bisnis" yaitu, 0.5357 atau 53.57%, pada query "aplikasi mobile" yaitu, 0.7824 atau 78.24%, pada query "teori UML" yaitu, 0.9545 atau 95.45%, pada query "penelitian di uhamka" yaitu, 0.9091 atau 90.91%, pada query "sistem akademik" yaitu, 0.6666 atau 66.66%, pada query "jaringan vpn yaitu, 0.1818 atau 18.18%, pada query "aplikasi pembelajaran" yaitu, 0.7689 atau 76.89%, pada query "animasi" yaitu, 0.7995 atau 79.95%, pada query "jaringan" yaitu, 0.75 atau 75%, pada query "server" yaitu, 0.7741 atau 77.41%.



Grafik 1 Hasil Perhitungan Relevansi Berdasarkan Keseluruhan Isi Dokumen

Pada hasil tersebut dapat disimpulkan bahwa nilai recall (tingkat keberhasilan sistem menemukan informasi) lebih

tinggi daripada nilai *precision* (tingkat ketepatan informasi yang ditemukan). Setelah didapatkan nilai *precision* dan *recall* pada setiap *query* tersebut, kemudian dicari nilai maksimum *precision* dan *recall*. Seperti gambar di bawah

Hasil grafik 1 di atas menjelaskan bahwa hasil dari nilai maksimum pada *precision* dan *recall* dengan *query* tersebut yaitu, dengan tingkat ketepatan akurat dokumen yang dipanggil sistem “teori UML” dan “penelitian di UHAMKA” ini dikarenakan satu kata kunci UML dan UHAMKA menjadi ciri khas dari masing-masing dokumen yang ter-retrieve. Sedangkan untuk nilai maksimum *recall* atau tingkat keberhasilan sistem dalam menemukan informasi dari (*information retrieval*) adalah *query* “sistem akademik”, “jaringan vpn”, “animasi”, “jaringan”, “server”, dalam 4 *query* tersebut dokumen yang ter-retrieve dari database tidak ada yang *miss* atau *terlewat*. Dari kedua hasil tersebut dapat disimpulkan bahwa ketepatan dalam menemukan dokumen relevan yang dilakukan mesin pencarian ini yaitu, tergantung *query* yang diinput apakah *query* tersebut mewakili ciri khas dari dokumen tersebut atau tidak, jika *query* yang diinput itu tidak mewakili ciri khas dari suatu dokumen maka nilai ketepatannya akan menurun.

6 Simpulan dan Saran

6.1. Kesimpulan

1. Mesin pencarian *full text* ini selesai dirancang dan diuji dengan baik berdasarkan pengujian dengan 10 *query* berbeda yang diinput ke dalam mesin pencarian dan mesin pencarian menampilkan masing-masing hasil dari 10 *query* tersebut.
2. Mesin pencarian *full text* ini mampu menampilkan dokumen dengan relevan berdasarkan dari 10 *query* yang diuji coba dan menampilkan dokumen sudah terurut. Nilai relevansi dari mesin pencarian yang sudah cukup baik, berdasarkan nilai rata-rata *precision* 0.63, *recall* 0.92 dan *f-measure* 0.71, dengan hasil tersebut membuat mesin pencarian dapat dijadikan alternatif mesin pencarian di perpustakaan Fakultas Teknik UHAMKA.

6.2. Saran

1. Mesin pencarian *full text* ini terbilang masih sederhana masih butuh tambahan fitur lagi misalnya,

fitur untuk mendownload dan untuk membaca dokumen yang diinput.

2. Untuk penyimpanan dokumen lebih baik tidak disimpan di database *mysql* dikarenakan pada proses indexing dan hitung bobot prosesnya cukup lama. Hal ini membuat dokumen yang terinput tidak bisa terlalu banyak.

Kepustakaan

- [1] F. Amin, “Implementasi Search Engine (Mesin Pencari) Menggunakan Metode Vector Space Model,” *1127-1627-1-PB*, P. 14, 2011.
- [2] F. Amin, “Sistem Temu Kembali Informasi Dengan Peningkatan Metode Vector Space Model,” *1700-1512-1-SM*, P. 8, 2013.
- [3] I. P. A. Waranu, “Implementasi Metode Generalized Vector Space Model Pada Information Retrieval System,” *Makalah-Siki-1204505042*, P. 21, 2015.
- [4] F. S. U. Giat Karyono1, “Temu Balik Informasi Pada Dokumen Teks Berbahasa,” *2012072020200380*, P. 8, 2012.
- [5] A. A. Abdillah, “Implementasi Vector Space Model Untuk Pencarian Dokumen,” P. 7.
- [6] W. Hardi, “Mengukur Kinerja Search Engine: Sebuah Eksperimentasi Penilaian Precision And Recall Untuk Informasi Ilmiah Bidang Ilmu Perpustakaan Dan Informasi,” *Search_Engine_Article*, P. 7, 2006.
- [7] L. Noviandi, “Search Engine Pada Dokumen RDF Tanaman Obat Menggunakan Sesame Dan Lucene,” *Seminar_Ekstensi*, P. 7, 2015.
- [8] D. T. F. Abidin, “Accuracy Measure Precision, Recall & F-Measure,” 2015.
- [9] B. Liu, *Web Data Mining 2nd Edition Exploring Hyperlinks Contents And Usage Data*, 2012.
- [10] F. F. Paiki, “Evaluasi Penggunaan Similarity Thesaurus Terhadap Ekspansi Kueri Dalam Sistem Temu Kembali Informasi Berbahasa Indonesia,” *Fridolin Paiki - Paper Evaluasi Penggunaan Similarity Thesaurus Terhadap Ekspansi Kueri*, P. 10, 2012.
- [11] F. A. Pratama, “Aplikasi Vector Space Model Dalam Pencarian Arsip Karya Tulis Mahasiswa ITB,” *Makalah-IF2123-2015-065*, P. 6, 2016.