

SKRIPSI

**PENERAPAN ALGORITMA *LATENT SEMANTIC INDEXING*
PADA SISTEM *SEARCH ENGINE* DATA CITRA SATELIT
LAPAN**

Disusun Oleh :

Dimas Priyandi

0651 15 144



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PAKUAN**

BOGOR

2019

HALAMAN PENGESAHAN

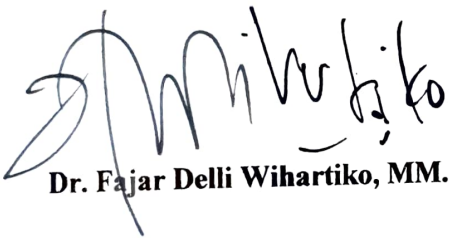
**JUDUL : PENERAPAN ALGORITMA LATENT SEMANTIC INDEXING
PADA SISTEM SEARCH ENGINE DATA CITRA SATELIT
LAPAN**

NAMA : DIMAS PRIYANDI

NPM : 065115144

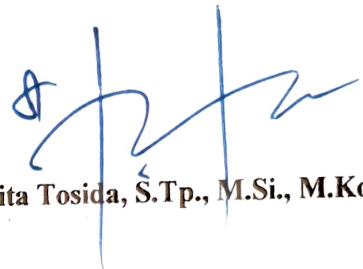
Mengesahkan,

Pembimbing Pendamping
Program Studi Ilmu Komputer
FMIPA-UNPAK



Dr. Fajar Delli Wihartiko, MM., M.Kom.

Pembimbing Utama
Program Studi Ilmu Komputer
FMIPA-UNPAK



Eneng Tita Tosida, S.Tp., M.Si., M.Kom.

Mengetahui,

Ketua Jurusan Ilmu Komputer
FMIPA – UNPAK



Dr. Prihastuti Harsani, M.Si.

Dekan
FMIPA – UNPAK



Dr. Prasetyorini, M.Si.

PERNYATAAN KEASLIAN KARYA TULIS SKRIPSI

Dengan ini saya menyatakan bahwa:

Sejauh yang saya ketahui, karya tulis ini bukan merupakan karya tulis yang pernah di publikasikan atau sudah pernah dipakai untuk mendapatkan gelar sarjana di Universitas lain, kecuali pada bagian-bagian di mana sumber informasinya dicantumkan dengan cara referensi yang semestinya.

Demikian pernyataan ini saya buat dengan sebenar-benarnya. Apabila kelak dikemudian hari terdapat gugatan, penulis bersedia dikenakan sanksi sesuai dengan peraturan yang berlaku.

Bogor, 5 Agustus 2019

Dimas Priyandi

0651 15 144

PERNYATAAN PELIMPAHAN SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA

Saya yang bertandatangan di bawah ini :

Nama : **Dimas Priyandi**
NPM : **0651151144**
Judul Skripsi : **PENERAPAN ALGORITMA *LATENT SEMANTIC INDEXING* PADA SISTEM *SEARCH ENGINE* DATA CITRA SATELIT LAPAN**

Dengan ini saya menyatakan bahwa Paten dan Hak Cipta dari produk Skripsi dan Tugas Akhir di atas adalah benar karya saya dengan arahan dari komisi pembimbing dan belum diajukan dalam bentuk apapun kepada perguruan tinggi manapun.

Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka di bagian akhir skripsi ini.

Dengan ini saya melimpahkan Paten, Hak Cipta dari karya tulis saya kepada Universitas Pakuan.

Bogor, 5 Agustus 2019

Dimas Priyandi
0651 15 144

RIWAYAT HIDUP



Dimas Priyandi tinggal di Kota Bogor. Lahir di Bogor pada tanggal 28 Oktober 1997 sebagai anak pertama dari dua bersaudara. Penulis memulai pendidikan di Sekolah Dasar yang bertempat di SDN 1 Pasireurih 01. Kemudian di tahun 2010, masuk SMPN 9 Bogor dan penulis adalah Alumni dari SMAN 4 Bogor, berlokasi di Kota Bogor. Pada tahun 2015 penulis meneruskan pendidikan ke Universitas Pakuan Bogor, Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam. Selama di Universitas Pakuan, penulis pernah aktif di Himpunan Mahasiswa Komputer (HIMAKOM) sebagai anggota dan sebagai Asisten Praktikum Labkom angkatan 2015. Pada bulan Agustus tahun 2019 penulis menyelesaikan penelitian dengan judul “**Penerapan Algoritma Latent Semantic Indexing Pada Sistem Search Engine Data Citra Satelit Lapan**”.

RINGKASAN

Dimas Priyandi. 2019. Penerapan Algoritma *Latent Semantic Indexing* Pada Sistem *Search Engine* Data Citra Satelit Lapan. Dibawah bimbingan Eneng Tita Tosida, S. Tp., M.Si., M.Kom dan Dr. Fajar Delli Wihartiko, MM., M.Kom.

Lembaga Penerbangan dan Antariksa Nasional adalah Lembaga Pemerintah Non Kementerian Indonesia yang melaksanakan tugas pemerintahan di bidang penelitian dan pengembangan kedirgantaraan dan pemanfaatannya serta penyelenggaraan keantariksaan sesuai dengan peraturan perundang-undangan. Salah satu fungsi yang diselenggarakan oleh LAPAN yaitu pelaksanaan penelitian dan pengembangan sains antariksa dan atmosfer, teknologi penerbangan dan antariksa, dan penginderaan jauh serta pemanfaatannya.

Maka dibuat penelitian dalam pencarian metadata citra satelit dapat dilakukan berdasarkan judul serta deskripsi sehingga menemukan dokumen citra yang relevan maka solusi untuk pencarian data citra satelit tersebut peneliti membuat “Penerapan Algoritma Latent Semantic Indexing Pada Sistem Search Engine Data Citra Satelit”. Dengan menerapkan algoritma Latent Semantic Indexing diharapkan pencarian data citra satelit akan semakin akurat dan optimal.

Berdasarkan hasil uji coba *precision* dan *recall* pada setiap *term* yang sudah dilakukan dapat terlihat nilai *recall* selalu 100 % dikarenakan dokumen yang relevan selalu terambil pada setiap *term*, sedangkan nilai rata-rata *precision* pada setiap *term* berbeda-beda. Pada 1 *term* didapatkan rata-rata *precision* 95% , 2 *term* didapatkan *precision* 53.65%, dan 3 sampai 5 *term* didapatkan *precision* sebesar 33.75%. Menurut Kartarina & Junaedi H (2017) Terkadang sebuah sistem memiliki nilai *recall* tinggi tetapi memiliki nilai *precision* rendah dan sebaliknya hal ini dikarenakan pengambilan data relevan oleh sistem tidak seimbang dengan yang manual atau sebaliknya. Maka sebaiknya pengukuran yang dilakukan tidak hanya mengambil salah satu nilai *precision* dan *recall* saja. Nilai *precision* dan *recall* dapat digunakan bersama-sama yang digabungkan dalam *F-Measure* (Winata F & Rainarli E, 2016). Adapun nilai *f-measure* dari pengujian 1 *term* yaitu 97.4%, pengujian 2 *term* yaitu 69.83%, sedangkan pada pengujian 3 sampai 5 *term* didapatkan nilai sebesar 50.46 %.

KATA PENGANTAR

Segala puji bagi Allah SWT. Tuhan semesta alam yang terus menerus melimpahkan rahmat serta memberikan kekuatan dan keimanan kepada kita semua selaku makhluk ciptaan-Nya. Shalawat serta salam semoga tercurah limpahkan kepada Nabi akhir zaman baginda Nabi Muhammad SAW, kepada keluarganya, sahabatnya, sampai kepada kita selaku umatnya.

Atas izin Allah SWT penulis dapat menyelesaikan Hasil Proposal tentang “Penerimaan Algoritma *Latent Sematic Indexing* Pada *Sistem Search Engine* Data Citra Satelit Lapan”. Penulis dengan senang hati ingin mengucapkan terimakasih yang sebesar-besarnya kepada :

1. **Eneng Tita Tosida, S. Tp., M.Si., M.Kom** sebagai Pembimbing I yang telah memberikan bimbingan serta motivasi kepada penulis dalam penyusunan hasil proposal ini.
2. **Dr. Fajar Delli Wihartiko, MM., M.Kom** sebagai Pembimbing II yang telah memberikan bimbingan serta motivasi kepada penulis dalam penyusunan hasil proposal ini.
3. **Dr. Prihastuti Harsani, M.Si**, selaku Kepala Program Studi Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam.
4. **Kedua Orang Tua** yang telah memberikan dukungan, motivasi dan doa yang terus menerus untuk keberhasilan penyusunan laporan ini.
5. Semua Team Analisis dan rekan kelas Ilmu Komputer angkatan 2015 yang selalu memberikan *support* dan bantuan dalam pengerjaan laporan ini.
6. Semua pihak yang telah membantu baik secara langsung maupun tidak langsung yang tidak dapat disebutkan satu persatu terimakasih atas segalanya.

Disadari Proposal ini masih kurang sempurna. Oleh karena itu, kritik dan saran dari semua pihak sangat diharapkan guna penyempurnaan Skripsi ini. Semoga Skripsi ini bermanfaat bagi pembacanya.

Bogor, 05 Agustus 2019

Dimas Priyandi
0651 15 144

DAFTAR ISI

HALAMAN PENGESAHAN	i
PERNYATAAN KEASLIAN KARYA TULIS	ii
PERNYATAAN PELIMPAHAN SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA	iii
DAFTAR RIWAYAT HIDUP	iv
RINGKASAN	v
KATA PENGANTAR	vi
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	x
DAFTAR LAMPIRAN	xi
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Tujuan	2
1.3 Ruang Lingkup	2
1.4 Manfaat Penelitian	2
BAB II TINJAUAN PUSTAKA	
2.1 Tinjauan Pustaka	3
2.1.1 <i>Latent Semantic Indexing</i>	3
2.1.2 <i>Singular Vector Dekomposition</i>	3
2.1.3 <i>Information Retrieval</i>	3
2.1.4 <i>Preprocessing</i>	3
2.1.5 <i>Extraction/Tokenization</i>	4
2.1.6 <i>Filtration</i>	4
2.1.7 <i>Stemming</i>	4
2.1.8 <i>TF (Term Frequency) an ID (Invers Document Frequency)</i>	4
2.1.9 <i>Search Engine</i>	5
2.1.10 <i>Citra Satelit</i>	5
2.1.11 <i>VSM (Vector Space Model)</i>	5
2.1.12 <i>Recision dan Recall</i>	6
2.2 Penelitian Terdahulu	6
2.3 Perbandingan Penelitian Terdahulu	7
BAB III METODE PENELITIAN	
3.1 Metode Penelitian	8
3.1.1 Perencanaan Sistem	8
3.1.2 Analisis Sistem	8
3.1.3 Perancangan Sistem	9
3.1.4 Implementasi Sistem	9
3.1.5 Uji Coba (<i>Testing</i>)	9
3.2 Penggunaan	9
3.2.1 Waktu dan Tempat Pelaksanaan	9
3.3 Alat dan Bahan	10
3.3.1 Alat Penelitian	10
3.3.2 Bahan Penelitian	10
BAB IV PERANCANGAN DAN IMPLEMENTASI	

4.1 Tahap Perencanaan	11
4.2 Analisis Sistem	11
4.2.1 Analisis Sistem yang Sedang Berjalan	11
4.2.2 Analisis Sistem yang Akan Dikembangkan	12
4.3 Perancangan Sistem	13
4.3.1 <i>Text Preprocessing</i>	14
4.3.2 Pembobotan TF - IDF	19
4.3.3 Pembobotan LSI	20
4.3.4 Pehitungan Kemiripan Menggunakan <i>Cosine Similarity</i>	22
4.4 Perancangan Basis Data	24
4.4.1 Perancangan <i>Entity Relationship Relation (ERD)</i>	24
4.4.2 Perancangan Secara Detail	24
4.4.2.1 Rancangan Antarmuka	24
4.5 Implementasi	25
4.5.1 Implementasi <i>Database</i> Menggunakan MySQL	26
4.5.2 Implementasi Sistem	26
BAB V HASIL DAN PEMBAHASAN	
5.1 Hasil Penelitian	27
5.1.1 Halaman Utama	27
5.1.2 Halaman Citra Satelit	27
5.1.3 Tampilan Citra Satelit	28
5.2 Pembahasan Sistem	28
5.2.1 Penginputan Dokumen Citra Satelit	28
5.2.2 <i>Processing</i> Pada Dokumen dan <i>Query</i>	29
5.2.3 Pembobotan TF - IDF	29
5.2.4 Pembobotan <i>Latent Semantic Indexing</i>	30
5.2.5 Proses Untuk Menghitung <i>Cosine Similarity</i>	31
5.2.6 Hasil Pencarian Data Satelit	32
5.3 Uji Coba Sistem	33
5.3.1 Uji Coba Struktural	33
5.3.2 Uji Coba Fungsional	34
5.3.3 Uji Coba Validasi	34
5.3.3.1 Uji Coba <i>Precision</i> dan <i>Recall</i>	34
BAB VI KESIMPULAN DAN SARAN	
6.1 Kesimpulan	39
6.2 Saran	39
DAFTAR PUSTAKA	40

DAFTAR GAMBAR

Gambar Halaman

Gambar 1. Tahapan <i>Preprocessing</i>	4
Gambar 2. Representasi Vector Dokumen dan <i>Query</i>	5
Gambar 3. Ilustrasi dan Perhitungan <i>Precision</i> dan <i>Recall</i>	6
Gambar 4. Diagram Tahapan SDLC	8
Gambar 5. Analisis Sistem yang Sedang Berjalan	12
Gambar 6. Analisis Sistem yang Akan Dikembangkan	13
Gambar 7. Flowchart <i>Information Retrieval System</i>	14
Gambar 8. Proses <i>Tokenization</i>	15
Gambar 9. Proses <i>Filtering</i>	16
Gambar 10. Flowchart Algoritma Nazief & Adriani	18
Gambar 11. Proses Pembobotan TF - IDF	19
Gambar 12. Perhitungan Kemiripan dengan <i>Cosine Similarity</i>	22
Gambar 13. ERD (<i>Entity Relationship Diagram</i>)	24
Gambar 14. Rancangan Halaman Utama	25
Gambar 15. Rancangan Hasil Pencarian Citra Satelit	25
Gambar 16. Implementasi <i>Database</i> PHPMyAdmin	26
Gambar 17. Halaman Utama	27
Gambar 18. Hasil Pencarian Citra Satelit	28
Gambar 19. Tampilan Citra Satelit	28
Gambar 20. Metadata Citra Satelit	29
Gambar 21. <i>Preprocessing</i> Tabel <i>Index</i>	30
Gambar 22. Hasil Perhitungan Tabel Bobot TF - IDF	30
Gambar 23. Hasil Proses SVD Disimpan Dalam <i>File Txt</i>	31
Gambar 24. Hasil Proses Similaritas	31
Gambar 25. Hasil Pencarian Pada Query 1 Term	32
Gambar 26. Hasil Pencarian Pada 3 sampai 5 Term Query	33
Gambar 27. Grafik Perbandingan Hasil Uji Coba <i>Precision</i> dan <i>Recall</i>	38

DAFTAR TABEL

Tabel Halaman

Tabel 1. Tabel Perbandingan Penelitian Terdahulu	7
Tabel 2. Hasil Tokenisasi	16
Tabel 3. Hasil <i>Filtering</i>	17
Tabel 4. Hasil <i>Stemming</i>	18
Tabel 5. Hasil Pembobotan TF - IDF	20
Tabel 6. Uji Coba Struktural	34
Tabel 7. Uji Coba Fungsional	34
Tabel 8. Pengujian 1 <i>Term</i> Pada Wilayah Indonesia	35
Tabel 9. Pengujian 1 <i>Term</i> Pada Wilayah Luar Negeri	35
Tabel 10. Pengujian 2 <i>Term</i> Pada Wilayah Indonesia	36
Tabel 11. Pengujian 2 <i>Term</i> Pada Wilayah Luar Negeri	36
Tabel 12. Pengujian 3 sampai 5 <i>Term</i> Pada Wilayah Indonesia	37
Tabel 13. Pengujian 3 sampai 5 <i>Term</i> Pada Wilayah Luar Indonesia	37

DAFTAR LAMPIRAN

Gambar Halaman

Lampiran 1. DBF Level 1	xii
Lampiran 2. Implementasi Sistem	xiii
Lampiran 3. TF-IDF dan LSI Tabel	xv

BAB I

PENDAHULUAN

1.1 Latar Belakang

Lembaga Penerbangan dan Antariksa Nasional adalah Lembaga Pemerintah Non Kementerian Indonesia yang melaksanakan tugas pemerintahan di bidang penelitian dan pengembangan kedirgantaraan dan pemanfaatannya serta penyelenggaraan keantariksaan sesuai dengan peraturan perundang-undangan. Salah satu fungsi yang diselenggarakan oleh LAPAN yaitu pelaksanaan penelitian dan pengembangan sains antariksa dan atmosfer, teknologi penerbangan dan antariksa, dan penginderaan jauh serta pemanfaatannya.

Penginderaan jauh adalah pengkajian atas informasi mengenai daratan dan permukaan air bumi dengan menggunakan citra yang diperoleh dari sudut pandang atas (overhead perspective), menggunakan radiasi elektromagnetik dalam satu beberapa bagian dari spektrum elektromagnetik yang dipantulkan atau dipancarkan dari permukaan bumi (Campbell, 2011). Citra satelit merupakan hasil masukan dari observasi dengan penginderaan jauh. Citra satelit tersebut dapat dimanfaatkan untuk mengamati objek atau fenomena bumi serta untuk pemantauan wilayah hutan, pemantauan lahan pertanian dan pangan.

Sejumlah citra satelit LAPAN diakuisisi dari satelit LAPAN-A2 / LAPAN-ORARI dan LAPAN-A3 / LAPAN-IPB. Saat ini, pengguna di seluruh Indonesia dapat dengan mudah untuk mengakses dan mengunduh citra satelit untuk keperluan penelitian melalui sistem katalog citra satelit LAPAN. Selain sebagai media penyimpanan, dengan katalog citra satelit berbasis web ini akan mempermudah Pusteksat untuk mendiseminasikan produk satelit LAPAN (Herawan A & Salaswati S, 2018). Pada sistem katalog citra satelit tersebut terdapat fasilitas pencarian yang dapat membantu pengguna untuk menemukan citra satelit melalui metadata. Pencarian citra satelit dilakukan dengan melalui pencocokan kata kunci dengan memasukkan kriteria yang tersedia, kemudian apabila kriteria yang dimasukkan sesuai dengan metadata pada citra satelit maka sistem akan menampilkan citra satelit tersebut. Metadata dapat didefinisikan sebagai data tentang data atau data yang berisi informasi mengenai data tersebut (Haynes, 2004). Metadata digunakan untuk mendokumentasikan produk data yang dihasilkan serta untuk menjawab pertanyaan apa, siapa, dimana, dan untuk apa data tersebut dipersiapkan. Metadata pada citra satelit biasanya berisi tanggal pengambilan citra, nama daerah atau lokasi, produk satelit, koordinat dan lain sebagainya.

Namun fasilitas pencarian satelit tersebut masih memiliki kekurangan dalam mencocokkan kata kunci yang dimasukkan dengan kata dalam dokumen, sehingga seringkali hasil pencarian pada dokumen belum relevan. Selain itu banyaknya kata yang memiliki kesamaan seperti nama provinsi dan nama daerah yang terdapat pada kriteria masukan sehingga jika hanya memanfaatkan pencocokkan kata pada dokumen sehingga membuat pencocokkan kata semakin tidak relevan. Untuk menemukan informasi dengan relevan antar kata kunci yang dimasukkan dengan dokumen maka diperlukan *Information Retrieval System*. *Information Retrieval System* merupakan studi tentang sistem pengindeksan, pencarian, dan mengingat data, khususnya teks atau bentuk tidak terstruktur lainnya dimana pengindeksan dan pengambilan informasi berasal dari sumber informasi heterogen (Prasetyo E, 2012).

Dalam pencarian dokumen pada Information retrieval terdapat banyak metode yang baik untuk dapat menemukan informasi yang relevan dengan cepat dan akurat. Salah satu metode dalam Information Retrieval yang cukup terkenal dan diterapkan oleh raksasa mesin pencarian google untuk penggunaan karakter string dalam dokumen untuk menetapkan relevansi semantik untuk istilah pencarian (keyword) yang digunakan atau dengan kata lain, untuk membantu membangun makna sebenarnya dari teks pada posting blog atau halaman web adalah metode Latent Semantic Indexing (LSI) (Alhensiri, 2010). LSI merupakan sebuah metode automatic indexing dan retrieval dengan memanfaatkan semantic structure (struktur asosiasi terms dengan dokumen) yang secara implisit terdapat dalam suatu dokumen untuk digunakan dalam pencarian dokumen yang relevan dengan terms dalam query (Muhammad et al, 2011).

Berdasarkan permasalahan di atas, agar dalam pencarian metadata citra satelit dapat dilakukan berdasarkan judul serta deskripsi sehingga menemukan dokumen citra yang relevan maka solusi untuk pencarian data citra satelit tersebut peneliti membuat “Penerapan Algoritma Latent Semantic Indexing Pada Sistem Search Engine Data Citra Satelit”. Dengan menerapkan algoritma Latent Semantic Indexing diharapkan pencarian data citra satelit akan semakin akurat dan optimal.

1.2 Tujuan Penelitian

Penelitian ini bertujuan untuk membuat Penerapan Algoritma Latent Semantic Indexing Pada Sistem Search Engine Data Citra Satelit.

1.3 Ruang Lingkup Penelitian

Sistem ini dibatasi pada ruang lingkup yaitu:

1. Algoritma yang digunakan untuk pencocokan string dengan menemukan hubungan tersembunyi antara kata yang dimasukkan pada kata kunci dengan kata dalam dokumen dengan memanfaatkan temu kembali informasi.

1.4 Manfaat Penelitian

Manfaat dari penelitian ini yaitu :

1. Mengembangkan sistem search engine atau mesin pencari pada citra satelit agar dapat berjalan dengan tepat dan akurat.
2. Dapat mengetahui metode pencocokan string yang tepat sehingga dapat menerapkannya pada data citra satelit agar pencarian menjadi optimal.

BAB II DAFTAR PUSTAKA

2.1 Tinjauan Pustaka

2.1.1 *Latent Semantic Indexing*

Latent Semantic Indexing (LSI) adalah sebuah teknik berbasis bidang vektor (*vector space*) yang dapat menciptakan asosiasi antara dokumen-dokumen yang berhubungan secara konseptual (Chen C, 2001). Pada model bagian vektor sekumpulan dokumen yang diindex direpresentasikan dengan bentuk model $m \times n$ *term document matrix* A yang dinotasikan sebagai berikut,

$$A = [a_{ij}] \dots\dots\dots (1)$$

Elemen-elemen di dalam matriks A merupakan bobot frekuensi *term* i yang muncul pada dokumen j . Kolom-kolom pada matriks A merepresentasikan n vektor dokumen dan baris-barisnya merepresentasikan m vektor *terms*. Fungsi temu-kembali membandingkan vektor kueri q dengan tiap kolom yang merepresentasikan dokumen tertentu dalam bidang vektor. Sebuah derajat kesamaan antara vektor kueri dan semua dokumen yang direpresentasikan dalam bidang vektor kemudian diukur untuk perangkingan (Alhensiri A.A, 2003).

2.1.2 *Singular Vector Dekomposition*

Dekomposisi Nilai Singular (*Singular Value Decomposition*) atau yang lebih dikenal sebagai SVD, adalah salah satu teknik dekomposisi berkaitan dengan nilai singular (*singular value*) suatu matriks yang merupakan salah satu karakteristik matriks tersebut (Ariyanti G, 2010). Dekomposisi nilai singular matriks riil A $m \times n$ adalah faktorisasi.

$$A_{m \times n} = USV^T$$

Matriks $A_{m \times n}$ dapat dinyatakan sebagai dekomposisi matriks yaitu matriks U, S dan V. Matriks S merupakan matriks diagonal dengan elemen diagonalnya berupa nilai-nilai singular matriks A, sedangkan matriks U dan V merupakan matriks-matriks yang kolom-kolomnya berupa vektor singular kiri dan vektor singular kanan dari matriks A untuk nilai singular yang bersesuaian (Ariyanti G, 2010).

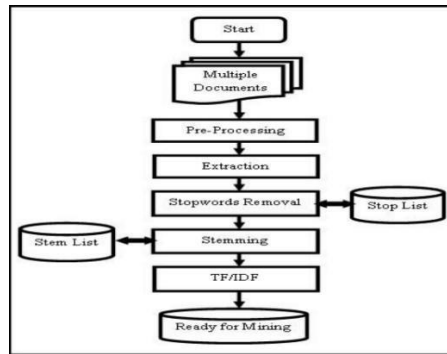
2.1.3 *Information Retrieval*

Menurut Manning, *et al* (2009), *information retrieval* (IR) adalah menemukan bahan (biasanya dokumen) yang bersifat terstruktur (biasanya teks) yang memenuhi kebutuhan informasi dari dalam koleksi besar (biasanya disimpan di komputer). *Information retrieval* secara umum terdapat 2 tahapan besar, yaitu *preprocessing/indexing* dan *retrieval*.

2.1.4 *Preprocessing*

Menurut Wahyuni, R.T *et al* (2017), *preprocessing* yaitu proses pendahulu yang diterapkan terhadap data teks yang bertujuan untuk menghasilkan data *numerik*.

Beberapa tahapan *preprocessing* diantaranya yaitu *tokenization*, *filtration* dan *weighthing*. Langkah kerja *preprocessing* dapat dilihat pada Gambar 1.



Gambar 1. Tahapan *Preprocessing*

(Sumber : Vijayarani et al, 2015)

2.1.5 Extraction / Tokenization

Tokenization merupakan proses memecah dokumen menjadi kumpulan kata. *Tokenization* dilakukan dengan menghilangkan tanda baca atau menghilangkan spasi dengan *white space* lalu mengubah hurufnya menjadi huruf kecil (Alamanda R et al, 2016).

2.1.6 Filtration

Filtration merupakan proses untuk menghilangkan kata yang tidak penting (*stop list*) seperti kata depan, kata sambung, kata ganti, dan lainnya. *Stoplist/stopword* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of-words* (Alamanda R et al, 2016).

2.1.7 Stemming

Stemming merupakan tahap mencari *root* kata dari tiap kata hasil *filtration*. Pada proses ini mengembalikan kata imbuhan menjadi kata baku (Alamanda R et al, unnamed2016).

2.1.8 TF (*Term Frequency*) dan IDF (*Invers Document Frequency*)

Menurut Robertson (2005), metode TF-IDF (*Term Frequency Inverse Document Frequency*) merupakan suatu cara untuk memberikan bobot hubungan suatu kata (*term*) terhadap dokumen. Metode ini menggabungkan dua konsep untuk perhitungan bobot yaitu, frekuensi kemunculan sebuah kata didalam sebuah dokumen tertentu dan *inverse* frekuensi dokumen yang mengandung kata tersebut. Metode ini akan menghitung bobot setiap token *t* di dokumen *d* dengan rumus:

$$W_{dt} = tf_{dt} * IDF_{t+1}$$

Dimana :

- d : dokumen ke-d
- W : bobot dokumen ke-d terhadap kata ke-t
- tf : banyaknya kata yang dicari pada sebuah dokumen
- IDF : inversed Document Frequency Nilai IDF didapatkan dari $IDF : \log_2(D/df)$
- D : total dokumen
- df : Banyak dokumen yang mengandung kata yang dicari

2.1.9 Search Engine

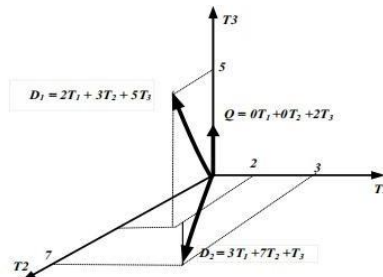
Search engine atau mesin pencari adalah program komputer yang dirancang untuk membantu seseorang menemukan file-file yang disimpan dalam komputer, misalnya dalam sebuah server umum di web (WWW) atau dalam komputer sendiri. Mesin pencari memungkinkan kita untuk meminta *content media* dengan kriteria yang spesifik (biasanya yang berisi kata atau frasa yang kita tentukan) dan memperoleh daftar file yang memenuhi kriteria tersebut (Yusuf S, 2010).

2.1.10 Citra Satelit

Citra Satelit merupakan citra yang diperoleh dari hasil penginderaan jauh. Penginderaan Jauh (*Remote Sensing*) merupakan pengamatan suatu obyek menggunakan sebuah alat dari jarak jauh (Campbell, 2011).

2.1.11 VSM (Vector Space Model)

Menurut Manning et al. (2009), *vector space model* adalah representasi kumpulan dokumen sebagai vektor dalam sebuah ruang vektor. *Vector space model* merupakan cara untuk memperoleh informasi yang relevan terhadap query yang dimasukkan dengan menghitung tingkat kesamaan antara *query* dengan dokumen yang tersimpan dalam sistem. Dokumen yang terambil dalam disortir sesuai urutan yang memiliki kemiripan. Sebuah dokumen d_j dan sebuah *query* di representasikan sebagai vektor t -dimensi seperti pada Gambar 2.



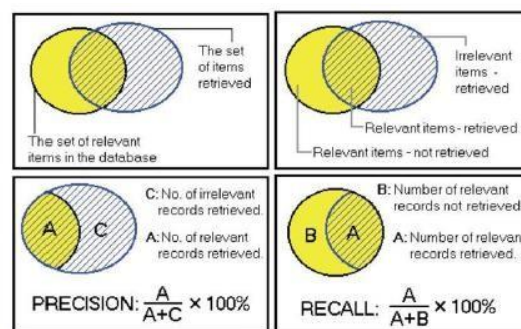
Gambar 2. Representasi *Vector* Dokumen dan *Query*
 Sumber : Kartarina & Junaedi H (2017).

Cosine Similarity atau *Sim* (q, dj) digunakan untuk mengevaluasi tingkat similaritas atau kemiripan dari dokumen (dj) berkaitan dengan *query* (q) sebagai korelasi antara vektor dj dan q (Amin F & Purwantiningtyas, 2015). Perhitungan similaritas dapat dilihat pada persamaan berikut :

$$Sim(q, d_j) = \frac{q \cdot d_j}{|q| * |d|} = \frac{\sum_{i=1}^n W_{iq} \cdot W_{ij}}{\sqrt{\sum_{i=1}^n (W_{iq})^2} \cdot \sqrt{\sum_{i=1}^n (W_{ij})^2}}$$

2.1.12 *Recision dan Recall*

Menurut Kurniawan (2010), *recall* adalah perbandingan jumlah dokumen relevan yang terambil sesuai dengan *query* yang diberikan dengan total kumpulan dokumen yang relevan dengan *query*. *Precision* adalah perbandingan jumlahdokumen yang relevan terhadap *query* dengan jumlah dokumen yang terambil dari hasil pencarian. Ilustrasi perhitungan *precision* dan *recall* dapat dilihat pada Gambar 3 dibawah ini :



Gambar 3. Ilustrasi dan Perhitungan *Precision* dan *Recall*.
(Sumber: www.hsl.creighton.edu/hsl/searching/recall-precision.html).

2.2 Penelitian Terdahulu

Ada beberapa penelitian yang terkait dengan penelitian ini diantaranya yaitu:

1. (Kartarina & Junaedi H, 2017). ”**Information Retrieval dengan Menggunakan Metode Latent Semantik Indexing (LSI) pada Proses Searching dan Klasifikasi Buku: Studi Kasus Perpustakaan STMIK Bumigor**”. Adanyapermasalahan dalam klasifikasi / memasukkan buku kedalam kelompoknya danmasalah pencarian buku oleh pengunjung yang memperoleh buku kadang bukuyang diperoleh dari tempatnya tidak sesuai dengan yang diharapkan misalnya judul dan isi tidak sesuai. Untuk mendapatkan *Information Retrieval* dalam indeks buku / dokumen di perpustakaan mencoba menggunakan metode *Latent Semantic Indexing* (LSI) yaitu dengan menerapkan model *Singular Value Decomposition* (SVD).
2. (Amin F, 2012) “**Sistem Temu Kembali Informasi dengan Metode Vector Space Model**”.STKI mampu melakukan pencarian dokumen teks bahasa Indonesia dan menampilkan hasil pencarian dokumen teks dengan disertai bobot tiap dokumen beserta letak dokumen dengan waktu komputasi rata-rata1,5 detik. Hasil Uji recall dan precision STKI menunjukkan hasil pencarian dokumen teks bahasa Indonesia

memiliki rata-rata recall = 0,19 dan rata-rata precision = 0,54.

3. (Pardede J *et al*, 2013).“**Implementasi Metode Generalized Vector Space Model Pada Aplikasi Information Retrieval**”. Generalized Vector Space Model (GVSM) adalah salah satu dari model sistem IR yang termasuk dalam model aljabar. Query yang dimasukkan oleh pengguna akan diproses terlebih dahulu. Pengolahan kata meliputi tokenizing, stop word removal, dan stemming. Aplikasi ini melakukan pencarian dokumen seperti pdf (*.pdf) dan Ms Word Doc (*.doc & *.docx.) yang relevan dengan query. Hasil pencarian ini disusun berdasarkan nilai kesamaan GVSM yang tertinggi. Dengan menggunakan Generalized Vector Space Model, hasil pencarian dokumen menjadi lebih relevan berdasarkan nilai perbandingan kemiripan.
4. (Melita R *et al*, 2013).”**Penerapan Metode Term Frequency Inverse Document Frequency (Tf-Idf) Dan Cosine Similarity Pada Sistem Temu Kembali Informasi Untuk Mengetahui Syarah Hadits Berbasis Web (Studi Kasus: Syarah Umdatil Ahkam)**”. Sistem Temu Kembali Informasi yang dapat dimanfaatkan untuk mengetahui syarah hadits, karena dapat memberikan alternatif berupa metode similarity yang dapat digunakan untuk melakukan pencarian dokumen relevan dengan yang kita inginkan. Metode *similarity* yang digunakan adalah cosine *similarity* dengan pembobotan kata menggunakan metode TFIDF dan menerapkan teks preprocessing terlebih dahulu untuk memperkecil term sehingga bisa mempercepat proses perhitungan term. Teks *preprocessing* tersebut meliputi *tokenizing*, *stopword removal* atau *filtering*, dan *stemming*. Hasil uji coba dengan pengujian *confusion matrix* didapatkan: recall 88.7%, precision 100%, accuracy 88,73 %, dan *error rate* 11,27 %.

2.3 Tabel Perbandingan Penelitian Terdahulu

Tabel penelitian terdahulu ini digunakan untuk melihat perbandingan dengan penelitian yang telah dilakukan pada sebelumnya untuk bahan referensi.

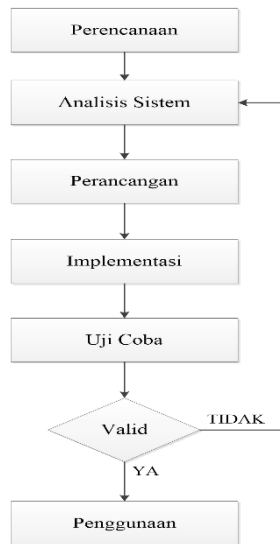
Tabel 1. Tabel Perbandingan Penelitian Terdahulu

No	Peneliti Dan Tahun	Metode				Basis	
		LSI	TF-IDF	VSM	GVSM	Desktop	Web
1	Kartarina & Junaedi H, 2017	√	-	√	-	-	√
2	Amin F, 2012	-	-	√	-	-	√
3	Pardede J <i>et al</i> , 2013	-	-	-	√	-	√
4	Melita R <i>et al</i> , 2013	-	√	-	-	-	√
5	Dimas Priyandi (2019)	√	√	√	-	-	√

BAB III METODE PENELITIAN

3.1 Metode Penelitian

Metodologi Penelitian yang diterapkan pada Penerapan Algoritma *Latent Semantic Indexing* Pada Sistem *Search Engine* Data Citra Satelit LAPAN ini adalah pola siklus pengembangan sistem SDLC (*System Development Life Cycle*). Adapun tahapan-tahapan SDLC secara lengkap disajikan pada Gambar 4:



Gambar 4. Diagram Tahapan SDLC.

3.1.1 Perencanaan Sistem

Tahap perencanaan merupakan suatu tahap pembelajaran mengenai kasus atau permasalahan yang akan diselesaikan. Kegiatan yang dilakukan pada tahap ini yaitu seperti mengambil data dengan melakukan observasi, wawancara, serta melakukan studi literatur untuk menguasai dan memahami dasar dasar teori yang mendukung penelitian :

1. Tahap Observasi.
Mempelajari sistem yang sedang berjalan yaitu sistem katalog citra satelit LAPAN.
2. Tahap wawancara.
Melakukan wawancara dengan melakukan tanya jawab kepada pihak-pihak yang terlibat sehingga mendapatkan informasi mengenai sistem yang akan dikembangkan.
3. Studi Literatur.
Studi literatur dilakukan dengan mencari referensi-referensi yang mendukung serta mempelajarinya agar dapat menyelesaikan masalah. Melalui studi literatur ini peneliti dapat mempelajari teori-teori pengenalan pola (*pattern*), algoritma *Latent Semantic Indexing*. Referensi tersebut diambil berdasarkan pada beberapa sumber yang relevan seperti buku, jurnal ilmiah, prosiding, maupun situs internet.

3.1.2 Analisis Sistem

Tahap analisis sistem dilakukan terhadap kebutuhan aplikasi yang akan dibangun. Dalam aplikasi ini dikhususkan pada fungsi *search engine* atau mesin pencari agar pencarian pada citra satelit LAPAN dapat akurat dan optimal. Pemilihan algoritma pencarian *string* yang ingin diterapkan dan dikembangkan pada sistem *search engine* ini yaitu algoritma *Latent Semantic Indexing* dikarenakan algoritma ini sangat efektif dalam melakukan pencarian satu *pattern* atau pola kata dengan melihat asosiasi tiap *term*.

3.1.3 Perancangan Sistem

Tahap perancangan sistem dilakukan melalui 3(tiga) tahap, yaitu:

1. Perancangan Basis Data.
Perancangan sistem secara basis data (*database*) dapat dilakukan menggunakan pendekatan terstruktur meliputi: ERD (*Entity Relationship Diagram*), Relasi *Entitas* dan spesifikasi *file*.
2. Perancangan Sistem Secara Umum.
Perancangan Sistem secara keseluruhan dalam penelitian ini dilakukan dengan menggunakan *Diagram Konteks*, *Data Flow Diagram (DFD)*, *flowchart* sistem, struktur navigasi dan tampilan sistem berikutnya yang merupakan simbol-simbol yang digunakan untuk menggambarkan urutan proses yang terjadi didalam suatu program komputer secara sistematis dan logis.
3. Rancangan Sistem Secara Detail.
Merancang *form-form* yang diperlukan dalam melakukan suatu sistem yang akan dikembangkan.

3.1.4 Implementasi Sistem

Tahap implementasi pada penelitian ini merupakan tahap pembuatan sistem, sistem yang akan dibangun dengan menggunakan bahasa pemrograman PHP dan HTML, untuk database menggunakan MySQL (*XAMPP*) sebagai web server (*localhost*) serta *text editor* menggunakan *Visual Studio Code*.

3.1.5 Uji Coba (*Testing*)

Tahap uji coba ini dilakukan untuk mengukur kemampuan sistem dalam melakukan pencarian dokumen yang relevan dengan kata kunci yang dimasukkan oleh pengguna. Pada pengujian ini mengikuti prinsip relevansi pada *information retrieval* yaitu menggunakan ukuran *recall*, *precision*. Ukuran *recall* dan *precision* itu bergantung pada apa yang sesungguhnya dimaksud dengan dokumen yang relevan itu dan memastikan relevan-tidaknya sebuah dokumen.

3.2 Penggunaan

Tahap penggunaan merupakan tahap tujuan dari pembuatan penelitian ini untuk mempermudah pengguna dalam menemukan data citra satelit dengan akurat dan optimal.

3.2.1 Waktu dan Tempat Penelitian

Waktu pelaksanaan penelitian ini dilakukan dari bulan Maret 2019 sampai Mei 2019 (3 Bulan). Penelitian ini dilaksanakan di Lembaga Penerbangan dan Antariksa

Nasional.

3.3 Alat dan Bahan

3.3.1 Alat Penelitian

Alat yang dibutuhkan dalam penelitian ini adalah hardware dan software meliputi :

Software : Visual Studio Code, XAMPP, OS Windows 8.164-bit, MS. Visio.

Hardware : Laptop Asus X455L, *Processor Intel (R), Core (TM)*, CPU @1.90GHz,ROM 500 GB HDD SATA, RAM 4,00 GB DDR3.

3.3.2 Bahan Penelitian

Bahan yang dibutuhkan dalam penelitian ini adalah jurnal terkait dan data citra satelit.

BAB IV

PERANCANGAN DAN IMPLEMENTASI

4.1 Tahap Perencanaan

Tahap awal pada penelitian kali ini dimulai dengan mencari sumber data dengan cara melakukan observasi dan wawancara kepada narasumber instansi LAPAN. Data citra satelit yang dapat digunakan yaitu data yang sudah diolah sebelumnya oleh para pusteksat yang berada di LAPAN. Data yang diberikan berbentuk citra dengan ukuran lebar dan panjang berbeda satu sama lain dengan format png. Data citra tersebut berupa citra yang menampilkan gunung, pulau serta wilayah di Indonesia dan Luar Negeri. Citra yang diakuisisi oleh LAPAN dihasilkan dari satelit LAPAN-A2 dan LAPAN-A3 dengan masing-masing produk satelit. Terdapat 2 produk satelit yang digunakan yaitu Spacecam dan LISA.

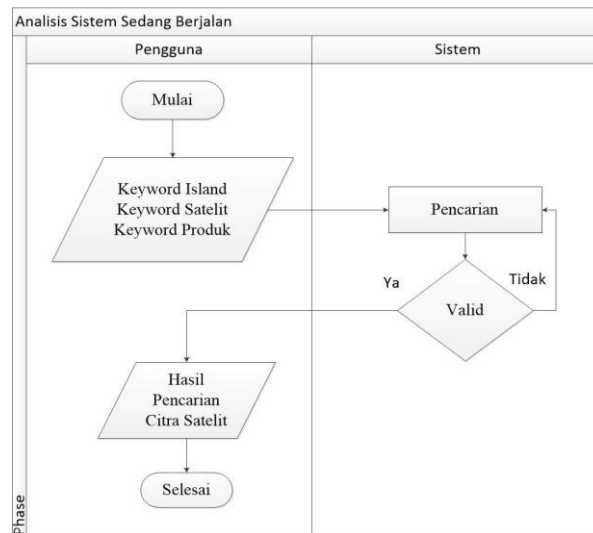
Sistem yang akan dikembangkan pada penelitian ini yaitu untuk mengoptimalkan pencarian pada metadata citra satelit bukan hanya berdasarkan judulnya melainkan juga deskripsi pada citra tersebut, sehingga sistem pencarian ini mampu mencari keterkaitan antara kata kunci yang dimasukkan pada pencarian dengan deskripsi mengenai citra satelit tersebut.

4.2 Analisis Sistem

Sebelum merancang sistem terlebih dahulu dilakukan analisis terhadap data yang sudah diperoleh dan mengamati sistem yang sedang berjalan yaitu pada basis data katalog citra satelit. Dimana pada sistem tersebut pengguna dapat melakukan pencarian dengan tiga cara yaitu memilih provinsi, memilih produk satelit dan memilih satelit berdasarkan kata kunci (*keyword*) yang dimasukkan pada menu *search* tersebut. Setelah itu, sistem akan memberikan keluaran berupa citra yang diinginkan oleh pengguna sesuai dengan *keyword* yang telah dimasukkan. Data dapat dilihat dan diunduh oleh pengguna.

4.2.1 Analisis Sistem yang Sedang Berjalan

Analisis ini berfungsi untuk mengetahui kekurangan dan permasalahan apa saja yang sering muncul dalam sistem yang sedang berjalan. Adapun *flowchart* dari sistem katalog citra satelit tersebut dapat dilihat pada Gambar 5.



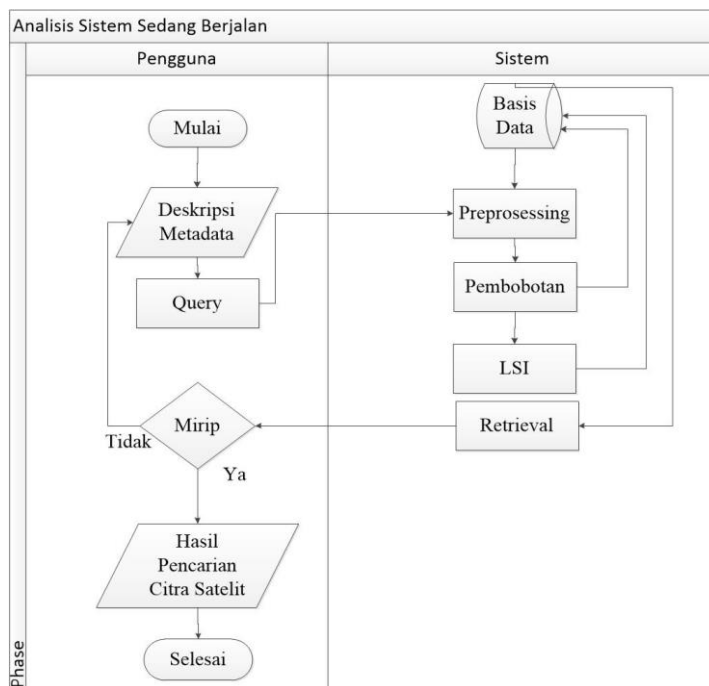
Gambar 5. Analisis Sistem Yang Sedang Berjalan.

4.2.2 Analisis Sistem yang Akan Dikembangkan

Analisis sistem yang akan dikembangkan yaitu pada proses pencarian string dengan menggunakan algoritma *Latent Semantic Indexing* dengan kaidah proses *retrieval* yaitu *preprocessing* teks, menghitung *index*, pemberian bobot pada setiap kata, dan melakukan perangkingan untuk melihat nilai kemiripan atau *similarity* antara kata yang dimasukkan pada kata kunci dengan data yang sudah diolah di dalam *database*. Langkah awal admin menambahkan data dokumen citra satelit ke dalam *database* untuk kemudian dilakukan pengolahan teks. Langkah berikutnya admin mengontrol untuk *preprocessing* pada dokumen sehingga menjadi beberapa kumpulan *term*.

Setelah dilakukan *preprocessing*, setiap *term* dibentuk menjadi sebuah matriks *term* agar dapat dilakukan proses *indexing* yaitu dengan menghitung jumlah *term* yang sama lalu memberikan bobot *term* pada dokumen menggunakan TF-IDF. Proses yang dilakukan pada tahap ini juga dilakukan pada setiap *term* yang dimasukkan dalam *query*. Langkah selanjutnya untuk meningkatkan relevansi hasil pencarian pada dokumen maka perlu ditingkatkan hubungan *semantic* antar *term*. Maka dipilihlah algoritma *Latent Semantic Indexing* yang dimana memiliki sifat utama yaitu untuk membangun hubungan antara istilah yang muncul dalam konteks yang serupa (Wahib *et al*, 2015).

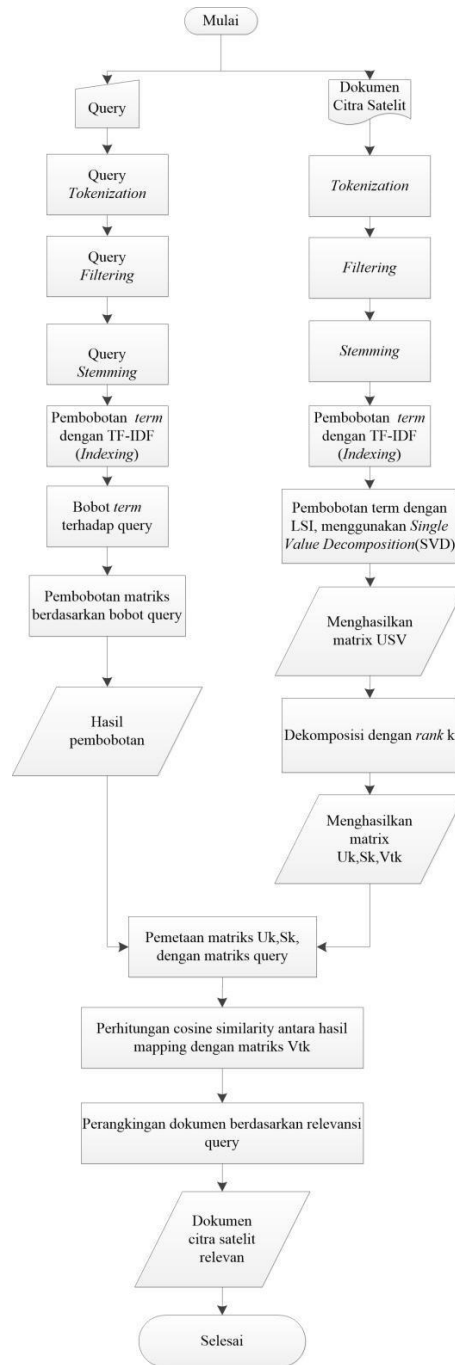
Adapun *flowchart* tahapan dari sistem yang akan dikembangkan ini dapat dilihat pada Gambar 6 dibawah ini.



Gambar 6. Analisis Sistem Yang Akan Dikembangkan.

4.3 Perancangan Sistem

Tahapan ini bertujuan untuk memberikan gambaran pada rancangan sistem yang akan dikembangkan berupa rancangan *Information Retrieval System*, perancangan basis data, dan perancangan sistem secara umum. Flowchart dari *Information Retrieval System* pada sistem ini dapat dilihat pada Gambar 7.



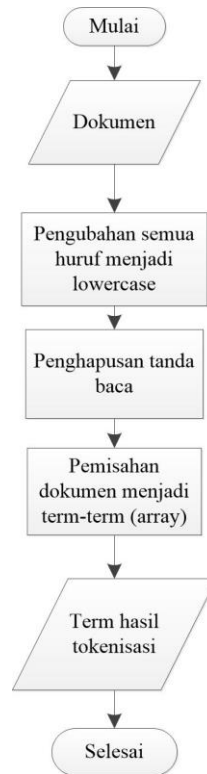
Gambar 7. Flowchart *Information Retrieval System*.

4.3.1 Text Preprocessing

Tahap awal dalam merancang sistem temu kembali informasi yaitu dengan melakukan *text preprocessing*. Sebelum melakukan *preprocessing*, terlebih dahulu dilakukan penginputan data dokumen citra satelit pada *database*. Kumpulan dokumen yang disimpan dianggap sebagai korpus dari citra satelit. Setelah data tersimpan di *database* maka akan dilakukan pengolahan teks diantaranya yaitu *tokenization*, *case folding* dan *stemming*.

1. Tokenization

Tokenization merupakan proses untuk pemisahan dokumen menjadi beberapa rangkaian *term* (Amin F, 2012). Berikut flowchart proses *tokenization* dapat dilihat pada Gambar 8.



Gambar 8. Proses *Tokenization*.

Dari gambar diatas dapat dijelaskan proses tokenisasi yaitu pertama mengubah huruf pada dokumen menjadi huruf kecil (*lowercase*). Kedua dilakukan penghapusan tanda baca seperti tanda tanya, tanda seru, tanda petik dan sebagainya. Contoh proses tokenisasi pada deskripsi dokumen citra satelit yaitu :

Dok 1 : Citra Gunung Bromo diakuisisi pada tanggal 26 April 2019 oleh satelit LAPAN-A2 dengan produk satelit SpaceCam.

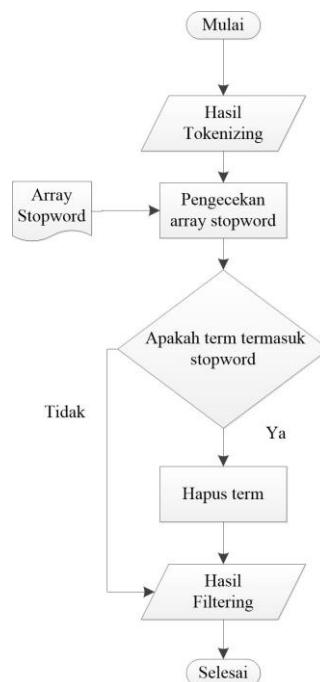
Dok 2 : Citra Gunung Kidul wilayah Provinsi Yogyakarta pada tanggal 03 April 2019 oleh satelit LAPAN-A3 dengan produk satelit LISA.

Tabel 2. Hasil Tokenisasi

Dok 1	Dok2
citra	citra
gunung	gunung
bromo	kidul
diakuisisi	wilayah
pada	provinsi
tanggal	yogyakarta
26	pada
april	tanggal
2019	03
oleh	april
satelit	2019
lapan-a2	oleh
dengan	satelit
produk	lapan-a3
satelit	dengan
spacecam	produk
	lisa

2. Filtering

Proses selanjutnya yaitu proses *filtering*. *Filtering* merupakan proses pembuangan term hasil tokenisasi yang termasuk kedalam stopwords (Amin F *et al*, 2016). *Stopword* merupakan kata penghubung atau dikatakan tidak memiliki arti karena tidak memiliki keterkaitan dengan topik tertentu (Kartarina & Junaedi H,2017). Berikut flowchart dari proses filtering dapat dilihat pada Gambar 9.



Gambar 9. Proses *Filtering*

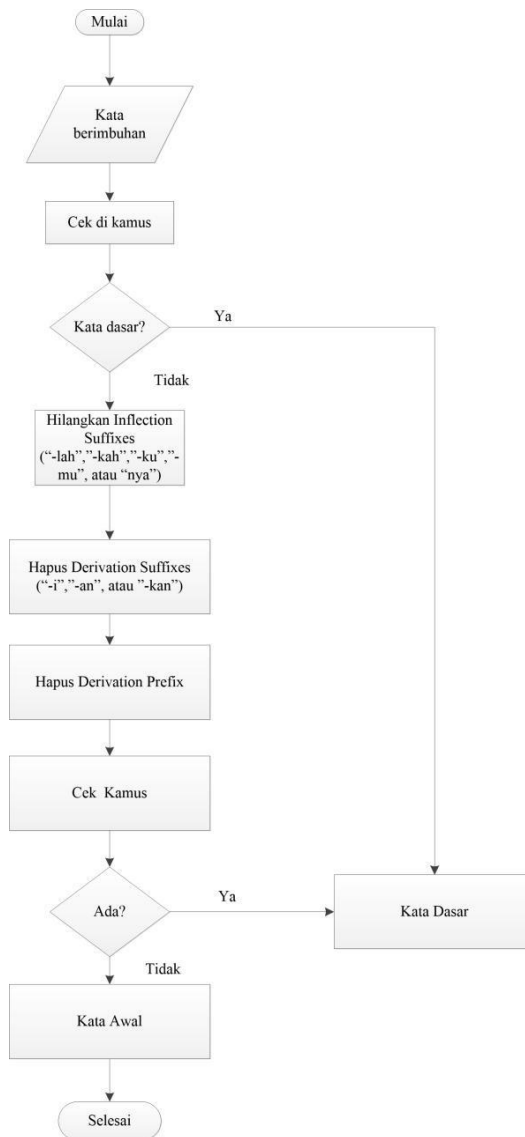
Contoh hasil filtering dapat dilihat pada tabel 3.

Tabel 3. Hasil *Filtering*.

Dok 1	Dok2
citra	citra
gunung	gunung
bromo	kidul
diakuisisi	wilayah
tanggal	provinsi
26	yogyakarta
april	tanggal
2019	03
satelit	april
lapan-a2	2019
produk	satelit
satelit	lapan-a3
spacecam	produk
-	lisa

3. *Stemming*

Proses *stemming* merupakan proses untuk mengubah *term* hasil *filtering* yang memiliki kata berimbuhan menjadi kata dasar (Hermawan & Sholihin F, 2012). Proses untuk mengambil kata dasar dengan cara memotong imbuhan pada kata tersebut, baik itu berupa awalan, akhiran, maupun sisipan. Pada penelitian ini menggunakan model *stemming* Sastrawi. Sastrawi merupakan library PHP sederhana yang dapat digunakan dalam proses *stemming*, yaitu mengubah kata berimbuhan menjadi kata dasar dalam bahasa Indonesia.. Dalam penelitian ini proses *stemming* menggunakan algoritma *Nazief & Adriani Stremmer* (2009). Berikut flowchart proses *stemming* dapat dilihat pada Gambar 10.



Gambar 10. Flowchart Algoritma Nazief & Adriani Stemmer (2009).

Dengan melihat contoh dari hasil filtering diatas maka hasil stemming dapat dilihat pada tabel 4.

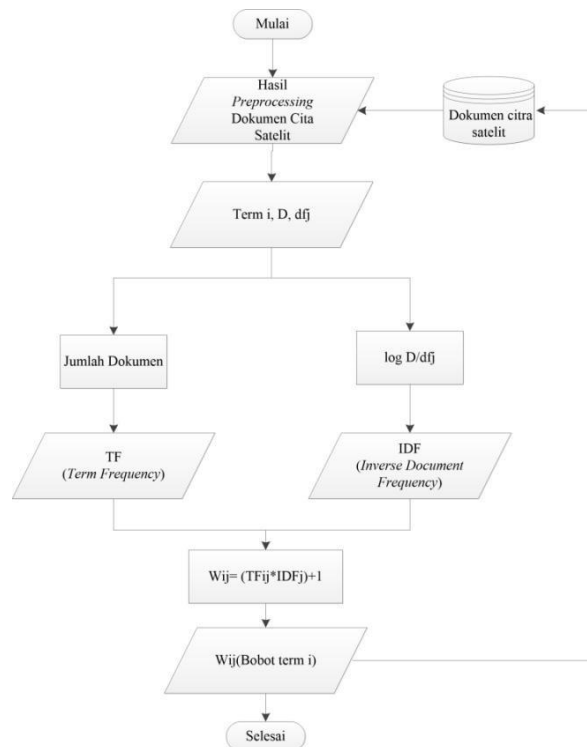
Tabel 4. Hasil *Stemming*

Dok 1	Dok2
citra	citra
gunung	gunung
bromo	kidul
akuisisi	wilayah

tanggal	provinsi
26	yogyakarta
april	tanggal
2019	03
satelit	april
lapan-a2	2019
produk	satelit
satelit	lapan-a3
spacecam	produk
	lisa

4.3.2 Pembobotan TF – IDF

Setelah *preprocessing text* dilakukan, kemudian dilanjutkan pada proses TDM (*Term Document Matrix*) yaitu suatu proses untuk membentuk matriks yang memuat nilai frekuensi kata pada masing-masing dokumen (Kartarina & Junaedi H, 2017). Dimana pembentukan TDM ini adalah bentuk *inverted index* atau file index dari VSM (*Vector Space Model*) yang digunakan dalam LSI. Selanjutnya pembobotan otomatis dilakukan terhadap *term* berdasarkan frekuensi kemunculan suatu kata dalam dokumen (*term document*) dan frekuensi kemunculan dalam dokumen (*inverse document frequency*) (Melita R *et al*, 2018). Flowchart proses pembobotan TF-IDF dapat dilihat pada Gambar 11.



Gambar 11. Proses Pembobotan TF-IDF

Berdasarkan Gambar diatas maka jika dilihat pada contoh hasil *preprocessing*, hasil pembobotan *term* dengan TF-IDF dapat dilihat pada tabel 5.

Tabel 5. Hasil Pembobotan TF-IDF

Term	D1	D2	DF	IDF+1	W=TF*IDF+1	
					D1	D2
citra	1	1	2	1	1	1
gunung	1	1	2	1	1	1
kidul	0	1	1	1.301029996	0	1.301029996
bromo	1	0	1	1.301029996	1.301029996	0
akuisisi	1	0	1	1.301029996	1.301029996	0
wilayah	0	1	1	1.301029996	0	1.301029996
provinsi	0	1	1	1.301029996	0	1.301029996
yogyakarta	0	1	1	1.301029996	0	1.301029996
tanggal	1	1	2	1	1	1
3	0	1	1	1.301029996	0	1.301029996
26	1	0	1	1.301029996	1.301029996	0
lapan-a2	1	0	1	1.301029996	1.301029996	0
produk	1	1	2	1	1	1
spacecam	1	0	1	1.301029996	1.301029996	0
april	1	1	2	1	1	1
2019	1	1	2	1	1	1
satelit	2	1	3	1.823908741	3.647817482	1.823908741
lapan-a3	0	1	1	1.301029996	0	1.301029996
lisa	0	1	1	1.301029996	0	1.301029996

4.3.3 Pembobotan LSI

Setelah dilakukan pembobotan pada setiap *term* dan pembentukan matriks, pengembangan selanjutnya dapat dilakukan proses dekomposisi matriks. Pada LSI proses tersebut dilakukan menggunakan *Singular Value Decomposition* (SVD) untuk

mendapatkan relasi asosiatif antara *term*. Secara konsep, SVD menerima kumpulan data dengan berdimensi tinggi dan variabel tinggi serta menguranginya ke ruangdimensi yang berukuran lebih rendah (Hasan I, 2018).

Proses perhitungan SVD menghasilkan 3 matriks yaitu U,S dan V^T. Kemudian dilakukan tahap pengurangan dimensi sebanyak k. Dimana penentuan nilai k optimal didapatkan dari hasil percobaan. Sehingga menghasilkan matriks U_k,S_k dan V^T_k.

Adapun langkah-langkah algoritma *singular value decomposition* adalah sebagai berikut:

1. Menghitung matriks AA^T.
2. Tentukan determinan |A-λI|=0, dimana λ adalah *eigenvalue* / akar karakteristik nya dan I adalah matriks Identitas. Hasil determinan digunakan untuk mendapatkan eigenvalue dan singular value yang akan bermanfaat dalam membentuk matriks S.
3. Hitung eigenvector dengan mengevaluasi.
4. Bentuk matriks V dengan menggunakan hasil kalkulasi normalisasi eigenvector sebagai kolom dari matriks sebagai kolom matriks V.
5. Bentuk matriks U dengan U=AVS⁻¹.

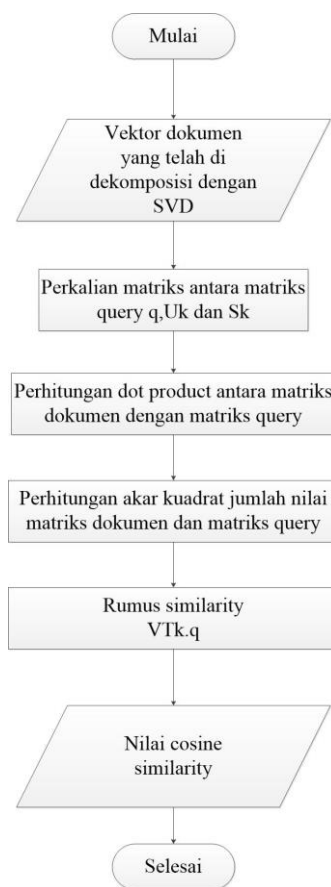
Berikut adalah hasil perhitungan berdasarkan langkah-langkah di atas didapatkan matriks U, S dan V^t yaitu sebesar :

$$\text{Matriks } U_k = \begin{bmatrix} 0.2289 & 0.0532 \\ 0.2289 & 0.0532 \\ 0.1298 & 0.3049 \\ 0.1679 & -0.2356 \\ 0.1679 & -0.2356 \\ 0.1298 & 0.3049 \\ 0.1298 & 0.3049 \\ 0.1298 & 0.3049 \\ 0.2289 & 0.0532 \\ 0.1298 & 0.3049 \\ 0.1679 & -0.2356 \\ 0.1679 & -0.2356 \\ 0.2289 & 0.0532 \\ 0.1679 & -0.2356 \\ 0.2289 & 0.0532 \\ 0.2289 & 0.0532 \\ 0.6530 & -0.233 \\ 0.1298 & 0.3049 \\ 0.1298 & 0.3049 \end{bmatrix} \quad \text{Matriks } S_k = \begin{bmatrix} 0.1631 & 0 \\ 0 & 0.2962 \end{bmatrix} \quad \text{Matriks } V_k^T = \begin{bmatrix} 0.7192 & -0.6114 \\ 0.6114 & 0.7192 \end{bmatrix}$$

4.3.4 Perhitungan Kemiripan Menggunakan *Cosine Similarity*

Setelah dilakukan perhitungan hubungan *semantic* antar *term* dengan menggunakan metode SVD dengan LSI. Selanjutnya hasil berupa vektor *query* dan vektor dokumen citra satelit dihitung kemiripannya menggunakan *cosine similarity*. *Cosine Similarity* digunakan untuk menghitung nilai *cosinus* dengan sudut yang dibentuk antara vektor *query* dan vektor dokumen (Aji R *et al*, 2011).

Dalam penelitian ini yaitu dokumen citra satelit dan *query* yang dimasukkan oleh user yang dipresentasikan dalam bentuk *vector*. Hasil dari proses LSI ini menghasilkan output berupa matriks U_k , S_k , dan V^T_k dengan nilai pengurangan dimensi sebanyak k . Matriks U_k dan S_k digunakan untuk menghitung bobot *query* sedangkan nilai V^T_k digunakan untuk menghitung nilai *cosine similarity*. Flowchart dari perhitungan kemiripan dengan *cosine similarity* dapat dilihat pada Gambar 12.



Gambar 12. Perhitungan kemiripan dengan *cosine similarity*.

Seperti yang sudah diketahui bahwa matriks A berisi sekumpulan n dokumen. Sedangkan matriks V harus berisi n baris, dimana setiap baris berisi koordinat dari sebuah vektor dokumen. Untuk sebuah dokumen, vektor d adalah $d = d^T U S^{-1}$. Dalam LSI, setiap *query* dapat diperlakukan sebagai sebuah dokumen, sehingga vektor q adalah $q = q^T U S^{-1}$. Terkait dengan reduksi dimensi sebesar k , maka vektor dokumen, d , dan vektor *query*, q , dapat dituliskan sebagai :

$$d = d^T U \& k^{-1}$$

$$q = q^T U S_k^{-1}$$

Dengan perlakuan tersebut, maka antara vektor *query*, *q*, dan sebuah vektor dokumen, *d*, dapat dihitung koefisien *similarity* dengan *cosinus* sebagai berikut :

$$\text{sim}(q,d) = \text{sim}(q^T U_k S_k^{-1}, d^T U_k S_k^{-1}).$$

Dengan rumus diatas maka kini dapat dihitung rangking semua dokumen terhadap *query* yang diberikan. Misal *query* yang diberikan adalah “citra yang diakuisisi lapan-a2”. Vektor *query* dapat dibentuk sebagai berikut :

$$q = q^T U_k S_k^{-1}$$

$$q = [11001000000010000000] \begin{bmatrix} 0.2289 & 0.0532 \\ 0.2289 & 0.0532 \\ 0.1298 & 0.3049 \\ 0.1679 & -0.2356 \\ 0.1679 & -0.2356 \\ 0.1298 & 0.3049 \\ 0.1298 & 0.3049 \\ 0.1298 & 0.3049 \\ 0.2289 & 0.0532 \\ 0.1298 & 0.3049 \\ 0.1679 & -0.2356 \\ 0.1679 & -0.2356 \\ 0.2289 & 0.0532 \\ 0.1679 & -0.2356 \\ 0.2289 & 0.0532 \\ 0.2289 & 0.0532 \\ 0.6530 & -0.2332 \\ 0.1298 & 0.3049 \\ 0.1298 & 0.3049 \end{bmatrix} \begin{matrix} \dots\dots\dots 0 \\ \dots\dots\dots 0.2962 \end{matrix} \quad q = [0.1295 \quad -0.1080]$$

Baris dari matriks V^T_k merupakan koordinat dari dokumen , sehingga

$$D_1 = (0.7192, 0.6114)$$

$$D_2 = (-0.6114, 0.7912)$$

Sekarang dapat dihitung hasil kali titik antara vektor *query* yang dipetakandengan masing-masing vektor dokumen. Hasilnya adalah sebagai berikut.

$$D_1 = \frac{(0.1295)(0.7192) + (-0.1080)(0.6114)}{\sqrt{(0.1295)^2 + (-0.1080)^2} \sqrt{(0.7192)^2 + (0.6114)^2}} = 0.9345$$

$$D_2 = \frac{(0.1295)(-0.6114) + (-0.1080)(0.7912)}{\sqrt{(0.1295)^2 + (-0.1080)^2} \sqrt{(0.6114)^2 + (0.7912)^2}} = -0.5411$$

D1 dan D2 merupakan hasil titik vektor *query* dengan vektor dokumen 1, dan dokumen 2 secara berurutan. Dari kedua nilai tersebut maka dapat disimpulkan bahwa urutan relevansi untuk dokumen yang paling relevan adalah

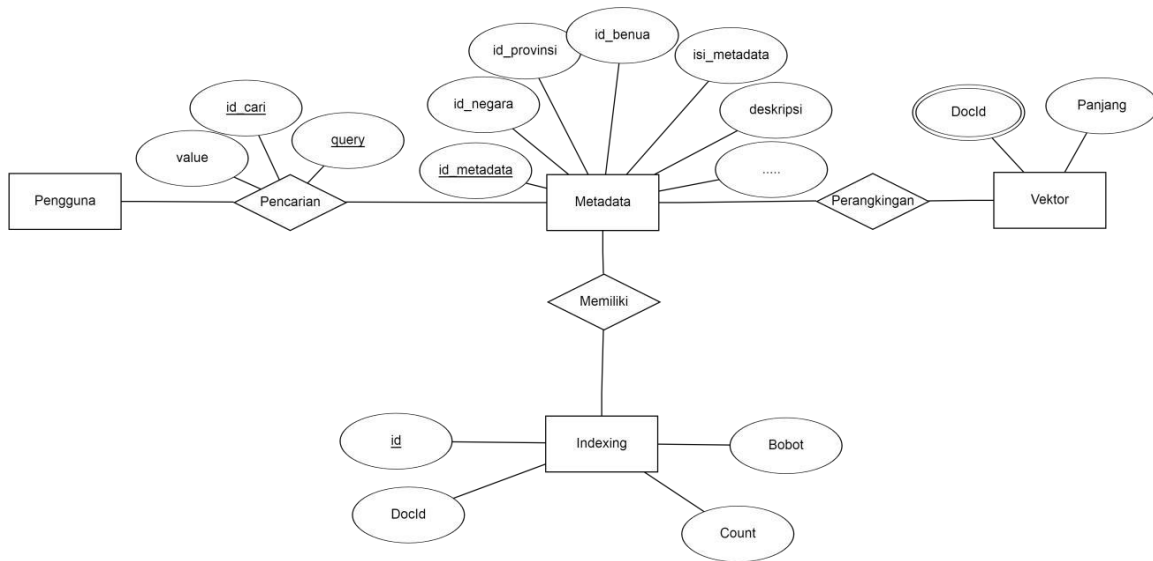
1. Dokumen ke -1
2. Dokumen ke -2

Dokumen ke-1 merupakan dokumen yang paling relevan dengan *query* karena memiliki nilai *cosinus* terbesar atau θ antara *query* dan D2 paling kecil.

4.4 Perancangan Basis Data

4.4.1 Perancangan *Entity Relationship Relation (ERD)*

ERD merupakan suatu model jaringan yang menggunakan susunan data yang disimpan oleh sistem secara abstrak. Tujuan utama adalah menunjukkan obyek data dan *relationship* pada obyek tersebut. Adapun ERD pada sistem ini dapat dilihat pada Gambar 13.



Gambar 13. ERD (*Entity Relationship Diagram*)

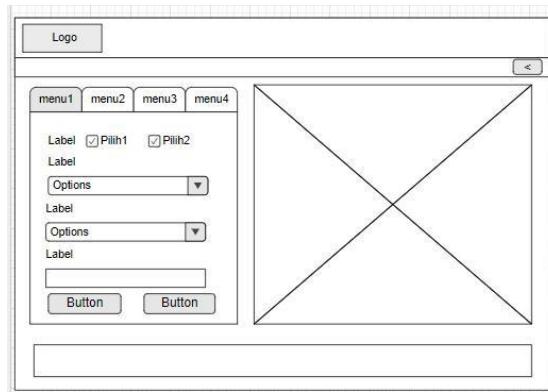
4.4.2 Perancangan Sistem Secara Detail

4.4.2.1 Rancangan Antarmuka

Perancangan antarmuka merupakan media komunikasi antara penggunadengan sistem. Pengguna dapat melakukan interaksi pada sistem sehingga membantu dalam proses menemukan informasi dan penelusuran masalah hingga menemukan solusi. Berikut rancangan antarmuka sistem pencarian citra satelit LAPAN.

1. Rancangan Halaman Utama

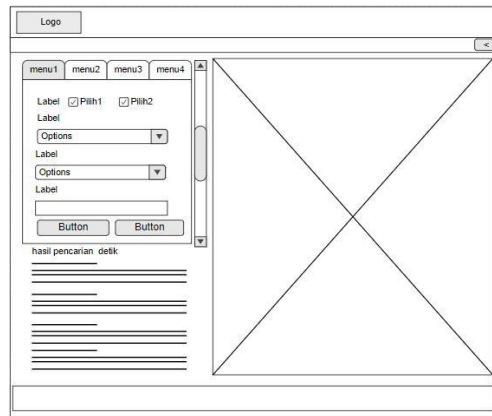
Halaman ini merupakan halaman awal ketika membuka sistem pencarian citra satelit. Pada rancangan halaman awal sistem ini terdapat *form* pencarian dengan tab-menu berisi menu awal pencarian berdasarkan lokasi, menu kedua berisi pencarian berdasarkan tanggal-pengambilan satelit, menu ketiga berisi pencarian berdasarkan produk satelit, dan keempat pencarian berdasarkan titik koordinat latitudedan longitude serta *maps* untuk menunjukkan lokasi citra yang akan ditampilkan. Perancangan halaman utama dapat dilihat pada Gambar 14.



Gambar 14. Rancangan Halaman Utama

2. Rancangan Hasil Pencarian Citra Satelit

Rancangan ini merupakan hasil pencarian citra satelit dari kata kunci yang dimasukkan oleh *user*. Rancangan hasil pencarian ini ditampilkan pada Gambar 15.



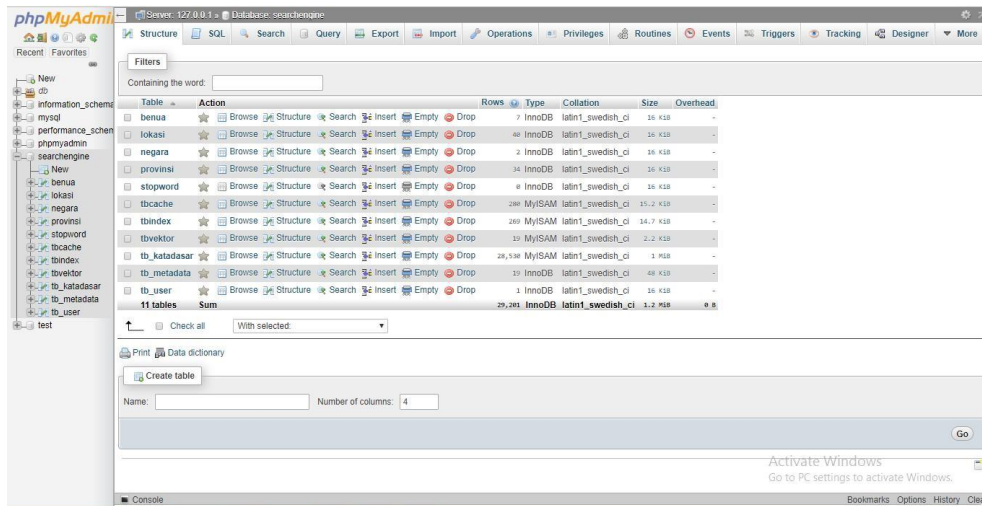
Gambar 15. Rancangan Hasil Pencarian Citra Satelit

4.5 Implementasi

Tahap implementasi merupakan tahap pembangunan sistem yang telah dirancang dengan menggabungkan komponen-komponen yang ada.

4.5.1 Implementasi *Database* Menggunakan *MySQL*

Tahap pembuatan *database* menggunakan *Mysql Server* yang digunakan untuk menyimpan data citra satelit. Tabel yang terdapat pada *database* tersebut diantaranya yaitu tabel negara, tabel benua, tabel provinsi, tabel metadata, tabel index, tabel pencarian, tabel vektor, tabel kata dasar dan tabel user. Implementasi *database* pada sistem ini dapat dilihat pada Gambar 16.



Gambar 16. Implementasi *Database* PHPMYAdmin

4.5.2 Implementasi Sistem

Tahap ini merupakan tahap untuk mengkombinasikan semua komponen yang akan digunakan ke dalam sistem. Tahap pemrosesan teks yang dilakukan yaitu tokenisasi, filterisasi dan stemming. Kemudian tahap *indexing* menggunakan metode *tf-idf*. Setelah itu dilakukan proses *Latent Semantic Indexing* dengan menggunakan matriks *SVD* (*Singular Vector Decomposition*) hingga dihitung kemiripan antara kata kunci yang dimasukkan dengan data citra dalam dokumen pada proses *retrieval*. Sistem dibuat dengan menggunakan bahasa pemrograman *PHP 7.1*. *Mysql Server* dan text editor *Visual Studio Code* serta dijalankan pada server local *XAMPP*.

BAB V

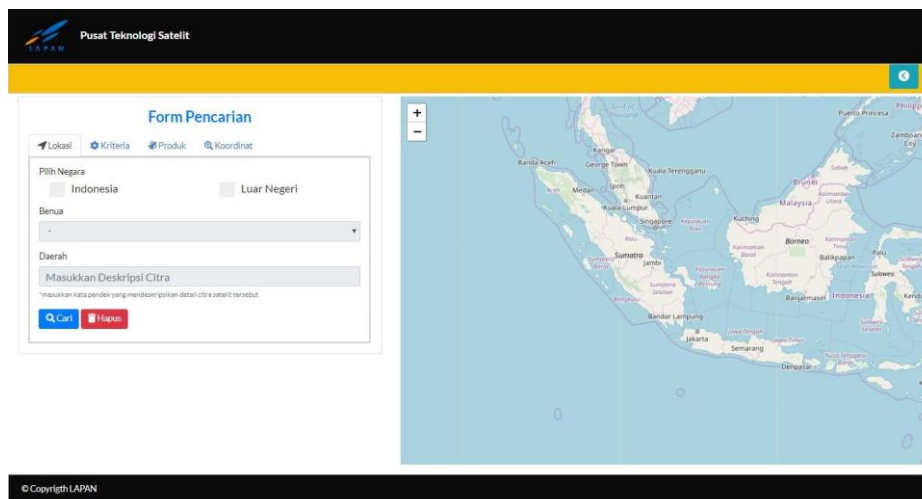
HASIL DAN PEMBAHASAN

5.1 Hasil Penelitian

Pada tahap perancangan dan implementasi sebelumnya sudah dibahas mengenai penerapan algoritma *latent semantic indexing* pada sistem *search engine* citra satelit. Berikut adalah hasil dan pembahasan sistem yang digambarkan pada sub poin dibawah ini:

5.1.1 Halaman Utama

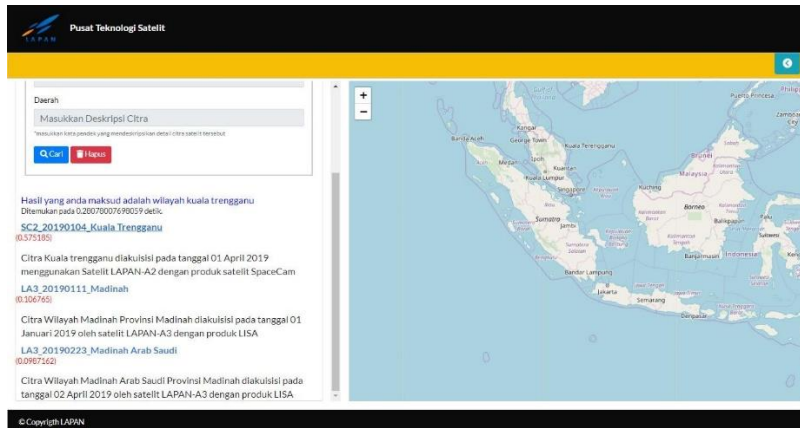
Halaman utama merupakan halaman awal ketika *user* membuka sistem pencarian citra satelit. Pada halaman ini terdapat fasilitas pencarian melalui *form* yang tersedia pada tab menu. *User* dapat mencari metadata citra satelit dengan memasukkan query kata kunci pada menu pencarian berdasarkan lokasi. Tampilan gambar halaman utama dapat dilihat pada Gambar 17.



Gambar 17. Halaman Utama

5.1.2 Halaman Hasil Citra Satelit

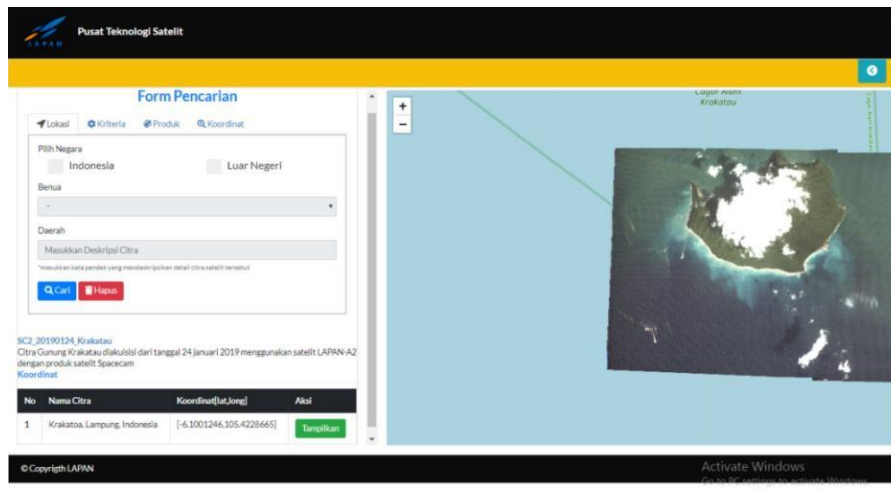
Halaman hasil pencarian citra satelit ini merupakan halaman yang menampilkan data citra satelit yang memiliki nilai kemiripan atau sesuai dengan kata kunci yang dimasukkan oleh *user*. Halaman ini berisi judul metadata citra satelit dan deskripsi detail dari citra satelit yang telah dimasukkan kedalam *database*. Tampilan hasil pencarian citra satelit dapat dilihat pada Gambar 18.



Gambar 18. Hasil Pencarian Citra Satelit

5.1.3 Tampilan Citra Satelit

Tampilan citra satelit dari hasil data yang ditemukan dapat dilihat pada Gambar 19.



Gambar 19. Tampilan Citra Satelit

5.2 Pembahasan Sistem

Pada tahap ini akan dibahas bagaimana cara kerja dari *Information Retrieval System* dengan menerapkan algoritma *Latent Semantic Indexing* pada sistem *search engine* data citra satelit LAPAN. Berikut langkah-langkah pembahasan sistem tersebut.

5.2.1 Penginputan Dokumen Citra Satelit

Pada tahap awal dokumen citra satelit berjumlah 21 dokumen dikumpulkan dan dimasukkan kedalam *database*. Data citra satelit ditambahkan oleh admin pusteksat kedalam *database* untuk nantinya dapat diolah menggunakan *text preprocessing*. Adapun data citra satelit yang berhasil ditambahkan dapat dilihat pada tabel metadata citra satelit dapat dilihat pada Gambar 20.

Show entries Search: +Tambah Data

No	Daerah	Deskripsi	File KML	Aksi
1	Gunung Kidul Yogyakarta	Citra Gunung Ki...	LA3_20190403_Gunung Kidul Yogyakarta.kml	
2	Madinah	Citra Wilayah M...	LA3_20190111_Madinah.kml	
3	Madinah Arab Saudi	Citra Wilayah M...	LA3_20190223_Madinah Arab Saudi.kml	
4	Kuala Trengganu	Citra Kuala tre...	SC2_20190104_Kuala Trengganu.kml	
5	Krakatau	Citra Gunung Kr...	SC2_20190124_Krakatau.kml	
6	Talaud	Citra Wilayah P...	SC2_20190905_Talaud.kml	
7	Nanasi	Citra Nanasi di...	SC2_20190409_Nanasi.kml	

Gambar 20. Metadata Citra Satelit

5.2.2 *Processing* Pada Dokumen dan *Query*

Kemudian setelah data citra satelit terdapat dalam *database*, dilakukan tahap *preprocessing* pada deskripsi dokumen citra satelit dan *query*. Pada tahap awal dengan proses *tokenizing* yaitu untuk memecah kalimat menjadi bentuk *term* dan *filtering* untuk membuang kata *stopword* pada *term* yang dilakukan 1 tahap pada 21 dokumen seperti yang dilakukan oleh Amin F & Purwatiningtas (2015). Proses *stemming* dilakukan untuk menghilangkan imbuhan pada *term* hasil *filtering* dengan menggunakan algoritma Nazief & Adriani Stremmer (2009) dalam waktu komputasi 3detik. Sehingga jumlah *term* yang terdapat hasil *preprocessing* yaitu sebanyak 305 *term*.

5.2.3 Pembobotan TF – IDF

Setelah dilakukan *preprocessing* dilanjutkan dengan tahap *indexing* untuk menghitung jumlah kata (*term*) yang sama dalam dokumen citra satelit. Pada tahap ini dilakukan perhitungan jumlah frekuensi *term* yang terdapat pada dokumen citra satelit dan dilanjutkan untuk pembentukan matriks *term* (Susanto G & Purwanto H.R, 2018). Proses ini dilakukan dengan menekan tombol hitung pada menu *indexing* sehingga hasil *indexing* dari setiap kata (*term*) pada dokumen dapat dilihat pada Gambar 21.

Mengindeks Sebanyak 21 Deskripsi

Show 10 entries Search: Tampilkan

No	Term	Id Dokumen	TF
1	citra	52	1
2	gunung	52	1
3	kidul	52	1
4	wilayah	52	1
5	provinsi	52	1
6	yogyakarta	52	1
7	tanggal	52	1
8	03	52	1
9	april	52	1
10	2019	52	1

Showing 1 to 10 of 305 entries Previous 1 2 3 4 5 ... 31 Next

Gambar 21. *Preprocessing Tabel Index.*

Dari hasil pembuatan matriks *term* pada setiap dokumen, lalu dilakukan proses pembobotan dengan menghitung TF (*Term Frequency*) dan IDF (*Inverse Document Frequency*) pada masing-masing *term*. Adapun hasil perhitungan bobot pada setiap kata (*term*) dalam dokumen ditunjukkan pada Gambar 22.

Terdapat 305 Term Yang Memiliki Bobot

Show 10 entries Search: Tampilkan

No	Id Dokumen	Term	Bobot TF IDF
1	52	citra	1.14613
2	52	gunung	1.8451
3	52	kidul	2.32222
4	52	wilayah	1.41913
5	52	provinsi	1.47712
6	52	yogyakarta	2.32222
7	52	tanggal	1
8	52	03	2.32222
9	52	april	1.14613
10	52	2019	1

Showing 1 to 10 of 305 entries Previous 1 2 3 4 5 ... 31 Next

Gambar 22. Hasil Perhitungan Tabel Bobot TF-IDF

5.2.4 Pembobotan *Latent Semantic Indexing*

Setelah dilakukan pembobotan TF-IDF pada setiap kata (*term*) di dalam dokumen, kemudian dihitung nilai *semantic* dengan pembobotan LSI menggunakan matriks SVD (*Singular Vector Decomposition*). Hasil pembentukan matriks pada dokumen yang telah diberikan bobot, lalu di proses menggunakan SVD yaitu matriks

A. Proses SVD ini menghasilkan 3 matriks yaitu U, S dan V^T lalu dilakukan pengurangan dimensi sebanyak k sehingga hasilnya berupa matriks U_k , S_k dan V_k^T dan disimpan dalam bentuk *file txt* untuk dapat digunakan pada tahap selanjutnya.

Hasil dari proses ini dapat dilihat pada Gambar 23.

The screenshot shows a web interface titled "Proses SVD". It features a search bar, a "Show 10 entries" dropdown, and a table with the following data:

No	Matriks	Aksi
1	Matriks U.txt	
2	Matriks S.txt	
3	Matriks V.txt	
4	Matriks Uk.txt	
5	Matriks Sk.txt	
6	Matriks Vkt.txt	

At the bottom, it says "Showing 1 to 6 of 6 entries" and includes "Previous", "1", and "Next" navigation buttons.

Gambar 23. Hasil Proses SVD Disimpan Dalam *File Txt*

5.2.5 Proses Untuk Menghitung *Cosine Similarity*

Pada halaman utama menu pencarian *user* memasukkan deskripsi pada *form* input kemudian *query* tersebut dilakukan *preprocessing* sama seperti pada halaman admin, setelah itu dilakukan pembobotan dengan TF-IDF kemudian dihitung nilai *similarity* antara *query* tersebut dengan deskripsi pada data dalam dokumen citra satelit. Proses similaritas dilakukan dengan menghitung panjang vektor matriks *query* dengan vektor V_k^T . Panjang vektor didapat dari nilai akar jumlah kuadrat matriks *query* baru dengan bobot matriks V^T . Menurut (Hasan I, 2018) hasil similaritas didapat dengan membagi jumlah perkalian bobot dengan perkalian panjang vektor *query* dengan V^T . Jika *query* tersebut sudah dimasukkan maka nilai similaritas akan ditambahkan pada tabel *cache* di dalam *database* dengan maksud agar hasil pencarian dari *query* tersebut dapat dipanggil kembali. Hasil proses similaritas antara *query* dengan dokumen dapat dilihat pada Gambar 24.

The screenshot shows a web interface titled "Tampilkan Similaritas". It features a search bar, a "Show 10 entries" dropdown, and a table with the following data:

No	Query	Id Document	Nilai Similaritas
1	pulau raya aceh	57	0.183354
2	bromo	61	0.455032
3	krakatau	56	0.433477
4	pulau raya aceh	63	0.67447
5	citra spacecam lapan a2	52	0.165983
6	citra spacecam lapan a2	53	0.187129
7	citra spacecam lapan a2	54	0.173023
8	citra spacecam lapan a2	55	0.447394
9	citra spacecam lapan a2	56	0.437712
10	citra spacecam lapan a2	57	0.405851

At the bottom, it says "Showing 1 to 10 of 312 entries" and includes "Previous", "1", "2", "3", "4", "5", "...", "32", and "Next" navigation buttons.

Gambar 24. Hasil Proses Similaritas

5.2.6 Hasil Pencarian Data Satelit

Pengujian awal dilakukan untuk mencari data citra satelit dengan 1 *term* yang dimasukkan. Misalkan query “Bromo” dimasukkan untuk melihat berapa dokumen relevan yang terambil dan nilai similaritas antara *query* yang dimasukkan dengan dokumen pada citra satelit. Hasil pencarian 1 *term* pada query dapat dilihat pada Gambar 25.



Hasil yang anda maksud adalah bromo
Ditemukan pada 0.26679301261902 detik.
[SC2_20190426_Gunung Bromo](#)
(Similarity: 0.455032)
Citra Gunung Bromo diakuisisi pada tanggal 26 April 2019 oleh satelit LAPAN-A2 dengan produk satelit SpaceCam

Gambar 25. Hasil Pencarian Pada *Query 1 Term*.

Dari hasil pengujian pada 1 *term* dapat ditemukan 1 hasil pencarian dengan dokumen yang relevan dimana nilai similaritas 0.455032 dalam waktu 0.2668 detik. Pengujian berikutnya dilakukan untuk mencari dokumen citra satelit pada 3 sampai 5 *term*. Misalkan query “Citra Gunung Yang Diakuisisi Oleh SpaceCam” dimasukkan maka hasil pencarian pada 3 sampai 5 *term* tersebut ditunjukkan pada Gambar 26.



Gambar 26. Hasil Pencarian Pada 3 Sampai 5 *Term Query*.

Pada pengujian diatas didapatkan berupa 2 hasil dokumen yang terambil dalam waktu 0.4407 detik yaitu dokumen pada urutan ke-1 dengan nilai similaritas 0.52117 dan dokumen pada urutan ke-2 dengan nilai similaritas 0.298745. Dokumen yang relevan dengan *query* yang dimasukkan yaitu dokumen ke-1 dengan nilai similaritas tertinggi, tetapi dokumen ke-2 tetap muncul walaupun bukan termasuk citra gunung. Hal ini terjadi dikarenakan terdapat 3 *term* yang memiliki pola yang sama yaitu citra, akuisisi dan spacecam.

5.3 Uji Coba Sistem

Uji coba sistem merupakan pengujian terhadap aplikasi yang telah dibuat, dengan uji coba ini dapat diketahui kelebihan dan kekurangan aplikasi yang telah dibuat. Selain itu juga dapat diketahui apakah setiap sistem telah berfungsi sesuai dengan tujuannya. Berikut ini adalah uji coba yang telah dilakukan pada sistem.

5.3.1 Uji Coba Struktural

Uji coba struktural ini dilakukan untuk memastikan apakah keadaan program ini terstruktur dengan baik sesuai yang diharapkan. Hal ini bertujuan agar didapatkan hasil yang diinginkan. Jika terjadi kesalahan atau hasil yang didapat tidak sesuai dengan yang diinginkan maka proses akan kembali. Hasil dari uji coba struktural dapat dilihat pada tabel 6.

Tabel 6. Uji Coba Struktural

No	Form/Halaman	Keterangan
1	Halaman Utama	Sesuai
2	Pencarian Berdasarkan Lokasi	Sesuai
3	Pencarian Berdasarkan Tanggal Pengambilan	Sesuai
4	Pencarian Berdasarkan Produk	Sesuai
5	Pencarian Berdasarkan Koordinat	Sesuai
6	Hasil Pencarian Citra Satelit	Sesuai
7	Halaman Login	Sesuai
8	Halaman Dashboard Admin	Sesuai
9	Halaman Metadata Citra Satelit	Sesuai
10	Halaman Tambah Metadata Citra Satelit	Sesuai
11	Halaman Edit Metadata Citra Satelit	Sesuai
12	Halaman Provinsi	Sesuai
13	Halaman Benua	Sesuai
14	Halaman Tambah Provinsi	Sesuai
15	Halaman Edit Provinsi	Sesuai
16	Halaman Tambah Benua	Sesuai
17	Halaman Edit Benua	Sesuai
18	Halaman <i>Indexing</i>	Sesuai
19	Halaman Bobot	Sesuai
20	Halaman Vektor	Sesuai

5.3.2 Uji Coba Fungsional

Uji coba fungsional dilakukan dengan cara mengklik setiap link dan melihat halaman yang akan terbuka. Hasil uji coba fungsional dapat dilihat pada tabel 8.

Tabel 7. Uji Coba Fungsional

No	Fungsi	Hasil yang diharapkan	Keterangan
1	Pencarian	Sistem menampilkan data citra satelit berdasarkan query yang dimasukkan pada kata kunci	Berfungsi
2	Tombol Login	Masuk ke dalam dashboard admin	Berfungsi
3	Tombol Tambah	Admin berhasil menambahkan data	Berfungsi
4	Tombol Edit	Admin berhasil mengubah data	Berfungsi
5	Tombol Hapus	Admin berhasil menghapus data	Berfungsi
6	<i>Tokenizing</i>	Mengubah dokumen menjadi potongan kata (<i>term</i>)	Berfungsi
7	<i>Filtering</i>	Menghapus kata ganti	Berfungsi

5.3.3 Uji Coba Validasi

Tahap uji coba validasi dilakukan untuk mengetahui apakah proses validasi dalam hasil pencarian data citra satelit sudah relevan atau belum. Uji coba ini menggunakan uji coba *precision* dan *recall*.

5.3.3.1 Uji Coba *Precision* dan *Recall*

Uji coba dalam *information retrieval* yaitu menggunakan ukuran *precision* dan *recall*. Pada pengujian ini diukur seberapa besar relevansi dokumen yang terambil apakah sudah relevan atau belum. Proses pengujian dilakukan dengan memasukkan 10 *query* pada masing-masing kawasan yaitu Indonesia atau Luar Negeri. Data dokumen yang diukur yaitu sebesar 21 dokumen citra satelit dengan jumlah *term*

sebanyak 305 *term*. Pengujian *recall* dan *precision* dilakukan dengan cara input *query* ke dalam *Information Retrieval System* input 1 *term*, 2 *term* dan 3 sampai 5 *term*. Berikut hasil pengujian *recall* dan *precision* dengan menguji 1 *term* ditunjukkan pada tabel 8 dan 9.

Tabel 8. Pengujian 1 *term* Pada Wilayah Indonesia

No	Query	Indonesia			Precision	Recall
		Relevan Terambil	Relevan Tidak terambil	Tidak Relevan Terambil	$[a/(a+c)] \times 100\%$	$[a/(a+b)] \times 100\%$
1	bromo	1	0	0	100	100
2	krakatau	1	0	0	100	100
3	kidul	1	0	0	100	100
4	nanasi	1	0	0	100	100
5	suramadu	1	0	0	100	100
6	aceh	1	0	0	100	100
7	pulau	2	0	0	100	100
8	gunung	3	0	0	100	100
9	spacecam	5	0	0	100	100
10	lisa	1	0	0	100	100
Total					1000	1000
MAP					100	100

Tabel 9. Pengujian 1 *Term* Pada Wilayah Luar Negeri

No	Query	Luar Negeri			Precision	Recall
		Relevan Terambil	Relevan Tidak terambil	Tidak Relevan Terambil	$[a/(a+c)] \times 100\%$	$[a/(a+b)] \times 100\%$
1	yunani	1	0	0	100	100
2	gibraltar	1	0	0	100	100
3	madinah	4	0	0	100	100
4	jeddah	1	0	0	100	100
5	kuala	1	0	0	100	100
6	turki	1	0	0	100	100
7	lapan-a2	2	0	9	18.18181818	100
8	lapan-a3	10	0	2	83.33333333	100
9	spacecam	7	0	0	100	100
10	lisa	4	0	0	100	100
Total					901.5151515	1000
MAP					90.15151515	100

Dari hasil pengujian pada tabel diatas untuk dokumen citra wilayah Indonesia yang relevan terambil dapat dilihat dari rata-rata nilai *precision* yaitu 100% dan *recall* 100% sedangkan untuk dokumen citra wilayah Luar Negeri yang relevan terambil dengan nilai rata-rata *precision* 90,15% dan *recall* 100%. Hasil ini diperoleh karena 1 *term* pada *query* yang dimasukkan dengan nama tempat hanya memiliki 1 hasil dokumen yang relevan. Pada *query* “gunung” terdapat lebih dari 1 dokumen yang relevan, dikarenakan *term* tersebut memiliki kesamaan pada 2 dokumen citra satelit. Sedangkan pada *query* produk dan satelit terdapat dokumen yang tidak relevan terambil sehingga menyebabkan nilai *precision* berkurang. Berikutnya pengujian *precision* dan *recall* pada 2 *term* dapat dilihat pada tabel 10 dan tabel 11.

Tabel 10. Pengujian 2 Term pada wilayah Indonesia

No	Query	Indonesia			Precision	Recall
		Relevan Terambil	Relevan Tidak terambil	Tidak Relevan Terambil	$[a/(a+c)] \times 100\%$	$[a/(a+b)] \times 100\%$
1	Gunung Krakatau	1	0	2	33.33333333	100
2	Gunung Bromo	1	0	2	33.33333333	100
3	Gunung Kidul	1	0	2	33.33333333	100
4	Pulau Talaud	1	0	1	50	100
5	Produk LISA	2	0	7	22.22222222	100
6	Produk Spacecam	8	0	2	80	100
7	Citra Gunung	3	0	7	30	100
8	Citra Pulau	3	0	0	100	100
9	Citra Suramadu	1	0	0	100	100
10	Satelit Lapan-a3	2	0	6	25	100
Total					507.2222222	1000
MAP					50.72222222	100

Tabel 11. Pengujian 2 Term pada wilayah Luar Negeri

No	Query	Luar Negeri			Precision	Recall
		Relevan Terambil	Relevan Tidak terambil	Tidak Relevan Terambil	$[a/(a+c)] \times 100\%$	$[a/(a+b)] \times 100\%$
1	Kuala Trengganu	1	0	0	100	100
2	Kozani Yunani	1	0	0	100	100
3	Gibraltar Spanyol	1	0	0	100	100
4	Lagos Nigeria	1	0	0	100	100
5	Citra Madinah	4	0	5	44.44444444	100
6	Citra Jeddah	1	0	9	10	100
7	spacecam lapan-a2	2	0	9	18.18181818	100
8	spacecam lapan-a3	6	0	6	50	100
9	lisa lapan-a3	4	0	11	26.66666667	100
10	akuisisi januari	2	0	10	16.66666667	100
Total					565.959596	1000
MAP					56.5959596	100

Dari hasil pengujian yang dilakukan pada tabel 11 dan tabel 12 dapat dilihat untuk dokumen citra wilayah Indonesia yang relevan terambil memiliki rata-rata nilai *precision* yaitu 50.7% dan *recall* 100% sedangkan untuk dokumen citra wilayah Luar Negeri yang relevan terambil dengan nilai rata-rata *precision* 56,59% dan *recall* 100%. Pada 2 term query yang dimasukkan ternyata dokumen yang tidak relevan banyak terambil pada term “gunung”, “citra”, “spacecam lapan-a2” dan selainnya dapat dilihat pada pengujian. Hal ini terjadi karena term pada deskripsi memiliki pola yang sama pada setiap dokumen, sehingga memiliki nilai similaritas tetapi dengan ukuran kecil. Berikutnya pengujian dilakukan lagi pada 3 sampai 5 term yang dapat dilihat pada tabel 12 dan tabel 13.

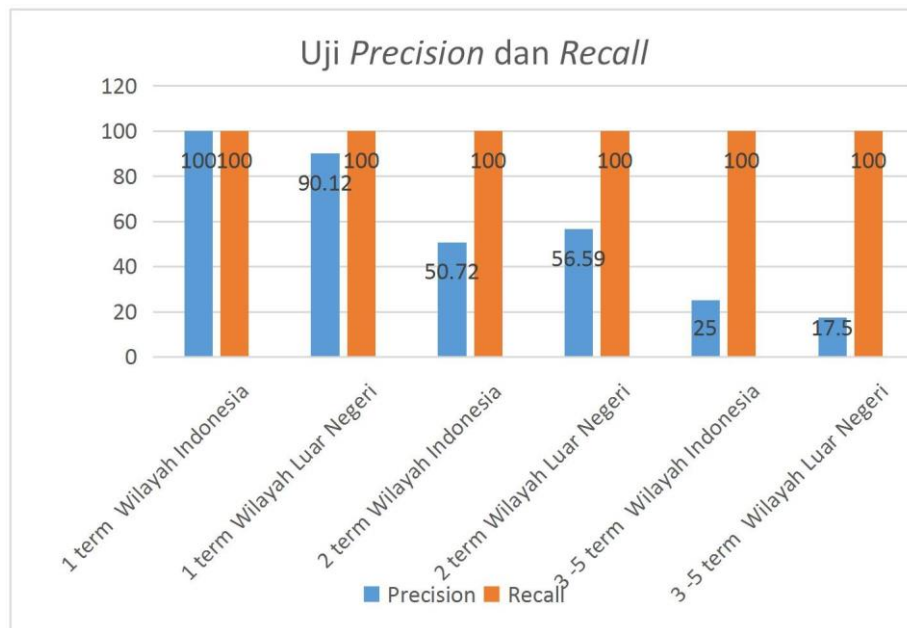
Tabel 12. Pengujian 3 sampai 5 *Term* pada wilayah Indonesia

No	Query	Relevan Terambil	Relevan Tidak Terambil	Tidak Relevan	<i>Precision</i>		<i>Recall</i>	
					$[a/(a+c)] \times 100\%$		$[a/(a+b)] \times 100\%$	
1	Citra Wilayah Nanasi	1	0	7	12.5		100	
2	Citra Gunung Krakatau	1	0	7	12.5		100	
3	Citra Gunung Bromo	1	0	7	12.5		100	
4	Citra Gunung Kidul Yogyakarta	1	0	7	12.5		100	
5	Citra Gunung pada januari	1	0	7	12.5		100	
6	Citra Wilayah suramadu	1	0	7	12.5		100	
7	Citra Gunung yang diakuisisi lapan-a2	2	0	6	25		100	
8	Citra Wilayah oleh produk spacecam pada april 2019	4	0	4	50		100	
9	Citra Wilayah yang diakuisisi oleh spacecam lapan-a2	6	0	2	75		100	
10	Citra yang diakuisisi oleh produk LISA lapan-a3	2	0	6	25		100	
Total					250		1000	
MAP					25		100	

Tabel 13. Pengujian 3 sampai 5 *Term* Pada Wilayah Luar Negeri

No	Query	Relevan Terambil	Relevan Tidak terambil	Tidak Relevan Terambil	<i>Precision</i>		<i>Recall</i>	
					$[a/(a+c)] \times 100\%$		$[a/(a+b)] \times 100\%$	
1	Citra Wilayah Kuala Trengganu	1	0	7	12.5		100	
2	Citra Wilayah Lagos Nigeria	1	0	7	12.5		100	
3	Citra Gibraltar Wilayah Spanyol	1	0	7	12.5		100	
4	Citra Kozani Wilayah Yunani	1	0	7	12.5		100	
5	Citra Wilayah Nigeria	1	0	7	12.5		100	
6	Citra Madinah Pada April 2019	1	0	7	12.5		100	
7	Citra yang diakuisisi lapan-a2 Pada Februari	1	0	10	9.090909091		100	
8	Citra yang diakuisisi oleh produk LISA lapan-a3	4	0	8	33.33333333		100	
9	Citra Wilayah yang diakuisisi oleh spacecam lapan-a2	2	0	10	16.66666667		100	
10	Citra Wilayah oleh produk spacecam Lapan-a3 pada april 2019	5	0	7	41.66666667		100	
Total					175.7575758		1000	
MAP					17.5		100	

Dari hasil pengujian yang dilakukan pada tabel 13 dan tabel 14 dapat dilihat untuk dokumen citra wilayah Indonesia yang relevan terambil memiliki rata-rata nilai *precision* yaitu 25% dan *recall* 100%, sedangkan untuk dokumen citra wilayah Luar Negeri yang relevan terambil dengan nilai rata-rata *precision* 17,5% dan *recall* 100%. Pada pengujian 3 sampai 5 *term* query yang dimasukkan, dokumen yang tidak relevan semakin banyak terambil. Hasil ini diperoleh karena semakin banyak *term* yang memiliki makna atau pola yang sama sehingga mempengaruhi nilai similaritas yang besar mendekati nilai similaritas pada dokumen yang relevan. Berikut grafik perbandingan yang menggambarkan hasil uji coba dengan *precision* dan *recall* pada setiap *term* dapat dilihat pada Gambar 27.



Gambar 27. Grafik Perbandingan Hasil Uji Coba *Precision* Dan *Recall*.

Berdasarkan grafik perbandingan hasil uji coba *precision* dan *recall* pada setiap *term* yang sudah dilakukan dapat terlihat nilai *recall* selalu 100 % dikarenakan dokumen yang relevan selalu terambil pada setiap *term*, sedangkan nilai rata-rata *precision* pada setiap term berbeda-beda. Pada 1 *term* didapatkan rata-rata *precision* 95% , 2 *term* didapatkan *precision* 53,65%, dan 3 sampai 5 *term* didapatkan *precision* sebesar 33,75%. Menurut Kartarina & Junaedi H (2017) Terkadang sebuah sistem memiliki nilai *recall* tinggi tetapi memiliki nilai *precision* rendah dan sebaliknya hal ini dikarenakan pengambilan data relevan oleh sistem tidak seimbang dengan yang manual atau sebaliknya. Maka sebaiknya pengukuran yang dilakukan tidak hanya mengambil salah satu nilai *precision* dan *recall* saja. Nilai *precision* dan *recall* dapat digunakan bersama-sama yang digabungkan dalam *F-Measure* (Winata F & Rainarli E, 2016). Adapun nilai *f-measure* dari pengujian 1 term yaitu 97,4%, pengujian 2 term yaitu 69,83%, sedangkan pada pengujian 3 sampai 5 term diidapatkan nilai sebesar 50,46 %. Dengan hasil diatas maka penerapan algoritma *latent semantic indexing* pada sistem search engine ini sudah baik dalam menemukan dokumen citra satelit yang relevan sesuai dengan kata kunci yang dimasukkan. Namun dikarenakan pola term pada setiap dokumen memiliki banyak jumlah kesamaan, sehingga banyak pula dokumen yang tidak relevan dengan memiliki nilai similaritas tertentu terambil pada hasil pencarian.

BAB IV

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Pencarian citra satelit LAPAN yang dilakukan dengan menerapkan algoritma latent semantic indexing pada pencarian berdasarkan deskripsi lokasi pada citra tersebut. Sistem pencarian ini mempermudah pusteksat dalam menemukan dokumen citra satelit yang ingin ditemukan dengan cepat dan optimal. Data dokumen citra yang dijadikan sebagai *sample* dikumpulkan sebanyak 21 data dokumen citra satelit dengan total 305 kata.

Berdasarkan hasil uji coba *precision* dan *recall* pada setiap *term* yang sudah dilakukan dapat terlihat nilai *recall* selalu 100 % dikarenakan dokumen yang relevan selalu terambil pada setiap *term*, sedangkan nilai rata-rata *precision* pada setiap *term* berbeda-beda. Pada 1 *term* didapatkan rata-rata *precision* 95% , 2 *term* didapatkan *precision* 53.65%, dan 3 sampai 5 *term* didapatkan *precision* sebesar 33.75%. Menurut Kartarina & Junaedi H (2017) Terkadang sebuah sistem memiliki nilai *recall* tinggi tetapi memiliki nilai *precision* rendah dan sebaliknya hal ini dikarenakan pengambilan data relevan oleh sistem tidak seimbang dengan yang manual atau sebaliknya. Maka sebaiknya pengukuran yang dilakukan tidak hanya mengambil salah satu nilai *precision* dan *recall* saja. Nilai *precision* dan *recall* dapat digunakan bersama-sama yang digabungkan dalam *F-Measure* (Winata F & Rainarli E, 2016). Adapun nilai *f-measure* dari pengujian 1 *term* yaitu 97.4%, pengujian 2 *term* yaitu 69.83%, sedangkan pada pengujian 3 sampai 5 *term* diidapatkan nilai sebesar 50.46 %.

Dengan hasil diatas maka penerapan algoritma *latent semantic indexing* pada sistem search engine ini sudah baik dalam menemukan dokumen citra satelit yang relevan sesuai dengan kata kunci yang dimasukkan. Namun dikarenakan pola *term* pada setiap dokumen memiliki banyak jumlah kesamaan, sehingga banyak pula dokumen yang tidak relevan dengan memiliki nilai similaritas tertentu terambil pada hasil pencarian.

6.2 Saran

Pada sistem pencarian citra satelit ini terdapat kekurangan pada kata kamus seperti nama wilayah atau nama daerah dimana diantaranya ada beberapa wilayah yang dianggap sebagai kata imbuhan pada kata depan, samping dan belakang. Selain itu jika terdapat kata yang tidak sesuai atau kata yang salah maka sistem tidak dapat mencari nilai kemiripan dikarenakan tidak ada pengoreksian kata pada sistem tersebut. Untuk pengembangan sistem pencarian ini dapat ditambahkan *spelling* padakata tertentu, kemudian juga menambahkan isi konten pada deskripsi dengan detail.

DAFTAR PUSTAKA

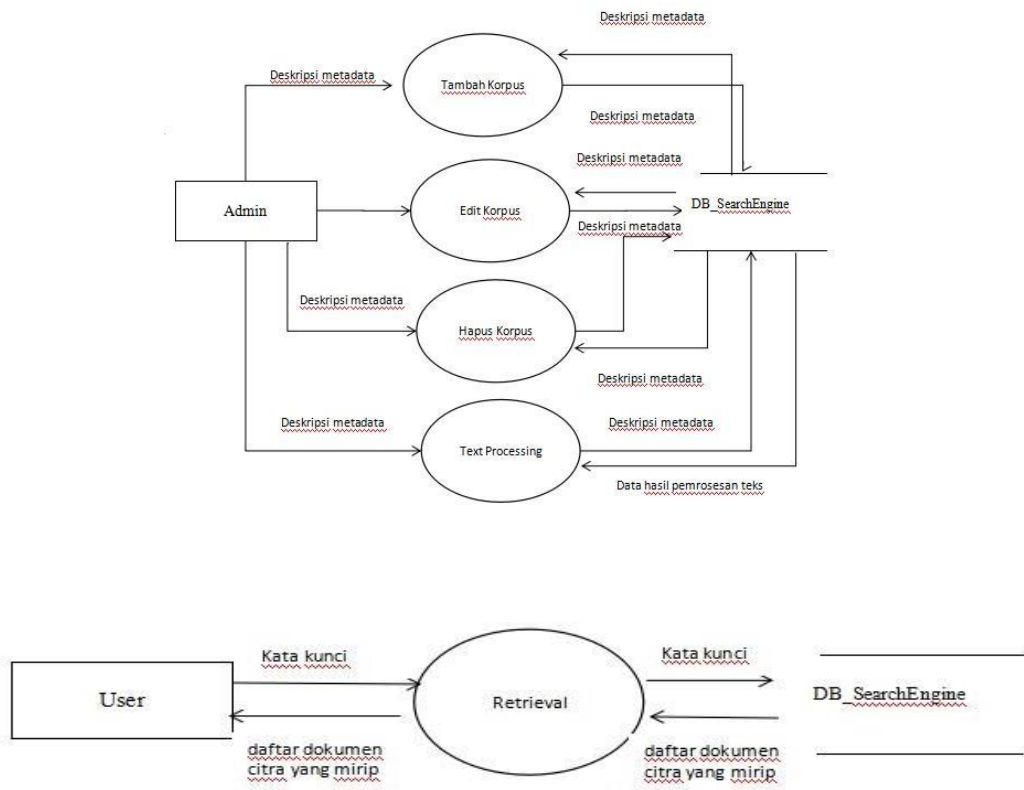
- Selamat, Muhammad Banda, Muhammad Farid Samawi, Zaiduddin, Arniati Massinai. 2015.** Aplikasi Sistem Informasi Geografis Penginderaan Jauh Satelit untuk Evaluasi Pemanfaatan Ruang Budidaya Rumput Laut di Pantai Amal, Kota Tarakan, Kalimantan Utara.
- Baum, Kristen A, Kyle J, Haynes, Forrest P. Dilemuth, James T. Cronin. 2004.** *The Matrix Examines The Effectiveness Of Corridors And Stepping Stones.*
- Prasetyo, Eko. 2012.** *Crowdsourcing System Success For Emergency Relief Operations (Case Study : Haiti).*
- Chen, Chen Tung. 2001.** *A Fuzzy Approach To Select The Location Of The Distribution Center.*
- Ariyanti, G. 2010.** Dekomposisi Nilai Singular dan Aplikasinya.
- Manning, Mark. 2010.** *The Effects Of Subjective Norms On Behaviour In The Theory OF Planned Behaviour : A Meta – Analysis.*
- Alamanda, Rio, Cucu Suhery, Yulrio Brianorman. 2016.** Aplikasi Pendeteksi Plagiat terhadap Karya Tulis Berbasis Web Menggunakan *Natural Language Processing* Dan Algoritma *Knuth Morris Pratt.*
- Wijaya, ET. 2013.** Perancangan Informasi Retrieval (IR) berbasis *Term Frequency Inverse Document Frequency (TF – IDF)* untuk Peringkasan Teks Tugas Khusus Berbahasa Indonesia.
- Wahyuni, RT, D Prastiyanto. 2017.** Penerapan Algoritma Cosine Similarity Dan Pembobotan *TF – IDF* Pada Sistem Klasifikasi Dokumen Skripsi.
- Kartarina, Kartarina, Hartanto Junaedi. 2017.** Information Retrieval dengan Menggunakan Metode Latent Semantic Indexing (LSI) pada Proses Searching dan Klasifikasi Buku: Studi Kasus Perpustakaan STMIK Bumigor.
- Amin, Fatkhul. 2017.** Sistem Temu Kembali Informasi Dengan Metode Vector Space Model.
- Pardede, Jasman, Mira Musrini Barmawi, Wildan Denny Pramono. 2013.** Implementasi Metode *Generalized Vector Space Model* Pada Aplikasi *Information Retrieval.*
- Melita, Ria. 2018.** Penerapan Metode *Term Frequency Inverse Document Frequency (Tf-Idf)* Dan *Cosine Similarity* Pada Sistem Temu Kembali Informasi Untuk Mengetahui Syarah Hadits Berbasis Web (Studi Kasus: Syarah Umdatil Ahkam).
- Kurniawan, Didik. 2010.** Evaluasi Sistem Temu Kembali Informasi Model Ruang *Vector* Dengan Pendekatan *User Judgement.*
- Muham** Sistem temu Kembali Informasi Dalam Dokumen Menggunakan Metode *Latent Semantic Indexing.*

LAMPIRAN

Lampiran 1. DFD

1. DFD Level 1

Pada saat admin menambahkan dokumen citra baru kedalam *database* , maka akan dilakukan pemrosesan teks seperti tokenisasi, filterisasi dan *stemming*. Sedangkan *user* memasukkan kata kunci dalam fasilitas pencarian untuk menelusuri kembali dokumen yang sesuai atau memiliki kemiripan dengan kata kunci yang dimasukkan. Tampilan DFD level 1 dapat dilihat pada Gambar 8.



Lampiran 2. Implementasi Sistem

1. Implementasi Preprocessing

Tahap awal setelah data dimasukkan oleh admin adalah melakukan preprocessing. Pertama dilakukan proses tokenisasi untuk mengubah kalimat pada deskripsi tersebut menjadi potongan kata. Kedua mengubah huruf besar atau kapital pada setiap kata tersebut menjadi huruf kecil dengan fungsi *lowercase*. Ketiga menghapus semua tanda baca. Keempat menghilangkan kata ganti atau kata yang tidak penting. Untuk potongan *script* tahap preprocessing dapat dilihat pada Gambar 1

```
//bersihkan tanda baca, ganti dengan space
global $conn;
$teks = str_replace(" ", " ", $teks);
$teks = str_replace("-", " ", $teks);
$teks = str_replace("'", " ", $teks);
$teks = str_replace("&", " ", $teks);
$teks = str_replace("&quot;", " ", $teks);
$teks = str_replace("/>", " ", $teks);
$teks = str_replace("=", " ", $teks);
$teks = str_replace(".", " ", $teks);
$teks = str_replace(",", " ", $teks);
$teks = str_replace(":", " ", $teks);
$teks = str_replace(";", " ", $teks);
$teks = str_replace("!", " ", $teks);
$teks = str_replace("?", " ", $teks);
$teks = str_replace("^", " ", $teks);
$teks = str_replace("@", " ", $teks);
$teks = str_replace("#", " ", $teks);
$teks = str_replace("~", " ", $teks);
$teks = str_replace("`", " ", $teks);
$teks = str_replace("[", " ", $teks);
$teks = str_replace("]", " ", $teks);
$teks = str_replace("{", " ", $teks);
$teks = str_replace("}", " ", $teks);
$teks = str_replace("<", " ", $teks);
$teks = str_replace(">", " ", $teks);
//ubah ke huruf kecil
$teks = strtolower(trim($teks));
//terapkan stop word removal
require 'vendor/autoload.php';
```

Gambar 1. Script Preprocessing

2. Implementasi Indexing

Setelah tahap preprocessing selanjutnya dilakukan pembobotan dengan menggunakan metode TF-IDF pada term yang terdapat pada setiap dokumen. Untuk potongan *script* tahap indexing dapat dilihat pada Gambar 2.

```
//fungsi hitungbobot, menggunakan pendekatan tf.idf
function hitungbobot() {
    global $conn;
    //berapa jumlah DocId total?, n
    $resn = mysqli_query($conn, "SELECT DISTINCT DocId FROM tbindex");
    $n = mysqli_num_rows($resn);
    //ambil setiap record dalam tabel tbindex
    //hitung bobot untuk setiap Term dalam setiap DocId
    $resBobot = mysqli_query($conn, "SELECT * FROM tbindex ORDER BY Id");
    $num_rows = mysqli_num_rows($resBobot);

    $result="Terdapat $num_rows term yang memiliki bobot";
    echo $result;
    while($rowbobot = mysqli_fetch_array($resBobot)) {
        // $w = tf * log (n/N)
        $term = $rowbobot['Term'];
        $tf = $rowbobot['Count'];
        $id = $rowbobot['Id'];
        //berapa jumlah dokumen yang mengandung term tersebut?, N
        $resNTerm = mysqli_query($conn, "SELECT Count(*) as N FROM tbindex WHERE Term = '$term'");
        $rowNTerm = mysqli_fetch_array($resNTerm);
        $NTerm = $rowNTerm['N'];
        $w = $tf * log($n/$NTerm);
        //update bobot dari term tersebut
        $resUpdateBobot = mysqli_query($conn, "UPDATE tbindex SET Bobot = $w WHERE Id = '$id'");
    } //end while $rowbobot
} //end function hitungbobot
//-----
//fungsi panjangvektor, jarak euclidean
```

Gambar 2. Script Indexing

3. Implementasi LSI

Setelah tahap indexing dilakukan selanjutnya tahap pemrosesan menggunakan *Latent Semantic Indexing*. Dimana pada tahapan ini dilakukan perhitungan matriks SVD pada kata (term) yang terdapat dalam dokumen. Untuk potongan *script* tahap LSI dapat dilihat pada Gambar 3.

```
function panjangvektor() {
    global $conn;
    //hapus isi tabel tbvektor
    mysqli_query($conn,"TRUNCATE TABLE tbvektor");
    //ambil setiap DocId dalam tbindex
    //hitung panjang vektor untuk setiap DocId tersebut
    //simpan ke dalam tabel tbvektor
    $resDocId = mysqli_query($conn,"SELECT DISTINCT DocId FROM tbindex");
    $num_rows = mysqli_num_rows($resDocId);

    $result="Terdapat $num_rows dokumen yang dapat dihitung panjang vektornya";
    echo $result;
    while($rowDocId = mysqli_fetch_array($resDocId)) {
        $docId = $rowDocId['DocId'];
        $resVektor = mysqli_query($conn,"SELECT Bobot FROM tbindex WHERE DocId = $docId");
        //jumlahkan semua bobot kuadrat
        $panjangVektor = 0;
        while($rowVektor = mysqli_fetch_array($resVektor)) {
            $panjangVektor = $panjangVektor + $rowVektor['Bobot'] * $rowVektor['Bobot'];
        }
        //hitung akarnya
        $panjangVektor = sqrt($panjangVektor);
        //masukkan ke dalam tbvektor
        $resInsertVektor = mysqli_query($conn,"INSERT INTO tbvektor (DocId, Panjang) VALUES ($docId, $panjangVektor)");
    } //end while $rowDocId
} //end function panjangvektor
//-----
Ln 214, Col 1 Space
```

Gambar 3. Script LSI

4. Implementasi VSM

Untuk menghitung nilai kemiripan antara query yang dimasukkan pada kata kunci dengan dokumen citra dalam database digunakan model VSM (*Vector Space Model*). Untuk potongan *script Vector Space Model* dapat dilihat pada Gambar 26.

```
//setiap dokumen dalam database (berdasarkan bobot di tbindex)
function hitungsim($query) {
    global $conn;
    //ambil jumlah total dokumen yang telah diindex (tbindex atau tbvektor), n
    $resn = mysqli_query($conn,"SELECT Count(*) as n FROM tbvektor");
    $rown = mysqli_fetch_array($resn);
    $n = $rown['n'];
    //terapkan preprocessing terhadap $query
    $aquery = explode(" ", $query);
    //hitung panjang vektor query
    $panjangQuery = 0;
    $aBobotQuery = array();
    for ($i=0; $i<count($aquery); $i++) {
        //hitung bobot untuk term ke-i pada query, log(n/N);
        //hitung jumlah dokumen yang mengandung term tersebut
        $resNTerm = mysqli_query($conn,"SELECT Count(*) as N from tbindex WHERE Term = '$aquery[$i]'");
        $rownTerm = mysqli_fetch_array($resNTerm);
        $NTerm = $rownTerm['N'];
        $idf = log10($n/$NTerm);
        //simpan di array
        $aBobotQuery[$i] = $idf;
        $panjangQuery = $panjangQuery + $idf * $idf;
    }
    $panjangQuery = sqrt($panjangQuery);
    $jumlahmirip = 0;
    //ambil setiap term dari DocId, bandingkan dengan Query
    $resDocId = mysqli_query($conn,"SELECT * FROM tbvektor ORDER BY DocId");
    while ($rowDocId = mysqli_fetch_array($resDocId)) {
        $dotproduct = 0;
        $docId = $rowDocId['DocId'];
        $panjangDocId = $rowDocId['Panjang'];
        Ln 214, Col 1 Space: 4 UTF-8
```

Gambar 4. Scrip VSM

