Pengaruh *Query Expansion* Terhadap Pendeteksian Kemiripan Teks Menggunakan *Cosine Similarity*

Diajukan untuk Menyusun Tugas Akhir
di Jurusan Teknik Informatika Fakultas Ilmu Komputer Unsri

Oleh :

Pipit Kurnia Sari
NIM : 09021181520025

JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER UNIVERSITAS SRIWIJAYA
2019

DAFTAR ISI

Halaman

DAFTAR TABEL

Halaman

DAFTAR GAMBAR

Halaman

BAB I

PENDAHULUAN

1.1 Pendahuluan

Pada bab ini membahas latar belakang masalah, rumusan masalah, tujuan dan manfaat penelitian serta batasan masalah. Bab ini akan memberikan penjelasan umum mengenai keseluruhan penelitian.

Pendahuluan dimulai dengan penjelasan mengenai tantangan dan tujuan proses menemukan pengetahuan baru pada deteksi kemiripan teks. Serta penelitian yang berkaitan dengan menerapkan *Query Expansion* yang menjadi latar belakang dari penelitian ini.

1.2 Latar belakang

Dalam perkembangan dunia teknologi informasi plagiarisme secara umum mengacu pada penyalinan informasi atau menggandakan karya seseorang yang tidak diketahui sumbernya, seperti dokumen dan program(Muhammad *et al.*, 2017). Serta penggunaan ulang materi miliknya sendiri (dikenal sebagai *self-plagiarism),* dan yang dihasilkan oleh orang lain(Muhammad *et al.*, 2017). Terutama dalam pendidikan tinggi, plagiarisme diakui sebagai masalah yang signifikan dan telah dilaporkan semakin meningkat (Park, 2003). Misalnya, (Citron and Ginsparg, 2015) menganalisis penggunaan kembali teks dalam korpus ilmiah ArXiv.org. Akibatnya, plagiarisme dan pendeteksiannya baru-baru ini menerima perhatian yang signifikan (Boisvert and Irwin, 2006). Sehingga perlu adanya tindakan pendeteksian

kemiripan teks dari karya-karya tulis supaya nilai keaslian dari teks tersebut dapat diketahui (Ryansyah and Andayani, 2017).

Berbagai faktor dapat menandakan plagiarisme, seperti referensi yang salah dan kesamaan yang sama dengan materi yang ada. Secara umum, pendekatan untuk mendeteksi plagiarisme (baik manual atau otomatis) dapat dikategorikan ke dalam dua masalah utama. Deteksi plagiarisme intrinsik berhubungan dengan mengindentifikasi inkonsistensi gaya dalam teks yang menimbulkan pertanyaan tentang kepenulisannya. Detekti plagiarisme ekstrinsik berkaitan dengan mengindentifikasi kemungkinan sumber dari dokumen yang mencurigakan (Stein, Eissen and Potthast, 2007). Maka pada penelitian ini menggunakan deteksi plagiarisme ekstrinsik.

Penelitian mengenai pendeteksian kemiripan teks bahasa Indonesia telah banyak diteliti sebelumnya. Salah satu metode yang digunakan adalah metode *Cosine Similarity.* Pada metode tersebut, didapatkan persentase kemiripan suatu teks berdasarkan perhitungan jumlah kemunculan kata pada teks pembanding (Imbar et al., n.d. 2014). Dalam penelitian lain yang dilakukan oleh (Firdaus, Ernawati and Vatresia, 2014) menjelaskan setiap kata harus diubah menjadi kata dasar terlebih dahulu pada tahap *preprocessing* sebelum melakukan perhitungan kemiripann teks. Hasil pengujian dari percobaan tersebut dihitung kemiripan teksnya menggunakan *Cosine Similarity* didapatkan persentase tingkat akurasi sebesar 87,83%, sedangkan untuk hasil pengujian setelah dilakukan tahap *Praprocessing* persentase tingkat akurasi menjadi 93,81% (Firdaus, Ernawati and Vatresia, 2014).

Salah satu faktor kendala menghitung kesamaan teks adalah ketika kata pada teks uji berbeda dengan kata pada teks yang aslinya. Kata tersebut tidak dapat dihitung. Kegagalan ini disebut ketidakcocokan daftar kata atau *vocabulary mismatch* (Carpineto and Romano, 2012). Untuk menutupi kendala ini, maka diperlukannya suatu metode untuk meningkatkan keefektifan dari ketidakcocokan daftar kata atau ketidakkonsistenan pada pengindeksan dokumen yaitu dengan menerapkan teknik *Query Expansion* menggunakan *Thesaurus*. *Thesaurus* dapat memecahkan masalah ketidakkonsistenan pada pengindeksan dokumen, dan juga dapat digunakan dengan pencarian dalam memformulasi ulang strategi pencarian yang tepat jika diperlukan (Cholifah, Purwanto and Bramanto, 2011). *Thesaurus* akan menyediakan daftar kata yang tepat dan terkontrol yang menunjukkan keterkaitan istilah dan dapat digunakan sebagai alat untuk memperluas *query*, juga dalam mengkoordinasikan pengindeksan maupun pencarian dokumen (Khafajeh, Refai and Yousef, 2013).

Rasyidi, Romadhony and Wibowo, (2013) dalam penelitiannya tentang sistem temu kembali informasi *hadits* menerapkan *Query Expansion* menggunakan *Thesaurus*. Pengujian tingkat akurasi dari hasil penelitian tersebut menggunakan MAP *(Mean Average Precision)* dan *Recall*. Dalam penelitian ini adanya peningkatan performansi sistem temu kembali informasi sebelum dan sesudah menggunakan *Thesaurus* dalam proses *Query Expansion* dengan peningkatan MAP sebesar 34% dan *Recall* sebesar 43%. Penelitian yang dilakukan oleh Muhammad et al. (2017) dalam penelitiannya menerapkan *Query Expansion* menggunakan UMLS Metathesaurus dan *MEDLINE*, pada sistem pendeteksian kemiripan teks.

Pendekatan berbasis IR yang diusulkan di sini yaitu *Cosine Similarity* mencapai hasil yang lebih tinggi daripada pendekatan *Kullback-Leibler Distance*. Penarikan tertinggi yang dicapai dengan metode *Kullback-Leibler Distance* adalah 0,8596 untuk 20 dokumen kandidat teratas, meskipun diharapkan kinerja akan turun ketika seluruh basis data MEDLINE digunakan. Pendekatan yang diusulkan (dengan *query* ekspansi dan WSD) mencapai penarikan sebesar 0,9077 untuk 1 dokumen, yang masih lebih tinggi dari penarikan maksimum yang diperoleh menggunakan metode Kullback-Leibler Distance.

Dari penelitian-penelitian yang telah dijabarkan diatas, sistem pendeteksian kemiripan teks menggunakan teknik *Query Expansion* dapat memberikan hasil tingkat akurasi yang lebih baik dibandingkan dengan sistem pendeteksian kemiripan teks tanpa menggunakan *Query Expansion*. Sehingga sistem pendeteksian kemiripan teks bahasa Indonesia dengan menerapkan *Query Expansion* menggunakan *Thesaurus* diharapkan mampu meningkatkan kinerja sistem pendeteksian kemiripan teks dalam menyajikan informasi lebih cepat dan akurat.

1.3 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah jelaskan, rumusan masalah pada penelitian ini adalah bagaimana pengaruh *Query Expansion* menggunakan *Thesaurus* terhadap tingkat akurasi sebuah sistem pendeteksian kemiripan teks bahasa Indonesia menggunakan *Cosine Similarity* ?

Untuk menjawab rumusan masalah tersebut, dibawah ini diuraikan beberapa *research question* sebagai berikut :

1. Bagaimana mekanisme *Cosine Similarity* dalam sistem pendeteksian kemiripan teks bahasa Indonesia?
2. Bagaimana mekanisme *Query Expansion* menggunakan *Thesaurus* dalam sistem pendeteksian kemiripan teks bahasa Indonesia menggunakan *Cosine Similarity*?
3. Bagaimana hasil persentase kemiripan teks dalam sistem pendeteksian kemiripan teks bahasa Indonesia dengan menerapkan *Cosine Similarity* dan *Query Expansion?*
4. Bagaimana hasil persentase kemiripan teks dalam sistem pendeteksian kemiripan teks bahasa Indonesia menggunakan *Cosine Similarity* tanpa *Query Expansion?*
5. Bagaimana hasil perbandingan *Cosine Similarity* dengan *Query Ekspansion* dan *Cosine Similarity* tanpa *Query Ekspansion* pada sistem pendeteksian kemiripan teks bahasa indonesia berdasarkan dari hasil persentase?

1.4 Tujuan penelitian

Tujuan dari penelitian ini adalah sebagai berikut :

1. Mengetahui mekanisme *Cosine Similarity* dalam sistem pendeteksian kemiripan teks bahasa Indonesia.

2. Mengetahui mekanisme *Query Expansion* menggunakan *Thesaurus* dalam sistem pendeteksian kemiripan teks bahasa Indonesia menggunakan *Cosine Similarity*.
3. Dapat mengetahui pengaruh *Query Expansion* jika tidak diterapkan pada *Cosine Similarity* dengan melihat hasil persentase kemiripan teks terhadap sistem pendeteksian kemiripan teks bahasa Indonesia.
4. Dapat mengetahui pengaruh *Query Expansion* yang diterapkan pada *Cosine Similarity* dengan melihat hasil persentase kemiripan teks terhadap sistem pendeteksian kemiripan teks bahasa Indonesia.
5. Menganalisis hasil perbandingan berdasarkan hasil persentase pada metode *Cosine Similarity* dengan *Query Ekspansion* dan *Cosine Similarity* tanpa *Query Ekspansion* pada sistem pendeteksian kemiripan teks bahasa Indonesia.

1.5 Manfaat penelitian

Adapun manfaat yang diperoleh dalam penelitian ini adalah sebagai berikut :

1. Memahami *Query Expansion* menggunakan *Thesaurus* sebagai metode pendeteksian kemiripan teks bahasa Indonesia.
2. Hasil penelitian dapat dijadikan sebagai rujukan bagi peneliti lain dalam mengembangkan sistem pendeteksian kemiripan teks bahasa Indonesia.

1.6 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut :

1. Data teks yang digunakan berupa teks (huruf) berbahasa Indonesia, Data teks tidak memperhitungkan *equation* (rumus), tabel, simbol dan gambar.
2. data teks tidak menggunakan singkatan, sudah sesuai dengan ejaan yang disempurnakan (EYD) dan kata-katanya sudah baku.
3. Metode pembobotan kata yang digunakan adalah TF-IDF.
4. Metode pendeteksian kemiripan teks yang digunakan adalah *Query Ekspansi* dan *Cosine Similarity.*

1.7 Sistematika Penulisan

Sistematika penulisan skripsi ini adalah sebagai berikut :

BAB I. PENDAHULUAN

Bab I ini menguraikan latar belakang, perumusan masalah, tujuan dan manfaat penelitian serta batasan masalah, dan sistematika penulisan penelitian.

BAB II. KAJIAN LITERATUR

Pada bab II ini akan membahas landasan teori yang digunakan dalam penelitian ini, seperti analisis *preprocessing, query ekspansi*, *thesaurus, term co-occurrence,* analisis metode *cosine similarity, term frequency/inverse document frequency*. Selain itu di bab ini akan dibahas penelitian-penelitian lain yang relevan dengan peneitian ini.

BAB III. METODOLOGI PENELITIAN

Pada bab III ini akan membahas mengenai tahapan yang akan dilaksanakan pada penelitian ini. Pada tahapan penelitian ini akan dibahas dengan rinci dengan

mengacu pada suatu kerangka kerja. Serta pada bagian akhir bab ini akan dijabarkan perancangan manajemen proyek perangkat lunak untuk pelaksanaan penelitian ini.

1.8 Kesimpulan

Pada penelitian ini berfokus pada deteksi plagiarisme ekstrinsik, dan untuk melihat ada tidaknya pengaruh dari menggunakan *Query Ekspansi* menggunakan *Thesaurus* terhadap pendeteksian kemiripan teks bahasa Indonesia. Metode yang digunakan untuk mendeteksi kemiripan teks bahasa Indonesia adalah *Query Ekspansi (Thesaurus)* dan *Cosine Similarity.* Kemudian pengujian akan dilakukan dengan melihat persentase keakurasian hasil kemiripan teks dari sistem pendeteksian kemiripan teks bahasa Indonesia dengan *Query Expansion* menggunakan *Thesaurus*. Kemudian hasil pengujian akan dibandingkan dengan hasil dari yang tidak menerapkan *Query Expansion* menggunakan *Thesaurus.*

BAB II

KAJIAN LITERATUR

2.1 Pendahuluan

Bab I telah menjelaskan bahwa rumusan masalah pada penelitian ini adalah bagaimana pengaruh *Query Expansion* menggunakan *Thesaurus* terhadap tingkat akurasi sebuah sistem pendeteksian kemiripan teks bahasa Indonesia. Maka pada bab ini akan dijabarkan landasan teori untuk memahami fundamental objek penelitian, penulis juga melakukan *literature review* terhadap jurnal, buku dan artikel yang terkait dengan teknik *Query Ekspansi* menggunakan *Thesaurus* pada sistem pengukuran kesamaan dokumen ini.

2.2 Plagiarisme

Plagiarisme atau plagiat adalah tindakan penjiplakan, atau pengandaan, pengambilan karangan, pendapat orang lain dan menjadikannya karangan tersebut seolah-olah milik sendiri tanpa menambahkan sumber referensi (KBBI, 1997: 775). Oleh karena itu dapat dikatakan bahwa tindakan plagiat adalah tindakan yang tidak baik, karena termasuk mencuri karya orang lain. Plagiarisme juga biasanya terjadi pada saat seseorang menggunakan pengulangan materi miliknya sendiri (*self-plagiarism),* serta yang dihasilkan oleh orang lain (Muhammad *et al.*, 2017).

Dijelaskan oleh (Soelistyo, 2011), di bawah ini merupakan tipe-tipe plagiarisme :

1. *Word-for-word plagiarism* : penulis yang langsung menyalin frase atau bagian dari teks yang diterbitkan tanpa diikuti sumbernya.

2. *Paraphrasing plagiarism* : sintaks atau kata-kata yang diubah (ditulis ulang), tetapi masih dapat dikenali dari teks aslinya.
3. *Plagiarism of secondary sources* : ketika penulis menuliskan sumber asli yang direferensikan atau dikutip, tetapi diperoleh dari teks sumber sekunder tanpa melihat sumber aslinya.
4. *Plagiarism of the forms of the sources* : struktur argumen dalam sebuah sumber disalin.
5. *Plagiarism of ideas* : penggunaan kembali pemikiran asli dari teks asli tanpa ketergantungan pada kata-kata atau bentuk dari aslinya.
6. *Plagiarism of authorship* : penulis mengganti nama pemilik asli dengan nama penulis.
7. *Self-Plagiarism* : penulis mendaur ulang karya tulis/ karya ilmiah dan dipublikasikan pada lebih dari satu redaksi publikasi.

Dari penjelasan tipe-tipe plagiarisme diatas pada penelitian ini tipe plagiarisme yang digunakan adalah *Word-for-word plagiarism* dan *Paraphrasing plagiarism.* Terdapat 3 teknik untuk menganalisis plagiarisme berdasarkan penelitian (Stein and zu Eissen, 2006), yaitu: *substring matching*, *keyword similarity* dan *fingerprint analysis.*

a. *Substring matching* – pendekatan ini mencoba untuk mengidentifikasi kesamaan antar string secara keseluruhan, yang kemudian digunakan sebagai indikator dari plagiarisme.

b. *Keyword similarity* – pada pendekatan ini topik utama akan diekstrak dan kemudian mengidentifikasi kata kunci dokumen. Perbandingan dilakukan dengan membandingkan kata kunci dari dokumen A dan dokumen B.
c. *Fingerprint analysis* – cara kerja pendekatan ini adalah dengan menggunakan teknik *hashing* yang mengkonversi setiap string menjadi bilangan.

Berdasarkan penjelasan mengenai teknik analisa plagiarisme, penelitian ini menggunakan teknik analisa *substring matching*. Untuk menentukan jenis plagiarisme antara dokumen yang diuji terdapat 5 jenis penilaian persentase dijabarkan oleh (Mutiara and Agustina, 2008) yaitu:

1. 0%, hasil uji 0% berarti kedua dokumen tersebut benar-benar berbeda baik dari segi isi dan kalimat secara keseluruhan.
2. < 15%, hasil uji 15% berarti kedua dokumen tersebut hanya mempunyai sedikit kesamaan.
3. 15 – 50 %, hasil uji 15-50% berarti menandakan dokumen tersebut termasuk plagiat tingkat sedang.
4. > 50%, hasil uji lebih dari 50% berarti dapat dikatakan bahwa dokumen tersebut mendekati plagiat.
5. 100%, hasil uji 100% menandakan bahwa dokumen tersebut adalah plagiat karena dari awal sampai akhir mempunyai isi yang sama persis.

Sedangkan untuk menyatakan suatu dokumen termasuk plagiarisme digunakan standar umum dengan *threshold* 50% (Rafles, 2013). Dengan kata lain, jika tingkat kemiripan dokumen mencapai atau melebihi 60% maka dokumen tersebut termasuk kedalam kategori plagiat atau plagiarisme.

2.3 Preprocessing

Preprocessing adalah proses awal mengelola data sebelum pengolahan data dilakukan (Vijayarani, Ilamathi and Nitya, 2015) . Tahapan-tahapan yang dilakukan pada *Preprocessing* diantaranya yaitu *casefolding*, *tokenizing*, *Stopword Removal* dan *stemming*.

Kalimat	**:** Algoritma Pemrograman adalah sebuah pengetahuan dan materi.			
Case folding	: algoritma pemrograman adalah sebuah pengetahuan dan materi			
Tokenizing	:			
	algoritma	pemograman	adalah	sebuah
	pengetahuan	dan	materi	
Stopword removal	:			
	algoritma	pemograman		
	pengetahuan	materi		
Stemming	:			
	algoritma	program		
	tahu	materi		

Gambar II-1 Contoh *preprocessing* teks

1) *Casefolding* adalah proses penyamaan case dalam sebuah dokumen, dari huruf besar ke huruf kecil. Hanya huruf ‘a’ sampai dengan ‘z’ yang diterima. Karakter selain huruf dianggap delimiter.
2) *Tokenizing* adalah proses memecah kalimat menjadi kumpulan kata,

3) *Stopword Removal* adalah proses menghapus kata umum yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna, contoh stopwords dalam bahasa Inggris diantaranya ‘of’, ‘the’, sedangkan dalam bahasa Indonesia diantaranya ‘yang’, ‘ke’.
4) *Stemming* adalah proses untuk menemukan kata dasar dari sebuah kata dengan 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 baik dan benar.

2.4 *Thesaurus*

Query Expansion merupakan suatu teknik perluasan *query* dengan menambahkan *keywords* baru ke dalam *query* awal yang dirasa memiliki keterkaitan untuk memperjelas *query* yang memungkinkan untuk terambilnya dokumen berisi informasi relevan sehingga meningkatkan performansi pencarian (Saneifar *et al.*, 2014). Thesaurus termasuk dalam metode *Query Expansion. Thesaurus* atau Tesaurus adalah kosa kata kontrol yang menunjukkan keterkaitan istilah dan dapat digunakan sebagai alat untuk memperluas kueri (Khafajeh, Refai and Yousef, 2013). Terdapat dua cara untuk membangun *Thesaurus* yaitu secara manual dan otomatis (Rahman, Bakar and Sembok, 2010). *Thesaurus* yang dilakukan secara manual dapat memiliki konten yang sangat komprehensif, sehingga membutuhkan banyak waktu dan *resource* dalam pembentukannya.

Untuk domain tertentu, *Thesaurus* yang dibangun secara manual akan meningkatkan performa sistem secara keseluruhan. Di sisi lain, pembangunan *Thesaurus* secara otomatis bisa dilakukan dengan tiga cara: Cara pertama adalah membangun *Thesaurus* dari koleksi dokumen. Cara kedua adalah menggabungkan *Thesaurus* yang ada, dan yang terakhir adalah menangkap informasi dari pengguna (Rahman, Bakar and Sembok, 2010). Dalam beberapa penelitian telah diterapkan pembangunan *Thesaurus* dengan menggabungkan pembangunan secara otomatis dan manual sehingga dihasilkan *Thesaurus* yang baik dengan tidak terlalu membutuhkan banyak waktu dalam proses pembentukannya.

Salah satu metode yang dapat digunakan dalam pembentukan *Thesaurus* secara otomatis adalah dengan analisis *term co-occurrence*. Metode ini berdasarkan pada perhitungan statistik dengan melihat kemunculan beberapa istilah kata secara bersama-sama dalam suatu dokumen untuk menetapkan keterkaitan antar istilah-istilah tersebut. Semakin dekat kehadiran beberapa kata secara bersama-sama, semakin tinggi nilai *co-occurrence*-nya. Langkah-langkah dalam pembentukan *Thesaurus* menurut (Khafajeh, Refai and Yousef, 2013) yaitu :

1. Setelah teks dalam suatu dokumen selesai di *praprocessing* maka selanjutnya dilakukan perhitungan frekuensi *term* pada setiap dokumen, dan menghitung IDF (*Inverse Document Frequency)* dari setiap *term*. Kemudian hasil dari kedua perhitungan tersebut ialah nilai bobot *term* tersebut, seperti persamaan (II-6) sub bab 2.6.
2. Kemudian untuk setiap *term* akan dibuat berpasangan. Cara yang digunakan untuk membentuk pasangan kata (*pair term)* yaitu *window size* kanan dan kiri.

Lei, Tang and Zeng, (2018) dalam penelitiannya menyatakan bahwa banyaknya *window size* dimulai dari 2 sampai 15, semakin kecil *window size* yang digunakan maka hubungan antar *term* semakin berkaitan dan *pair term* yang dihasilkan sedikit. Sedangkan, semakin besar *window size* yang digunakan maka hubungan antar *term* akan kurang atau tidak berkaitan. Jadi pada penelitian ini digunakan *window size* dari 4 sampai 7. Setelah semua *term* telah berpasangan, setiap pasangan *term* atau *pair term* akan dihitung frekuensi kemunculan bersamanya dilihat dari frekuensi terkecil dari kedua *term* tersebut. Maka bobot pasangan kedua *term* tersebut dihitung dengan mengalikan *term* frekuensi pasangan *term* dan IDF dari pasangan *term* tersebut, seperti persamaan (II-1).

$$Wijk = tfijk \; x \; log \frac{N}{dfjk} \qquad \text{(II-1)}$$

Keterangan :

W_{ijk} = bobot *term j* dan *term k* terhadap dokumen *i.*

tf_{ijk} = frekuensi terkecil dari *term j* dan *term k* pada dokumen *i.*

df_{jk} = jumlah dokumen yang mengandung *term j* dan *term k.*

N = jumlah dari keseluruhan dokumen yang ada.

3. Untuk menentukan nilai kemiripan dari pasangan antar *term* atau *pair term*, maka dapat dilakukan perhitungan seperti persamaan (II-2) berikut.

$$ClusterWeight\left(t_j,\ t_k\right) = \frac{\sum_{i=1}^{n} W_{ijk}}{\sum_{i=1}^{n} W_{ij}} \; x \; WeightingFactor(t_k) \qquad \text{(II-2)}$$

Keterangan :

cluster weight = nilai kemiripan antara *term j* dan *term k* .

W_{ijk} = bobot *term j* dan *term k* pada dokumen *i.*

W_{ij} = bobot *term j* pada dokumen *i.*

Sedangkan, *weighting factor* didapat dengan persamaan (II-3).

$$WeightingFactor(t_k) = \frac{\log \frac{N}{df_k}}{\log N} \quad \text{(II-3)}$$

Keterangan :

N = jumlah dokumen keseluruhan

df_k = jumlah dokumen yang mengandung *term k.*

4. Dari hasil perhitungan *cluster weight* tersebut akan di *filter* untuk mendapatkan pasangan *term* yang dianggap mirip berdasarkan persamaan (II-4) dimana x merupakan nilai *cluster weight. Term* dengan bobot kesamaan > 0 akan dipilih menjadi kandidat *Thesaurus*.

$$x > 0 \quad \text{(II-4)}$$

Berikut contoh perhitungan dari pembentukan *Thesaurus :*

Dokumen 1 : Algoritma Pemograman adalah sebuah pengetahuan dan materi ....

Hasil preprocessing : algoritma program tahu materi

1. Dilakukan perhitungan pembobotan Tf-Idf dengan menggunakan persamaan (II-6) pada sub bab 2.6. Didapatkan hasil perhitungan seperti berikut :

   Tabel II-1. Hasil Perhitungan Pembobotan Tf-Idf

No	All term	W_{ij}
1.	Algoritma	0.875

2.	Program	0.875
3.	Tahu	0.6020
4.	materi	1.204
:	:	:

2. Kemudian dilakukan pembentukan pair term dan dihitung pembobotan term untuk *pair term* menggunakan persamaan (II-1) pada sub 2.5.

Tabel II-2. Hasil Perhitungan Pembobotan *Pair Term*

No	All term	W_{ijk}
1.	(algoritma, program)	0.875
2.	(algoritma, tahu)	0.6020
3.	(algoritma, materi)	0,903
4.	(program, algoritma)	0,87
5.	(program, tahu)	0.6020
6.	(program, materi)	1.204
7.	(tahu, algoritma)	0.6020
8.	(tahu, program)	0.6020
9.	(tahu, materi)	0.6020
10.	(materi, program)	1.204
:	:	:

3. Setelah didapatkan hasil dari perhitungan pembobotan *pair term.* Dilakukan perhitungan *weight factor* menggunakan rumus pada persamaan (II-3).

   Tabel II-3. Hasil Perhitungan *Weight Factor*

No	All term	Weight Factor
1.	(algoritma, program)	0.2076
2.	(algoritma, tahu)	1
3.	(algoritma, materi)	1
4.	(program, algoritma)	0.2076
5.	(program, tahu)	1
6.	(program, materi)	1
7.	(tahu, algoritma)	0.2076
8.	(tahu, program)	0.2076
9.	(tahu, materi)	1
10.	(materi, program)	0.2076
:	:	:

4. Kemudian dilakukan perhitungan *cluster weight* menggunakan rumus pada persamaan (II-2).

   Tabel II-4. Hasil Perhitungan Cluster Weight

No	All term	Cluster Weight
1.	(algoritma, program)	0.2076
2.	(algoritma, tahu)	0.688

3.	(algoritma, materi)	1,032
4.	(program, algoritma)	0.2076
5.	(program, tahu)	0.688
6.	(program, materi)	1,376
7.	(tahu, algoritma)	0.2076
8.	(tahu, program)	0.2076
9.	(tahu, materi)	1
10.	(materi, program)	0.2076
:	:	:

5. Setelah itu dari hasil perhitungan *cluster weight* tersebut akan di *filter* berdasarkan ketentuan dari persamaan (II-4), dimana x > 0 jadi hasil *cluster weight* diatas dipilih semua.

2.5 *Cosine Similarity*

Salah satu metode yang umumnya digunakan untuk menghitung kemiripan antar dokumen adalah ukuran kemiripan (*cosine similarity*). Metode ukuran kemiripan berfungsi untuk menghitung nilai kosinus sudut antara dua vektor. Semakin besar sudut antara dua vektor dari dokumen yang berbeda didalam suatu model ruang vektor maka semakin kecil tingkat kemiripan antara dua dokumen tersebut dan sebaliknya. Perhitungan kemiripan untuk vektor dokumen dan vektor *query* didapat dari hasil kali vektor dokumen dengan vektor *query* dalam 1 ruang vektor dimana didapatkan nilai kosinus yang merupakan hasil kemiripan dari kedua vektor tersebut. Untuk

menentukan nilai kemiripan dokumen dengan *query* masukan, dilakukan perhitungan menggunakan rumus seperti dibawah ini.

$$similarity = \cos(\theta) = \frac{A \,.\, B}{\|A\| \, \|B\|} = \frac{\sum_{i=1}^{n} A_i \, x \, B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \; x \sqrt{\sum_{i=1}^{n} (B_i)^2}} \qquad \text{(II-5)}$$

Keterangan:

A = jumlah kemunculan kata indeks ke-n dari daftar kata pada kalimat A.

B = jumlah kemunculan kata indeks ke-n dari daftar kata pada kalimat B

Perhitungan dari rumus *Cosine Similarity* akan digunakan pada perhitungan kemiripan teks asli dengan teks pembanding, dimana digunakan persamaan (II-5) seperti berikut :

Tabel II-5. Hasil Perhitungan *Cosine Similarity*

Term		algoritma	pegang	peran	bidang	materi	program	...
Tf	D Asli	3	0	0	0	2	3	...
	D Uji	28	1	1	1	33	24	...

$$\sum_{i=1}^{n} A_i \, x \, B_i = (3 \, x \, 28) + (\, 0 \, x \, 1) + (0 \, x \, 1) + (0 \, x \, 1) + (2 \, x \, 33) + (3 \, x \, 24) + \cdots$$

$$= 555$$

$$\sqrt{\sum_{i=1}^{n} (A_i)^2} = \sqrt{(3^2 + 0^2 + 0^2 + 0^2 + 2^2 + 3^2 + \cdots)}$$

$$= 7{,}0710678$$

$$\sqrt{\sum_{i=1}^{n} (B_i)^2} = \sqrt{(28^2 + 0^2 + 0^2 + 0^2 + 33^2 + 24^2 + \cdots)}$$

= 87,58425

$$similarity = \frac{555}{7{,}0710678\ x\ 87.58245} = 0{,}8962$$

2.6 *Term Frequency -Inverse Document Frequeny*

Metode *tf-idf* merupakan suatu metode pembobotan kata dengan cara memberikan bobot hubungan suatu *term* terhadap dokumen untuk mengindikasikan ada atau tidaknya suatu *term* pada suatu dokumen yang digunakan untuk pembanding dokumen (Purwarianti and Yusliani, 2012). *Term frequency* (*tf*) berfungsi untuk mengetahui frekuensi kemunculan sebuah *term* didalam sebuah dokumen. Semakin tinggi tingkat kemuculan suatu *term* didalam suatu dokumen maka akan semakin penting *term* tersebut. Sedangkan *Inverse document frequency* (*idf*) merupakan teknik pembobotan *term* yang berfungsi mengetahui frekuensi kemunculan suatu *term* didalam sekumpulan dokumen yang digunakan (Purwarianti and Yusliani, 2012). Kesesuaian dokumen sangat tergantung pada nilai *tf* dan *idf* dimana semakin besar nilai *tf* maka tingkat kecocokan dokumen semakin besar dan semakin besar nilai *idf* pada suatu *term* maka semakin penting *term* tersebut. Pada penelitian ini, metode *tf-idf* digunakan pada perhitungan analisis *co-occurrence* dan pencarian dokumen jawaban. Untuk menghitung nilai bobot dari *term j* digunakan persamaan (II-6).

$$W_{ij} = tf_{ij}\ x\ idf$$

$$W_{ij} = tf_{ij}\ x\ \log\frac{N}{df_j} \qquad \text{(II-6)}$$

Keterangan:

W_{ij} = bobot *term j* terhadap dokumen *i*.

tf_{ij} = frekuensi *term j* pada dokumen *i*.

df_j = jumlah dokumen yang mengandung *term j*.

N = jumlah dari keseluruhan dokumen yang ada.

Dilakukan perhitungan pembobotan Tf-Idf dengan menggunakan persamaan (II-6) pada sub bab 2.6. Didapatkan hasil perhitungan seperti berikut :

Tabel II-6. Hasil Perhitungan Pembobotan Tf-Idf

No	All term	W_{ij}
1.	Algoritma	0.875
2.	Program	0.875
3.	Tahu	0.6020
4.	materi	1.204

2.7 Penelitian Lain Yang Relevan

Pada bagian ini memuat landasan teori dan beberapa penelitian yang telah dilakukan oleh peneliti sebelumnya. Hal ini dibuat untuk memperkuat penalaran dan rasionalitas keterlibatan sejumlah variabel pada penelitian ini. Selain itu juga difungsikan sebagai pendapat ilmiah yang dipadukan dengan hasil kajian pustaka untuk membangun kerangka berpikir peneliti dalam kaitannya dengan masalah yang sedang diteliti.

2.7.1 Muhammad *et al.*, (2017) : An IR *Base*d *Approach Utilising Query Expansion for Plagiarism Detection in MEDLINE*, **Department Computational Science, COMSATS Institute of Information Technology , and University of Sheffield, Defence Road, Off Raiwind Road, Lahore, Pakistan, Regent Court, 211 Portobello, Sheffield, S1 4DP United Kingdom.**

Pada penelitian yang dilakukan oleh (Muhammad et al., 2017) dalam penelitiannya menerapkan *Query Expansion* menggunakan UMLS Metathesaurus dan *MEDLINE Corpus* pada sistem pendeteksian kemiripan teks. Pendekatan berbasis IR yang diusulkan di sini yaitu *Cosine Similarity* mencapai hasil yang lebih tinggi daripada pendekatan *Kullback-Leibler Distance*. Penarikan tertinggi yang dicapai dengan metode *Kullback-Leibler Distance* adalah 0,8596 untuk 20 dokumen kandidat teratas, meskipun diharapkan kinerja akan turun ketika seluruh basis data MEDLINE digunakan. Pendekatan yang diusulkan (dengan *query* ekspansi dan WSD) mencapai penarikan sebesar 0,9077 untuk 1 dokumen, yang masih lebih tinggi dari penarikan maksimum yang diperoleh menggunakan metode Kullback-Leibler Distance.

2.7.2 Sholikah *et al.*, (2017) : *Co-occurrence Technique And Dictionary Based Method For Indonesian Thesaurus Construction,* **Informatics Department, Faculty of Information Technology and Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia.**

(Sholikah et al., 2017) pada penelitiannya tentang konstruksi *Thesaurus* otomatis bahasa Indonesia menggunakan teknik *co-occurrence* dan metode berbasis kamus Indonesia untuk mengeskstrak istilah terkait sebagai proses *Query Expansion* untuk meningkatkan kinerja sistem temu kembali informasi artikel berita online bahasa indonesia. Untuk menghasilkan kandidat *term Thesaurus* menggunakan teknik *co-occurrence,* dokumen artikel berita terlebh dahulu di-praproses menggunakan *normalization, tokenization, stopword removal* dan *stemming.* Setelah itu dilakukan perhitungan dengan teknik *co-occurrence*, dan hasilnya akan didapat kandidat relasi *term* 1 untuk *Thesaurus*. Untuk menghasilkan kandidat term *Thesaurus* menggunakan metode berbasis kamus, dokumen artikel berita terlebih dahulu di-praposes menggunakan *normalization, tokenization, stopword removal* dan *stemming*. Kemudian dilakukan pembobotan pada setiap *term* dan perhitungan *cosine similarity* untuk menemukan kemiripan antar *term,* dan hasilnya akan didapat kandidat relasi *term* 2 untuk *Thesaurus*. Istilah kandidat *term* yang terkait untuk *query expansion* diambil dari kedua metode tersebut dengan menggunakan operator gabungan, maka hasil dari gabungan kedua kandidat *term* tersebut akan menghasilkan *Thesaurus* Akhir. Pengujian dilakukan dengan menggunakan *Precision* dan *Recall* dengan membandingkan sistem sebelum dan sesudah menggunakan *Query Expansion* adanya peningkatan akurasi dengan nilai

Precision dari 51,33% menjadi 54,00% serta nilai *Recall* dari 80,22% menjadi 84,42%.

2.7.3 Rasyidi, I., Romadhony, A. & Wibowo, A. T. (2013) : *Indonesian Hadith Retrieval System using thesaurus*. Computer, Control, Informatics and Its Applications (IC3INA), International Conference on, 2013. IEEE, 285-288.

Rasyidi et al. (2013) pada penelitiannya tentang sistem temu kembali informasi pada Hadist Bahasa Indonesia menggunakan *Thesaurus* dalam proses *Query Expansion*. *Thesaurus* dalam format digital di dibagun dengan 2 tahap yaitu a) *Thesaurus* dibagun secara otomatis menggunakan analisis *co-coccurence*, dimana range nilai kemiripan dari hasil analisis *co-occurrence* diatas 0,5 akan dijadikan kandidat *term Thesaurus*. b) setelah *Thesaurus* selesai maka keabsahannya akan divalidasi secara manual berdasarkan buku Tesaurus Bahasa Indonesia, Kamus Bahasa Indonesia (KBBI), serta situs menyedia kamus, Tesaurus dan glossary bahasa Indonesia. *Thesaurus* berisi sinonim atau istilah yang berkaitan dengan *query* yang diajukan dan digunakan dalam pengindeksan serta proses pencarian. Pengujian dilakukan dengan menggunakan *Mean Average Precision* (MAP) dan *Recall* dengan membandingkan sistem sebelum menggunakan *Query Expansion* memberikan nilai MAP sebesar 0,70 serta nilai *Recall 0,57* dan sesudah menggunakan *Query Expansion* memberikan nilai MAP sebesar 1 serta nilai *Recall 0,83*. Dalam penelitian ini adanya peningkatan performansi sistem temu kembali

informasi setelah menggunakan *Thesaurus* dalam proses *Query Expansion* dengan meningkatan MAP sebesar 34% dan Recall sebesar 43%.

2.8 Kesimpulan

Berdasarkan uraian di atas, *co-occurence technique* adalah salah satu tahapan dalam yang akan dilakukan pada penelitian ini. Metode pada pendeteksian kemiripan teks yang digunakan adalah *Query Expansion* menggunakan *Thesaurus* yang akan dilakukan dengan cara menganalisis co-occurence term dan Algoritma *Cosine Similarity*, akan dilakukan proses pembobotan kata dengan TF-IDF. Pada penelitian ini juga menjelasan landasan teori dan hasil dari penelitian yang relevan.

BAB III

METODOLOGI PENELITIAN

3.1 Pendahuluan

Bab ini menjelaskan unit penelitian, tahapan-tahapan penelitian yang akan diimplementasikan, metodologi penelitian, serta penjadwalan penelitian. Tahapan penelitian dijadikan sebagai acuan pada setiap fase pengembangan, memberikan sebuah solusi untuk rumusan masalah dan mencapai tujuan penelitian. Pada akhir bab ini berisi perancangan manajemen proyek pada pelaksanaan penelitian.

3.2 Pengumpulan Data

3.2.1 Jenis dan Sumber Data

Jenis data yang digunakan sebagai objek penelitian adalah data sekunder berupa file teks atau dokumen berbahasa Indonesia yang diperoleh dari penelitian sebelumnya yaitu dalam tugas akhir mahasiswa Fakultas Ilmu Komputer Universitas Sriwijaya, Yeri Saputra tahun 2017. Sehingga dapat digunakan sebagai data masukan pada saat percobaan perhitungan kemiripan teks. Dokumen yang digunakan sebagai data masukan merupakan dokumen tanpa angka dan tanpa gambar yang disusun menjadi beberapa paragraf.

3.2.2 Metode Pengumpulan Data

Data didapatkan secara manual dengan mengunduh berkas atau mengopi data dari penelitian sebelumnya yaitu dalam tugas akhir yang disusun oleh mahasiswa Fakultas Ilmu Komputer Universitas Sriwijaya, Yeri saputra tahun 2017. Kemudian

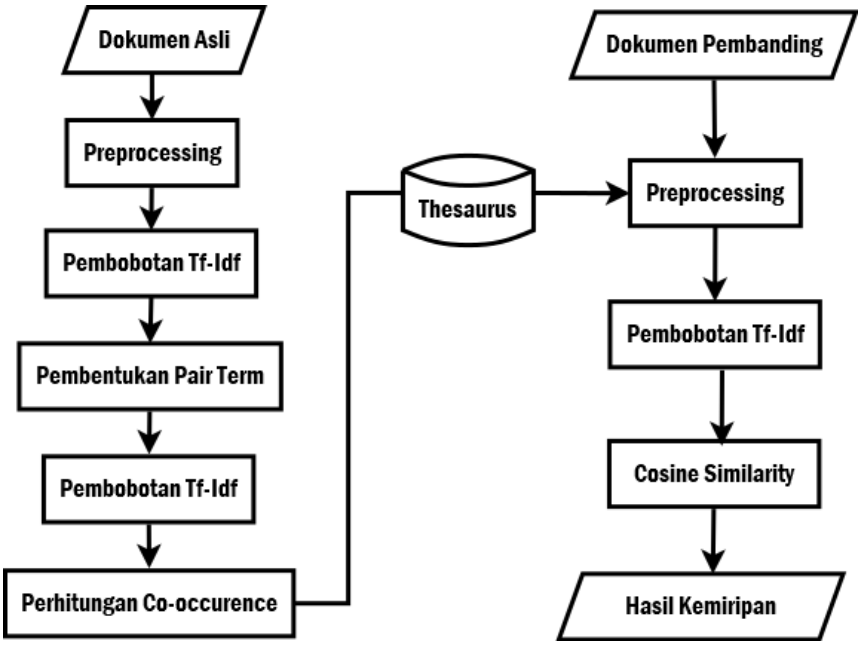
data tersebut disimpan ke dalam berkas bereksistensi *.txt*. Penelitian ini menggunakan dokumen sebanyak 114 dokumen, dengan 19 dokumen sebagai dokumen sumber atau dokumen asli dan 95 dokumen sebagai dokumen uji atau dokumen pembanding.

3.3 Tahap Penelitian

Untuk mencapai tujuan dari penelitian, maka penelitian ini akan dilakukan dengan tahapan-tahapan yang akan dijelaskan dalam subbab 3.3.1 sampai dengan 3.3.6.

3.3.1 Menetapkan Kerangka Kerja / Framework

Kerangka Kerja dari sistem dapat dilihat pada gambar III-1 yang mengilustrasikan proses pengujian dari sistem yang akan dikembangkan.



Gambar III-1. Diagram Tahapan Perangkat Lunak

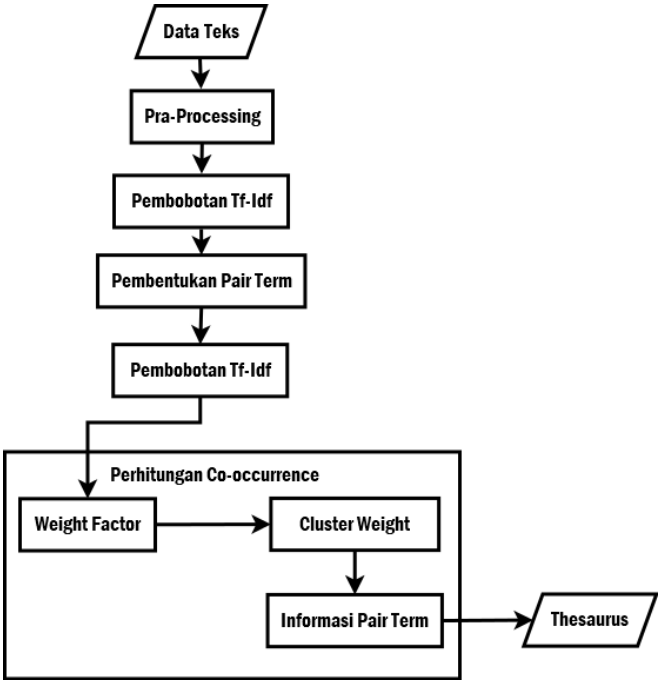
a. Preprocessing

Tahapan *preprocessing* merupakan tahapan awal pengolahan data. Pada tahapan ini dokumen asli dan dokumen uji akan dilakukan empat tahap *preprocessing* yaitu *case folding* untuk penyamarataan karakter pada kalimat pertanyaan masukan berupa perubahan huruf kecil dan penghapusan karakter selain huruf, *tokenizing* untuk mengenali batas-batas antara kata-kata, *stopwords removal* untuk menghapus kata-kata yang sering muncul dan tidak mengandung makna penting pada kalimat masukan yang dapat diketahui dengan menggunakan daftar berisi semua stopwordslist dan *stemming* menggunakan library Sastrawi untuk mengubah kata yang memiliki imbuhan menjadi kata dasar.

b. *Query Expansion* dan Pembentukan *Thesaurus*

Setelah dilakukan praproses pada dokumen uji akan diperluas atau tambah query dengan menerapkan *query expansion* menggunakan *Thesaurus*. Pada penelitian ini metode didasarkan pada analisis terhadap kemunculan pasangan kata (*co-word*) dalam kumpulan dokumen. Dalam penelitian ini pembangunan *Thesaurus* dibagi menjadi dua tahapan: a) pembangunan *Therausus* secara otomatis dengan menerapkan analisis *term co-occurrence*. b) Setelah itu *Thesaurus* yang dihasilkan secara otomatis dikembangkan secara manual. Untuk membangun *Thesaurus* otomatis memperhitungkan nilai *similarity* atau kemiripan *term* dari seluruh dokumen menggunakan analisis *co-occurrence*. Range nilai kemiripan dari hasil analisis *co-occurrence* adalah antara $\chi > 0$. Dimana χ merupakan nilai hasil dari perhitungan kemiripan *term* pada tahapan teknik *co-occurrence.*. Setelah

dilakukan pembangunan secara otomatis, data *Thesaurus* dikembangkan secara manual dengan melakukan validasi/filtering terhadap hasil *Thesaurus* sebelumnya. Kata-kata yang tidak terlalu memiliki keterkaitan dengan kata asal akan dihilangkan, lalu kemudian kata yang memiliki keterkaitan dekat dengan kata asal namun belum terdaftar akan dimasukan. Pembangunan *Thesaurus* manual merujuk pada http://kateglo.bahtera.org sebagai situs rujukan kamus, *thesaurus*, dan glosarium bahasa Indonesia. Perhitungan *Thesaurus* telah dijelaskan pada bab II subbab 2.4.



Gambar III-2. Diagram Tahapan Proses Metode *Query Ekpansi*(*Thesaurus)*

c. Pembobotan *TF-IDF*

Setelah dilakukan perluasan *query* menggunakan *Thesaurus*, selanjutnya dokumen uji yang telah diperluas dengan query dan dokumen asli akan diberi nilai atau bobot. Metode *tf-idf* merupakan suatu metode pembobotan kata dengan cara

memberikan bobot hubungan suatu *term* terhadap dokumen untuk mengindikasikan ada atau tidaknya suatu *term* pada suatu dokumen yang digunakan untuk pembanding dokumen (Purwarianti and Yusliani, 2012). *Term frequency* (*tf*) berfungsi untuk mengetahui frekuensi kemunculan sebuah *term* didalam sebuah dokumen. Semakin tinggi tingkat kemuculan suatu *term* didalam suatu dokumen maka akan semakin penting *term* tersebut. Sedangkan *Inverse document frequency* (*idf*) merupakan teknik pembobotan *term* yang berfungsi mengetahui frekuensi kemunculan suatu *term* didalam sekumpulan dokumen yang digunakan (Purwarianti and Yusliani, 2012). Kesesuaian dokumen sangat tergantung pada nilai *tf* dan *idf* dimana semakin besar nilai *tf* maka tingkat kecocokan dokumen semakin besar dan semakin besar nilai *idf* pada suatu *term* maka semakin penting *term* tersebut.

d. Perhitungan Kemiripan Teks menggunakan *Cosine Similarity*

Hasil pembobotan pada dokumen dan *query* menggunakan metode TF/IDF akan dilakukan proses perhitungan kemiripan dokumen dan *query* menggunakan metode *Cosine Similarity*. Tahap pertama, hitung jarak dua vektor antara dokumen dan *query*. Sebelum menghitung jarak antara dokumen dan query, harus dicari terlebih dahulu total bobot pada *query*. Setelah itu, akan dicari nilai *dot product*. Tahap terakhir, *Cosine Similarity* melakukan perhitungan kemiripan pada nilai vektor dokumen dan *query* yang telah didapatkan. *dot product* dibagi jarak antara tiap dokumen dan query. Proses ini akan terus diulang sesuai jumlah dokumen(n) yang ada. Dari definisi *Cosine Similarity* yaitu jika kedua sudut vektor tersebut

mendekati sudut 0 derajat maka kedua vektor tersebut akan semakin mirip. Kedua sudut vektor yang dimaksud yaitu nilai vektor dokumen dan nilai vektor *query*.

3.3.2 Menetapkan Kriteria Pengujian

Dalam pengujian penelitian ini, terdapat beberapa kriteria yang perlu didefinisikan, yaitu teknik *Query Expansion* yang digunakan, data dan jumlah data yang digunakan. Untuk mengetahui pengaruh maka pengujian ini dilakukan dengan cara melihat hasil dari deteksi kemiripan teks bahasa Indonesia tanpa menggunakan *query expansion* dan hasil deteksi kemiripan teks bahasa Indonesia menggunakan *query expansion* menggunakan *Thesaurus.*

Data yang dijadikan bahan penelitian, ialah dokumen yang berfungsi sebagai sumber sistem pendeteksian kemiripan teks bahasa Indonesia dengan format teks (*.txt). Dokumen yang digunakan sebanyak 114 dokumen, dengan 19 dokumen sebagai dokumen asli atau dokumen sumber dan 95 dokumen pembanding atau dokumen uji dengan masing-masing 5 dokumen dari dokumen asli dengan rincian sebagai berikut :

1. Dokumen dengan 100% sama dari dokumen aslinya
2. Dokumen dengan 80% sama dari dokumen aslinya
3. Dokumen dengan 60% sama dari dokumen aslinya
4. Dokumen dengan 40% sama dari dokumen aslinya
5. Dokumen dengan 20% sama dari dokumen aslinya

3.3.3 Menetapkan Format Data Pengujian

Hasil pengujian pendeteksian kemiripan teks bahasa Indonesia akan digambarkan dalam tabel III-1.

Tabel III-1. Rancangan Hasil Pendeteksian Kemiripan Teks

No	Teks 1	Teks 2	*Query Ekspansi*	Tanpa *Query Ekspansi*
			Similarity (%)	Similarity (%)
1	Dokumen A	100%		
2		80%		
3		60%		
4		40%		
5		20%		
: :	: :	: :	: :	: :
1	Dokumen S	100%		
2		800%		
3		60%		
4		40%		
5		20%		

Keterangan :

Teks 1 : Dokumen Asli

Teks 2 : Dokumen Uji

3.3.4 Menentukan Alat yang digunakan dalam Pelaksanaan Penelitian

Untuk mengetahui penelitian mengenai pengaruh *Query Ekspansi* menggunakan *Thesaurus* terhadap hasil akurasi pendeteksian kemiripan teks bahasa Indonesia. Oleh karena itu, penulis akan mengembangkan sebuah perangkat lunak sistem pendeteksian kemiripan teks bahasa Indonesia dengan menerapkan *Query Ekspansi* menggunakan *Thesaurus* yang hasilnya dapat dibandingkan dengan sistem pendeteksian kemiripan teks bahasa Indonesia tanpa menggunakan *Query Ekspansi*. Perangkat keras yang digunakan dalam penelitian ini adalah *Laptop* dengan spesifikasi sebagai berikut :

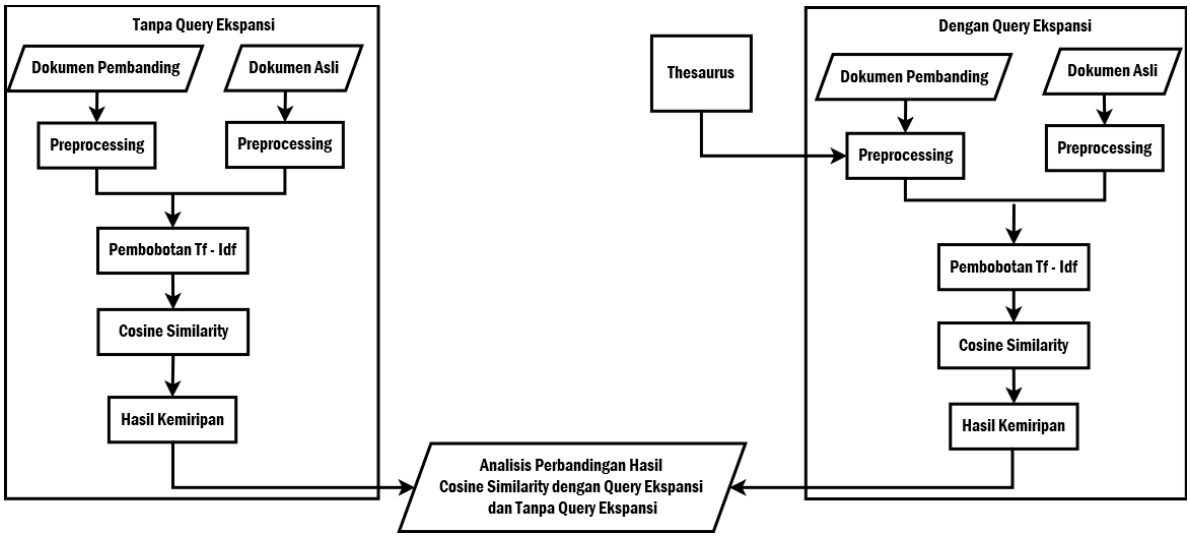
1. *Processor* : Intel(R) Core(TM) i7-4720HQ CPU @ 2.60GHz
2. Memory RAM : 8 GB
3. *Harddisk* : 1 TB

Perangkat lunak yang digunakan pada penelitian ini adalah sebagai berikut :

1. Sistem Operasi Windows 10 Ultimate 64-bit
2. Netbeans
3. *Java Development Kit* (JDK)

3.3.5 Melakukan Pengujian Penelitian

Tahapan pengujian yang akan dilakukan pada penelitian ini dapat dilihat pada Gambar III-3 di bawah ini.



Gambar III-3. Diagram Tahapan Pengujian

3.3.6 Melakukan Analisis Hasil Pengujian dan Membuat Kesimpulan

Untuk mengetahui pengaruh *Query Ekspansi* menggunakan *Thesaurus* terhadap hasil akurasi pendeteksian kemiripan teks dengan *Query Ekspansi* menggunakan *Thesaurus*, maka analisa hasil pengujian dilakukan dengan menganalisa persentase akurasi menggunakan sampel data teks yang sudah dideteksi kemiripannya secara manual. Selanjutnya berdasarkan (Gusmita *et al.*, 2014) persentase dihitung dengan persamaan (III-1).

$$Persentase = \frac{hasil\ kemiripan\ dari\ sistem}{hasil\ kemiripan\ yang\ sudah\ ditetapkan}\ x\ 100\%$$

Data teks yang terdeteksi plagiat yang dihasilkan oleh sistem dicocokan dengan sampel data teks manual, data teks yang dihasilkan oleh sistem dianggap plagiat

jika data teks tersebut sesuai dengan sampel data teks manual. Setelah hasil analisis klasifikasi didapatkan, pada bab V nanti akan ditulis kesimpulan mengenai hasil analisis tersebut.

3.4 Metode Pengembangan Perangkat Lunak

Metodologi yang diterapkan dalam pengembangan perangkat lunak sebagai alat penelitian berorientasi pada objek menggunakan metode *Rational Unified Process* (RUP). Secara umum, langkah-langkah yang akan dilakukan pada pengembangan perangkat lunak adalah fase insepsi, elaborasi, konstruksi, dan transisi (Pressman and Maxim, 2005)

3.4.1 Fase Insepsi

Pada tahapan pemodelan bisnis, penulis menentukan *user requirements* dan fungsionalitas atau fitur-fitur yang dibutuhkan pada perangkat lunak. Pada tahapan pengumpulan kebutuhan, penulis mengumpulkan data penelitian berupa data teks berbahasa Indonesia, kamus dasar bahasa Indonesia dan kamus *stopword*. Pada tahap analisis dan desain, penulis membuat diagram *usecase*. Pada tahap implementasi, penulis mendokumentasikan *user requirements,* fungsionalitas perangkat lunak dan diagram usecase. Pada tahap pengujian, penulis memastikan apakah *user requirements* dan fungsionalitas perangkat lunak sudah valid.

3.4.2 Fase Elaborasi

Pada tahapan pemodelan bisnis, penulis menentukan arsitektur perangkat lunak, desain basis data, dan desain antar muka sesuai dengan *user requirements* dan fungsionalitas perangkat lunak yang telah didapatkan. Penulis dapat melengkapi *user requirement*, apabila dirasa belum lengkap, pada tahap pengumpulan kebutuhan. *Activity diagram* dan *sequence diagram* dibuat pada tahap analisis dan desain. Penulis menyusun dokumentasi yang memuat arsitektur perangkat lunak, desain basis data, desain antar muka, *activity diagram*, dan sequence diagram pada tahap implementasi lalu memastikan seluruhnya sudah valid pada tahap pengujian.

3.4.3 Fase Konstruksi

Pada tahapan pemodelan bisnis, penulis menentukan kelas-kelas yang dibutuhkan pada perangkat lunak. Pada tahap pengumpulan kebutuhan, ditentukan bahasa pemrograman yang digunakan untuk mengembangkan perangkat lunak, yaitu Java. Kebutuhan lain dalam proses pengembangan perangkat lunak juga diidentifikasi, seperti perangkat keras dengan *Processor* Intel(R) Core(TM) i7-4720HQ CPU @ 2.60GHz, RAM 8 GB, Harddisk 1 TB, plantuml, dan Netbeans. Class diagram dibuat pada tahap analisis dan desain. Pada tahapan implementasi, penulis mengembangkan perangkat lunak dengan mengimplementasi kelas-kelas yang telah ditentukan ke kode program dalam bahasa Java. Selanjutnya, penulis melakukan *unit testing* terdapat perangkat lunak yang telah dikembangkan.

3.4.4 Fase Transisi

Pada tahapan pemodelan bisnis, penulis membuat rencana atau skenario pengujian terhadap perangkat lunak. Penulis menentukan *tools* pengujian yang diperlukan dan dokumen yang akan diuji pada sistem pendeteksian kemiripan teks di tahap pengumpulan kebutuhan. *Tools* pengujian merupakan perangkat keras yang sama saat digunakan untuk pengembangan perangkat lunak yaitu laptop dengan *Processor* Intel(R) Core(TM) i7-4720HQ CPU @ 2.60GHz, RAM 8 GB, Harddisk 1 TB . Penulis lalu mendesain tabel skenario pengujian pada tahap analisis dan desain. Pada tahapan implementasi, penulis melakukan pengujian terhadap perangkat lunak berdasarkan skenario atau rencana pengujian. Skenario pengujian dan deployment diagram ditinjau ulang pada tahapan pengujian.

3.5 Manajemen Proyek Penelitian

Manajemen Proyek merupakan perencanaan aktivitas penelitian dari tahap inisialisasi masalah sampai dengan pada tahap kesimpulan dari penelitian. Adapun kegiatan-kegiatan yang berlangsung selama penelitian dapat dilihat dalam Work Breakdown Structure (WBS) yang tertera pada Tabel III-2, dan Gantt Chart pada Gambar III-3, Gambar III-4, Gambar III-5, Gambar III-6, Gambar III-7, Gambar III-8, dan Gambar III-9.

Tabel III-2 Tabel Penjadwalan Penelitian dalam Bentuk *Work Breakdown Structure* (WBS)

Id	*Task Name*	*Duration*	Start	Finish	*Predecessors*
1	**Pengaruh *Query Ekspansi* Terhadap Pendeteksian Kemiripan Teks Menggunakan *Cosine Similarity***	210 days	Mon 08/10/18	Fri 26/07/19	
2	**Menentukan Ruang Lingkup dan Unit Penelitian**	50 days	Mon 08/10/18	Fri 14/12/18	
3	Menentukan masalah penelitian	15 days	Mon 08/10/18	Fri 26/10/18	
4	Membuat latar belakang dan rumusan masalah	15 days	Mon 29/10/18	Fri 16/11/18	3
5	Menentukan tujuan dan manfaat penelitian	8 days	Mon 19/11/18	Wed 28/11/18	4
6	Menentukan batasan masalah	7 days	Thu 29/11/18	Fri 07/12/18	5
7	Menentukan unit penelitian	5 days	Mon 10/12/18	Fri 14/12/18	6
8	Tersedia dokumen hasil tahapan penelitian	1 day	Mon 17/12/18	Mon 17/12/18	7
9	**Menentukan Dasar Teori yang Berkaitan dengan Penelitian**	26 days	Mon 17/12/18	Mon 21/01/19	
10	Mengumpulkan jurnal, paper, dan literatur ilmiah yang berkaitan dengan penelitian	15 days	Mon 17/12/18	Fri 04/01/19	3
11	Mempelajari metode *Query Ekspansi* dan *Cosine Similarity*	10 days	Mon 07/01/19	Fri 18/01/19	10
12	Tersedia dokumen hasil tahapan penelitian	1 day	Mon 21/01/19	Mon 21/01/19	11

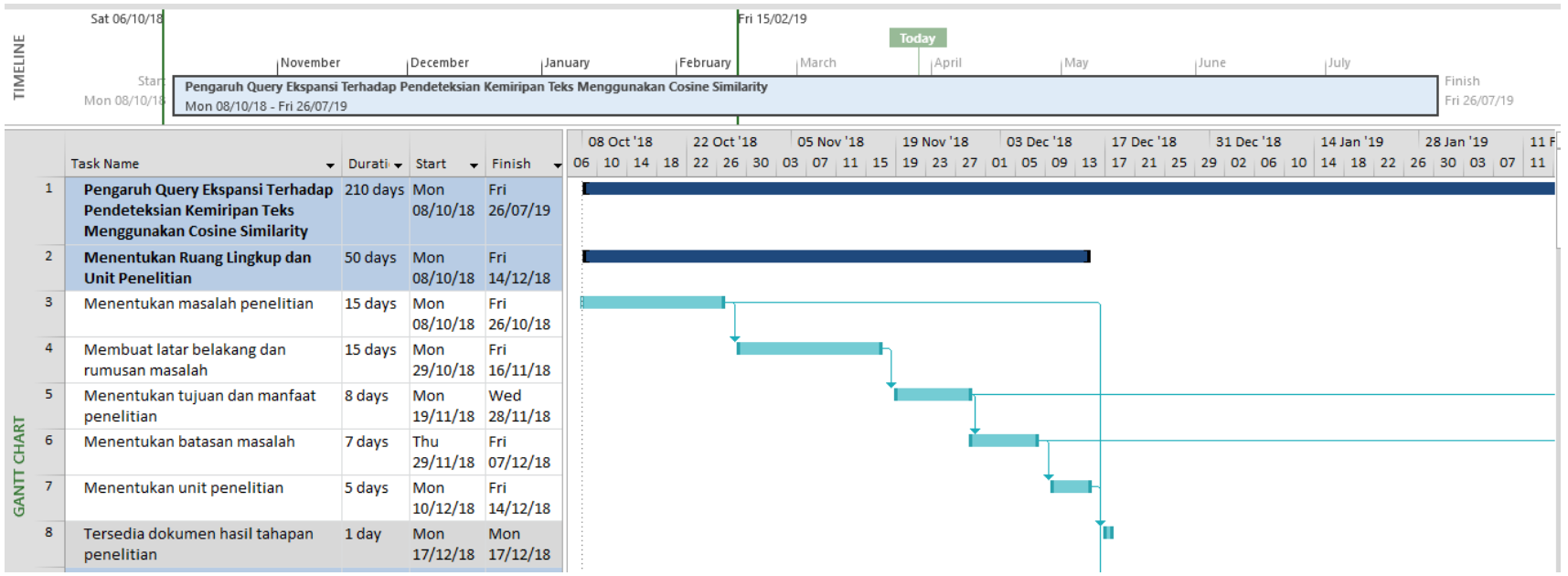
13	**Menentukan Kriteria Pengujian**	30 days	Tue 22/01/19	Mon 04/03/19	
14	Menentukan teknik yang digunakan untuk pembobotan kata	10 days	Tue 22/01/19	Mon 04/02/19	10
15	Menentukan tahapan praproses yang akan dilakukan	10 days	Tue 05/02/19	Mon 18/02/19	14
16	Menentukan teknik yang digunakan untuk pembentukkan *pair term*	8 days	Tue 19/02/19	Thu 28/02/19	15
17	Tersedia dokumen hasil tahapan penelitian	2 days	Fri 01/03/19	Mon 04/03/19	16
18	**Menentukan Alat yang Digunakan untuk Pelaksanaan Penelitian**	90 days	Tue 05/03/19	Mon 08/07/19	
19	***Inception***	31 days	Tue 05/02/19	Tue 19/03/19	
20	***Business Modelling***	7 days	Tue 05/03/19	Wed 13/03/19	
21	Menentukan user requirements dan fungsionalitas perangkat lunak	7 days	Tue 05/03/19	Wed 13/03/19	6
22	***Requirements***	7 days	Thu 14/03/19	Fri 22/03/19	
23	Mengumpulkan data set penelitian	7 days	Thu 14/03/19	Fri 22/03/19	6
24	***Construction***	7 days	Mon 25/03/19	Tue 02/04/19	
25	Membuat use case diagram	7 days	Mon 25/03/19	Tue 02/04/19	21
26	***Implementation***	7 days	Wed 03/04/19	Thu 11/04/19	

27	Membuat dokumentasi	7 days	Wed 03/04/19	Thu 11/04/19	21
28	***Testing***	3 days	Fri 12/04/19	Tue 16/04/19	
29	Memastikan user requirements dan fungsionalitas sudah valid	3 days	Fri 12/04/19	Tue 16/04/19	27
30	***Elaboration***	10 days	Wed 17/04/19	Tue 30/04/19	
31	***Business Modelling***	2 days	Wed 17/04/19	Thu 18/04/19	
32	Menentukan arsitektur perangkat lunak, desain basis data, dan desain antarmuka	2 days	Wed 17/04/19	Thu 18/04/19	28
33	***Requirements***	2 days	Fri 19/04/19	Mon 22/04/19	
34	Melengkapi user requirements yang telah didefinisikan di fase inception	2 days	Fri 19/04/19	Mon 22/04/19	21
35	***Analysis & Design***	3 days	Tue 23/04/19	Thu 25/04/19	
36	Membuat activity dan sequence diagram	3 days	Tue 23/04/19	Thu 25/04/19	32
37	***Implementation***	2 days	Fri 26/04/19	Mon 29/04/19	
38	Membuat dokumentasi	2 days	Fri 26/04/19	Mon 29/04/19	34
39	***Testing***	1 day	Tue 30/04/19	Tue 30/04/19	
40	Memastikan arsitektur perangkat lunak, desain basis data, dan desain antarmuka sudah valid	1 day	Tue 30/04/19	Tue 30/04/19	36
41	***Construction***	35 days	Wed 01/05/19	Tue 18/06/19	

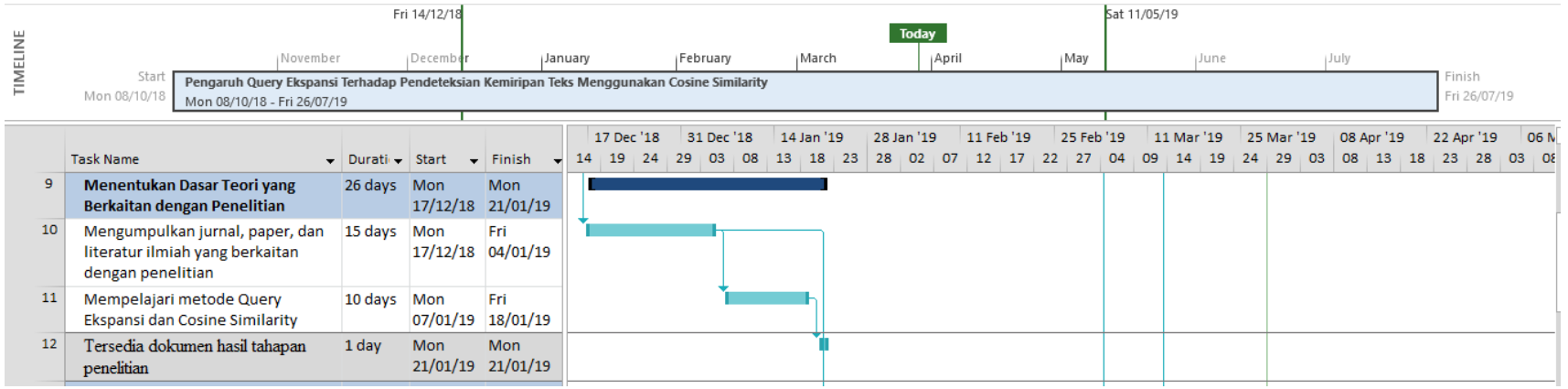
42	***Business Modelling***	2 days	Wed 01/05/19	Thu 02/05/19	
43	Menentukan kelas-kelas pada perangkat lunak	2 days	Wed 01/05/19	Thu 02/05/19	38
44	***Requirements***	2 days	Fri 03/05/19	Mon 06/05/19	
45	Menentukan bahasa pemrograman yang digunakan untuk mengembangkan perangkat lunak	1 day	Fri 03/05/19	Fri 03/05/19	16
46	Menentukan kebutuhan perangkat keras yang digunakan	1 day	Mon 06/05/19	Mon 06/05/19	43
47	***Analysis & Design***	2 days	Tue 07/05/19	Wed 08/05/19	
48	Membuat class diagram	2 days	Tue 07/05/19	Wed 08/05/19	43
49	***Implementation***	25 days	Thu 09/05/19	Wed 12/06/19	
50	Mengimplementasikan kelas-kelas ke dalam kode program	25 days	Thu 09/05/19	Wed 12/06/19	46
51	***Testing***	4 days	Thu 13/06/19	Tue 18/06/19	
52	Melakukan unit testing	4 days	Thu 13/06/19	Tue 18/06/19	46
53	***Transition***	14 days	Wed 19/06/19	Mon 08/07/19	
54	***Business Modelling***	2 days	Wed 19/06/19	Thu 20/06/19	
55	Membuat rencana atau skenario pengujian	2 days	Wed 19/06/19	Thu 20/06/19	50
56	***Requirements***	2 days	Fri 21/06/19	Mon 24/06/19	
57	Menentukan tools pengujian yang diperlukan dan contoh kasus	2 days	Fri 21/06/19	Mon 24/06/19	50

58	***Analysis & Design***	2 days	Tue 25/06/19	Wed 26/06/19	
59	Membuat tabel skenario pengujian	2 days	Tue 25/06/19	Wed 26/06/19	52
60	***Implementation***	4 days	Thu 27/06/19	Tue 02/07/19	
61	Melakukan pengujian terhadap perangkat lunak berdasarkan skenario atau rencana pengujian	4 days	Thu 27/06/19	Tue 02/07/19	55
62	***Testing***	4 days	Wed 03/07/19	Mon 08/07/19	
63	Meninjau atau menguji skenario pengujian	3 days	Wed 03/07/19	Fri 05/07/19	57
64	Tersedia dokumen hasil tahapan penelitian	1 day	Mon 08/07/19	Mon 08/07/19	61
65	**Melakukan Pengujian Penelitian**	7 days	Tue 09/07/19	Wed 17/07/19	
66	Menentukan rancangan hasil penelitian	4 days	Tue 09/07/19	Fri 12/07/19	5
67	Melakukan pengujian penelitian berdasarkan hasil pengujian perangkat lunak	2 days	Mon 15/07/19	Tue 16/07/19	59
68	Tersedia dokumen hasil tahapan penelitian	1 day	Wed 17/07/19	Wed 17/07/19	
69	**Membuat Analisa Hasil Pengujian dan Membuat Kesimpulan**	7 days	Thu 18/07/19	Fri 26/07/19	
70	Melakukan analisa terhadap hasil pengujian penelitian dengan menghitung perbandingan selisih evaluasi cluster	4 days	Thu 18/07/19	Tue 23/07/19	66
71	Membuat kesimpulan dan saran berdasarkan analisa terhadap hasil pengujian	2 days	Wed 24/07/19	Thu 25/07/19	67
72	Tersedia dokumen hasil tahapan penelitian	1 day	Fri 26/07/19	Fri 26/07/19	70

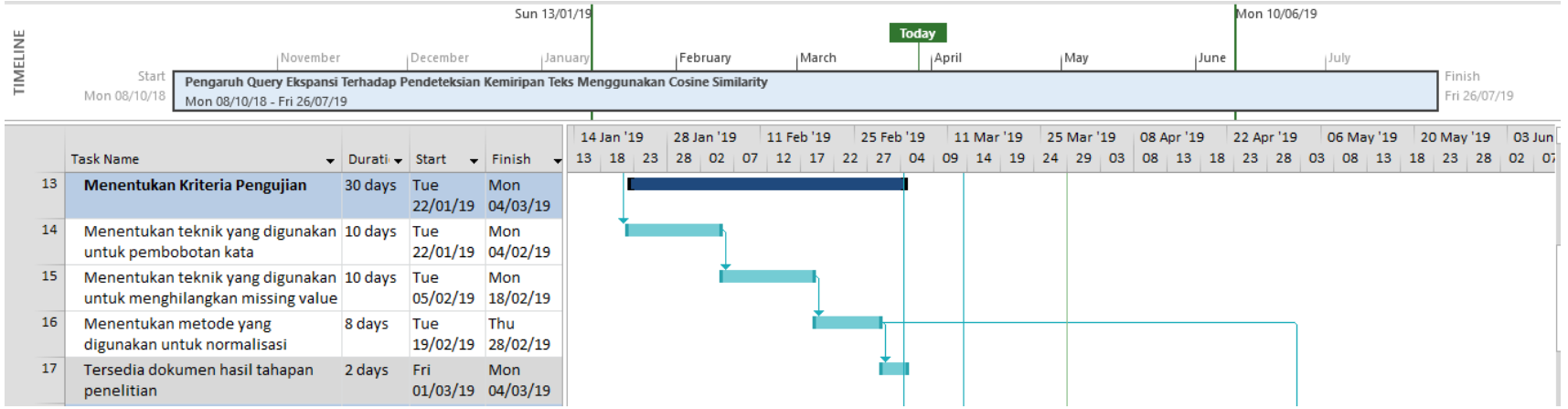
Penjadwalan penelitian dalam bentuk *Gantt Chart* dibuat dengan *tool* Microsoft Project 2013. Gambar III-4, Gambar III-5, Gambar III-6, Gambar III-7, Gambar III-8, Gambar III-9, Gambar III-10, Gambar III-11 dan Gambar III-12 menampilkan *Gantt Chart* untuk penjadwalan penelitian.



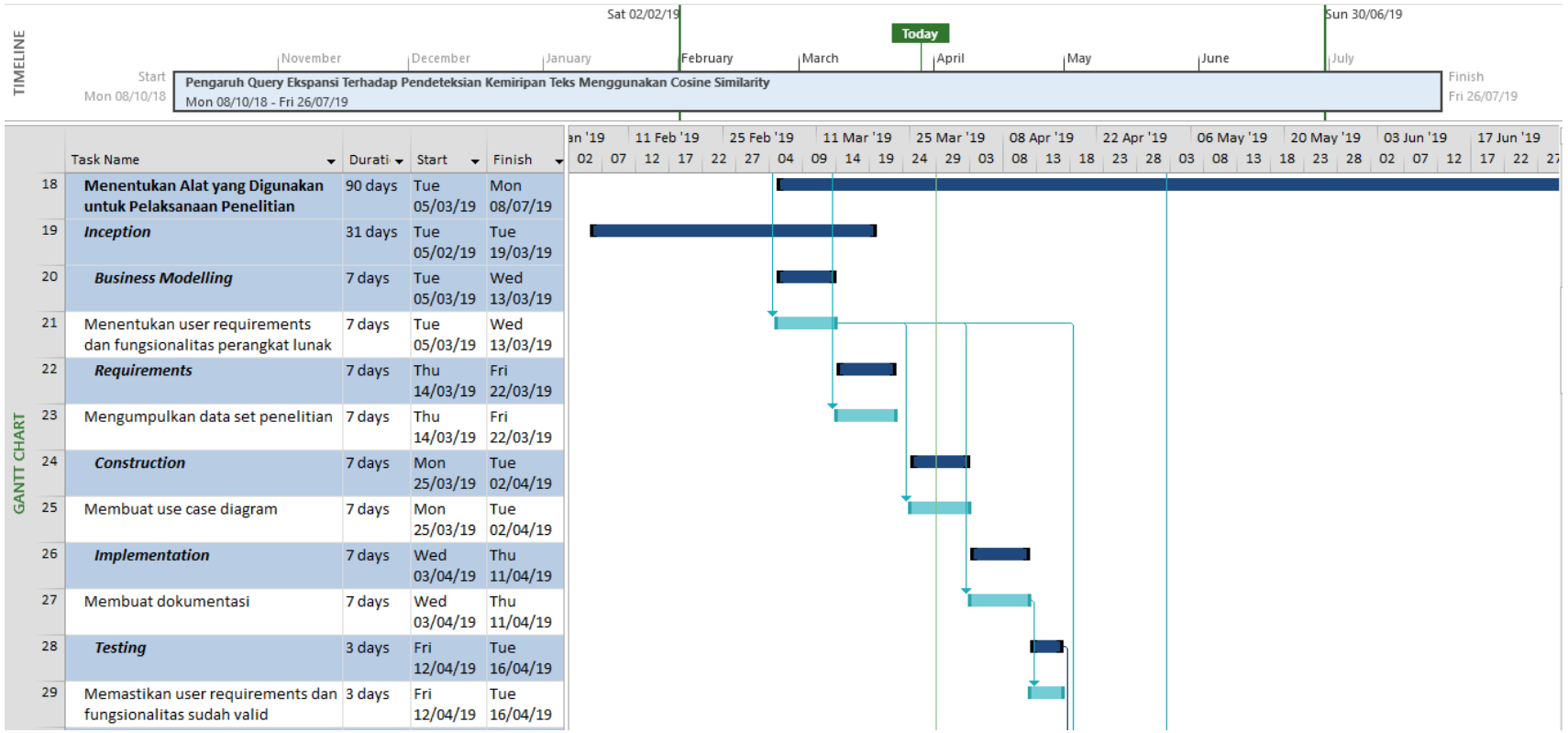
Gambar III-4. Penjadwalan untuk Tahap Menentukan Ruang Lingkup dan Unit Penelitian



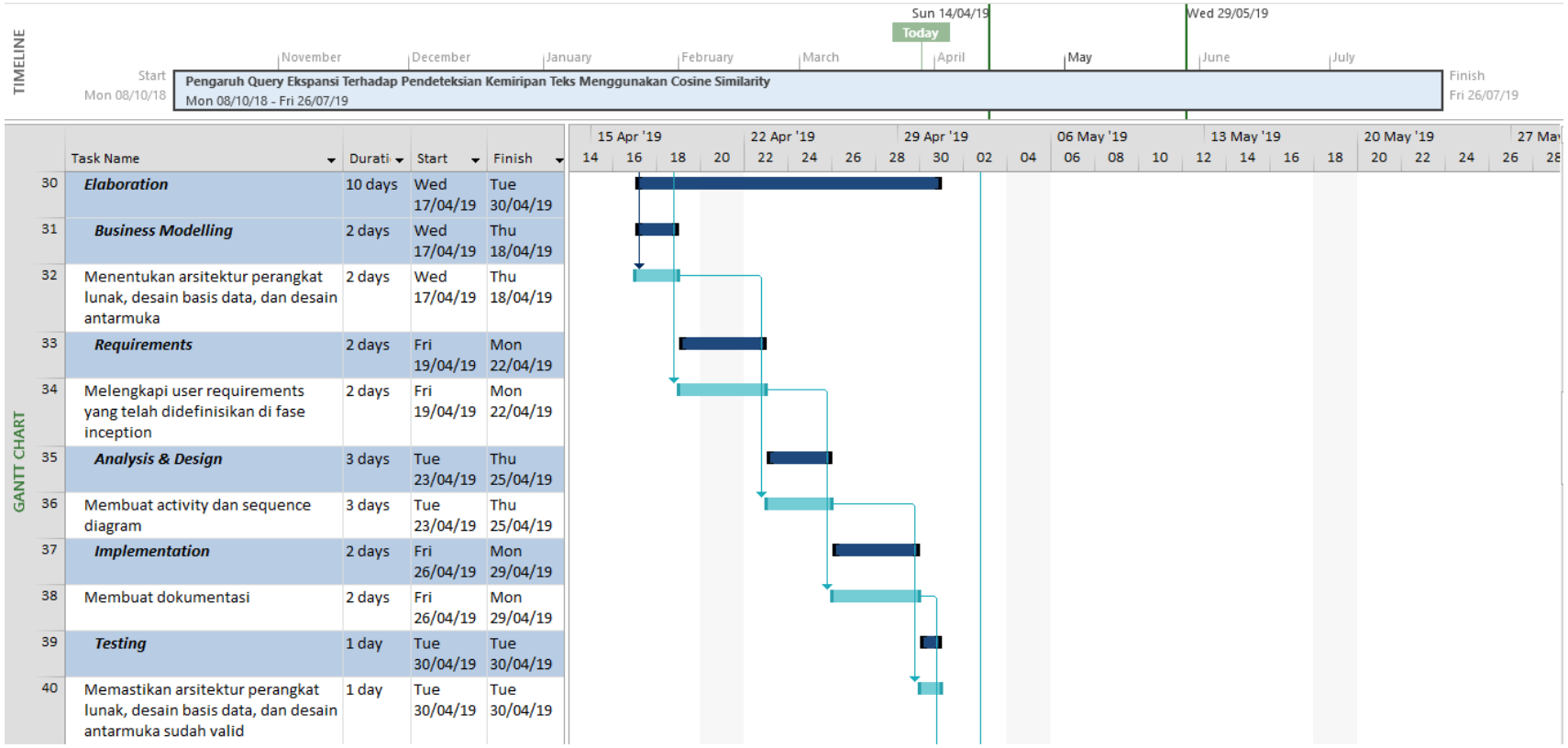
Gambar III-5. Penjadwalan untuk Tahap Menentukan Dasar Teori yang Berkaitan dengan Penelitian



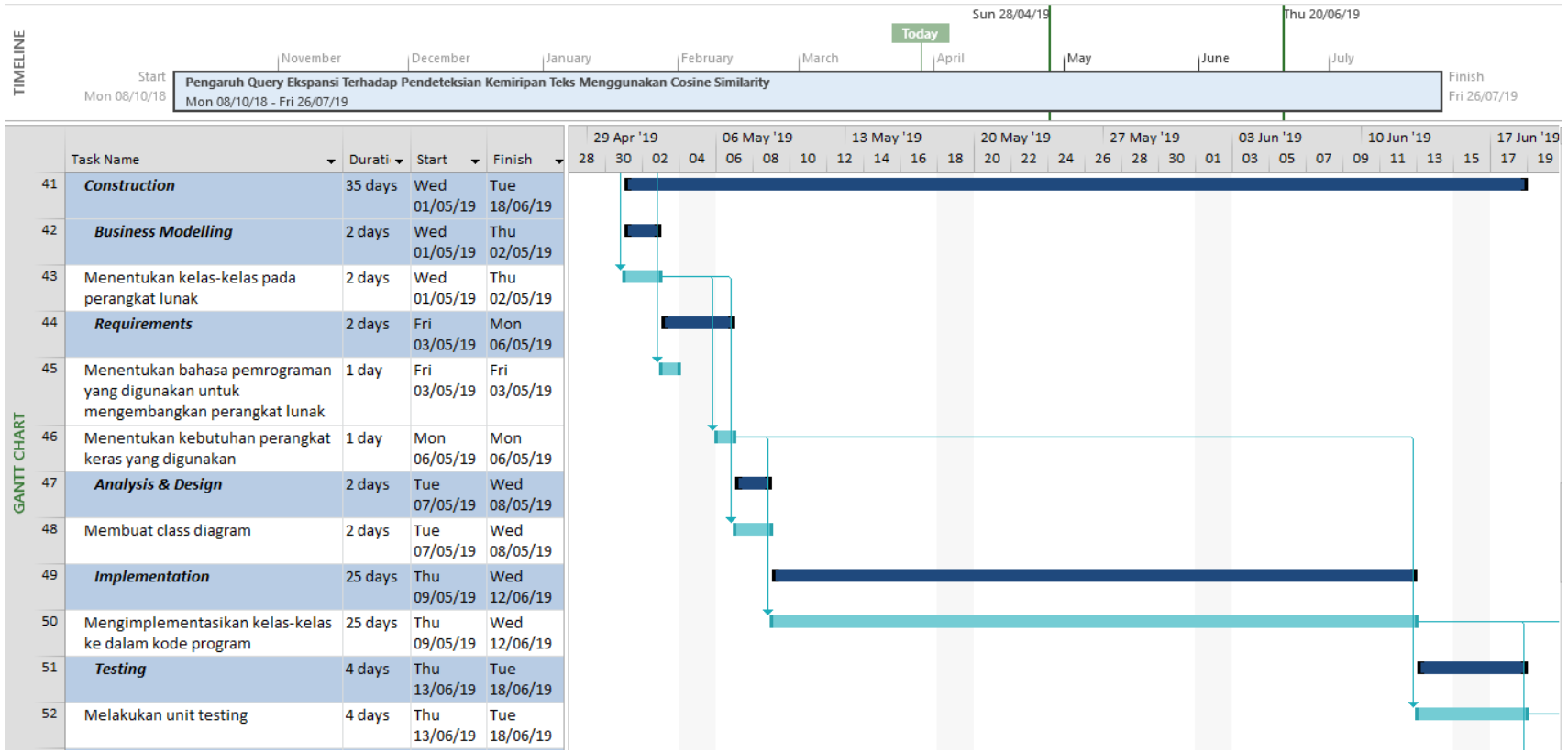
Gambar III-6. Penjadwalan untuk Tahap Menentukan Kriteria Pengujian



Gambar III-7. Penjadwalan untuk Tahap Menentukan Alat yang Digunakan untuk Pelaksanaan Penelitian Fase Insepsi

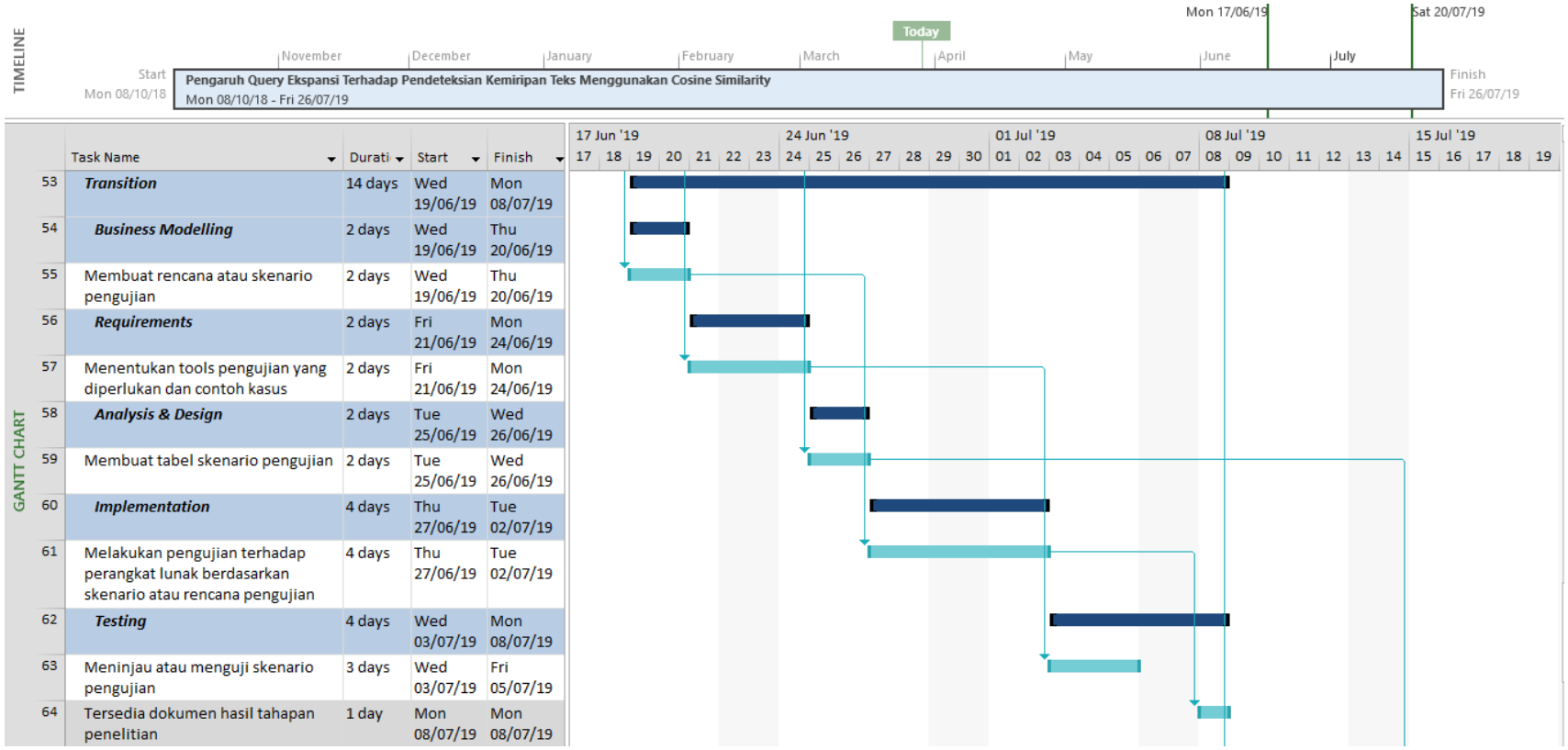


Gambar III-8. Penjadwalan untuk Tahap Menentukan Alat yang Digunakan untuk Pelaksanaan Penelitian Fase Elaborasi



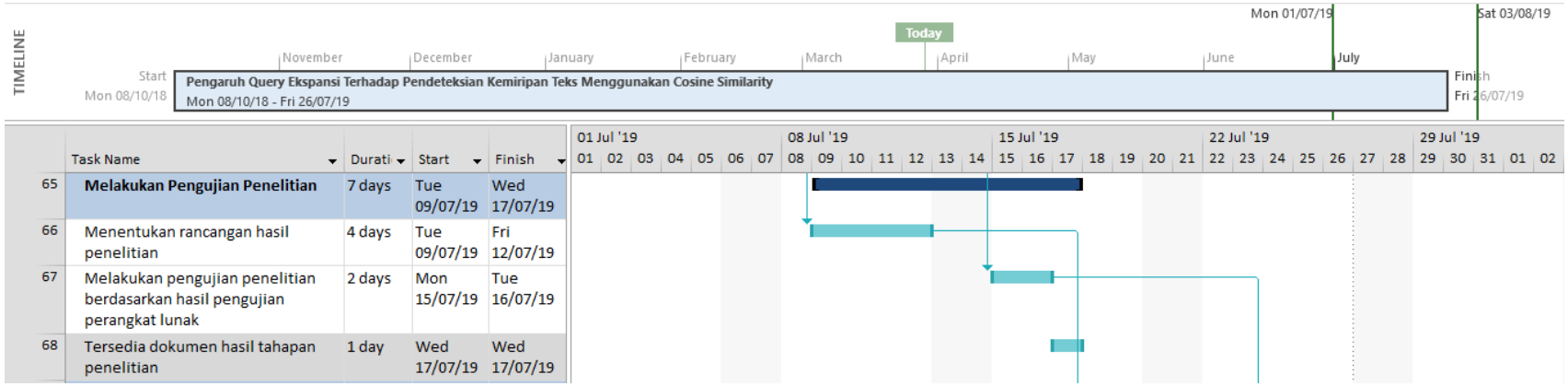
	Task Name	Durati	Start	Finish
41	***Construction***	35 days	Wed 01/05/19	Tue 18/06/19
42	***Business Modelling***	2 days	Wed 01/05/19	Thu 02/05/19
43	Menentukan kelas-kelas pada perangkat lunak	2 days	Wed 01/05/19	Thu 02/05/19
44	***Requirements***	2 days	Fri 03/05/19	Mon 06/05/19
45	Menentukan bahasa pemrograman yang digunakan untuk mengembangkan perangkat lunak	1 day	Fri 03/05/19	Fri 03/05/19
46	Menentukan kebutuhan perangkat keras yang digunakan	1 day	Mon 06/05/19	Mon 06/05/19
47	***Analysis & Design***	2 days	Tue 07/05/19	Wed 08/05/19
48	Membuat class diagram	2 days	Tue 07/05/19	Wed 08/05/19
49	***Implementation***	25 days	Thu 09/05/19	Wed 12/06/19
50	Mengimplementasikan kelas-kelas ke dalam kode program	25 days	Thu 09/05/19	Wed 12/06/19
51	***Testing***	4 days	Thu 13/06/19	Tue 18/06/19
52	Melakukan unit testing	4 days	Thu 13/06/19	Tue 18/06/19

Gambar III-9. Penjadwalan untuk Tahap Menentukan Alat yang Digunakan untuk Pelaksanaan Penelitian Fase Konstruksi



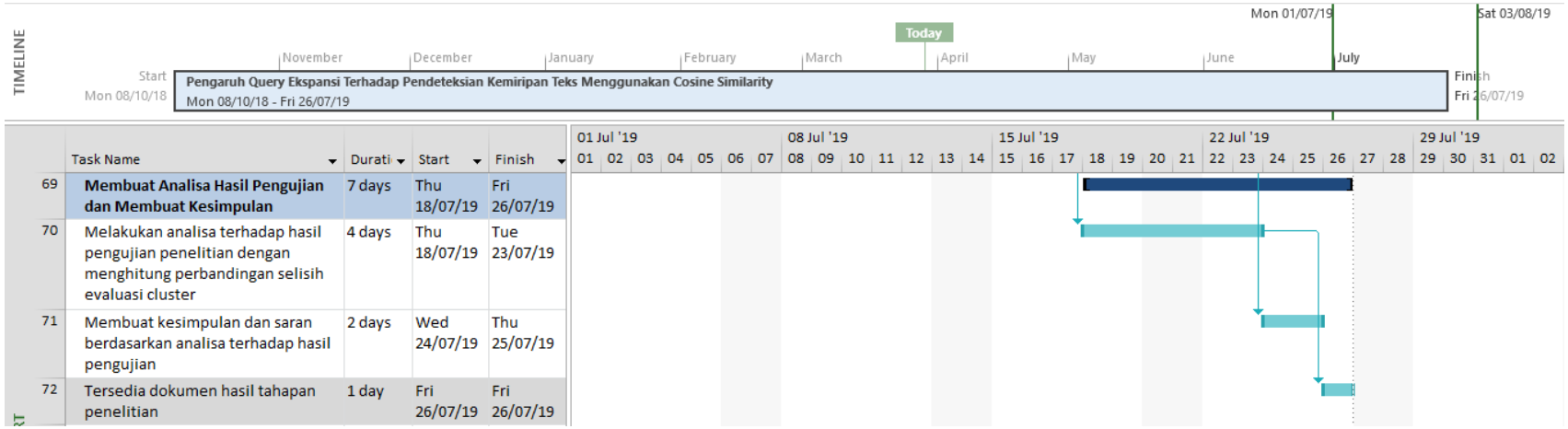
	Task Name	Durati	Start	Finish
53	***Transition***	14 days	Wed 19/06/19	Mon 08/07/19
54	***Business Modelling***	2 days	Wed 19/06/19	Thu 20/06/19
55	Membuat rencana atau skenario pengujian	2 days	Wed 19/06/19	Thu 20/06/19
56	***Requirements***	2 days	Fri 21/06/19	Mon 24/06/19
57	Menentukan tools pengujian yang diperlukan dan contoh kasus	2 days	Fri 21/06/19	Mon 24/06/19
58	***Analysis & Design***	2 days	Tue 25/06/19	Wed 26/06/19
59	Membuat tabel skenario pengujian	2 days	Tue 25/06/19	Wed 26/06/19
60	***Implementation***	4 days	Thu 27/06/19	Tue 02/07/19
61	Melakukan pengujian terhadap perangkat lunak berdasarkan skenario atau rencana pengujian	4 days	Thu 27/06/19	Tue 02/07/19
62	***Testing***	4 days	Wed 03/07/19	Mon 08/07/19
63	Meninjau atau menguji skenario pengujian	3 days	Wed 03/07/19	Fri 05/07/19
64	Tersedia dokumen hasil tahapan penelitian	1 day	Mon 08/07/19	Mon 08/07/19

Gambar III-10. Penjadwalan untuk Tahap Menentukan Alat yang Digunakan untuk Pelaksanaan Penelitian Fase Transisi



	Task Name	Durati	Start	Finish
65	**Melakukan Pengujian Penelitian**	7 days	Tue 09/07/19	Wed 17/07/19
66	Menentukan rancangan hasil penelitian	4 days	Tue 09/07/19	Fri 12/07/19
67	Melakukan pengujian penelitian berdasarkan hasil pengujian perangkat lunak	2 days	Mon 15/07/19	Tue 16/07/19
68	Tersedia dokumen hasil tahapan penelitian	1 day	Wed 17/07/19	Wed 17/07/19

Gambar III-11. Penjadwalan untuk Tahap Melakukan Pengujian Penelitian



	Task Name	Durati	Start	Finish
69	**Membuat Analisa Hasil Pengujian dan Membuat Kesimpulan**	7 days	Thu 18/07/19	Fri 26/07/19
70	Melakukan analisa terhadap hasil pengujian penelitian dengan menghitung perbandingan selisih evaluasi cluster	4 days	Thu 18/07/19	Tue 23/07/19
71	Membuat kesimpulan dan saran berdasarkan analisa terhadap hasil pengujian	2 days	Wed 24/07/19	Thu 25/07/19
72	Tersedia dokumen hasil tahapan penelitian	1 day	Fri 26/07/19	Fri 26/07/19

Gambar III-12. Penjadwalan untuk Tahap Analisa Hasil Pengujian Penelitian dan Membuat Kesimpulan

DAFTAR PUSTAKA

Boisvert, R. F. and Irwin, M. J. (2006) 'Plagiarism on the rise', *Communications of the ACM*, 49(6), p. 23. doi: 10.1145/1132469.1132487.

Carpineto, C. and Romano, G. (2012) 'A Survey of Automatic Query Expansion in Information Retrieval', *ACM Computing Surveys*, 44(1), pp. 1–50. doi: 10.1145/2071389.2071390.

Cholifah, Purwanto, Y. and Bramanto, A. (2011) 'Aplikasi Information Retrieval untuk pembentukan Tesaurus Berbahasa Indonesia secara otomatis'. Surabaya, pp. 41–48.

Citron, D. T. and Ginsparg, P. (2015) 'Patterns of text reuse in a scientific corpus', *Proceedings of the National Academy of Sciences*, 112(1), pp. 25–30. doi: 10.1073/pnas.1415135111.

Firdaus, A., Ernawati and Vatresia, A. (2014) 'Aplikasi Pendeteksi Kemiripan pada Dokumen Teks Menggunakan Algoritma Nazief & Andriani Dan Metode Cosine Similirity', *Jurnal Teknologi Informasi*, 10(April), pp. 96–109.

Gusmita, R. H. *et al.* (2014) 'A rule-based question answering system on relevant documents of Indonesian Quran Translation', *2014 International Conference on Cyber and IT Service Management, CITSM 2014*, pp. 104–107. doi: 10.1109/CITSM.2014.7042185.

Imbar, R. V. *et al.* (no date) 'Implementasi Cosine Similarity dan Algoritma Smith-Waterman untuk Mendeteksi Kemiripan Teks', pp. 31–42.

Khafajeh, H., Refai, M. and Yousef, N. (2013) 'Building Arabic Automatic Thesaurus Using Co-occurence Technique', *Proceedings of International Conference on Communication, Media, Technology and Design*, (April 2013), pp. 28–32.

Lei, K., Tang, H. and Zeng, Y. (2018) 'Keywords Extraction via Multi-relational Network Construction Keywords Extraction via Multi-relational Network', (January 2013). doi: 10.1007/978-3-319-00951-3.

Muhammad, R. *et al.* (2017) 'This is a repository copy of An IR-based Approach

Utilising Query Expansion for Plagiarism Detection in MEDLINE. An IR-based Approach Utilising Query Expansion for Plagiarism Detection in MEDLINE', *IEEE/ACM Transactions on Computational Biology and Bioinformatics JOURNAL OF COMPUTATIONAL BIOLOGY AND BIOINFORMATICS*, 14(4), pp. 796–804. doi: 10.1109/TCBB.2016.2542803.

Mutiara, A. B. and Agustina, S. (2008) 'Anti Plagiarism Application with Algorithm Karp-Rabin at Thesis in Gunadarma University', *arXiv preprint arXiv:0811.4349*, p. 9. Available at: http://arxiv.org/abs/0811.4349.

Park, C. (2003) 'Assessment & evaluation in higher education in other (people' s) words : plagiarism by university students--literature and', *Assessment & Evaluation in Higher Education*, 28(5), pp. 241–288. doi: 10.1080/0260293032000120352.

Pressman, R. S. and Maxim, B. R. (2005) 'Software Engineering', p. 976.

Purwarianti, A. and Yusliani, N. (2012) 'Sistem Question Answering Bahasa Indonesia untuk Pertanyaan Non-Factoid', *Jurnal Ilmu Komputer dan Informasi*, 4(1), p. 10. doi: 10.21609/jiki.v4i1.151.

Rafles, A. (2013) 'Plagiarisme Dokumen Dengan Pendekatan K-Gram Berbasis Frasa K-Gram Berbasis Frasa'. doi: 10.1186/1478-4491-13-2.

Rahman, N. A., Bakar, Z. A. and Sembok, T. M. T. (2010) 'Query Expansion using Thesaurus in Improving Malay Hadith Retrieval System', pp. 1404–1409.

Rasyidi, I., Romadhony, A. and Wibowo, A. T. (2013) 'Indonesian Hadith Retrieval System using Thesaurus', pp. 285–288.

Ryansyah, A. and Andayani, S. (2017) 'Implementasi Algoritma TF-IDF Pada Pengukuran Kesamaan Dokumen', *Jurnal Sistem & Teknologi Informasi Komunikasi*, 1(1), pp. 1–10.

Saneifar, H. *et al.* (2014) 'Enhancing passage retrieval in log files by query expansion based on explicit and pseudo relevance feedback', *Computers in Industry*, 65(6), pp. 937–951. doi: 10.1016/j.compind.2014.02.010.

Soelistyo, H. (2011) 'Plagiarisme : Pelanggaran Hak Cipta dan Etika', *Plagiarisme : Pelanggaran Hak Cipta dan Etika*, (Yogyakarta: Kanisius), p. No Pages. Available at:

http://www.dt.co.kr/contents.html?article_no=2012071302010531749001.

Stein, B., Eissen, S. M. zu and Potthast, M. (2007) ‘Strategies for retrieving plagiarized documents’, *Kidney and Blood Pressure Research*, 24(2), pp. 84–91. doi: 10.1159/000054212.

Stein, B. and zu Eissen, S. M. (2006) ‘Near Similarity Search and Plagiarism Analysis’, (1993), pp. 430–437. doi: 10.1007/3-540-31314-1_52.

Vijayarani, S., Ilamathi, J. and Nitya (2015) ‘Preprocessing Techniques for Text Mining - An Overview’, 5(1), pp. 7–16. doi: 10.1016/j.procs.2013.05.286.