



**PENGEMBANGAN SISTEM DETEKSI *TIME FRAME* DOKUMEN
BERITA BERBASIS *VECTOR SPACE MODEL***

SKRIPSI

Oleh
Ferry Wiranto
NIM 152410101157

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS JEMBER
2019**



**PENGEMBANGAN SISTEM DETEKSI *TIME FRAME* DOKUMEN
BERITA BERBASIS *VECTOR SPACE MODEL***

SKRIPSI

Diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan Pendidikan Sarjana (S1) Program Studi Sistem Informasi Universitas Jember dan mencapai gelar Sarjana Komputer

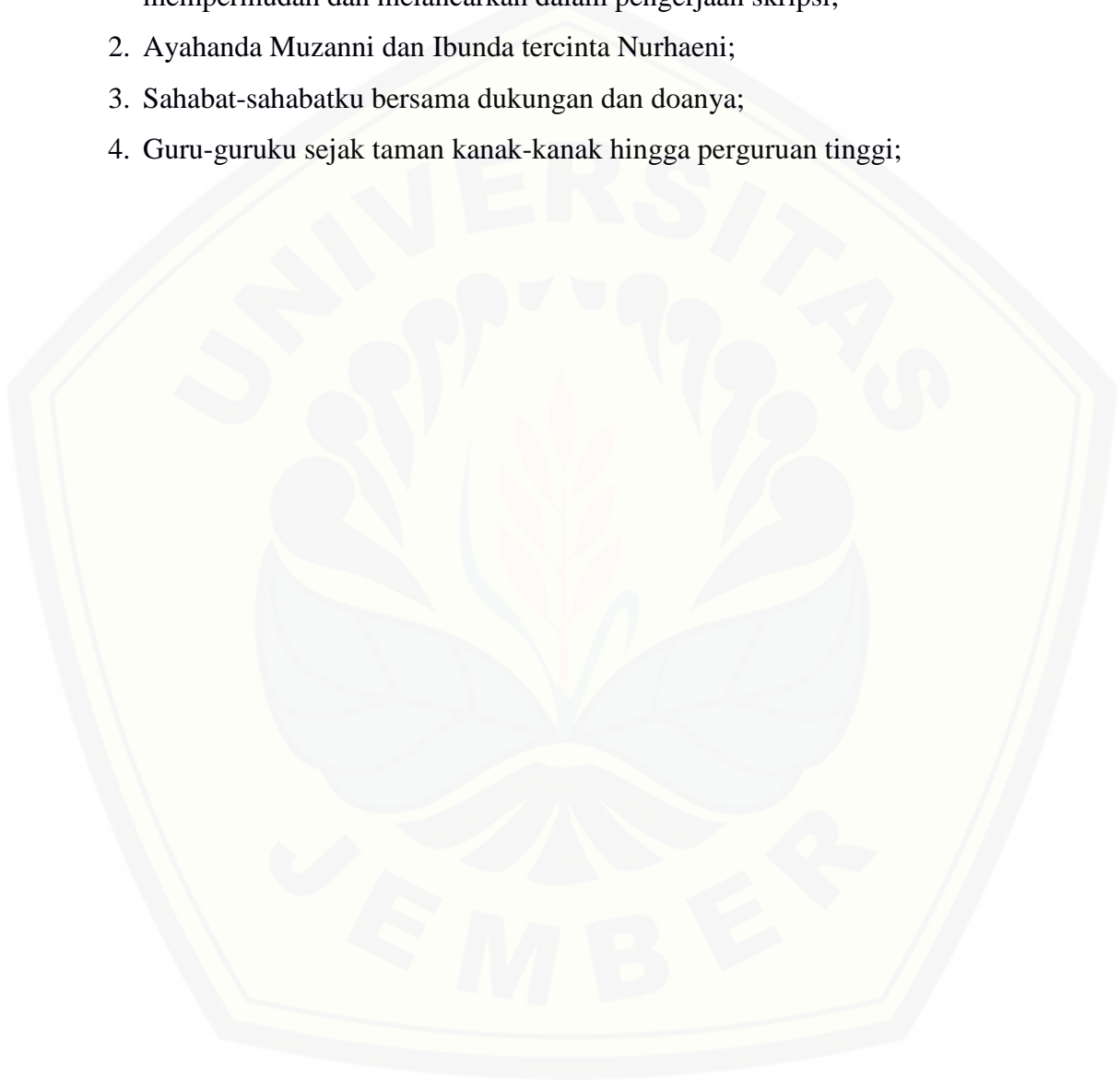
Oleh
Ferry Wiranto
NIM 152410101157

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS JEMBER
2019**

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Allah SWT yang senantiasa memberikan rahmat dan hidayah-Nya untuk mempermudah dan melancarkan dalam pengerjaan skripsi;
2. Ayahanda Muzanni dan Ibunda tercinta Nurhaeni;
3. Sahabat-sahabatku bersama dukungan dan doanya;
4. Guru-guruku sejak taman kanak-kanak hingga perguruan tinggi;



MOTO

“Jangan menunggu, Takkan pernah ada waktu yang tepat, Kemauanmu untuk berhasil harus lebih besar dari ketakutanmu untuk gagal.
Terbentur terbentur dan terbentur kemudian terbentuk”



PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Ferry Wiranto

NIM : 152410101157

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Pengembangan Sistem Deteksi *Time Frame* Dokumen berita berbasis *Vector Space Model*” adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 27 Maret 2019

Yang menyatakan,

Ferry Wiranto

NIM 152410101157

SKRIPSI

**PENGEMBANGAN SISTEM DETEKSI *TIME FRAME* DOKUMEN
BERITA BERBASIS *VECTOR SPACE MODEL***

Oleh :

Ferry Wiranto

NIM 152410101157

Pembimbing :

Dosen Pembimbing Utama : Achmad Maududie, ST., M.Sc

Dosen Pembimbing Pendamping : Tio Dharmawan, S.Kom., M.Kom

PENGESAHAN PEMBIMBING

Skripsi berjudul “Pengembangan Sistem Deteksi *Time Frame* Dokumen berita berbasis *Vector Space Model*”, telah diuji dan disahkan pada:

hari, tanggal : Jumat, 12 April 2019

tempat : Universitas Jember, Fakultas Ilmu Komputer, Program Studi Sistem Informasi

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Achmad Maududie, ST., M.Sc

NIP 198403052010122002

Tio Dharmawan, S.Kom., M.Kom

NIP 760016851

PENGESAHAN

Skripsi berjudul “Pengembangan Sistem Deteksi *Time Frame* Dokumen berita berbasis *Vector Space Model*”, telah diuji dan disahkan pada:

hari, tanggal : Jumat, 12 April 2019

tempat : Universitas Jember, Fakultas Ilmu Komputer, Program Studi Sistem Informasi

Tim Penguji,

Penguji I,

Penguji II,

Anang Andrianto, ST., MT
NIP 196906151997021002

Diksy Media Firmansyah, S.Kom., M.Kom
NIP 760016853

Mengesahkan

Penjabat Dekan Fakultas Ilmu Komputer,

Prof. Dr. Saiful Bukhori, ST.,M.Kom
NIP. 196811131994121001

RINGKASAN

Pengembangan Sistem Deteksi *Time Frame* Dokumen berita berbasis *Vector Space Model* Ferry Wiranto, 152410101157; 2019, 118 HALAMAN; Program Studi Sistem Informasi Universitas Jember.

Berkembangnya website sebagai salah satu bentuk media massa mengakibatkan tajamnya peningkatan jumlah informasi berupa artikel berita. Hasil pengamatan dari tiga situs berita (yaitu Tribunnews.com, Detik.com, dan Liputan6.com) dengan teknik *scraping*, pada bulan Agustus sampai dengan Desember 2018 ditemukan sebanyak 109.061 berita yang telah terpublikasi. Apabila dilihat dari sisi jumlah, pembaca berita sangat cukup mendapatkan referensi dari informasi yang ingin diketahuinya. Namun demikian, dengan banyaknya informasi tersebut, pembaca juga menemui kesulitan ketika ingin mengetahui rentang waktu sebuah topik yang diberitakan. Secara manual, pembaca harus mencari semua dokumen pada situs berita yang ada serta menentukan apakah dokumen berita tersebut sesuai dengan topik yang dimaksud atau tidak. Selanjutnya, pembaca juga diharuskan memetakan seluruh dokumen yang sesuai topik berdasarkan tanggal terbitnya. Langkah tersebut dilakukan untuk mengetahui durasi kejadian secara manual, Dengan demikian untuk mengetahui rentang sebuah topik yang diberitakan akan memakan waktu yang lama.

Sistem pencarian merupakan salah satu solusi yang dapat membantu dalam menemukan kembali informasi yang diinginkan (Dasar, Wicaksono, & Sihwi, 2015). Sistem ini berusaha menemukan dokumen yang relevan sesuai dengan *query* masukannya. Salah satu model yang sering digunakan dalam proses pencarian adalah *Vector Space Model (VSM)*. *VSM* merupakan suatu metode yang merepresentasikan sistem temu kembali (*information retrieval*) kedalam vektor serta memperhitungkan fungsi *similarity* dalam pencocokan beberapa vektor.

Pada penelitian ini menggunakan dokumen berita yang digunakan yang sudah tersimpan kedalam bentuk *JSON* dengan total berita 4812 berita dari situs berita Tribunnews.com, Detik.com, dan Liputan6.com. Peneliti akan membangun desain dan sistem yang akan membantu sistem pendeteksi rentang waktu pemberitaan

topik dalam dokumen berita atau dalam penelitian ini disebut dengan *time-frame* dokumen berita. Sistem ini didasarkan pada sistem temu kembali dengan menggunakan model *Vector Space Model*. Dokumen berita yang ditemu kembali adalah dokumen yang memiliki *similarity* sesuai dengan *threshold* (nilai minimal) yang ditentukan. Sistem ini diharapkan dapat mendeteksi *time-frame* dokumen berita dari tiga situs berita secara akurat sehingga dapat menjawab kebutuhan informasi tentang rentang waktu sebuah topik yang diberitakan, serta mengetahui akurasi dari sistem yang dibangun.

Berdasarkan tahapan analisis kebutuhan pengguna, desain sistem deteksi sistem *time frame* konten berita berbasis *vector space model* untuk menampilkan *time frame* dokumen berita berdasarkan pencarian berita pengguna membutuhkan 15 proses serta melibatkan 3 aktor. Berdasarkan desain tersebut yang telah berhasil dibangun, nilai *cosine similarity* minimal (*threshold*) yang dibutuhkan untuk mendapatkan dokumen yang diinginkan adalah 0,788. Nilai akurasi yang dihasilkan berdasarkan desain yang dibangun menggunakan teknik perhitungan *recall* mendapatkan nilai 100%, *precision* 90% dan *f-measure* sebesar 94,7%.

PRAKATA

Puji syukur kehadiran Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Pengembangan Sistem Deteksi *Time Frame* Dokumen berita berbasis *Vector Space Model*”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Ayahanda Muzanni, Ibunda Nurhaeni dan kakak Zainul Ihsan yang selalu mendukung dan mendoakan proses pengerjaan skripsi;
2. Prof. Dr. Saiful Bukhori, ST.,M.Kom., selaku Dekan Fakultas Ilmu Komputer Universitas Jember;
3. Achmad Maududie, ST, M.Sc selaku Dosen Pembimbing Utama dan Tio Dharmawan, S.Kom., M.Kom selaku Dosen Pembimbing Pendamping yang telah meluangkan waktu, pikiran, dan perhatian dalam membantu penulisan skripsi;
4. Anang Andrianto ST.,MT., selaku Dosen Pembimbing Akademik (DPA), yang telah mendampingi penulisan skripsi;
5. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember;
6. Teman-teman seperjuangan SELECTION angkatan 2015;
7. Keluarga besar asisten Laboratorium Rekaya Perangkat Lunak;
8. Semua mahasiswa Fakultas Ilmu Komputer yang telah menjadi keluarga kecil bagi penulis selama menempuh pendidikan S1;
9. Semua pihak yang tidak dapat disebutkan satu-persatu.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna, oleh sebab itu penulis mengharapkan adanya masukan yang bersifat membangun dari semua pihak. Penulis berharap skripsi ini dapat bermanfaat bagi semua pihak.

Jember, 05 Januari 2019

Penulis



DAFTAR ISI

SKRIPSI.....	i
PERSEMBAHAN	ii
MOTO.....	iii
PERNYATAAN.....	iv
SKRIPSI.....	v
PENGESAHAN PEMBIMBING.....	vi
PENGESAHAN	vii
RINGKASAN	viii
PRAKATA	x
DAFTAR ISI.....	xii
DAFTAR GAMBAR	xv
DAFTAR TABEL.....	xviii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian.....	3
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	4
1.6 Sistematika Penulisan.....	5
BAB 2 TINJAUAN PUSTAKA	6
2.1 Penelitian Terdahulu.....	6
2.2 Software Development Life Cycle (SDLC) Personal Extreme Programming.....	7
2.3 Situs Berita Online	12
2.4 Text Mining	12
2.5 <i>Stemming TALA</i>	15
2.6 <i>Information Retrieval</i>	17
2.7 <i>Vector Space Model</i>	18

2.8	<i>Cosine Similarity</i>	20
2.9	<i>Time frame</i> dokumen berita.....	21
2.10	Pengujian kualitas <i>information retrieval</i>	21
BAB 3 METODOLOGI PENELITIAN.....		23
3.1	Jenis Penelitian	23
3.2	Waktu Penelitian	23
3.3	Tahapan Penelitian	23
BAB 4 PENGEMBANGAN SISTEM.....		31
4.1	<i>Requirement</i>	31
4.2	<i>Planning</i>	31
4.3	<i>Iteration Initialization</i> sistem	38
4.4	Desain Sistem	40
4.4.1	<i>Diagram input output</i>	40
4.4.2	<i>Use Case Diagram</i>	40
4.4.3	<i>Business Process Model and Notation (BPMN)</i>	44
4.4.4	<i>Class diagram</i>	45
4.4.5	<i>CRC Card</i>	50
4.4.6	<i>Entity Relationship Diagram (ERD)</i>	55
4.5	<i>Implementasi</i>	56
4.6	<i>System Testing</i>	56
4.7	<i>Retrospective</i>	56
4.8	Pengujian Integrasi akhir	57
4.9	Penentuan batas minimal <i>cosine similarity</i> dokumen dengan <i>query</i>	57
BAB 5 HASIL DAN PEMBAHASAN.....		62
5.1	Desain kerangka sistem deteksi <i>time frame</i> konten berita berbasis <i>vector space model</i>	62

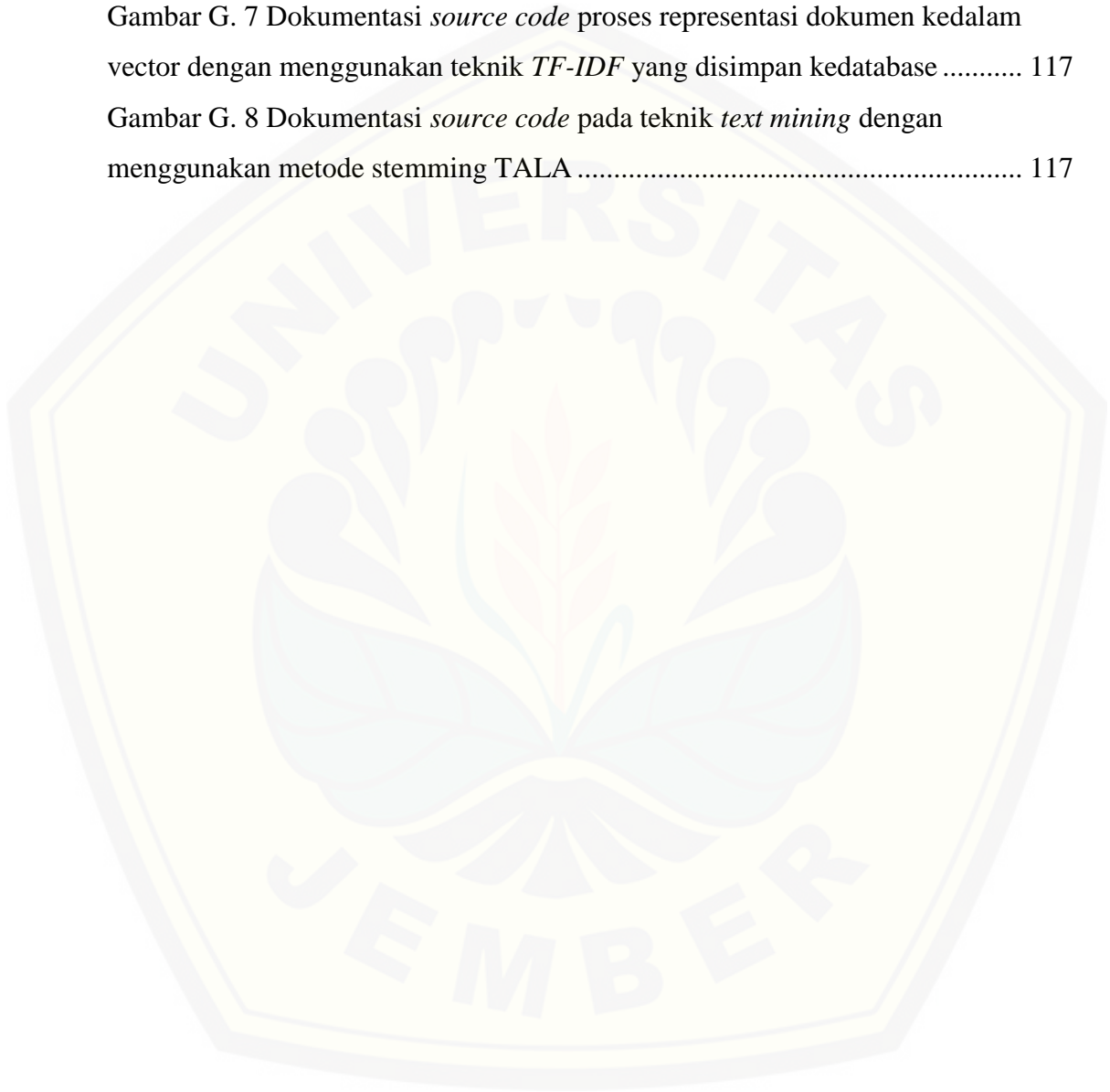
5.2	Perhitungan Akurasi <i>Vector Space Model</i> terkait dengan dokumen yang ditampilkan.....	66
5.3	Hasil Pembangunan Sistem	70
5.3.1	<i>Frontpage Sistem Siteframe</i>	70
5.3.2	Halaman Masuk Sistem.....	72
5.3.3	Halaman Daftar	73
5.3.4	Halaman <i>dashboard</i> Admin	73
5.3.5	Halaman Melihat Daftar Berita dari Sumber Berita	74
5.3.6	Halaman melihat dokumen hasil pencarian berita	75
5.3.7	Halaman <i>Feed Activity Member</i>	75
5.3.8	Halaman Pengguna.....	76
BAB 6	PENUTUP	78
6.1	Kesimpulan.....	78
6.2	Saran	79
DAFTAR	PUSTAKA	80
LAMPIRAN	82
A.	UNIT TESTING.....	82
B.	DOKUMEN <i>SYSTEM TESTING</i>	87
C.	DOKUMEN PENGUJIAN INTEGRASI AKHIR.....	95
D.	TAMPILAN SISTEM	100
E.	Perhitungan <i>Similarity</i> Antara Dokumen Berita dengan <i>Query</i>	103
F.	<i>User Story Site Time Frame</i>	109
G.	Dokumentasi <i>code</i> program sistem <i>time frame</i> & <i>ConsoleAPs</i>	114
H.	Dokumen <i>Retrospective</i>	118

DAFTAR GAMBAR

Gambar 2.1 Kerangka Kerja <i>Personal Extreme Programming</i>	9
Gambar 2.2 Contoh Proses Tokenizing	13
Gambar 2.3 Proses Algoritma <i>Stemming Tala</i>	16
Gambar 2.4 Diagram alur <i>Information Retrieval</i>	18
Gambar 2.5 Representasi dokumen dan <i>query</i> pada ruang vektor.....	19
Gambar 3.1 Diagram Alur Tahapan Penelitian.....	24
Gambar 4.1 Alur kerja sub sistem <i>consoleAps</i>	32
Gambar 4.2 Diagram Alir Tahapan Representasi Dokumen kedalam ruang <i>n</i> - dimensi	33
Gambar 4.3 Alur kerja sub sistem <i>site time frame</i>	35
Gambar 4.4 <i>Flowchart</i> Tahapan Pencarian Dokumen dan Durasi Berita.....	37
Gambar 4.5 <i>Diagram input output</i>	40
Gambar 4.6 <i>Use Case Diagram</i>	41
Gambar 4.7 <i>Business Process Model and Notation</i>	44
Gambar 4.8 <i>Class Diagram ConsoleAps</i>	46
Gambar 4.9 <i>Class Diagram Site Time Frame</i>	49
Gambar 4.10 <i>Entity Relationship Diagram (ERD)</i>	55
Gambar 4.11 Sampel dokumen berita yang akan dilihat nilai <i>similarity</i> dengan <i>query</i> “kecelakaan lalu lintas”	57
Gambar 4.12 hasil pengujian nilai <i>similarity</i> dokumen berita dengan <i>query</i>	59
Gambar 5.1 Desain kerangka sistem deteksi <i>time frame</i>	62
Gambar 5.2 Representasi dokumen berita dan kata pada sistem	64
Gambar 5.3 Pemrosesan kata pencarian.....	65
Gambar 5.4 pencarian dokumen berita dengan <i>query</i> “kecelakaan lalu lintas”...	67
Gambar 5.5 Dokumen berita ke-1 yang kurang sesuai dengan <i>query</i> “kecelakaan lalu lintas”	67
Gambar 5.6 Dokumen berita ke-2 yang kurang sesuai dengan <i>query</i> “kecelakaan lalu lintas”	68
Gambar 5.7 Konten <i>site time frame</i>	70

Gambar 5.8 Konten <i>about</i>	71
Gambar 5.9 konten layanan.....	71
Gambar 5.10 konten pengetahuan.....	72
Gambar 5.11 konten <i>contact</i>	72
Gambar 5.12 <i>Halaman Login Sistem</i>	73
Gambar 5.13 <i>Halaman Daftar Member</i>	73
Gambar 5.14 <i>Halaman Dashboard Admin</i>	74
Gambar 5.15 <i>Halaman Melihat Daftar Berita dari Sumber Berita</i>	74
Gambar 5.16 <i>Halaman melihat dokumen hasil pencarian berita</i>	75
Gambar 5.17 <i>Halaman Feed Activity Member</i>	76
Gambar 5.18 <i>Halaman Pengguna dan Log Activity Member</i>	77
Gambar H.1 <i>Catatan Hasil Retrospective</i>	118
Gambar C.1 <i>Daftar Modul site time frame yang akan diuji integrasi akhir</i>	95
Gambar D.1 <i>Dashboard Member</i>	100
Gambar D.2 <i>Dokumen Berita member pada Sumber Tribunnews.com Member</i>	100
Gambar D.3 <i>Dokumen pencarian berita member pada semua situs berita, beserta grafik time frame dan durasi kejadian.</i>	101
Gambar D.4 <i>Dokumen pencarian berita member pada situs berita detik.com, beserta grafik time frame dan durasi kejadian.</i>	101
Gambar D.5 <i>Dokumen pencarian berita member pada situs berita liputan6.com, beserta grafik time frame dan durasi kejadian.</i>	102
Gambar D.6 <i>Dokumen pencarian berita member pada situs berita tribunnews.com, beserta grafik time frame dan durasi kejadian.</i>	102
Gambar G. 1 <i>Dokumentasi source code pada pencarian dokumen berita</i>	114
Gambar G. 2 <i>Dokumentasi source code lanjutan pada pencarian dokumen berita</i>	114
Gambar G. 3 <i>Dokumentasi source code untuk menampilkan data pada grafik time series</i>	115
G. 4 <i>Dokumentasi source code lanjutan untuk menampilkan data pada grafik time series</i>	115

Gambar G. 5 Dokumentasi <i>source code</i> proses representasi kata pada dokumen ke database	116
Gambar G. 6 Dokumentasi <i>source code</i> lanjutan proses representasi kata pada dokumen ke database	116
Gambar G. 7 Dokumentasi <i>source code</i> proses representasi dokumen ke dalam vector dengan menggunakan teknik <i>TF-IDF</i> yang disimpan ke database	117
Gambar G. 8 Dokumentasi <i>source code</i> pada teknik <i>text mining</i> dengan menggunakan metode stemming TALA	117



DAFTAR TABEL

Tabel 2.1 Perbedaan fokus penelitian yang dilakukan dengan penelitian terdahulu	7
Tabel 2.2 hasil proses <i>Filtering</i>	14
Tabel 2.3 hasil proses <i>Stemming</i>	15
Tabel 2.4 (Tabel Aturan Melakukan Pemotongan Kata.....	17
Tabel 4.1 Tabel perulangan pengembangan sistem	38
Tabel 4.2 <i>Definisi Aktor</i>	42
Tabel 4.3 <i>Definisi Use Case</i>	42
Tabel 4.4 perhitungan minimal <i>cosine similarity</i> dengan 5 kali percobaan.....	60
Tabel 5.1 Rangkuman perhitungan akurasi pada masing-masing percobaan	68
Tabel A.1 <i>Unit testing Iterasi 01</i>	82
Tabel A.2 <i>Unit testing Iterasi 03</i>	82
Tabel A.3 <i>Unit testing Iterasi 04</i>	83
Tabel A.4 <i>Unit testing Iterasi 05</i>	83
Tabel A.5 <i>Unit testing Iterasi 06</i>	83
Tabel A.6 <i>Unit testing Iterasi 07</i>	83
Tabel A.7 <i>Unit testing Iterasi 01</i>	84
Tabel A.8 <i>Unit testing Iterasi 02</i>	84
Tabel A.9 <i>Unit testing Iterasi 03</i>	85
Tabel A.10 <i>Unit testing Iterasi 01</i>	85
Tabel A.11 <i>Unit testing Iterasi 01</i>	86
Tabel A.12 <i>Unit testing Iterasi 01</i>	86
Tabel A.13 <i>Unit testing Iterasi 02</i>	86
Tabel B.1 <i>Black Box System testing 01</i>	87
Tabel B.2 <i>Black Box System testing 02</i>	89
Tabel B. 3 <i>Black Box System testing 03</i>	92
Tabel B.4 <i>Black Box System testing 04</i>	93
Tabel B.5 <i>Black Box System testing 05</i>	94
Table C. 1 Integrasi modul <i>Text mining with TALA & VSM</i>	96

Tabel C. 2 Integrasi modul <i>Text mining with TALA & Read Json</i>	96
Tabel C. 3 Integrasi modul <i>Split Query & Cosine Similarity With Query</i>	97
Tabel C. 4 Integrasi modul <i>Show Grafik With Data & Cosine Similarity With Query</i>	97
Tabel C.5 Integrasi modul <i>Show Document & Cosine Similarity With Query</i>	98
Tabel C.6 Integrasi modul <i>Show Document & Log Member</i>	99
Tabel E.1 Contoh beberapa dokumen beserta <i>term</i> kata yang dikandung	104
Tabel E.2 Proses Penentuan bobot setiap term dari semua dokumen tersebut. ..	105
Tabel E.3 Proses menghitung kemiripan dokumen dengan Query Q berdasarkan dokumen yang ada $n=5$	106
Tabel E.4 Proses menghitung pembilang semua dokumen dan penyebut antara query dengan semua dokumen $n=5$	107
Table F.1 <i>User Story as Admin</i>	109
Tabel F.2 <i>User Story as Member (Anggota)</i>	111
Tabel F. 3 <i>User Story as Guest (Pengunjung)</i>	113

BAB 1 PENDAHULUAN

Bab ini merupakan langkah awal dari penulisan skripsi ini. Bab ini berisi latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi penelitian, dan sistematika penulisan.

1.1 Latar Belakang

Kemajuan teknologi dan informasi saat ini menyebabkan penggunaan *website* sudah menjadi suatu perhatian oleh banyak orang, mulai dari pengusaha, akademisi, pemasaran, dan media massa. *Website* adalah salah satu teknologi yang saat ini telah berkembang sebagai sarana penyebaran informasi. Sekarang ini, web telah menjadi media yang sangat penting bagi periklanan, dan juga menyediakan berbagai informasi, mulai dari *company profile*, *e-commerce*, *blog*, *news*, *web portal*, dan lain sebagainya. Berdasarkan hasil survey Asosiasi Pengusaha Jasa Internet Indonesia (APJII) tahun 2018, dari sekitar 143 juta orang telah terhubung jaringan internet, 58,01% memanfaatkannya untuk membaca berita *entertainment*, 50,48% untuk membaca berita olahraga, 50,26% untuk membaca berita sosial/lingkungan, 41,55% untuk membaca informasi keagamaan, 36,94% untuk membaca berita politik dan 51,06% untuk membaca berita informasi kesehatan. Selain itu, pentingnya *website* sebagai sarana penyebar informasi juga ditunjukkan dari besarnya kunjungan (*traffic*). Berdasarkan *Alexa Traffic Rank* (ATR), ada empat situs berita di Indonesia yang berada dalam 10 situs teratas di Indonesia dilihat dari aksesibilitasnya, yaitu *Tribunnews.com*, *Detik.com*, *Liputan6.com*, dan *Kompas.com* (www.alexacom, 2018)

Berkembangnya *website* sebagai salah satu bentuk media massa mengakibatkan tajamnya peningkatan jumlah informasi berupa artikel berita. Hasil pengamatan dari tiga situs berita (yaitu *Tribunnews.com*, *Detik.com*, dan *Liputan6.com*) dengan teknik *scraping*, pada bulan Agustus sampai dengan Desember 2018 ditemukan sebanyak 109.061 berita yang telah terpublikasi. Apabila dilihat dari sisi jumlah, pembaca berita sangat cukup mendapatkan referensi dari informasi yang ingin diketahuinya. Namun demikian, dengan

banyaknya informasi tersebut, pembaca juga menemui kesulitan ketika ingin mengetahui rentang waktu sebuah topik yang diberitakan. Proses untuk mengetahui durasi kejadian yang sering dilakukan yaitu, pembaca harus mencari semua dokumen pada situs berita yang ada serta menentukan apakah dokumen berita tersebut sesuai dengan topik yang dimaksud atau tidak. Selanjutnya, pembaca juga diharuskan memetakan seluruh dokumen yang sesuai topik berdasarkan tanggal terbitnya. Langkah tersebut dilakukan untuk mengetahui durasi kejadian secara manual, dengan demikian untuk mengetahui rentang sebuah topik yang diberitakan akan memakan waktu yang lama.

Sistem pencarian merupakan salah satu solusi yang dapat membantu dalam menemukan kembali informasi yang diinginkan (Dasar, Wicaksono, & Sihwi, 2015). Sistem ini berusaha menemukan dokumen yang relevan sesuai dengan *query* masukannya. Salah satu model yang sering digunakan dalam proses pencarian adalah *Vector Space Model (VSM)*. *VSM* merupakan suatu metode yang merepresentasikan sistem temu kembali (*information retrieval*) kedalam vektor yang dilanjutkan dengan proses perhitungan fungsi *similarity* dalam pencocokan beberapa vektor untuk mendapatkan kemiripan dari beberapa vektor yang digunakan. Model ini dipilih karena cara kerja model ini efisien, mudah dalam representasinya serta dapat diimplementasikan pada pencocokan dokumen (*document-matching*). Berdasarkan permasalahan yang telah dijelaskan sebelumnya maka peneliti mencoba mengembangkan sistem pendeteksi rentang waktu pemberitaan topik dalam dokumen berita atau dalam penelitian ini disebut dengan *time-frame* dokumen berita. Sistem ini didasarkan pada sistem temu kembali dengan menggunakan model *Vector Space Model*. Dokumen berita yang ditemu kembali adalah dokumen yang memiliki *similarity* sesuai dengan *threshold* (nilai minimal) yang ditentukan. Sistem ini diharapkan dapat mendeteksi *time-frame* dokumen berita dari tiga situs berita secara akurat dengan cara mengambil tanggal publikasi pada setiap dokumen berita yang relevan untuk ditampilkan durasi kejadiannya, sehingga dapat menjawab kebutuhan informasi tentang rentang waktu sebuah topik yang diberitakan.

1.2 Rumusan Masalah

Berdasarkan uraian yang telah disampaikan dalam latar belakang, penulis mendefinisikan beberapa permasalahan yang harus diselesaikan dalam penulisan ini adalah sebagai berikut:

1. Bagaimana desain sistem deteksi *time frame* konten berita berbasis *vector space model* untuk menampilkan *time frame* dokumen berita berdasarkan pencarian berita pengguna?
2. Berapa nilai *threshold similarity* yang sesuai untuk sistem pencarian dokumen berita berbasis *vector space model*?
3. Berapa tingkat akurasi sistem pencarian dokumen berita berbasis *vector space model*?

1.3 Tujuan Penelitian

Tujuan dalam penulisan ini merupakan jawaban dari perumusan masalah yang telah disebutkan. Tujuan yang ingin dicapai dalam penelitian ini adalah:

1. Mengetahui bagaimana desain sistem deteksi *time frame* konten berita berbasis *vector space model* untuk menampilkan *time frame* dokumen berita berdasarkan pencarian berita pengguna.
2. Mendapatkan nilai *threshold similarity* yang sesuai untuk sistem pencarian dokumen berita.
3. Mengetahui tingkat akurasi sistem pencarian dokumen berita berbasis vektor menggunakan metode *vector space model*.

1.4 Manfaat

Manfaat penelitian ini adalah sebagai berikut:

1. **Bagi Akademis.** Penelitian yang dilakukan memberikan hasil yang mampu menjadi masukan informasi yang terkait dengan judul penelitian

kepada pada umumnya dan kepada Program Studi Sistem Informasi Universitas Jember.

2. **Bagi Peneliti.** Melatih kemampuan dan menerapkan ilmu pengetahuan yang telah diperoleh di Program Studi Sistem Informasi untuk membangun sebuah sistem yang bisa mempermudah proses pencarian dan mengetahui durasi kejadian para pembaca situs berita.
3. **Bagi Pengguna.** Membantu para pencari berita untuk mendapatkan berita yang mereka inginkan secara cepat dan akurat serta mengetahui durasi kejadian yang diinginkan.

1.5 Batasan Masalah

Penulis memberikan batasan masalah untuk objek dan tema yang dibahas sehingga tidak terjadi penyimpangan dalam proses penulisan dan pembuatan aplikasi. Berikut adalah batasan masalah yang dicantumkan:

1. Situs yang menjadi objek penelitian adalah 3 situs berita yaitu www.detik.com, www.tribunnews.com dan www.liputan6.com melalui sistem web *scraping* yang dikembangkan oleh rohim pada tahun 2018.
2. Nilai *threshold* ditentukan berdasarkan *sample* penelitian sejumlah 4812 dokumen yang berasal dari tiga situs berita seperti yang dimaksud pada point satu.
3. Dokumen berita yang menjadi *sample* penelitian dalam proses untuk menguji proses pencarian berita berjumlah 4812 dokumen berita.
4. Nilai akurasi sistem akan diukur menggunakan nilai *precision*, *recal* dan *f-measure*.
5. Pengembangan sistem menggunakan dua program yang berbeda dari fungsionalitasnya untuk efisiensi sumber daya peneliti.
6. Durasi kejadian akan ditampilkan dalam *format* bulan dan hari.

1.6 Sistematika Penulisan

Sistematika penulisan dan keruntutan skripsi ini disusun sebagai berikut :

1. **Pendahuluan.** Bab ini menjelaskan tentang latar belakang, perumusan masalah, tujuan, dan manfaat, ruang lingkup studi dan sistematika penulisan skripsi yang masing-masing tertuang secara eksplisit dalam subbab tersendiri.
2. **Tinjauan pustaka.** Bab ini menjelaskan tentang materi, informasi, tinjauan pustaka, dan studi terdahulu yang menjadi kerangka pemikiran dalam penelitian.
3. **Metodologi penelitian.** Bab ini menjelaskan tentang metode penelitian yang digunakan dalam penelitian.
4. **Pengembangan sistem.** Bab ini menjelaskan tentang perancangan sistem yang dikembangkan. Perancangan sistem dimulai dari analisis sistem, kemudian merancang *business process*, *use case diagram*, *use case scenario*, *activity diagram*, *class diagram* dan *entity relationship diagram* (ERD).
5. **Hasil dan pembahasan.** Bab ini menjelaskan tentang hasil dan pembahasan dari penelitian yang dilakukan.
6. **Penutup.** Bab ini berisi kesimpulan dari penelitian dan saran untuk penelitian selanjutnya.

BAB 2 TINJAUAN PUSTAKA

Bagian ini memaparkan tinjauan yang berkaitan dengan masalah yang dibahas, kajian teori yang berkaitan dengan masalah, kerangka pemikiran yang merupakan sintesis dari kajian teori yang dikaitkan dengan permasalahan yang dihadapi. Teori-teori ini diambil dari buku, literature, jurnal, dan internet.

2.1 Penelitian Terdahulu

Penelitian sebelumnya terkait dengan penerapan model *VSM* adalah penelitian berjudul “Implementasi *Vector Space Model* pada Sistem Pencarian Mesin Karaoke” yang dilakukan oleh Anna & Hendini tahun 2018. Penelitian ini untuk membangun sistem temu kembali pencarian lagu berdasarkan judul, nama penyanyi dan kategori lagu pada mesin pencarian karaoke. Pengguna bisa melakukan pencarian lagu yang relevan sesuai dengan *query* atau pencarian dari pengguna secara cepat dan akurat. Penelitian ini tidak menerapkan model *VSM* untuk pencarian lagu berdasarkan lirik lagu, namun penerapan model *VSM* hanya berdasarkan judul lagu, asal negara dan nama penyanyi saja. Penelitian tidak dilengkapi dengan proses pengukuran nilai akurasi hasil akhir seperti *recall*, *precision*, dan *f-measure*.

Penelitian lainnya mengenai penggunaan metode *VSM* yaitu, “Aplikasi Penentuan Dosen Penguji Skripsi Menggunakan Metode *tf-idf* dan *Vector Space Model*” (Siregar, Sinaga, Arianto, Studi, & Teknik, 2017). Dalam penelitian dilakukan oleh Siregar dkk telah menunjukkan bahwa, implemmtasi *VSM* mampu menentukan dosen penguji skripsi untuk mahasiswa Teknik Informatika STT-PLN berdasarkan kompetensi dosen penguji dan topik penelitian skripsi penelitian mahasiswanya. Sistem yang dikembangkan pada penelitian tersebut menampilkan 3 ranking urutan teratas dosen penguji yang mirip atau sesuai antara kompetensi dosen dan topik skripsi mahasiswanya. Hasil pengujian keakuratan penelitian siregar dkk hanya menggunakan nilai *accuracy*, dengan nilai *accuracy* sebesar 93,22%.

Penelitian lainnya mengenai pengguna metode *VSM* yaitu “Pemanfaatan Sistem Temu Kembali Informasi dalam Pencarian Dokumen Menggunakan Metode Vector Space Model” (Sanjaya, 2018). Penelitian yang dilakukan ini, implementasi *VSM* berhasil mendapatkan dokumen pencarian sesuai dengan kata pencarian yang dimasukkan. Penelitian yang dilakukan tersebut mendapatkan nilai akhir *precision* sebesar 0.44 atau sebesar 44% persen dari menghitung *average precision* 5 hasil pengujian *recall & precision*

Tabel 2.1 Perbedaan fokus penelitian yang dilakukan dengan penelitian terdahulu

Peneliti	Perangkingan dokumen	Tipe pencarian Dinamis	Implementasi pada dokumen besar	Pengujian <i>information retrieval</i>
(Anna & Hendini, 2018)	✓	✓	✗	✗
(Siregar et al., 2017)	✓	✗	✗	✓
(Sanjaya, 2018)	✓	✓	✗	✓
(Wiranto, 2019)	✓	✓	✓	✓

2.2 Software Development Life Cycle (SDLC) Personal Extreme Programming

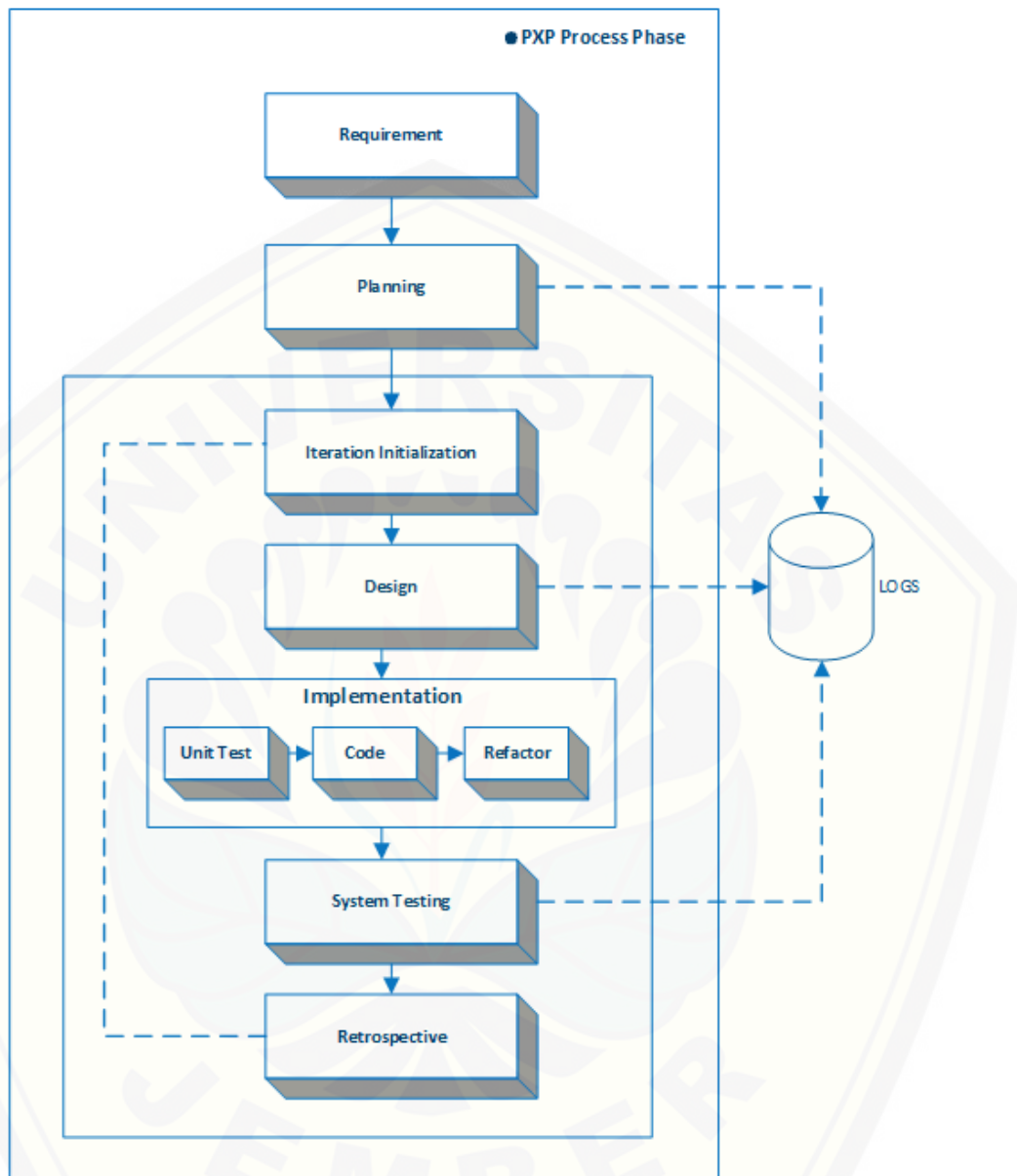
Pembangunan aplikasi dilakukan berdasarkan beberapa proses tahapan yaitu, analisis kebutuhan, desain perancangan aplikasi dan terakhir melakukan proses implementasi atau pembangunan aplikasi yang direncanakan. Beberapa pendekatan yang bisa digunakan untuk melakukan sebuah perancangan aplikasi seperti *software development life cycle*. Peranan SDLC yaitu, mampu menyatakan secara spesifik sasaran apa yang harus dicapai untuk memenuhi kebutuhan proses perancangan aplikasi. Terdapat beberapa jenis pendekatan SDLC yang bisa digunakan untuk membantu proses perancangan aplikasi, diantaranya adalah *personal extreme programming* (PXP).

PXP merupakan salah satu model proses dari *Agile Software Development* yang merupakan salah satu metode dalam pengembangan sistem berbasis *Software*

Development Life Cycle (SDLC). Model pengembangan sistem ini menyederhanakan berbagai tahapan proses pengembangan tersebut agar tercapainya peningkatan efisiensi dan fleksibilitas sebuah proyek pengembangan perangkat lunak. Hal ini sesuai dengan pendapat Dzhurov, Krasteva, & Ilieva tahun 2014 dalam penelitiannya yang berjudul “*Personal Extreme Programming – An Agile Process for Autonomous Developers*” menjelaskan bahwa *personal extreme programming* merupakan proses pengembangan perangkat lunak yang berbasis individual yang mengurangi jumlah dokumentasi dan pemeliharaan.

Model ini memastikan kualitas produk yang baik sesuai dengan perencanaan proyek. Menurut Shahriar Dzhurov, Krasteva, & Ilieva (2014) menjelaskan ada enam prinsip dalam model *personal extreme programming*, yaitu:

1. *Personal extreme programming* membutuhkan pendekatan yang disiplin dan tanggung jawab pada proses pengembangannya
2. Pengembang harus mengukur, melacak, dan menganalisis setiap pekerjaannya.
3. Pengembang harus terus meningkatkan skill kerja berdasarkan kompetensi data proyek yang didapatkan.
4. *Personal extreme programming* melibatkan pengujian berkelanjutan.
5. Secepat mungkin menyelesaikan setiap kesalahan yang ada pada setiap pengerjaannya, untuk meminimalisir biaya yang diperlukan pada akhir proyek.
6. Pengembang harus berusaha melakukan kerja efisien dalam tahap pengembangannya.



Gambar 2.1 Kerangka Kerja *Personal Extreme Programming* (Sumber : Dzhurov, Krasteva, & Ilieva, 2014)

Proses perencanaan pengembangan aplikasi perangkat lunak harus berdasarkan dokumen permintaan yang dikelola pada *Logs*. PXP diawali dengan fase mencari *requirement* dan dilanjutkan dengan menentukan *planning* awal. Proses berikutnya adalah membuat inisialisasi perulangan (*iterasi initialization*) yang menunjukkan awal dari setiap perulangan kemudian membuat design sederhana yang dipilih. Proses ini kemudian dilanjutkan dengan melakukan implementasi yang terdiri dari unit *testing*, *code* dan *refactor*. Setelah proses implementasi selesai, maka dilanjutkan dengan *system testing* untuk menguji semua fitur yang telah dikembangkan. Proses terakhir *retrospective* yang menindak lanjuti apakah semua yang sudah diterapkan memenuhi kebutuhan awal.

2.2.1 Requirement

Proses *requirement* merupakan proses awal pengembangan system. Dalam proses ini, peneliti mencari kebutuhan pengembangan kebutuhan sistem yang akan dikembangkan. Membagi setiap tugas atau *requirement* yang ada menjadi beberapa bagian besar dan kecil. Hal tersebut dilakukan sebagai proses awal proses implementasi sistem akan dilakukan.

2.2.2 Planning

Proses *planning* dimulai dengan membuat rencana dan gambaran sistem yang akan dikembangkan berdasarkan *requirement* yang dihasilkan. Proses ini akan membuat kerangka alur sistem yang akan dikembangkan, sekaligus menentukan mekanisme proses pengembangan yang akan dikerjakan. Oleh karena itu pada proses ini sudah mulai terlihat gambaran sistem yang akan dikembangkan, alur kerja sistem yang akan dikembangkan sekaligus teknik pengembangan yang akan dikerjakan.

2.2.3 Iteration Initialization

Iterasi initialization adalah awal dari setiap perulangan yang ada. Perulangan akan dimulai dengan pemilihan tugas yang akan menjadi fokus pada perulangan tertentu. Perulangan ini bervariasi mulai dari 1 minggu sampai 3 minggu tergantung pada ruang lingkup proyek yang dikerjakan. Setiap perulangan dapat menghasilkan kandidat produk yang akan rilis.

2.2.4 *Design*

Proses *design* bertujuan untuk memodelkan sistem yang akan dikembangkan, mulai dari *class* dan modul yang akan dikerjakan pada fase perulangaan yang sedang berlangsung. Model dikembangkan berdasarkan kebutuhan yang sekarang, tanpa melihat kebutuhan yang akan mendatang. model desain dipilih oleh pengguna, namun disarankan untuk menggunakan alat sesederhana mungkin.

2.2.5 *Implementation*

Proses ini merupakan proses dimana pembuatan kode program dari sistem sesuai dengan desain dan perulangan yang berlangsung. Proses ini terdiri dari 3 sub proses yaitu:

1. *Unit testing* merupakan proses pemeriksaan berdasarkan *test unit* pada sistem yang dikembangkan pada perulangan yang berlangsung.
2. *Code* merupakan proses eksekusi desain sistem kedalam *code* program yang sudah didefinisikan pada fase *planning*.
3. *Refactor* merupakan proses verifikasi apakah solusi yang ditawarkan memenuhi kebutuhan awal pada perulangan yang berlangsung.

2.2.6 *System testing* merupakan proses untuk memastikan semua fitur yang telah dihasilkan pada perulangan tersebut sudah berjalan sesuai yang diinginkan. Mekanisme pengujian ini membantu memvalidasi fungsi keseluruhan sistem. Pada penelitian ini menggunakan *blackbox testing*, *testing* ini dilakukan berdasarkan *requirement*, sehingga luaran yang dihasilkan dari sistem yang dikembangkan sesuai dengan *requirement*.

2.2.7 *Retrospective* merupakan proses menandai akhir dari setiap perulangan. Melakukan proses validasi dari serangkaian kegiatan perulangan yang sudah berlangsung. Proses ini juga melihat apakah waktu pengerjaan sudah sesuai dengan estimasi waktu yang direncanakan. Mekanisme pekerjaan ini untuk mengoreksi apa saja agenda kegiatan yang telah terjadi pada setiap akhir perulangan, dan mengidentifikasi tindakan untuk perbaikan kedepannya. Peneliti menggunakan media membuat dokumentasi ini di website <https://funretro.io>, gambar hasil dokumentasi *retrospective* menggunakan *funretro* bisa dilihat di Lampilan H.

2.3 Situs Berita Online

Seiring dengan semakin pesatnya penggunaan internet, proses untuk mendapatkan informasi yang kita inginkan semakin cepat. Informasi termasuk erat kaitannya dengan suatu instansi yang menyediakan media untuk memaparkan informasi ke publik seperti media pers dan jurnalistik. Situs berita online merupakan salah satu media online yang secara aktif melibatkan media pers dan jurnalistik yang menyajikan suatu konten karya jurnalistiknya berupa berita dan artikel melalui laman website. Karya jurnalistik yang dihasilkan bisa termasuk pelaporan suatu peristiwa yang diproduksi oleh unit atau instansi terkait yang nantinya akan diketahui oleh publik. Situs berita online sangat digemari oleh para pembaca berita, karena dapat diakses oleh siapa saja dan dimana saja hanya dengan terhubung ke jaringan internet kita sudah mengetahui suatu peristiwa yang sedang kita baca. Hal tersebut sesuai dengan hasil jurnal dewan pers pada tahun 2011 yang menunjukkan bahwa masyarakat dalam waktu mendatang, berdasarkan perspektif kebutuhan masyarakat akan terus berkembang dan diminati oleh masyarakat.

Masyarakat tidak hanya membaca berita dari situs berita online, masyarakat bisa mendapatkan informasi melalui media cetak. Media jurnalistik dibagi menjadi tiga kategori yaitu media cetak, media digital dan media online. Media cetak menyebarluaskan informasi kepada publik melalui media tertulis atau gambar pada media cetak sebagai alat penyajian data, media digital menyebarkan informasi melalui perangkat elektronik sebagai alat penyajian data, sedangkan media online menyebarluaskan informasi menggunakan laman website dan internet sebagai alat penyajian datanya.

2.4 Text Mining

Text mining merupakan variasi dari data *mining* yang berusaha menemukan pola yang menarik dari sekumpulan data tekstual yang berjumlah besar (Feldman, 2007). Proses langkah dalam *text mining* terdapat tahapan pemrosesan awal terhadap suatu teks yang disebut *text preprocessing*. *Text preprocessing* merupakan suatu proses pengubahan bentuk data tekstual yang belum terstruktur menjadi data yang terstruktur. Proses *preprocessing* ini meliputi empat tahapan yaitu *casefolding*, *tokenizing*, *filtering*, dan *stemming* (Pressman, 2015).

1. *Casefolding*

Merupakan tahapan awal dari proses *text mining* yang merubah semua karakter yang ada pada data menjadi huruf kecil (*lower case*). Tidak semua karakter pada setiap kata mengandung huruf kecil semua dan mengandung huruf besar semuanya, bahkan sangat memungkinkan kata terbentuk dari kombinasi huruf kecil dan besar. Pada proses implementasi nantinya kesamaan kata jika akan dibandingkan dengan kata lain harus mempunyai kesamaan kata dari segi huruf yang membangun dan dari segi huruf besar dan kecilnya huruf yang dimaksud. Sehingga untuk mengatasi hal semacam itu perlu mengubah karakter diawal agar kesesuaian data dan logika pada proses implementasi tidak terjadi kesalahan.

2. *Tokenizing*

Merupakan tahapan pemotongan string input berdasarkan tiap kata yang menyusunnya. *Tokenizing* secara garis besar memecah sekumpulan karakter dalam suatu teks ke dalam satuan kata, pemotongan yang dilakukan berdasarkan karakter penghubung antara satu kata dengan kata yang lainnya dalam satu kalimat atau dokumen penuh.



Gambar 2.2 Contoh Proses Tokenizing (Sumber : <https://www.geeksforgeeks.org/stringtokenizer-class-java-example-set-1-constructors/>)

3. *Filtering*

Tahap *Filtering* adalah tahap mengambil kata-kata penting dari hasil token yang dihasilkan. Proses ini bisa menggunakan algoritma *stoplist* yaitu membuang kata yang kurang penting. Kata yang kurang penting disini yang dimaksud ialah seperti kata sambung dan keterangan. Semua data pada dasarnya akan mempunyai kata penyusun yang terdiri dari kata sambung dan kata keterangan, sehingga hal ini dianggap termasuk kata tidak penting dan juga nantinya akan mempengaruhi komputasi nilai data yang diproses. Berikut adalah contoh tabel proses *filtering* yang bisa dilihat pada Tabel 2.2.

Tabel 2.2 hasil proses *Filtering*

Kata pada dokumen (hasil token)	Hasil <i>Filtering</i>
Politik	Politik
Indonesia	Indonesia
sudah	mengalami
mengalami	polemik
polemik	tahun
mulai	lalu
dari	sekarang
tahun	
lalu	
sampai	
sekarang.	

4. *Stemming*

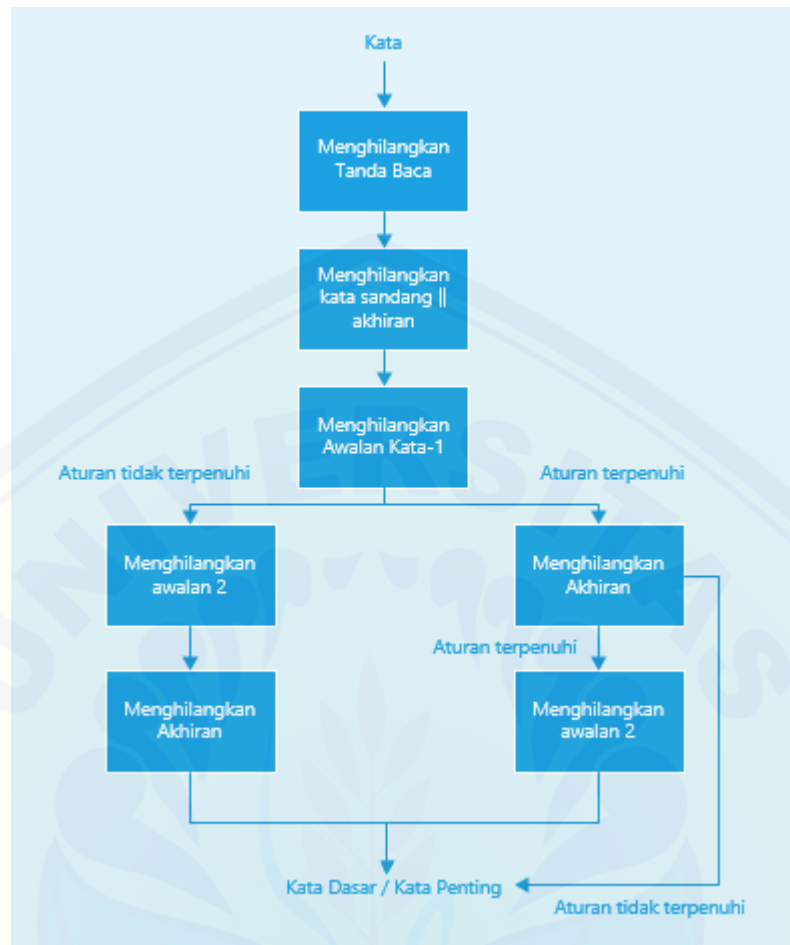
Teknik *Stemming* diperlukan untuk memperkecil jumlah indeks yang berbeda dari suatu dokumen, berdasarkan kata penyusun dari dokumen. *Stemming* pada akhirnya akan memperoleh kata dasar atau kata yang bukan tergolong dalam kata sambung dan kata keterangan, karena sudah mengalami proses pemeriksaan pada proses *filtering*. Kata akhir yang dihasilkan pada proses ini sudah dianggap kata penting yang menggambarkan dokumen pembentuknya. Berikut tabel contoh hasil akhir teknik *stemming* yang bisa dilihat pada Tabel 2.3.

Tabel 2.3 hasil proses Stemming

Kata pada dokumen (hasil <i>filtering</i>)	Hasil <i>Stemming</i>
Bencana	Bencana
Mengakibatkan	Akibat
Warga	Warga
Sekitar	Sekitar
Dievakuasi	Evakuasi

2.5 *Stemming TALA*

TALA merupakan adopsi dari algoritma stemmer bahasa inggris terkenal porter stemmer. Stemmer ini menggunakan rule base analisis untuk mencari *root* sebuah kata (Utomo, 2013). *Stemmer* ini sama sekali tidak menggunakan kamus sebagai acuan, yaitu hanya berdasarkan proses pemenggalan kata berdasarkan bahasa yang digunakan. Setiap bahasa yang digunakan untuk melakukan proses *text mining* mempunyai aturan proses pemenggalan kata menjadi kata pembangunnya. Berikut adalah alur proses pemenggalan kata pada proses *stemming* menggunakan metode *TALA* dapat dilihat pada Gambar 2.3.



Gambar 2.3 Proses Algoritma *Stemming Tala* (Sumber : Utomo, 2013)

Stemming TALA mengikuti aturan tata bahasa yang digunakan pada saat melakukan proses *teks mining*. Berikut adalah komponen struktur pembentukan kata bahasa Indonesia.

[awalan-1] + [awalan-2] + dasar + [akhiran] + [kepunyaan] + [sandang]

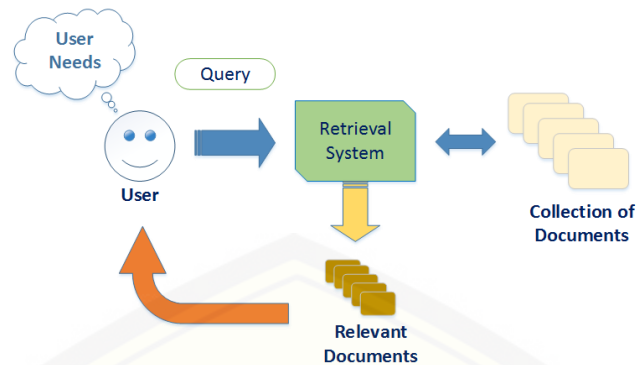
Susunan komponen struktur kata diatas bisa berubah sesuai dengan jenis bahasa yang digunakan. Oleh karena itu berdasarkan struktur susunan struktur kata diatas metode *TALA* diharapkan mampu memperoleh kata hasil *stemming* yaitu kata dasar. Aturan yang digunakan pada *Stemming TALA* dapat dilihat pada Tabel 2.4

Tabel 2.4 (Tabel Aturan Melakukan Pemotongan Kata (Sumber : Utomo, 2013)

Nama Aturan	Kata	Ganti dengan
Menghilangkan kata sandang	“Lah”, “kah”, “pun”, “nya”, “ku”, “mu”	“ ” “ ” “ ” “ ” “ ” “ ” “ ” “ ” “ ”
Menghilangkan awalan 1	“meng”, “menya”, “menyi”, “menyu”, “menye”, “menyo”, “meny”, “men”, “mema”, “memi”, “memu”, “memo”, “mem”, “me”, “peng”, “penya”, “penyi”, “penyu”, “penye”, “penyo”, “peny”, “pen”, “pema”, “pemi”, “pemu”, “peme”, “pemo”, “pem”, “di”, “ter”, “ke”, “ber”, “bel”, “be”, “per”, “pel”, “pe”	“ ” “s” “s” “s” “s” “s”, “s” “ ” “p” “p” “p” “p”, “p” “ ” “ ” “ ” “s” “s”, “s” “s” “s” “s” “ ” “p”, “p” “p” “p” “p” “ ” “ ” “ ” “ ” “ ” “ ” “ ” “ ” “ ”
Menghilangkan akhiran	“Kan”, “an”, “i”	“ ” “ ” “ ” “ ”

2.6 Information Retrieval

Information Retrieval (IR) atau sering disebut “temu kembali informasi” adalah ilmu yang mempelajari prosedur-prosedur dan metode-metode untuk menemukan kembali informasi yang tersimpan dari berbagai sumber (*resources*) yang relevan dari koleksi sumber informasi yang dicari atau dibutuhkan. Dalam pencarian data, beberapa jenis data dapat ditemukan diantaranya teks, tabel, gambar, video, suara. Adapun tujuan dari *Information Retrieval* ialah untuk memenuhi informasi pengguna dengan cara menemukan kembali dokumen yang relevan atau mengurangi dokumen pencarian yang tidak relevan. Secara umum gambar alur *information retrieval* dapat dilihat pada Gambar 2.4.

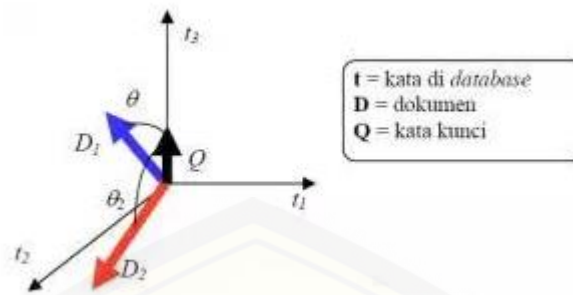


Gambar 2.4 Diagram alur *Information Retrieval* (Sumber : <http://ir.cs.ui.ac.id/new/>)

Berdasarkan gambar diatas, proses menampilkan dokumen yang relevan kepada *user* dengan cara pertama mengambil *query*, dilanjutkan dengan sistem yang menggunakan model *information retrieval* kemudian *query* akan diperiksa ke dokumen yang disimpan, kemudian dikembalikan kembali. Pada saat dokumen dari *collection of Document* didapatkan maka akan diperiksa pada sistem untuk dilihat sejauh mana kemiripan dokumen dengan *query*, lalu sistem akan menampilkan dokumen yang relevan sesuai dengan *query user*nya.

2.7 Vector Space Model

Salah satu cara untuk mencari dokumen dari beberapa dokumen yang ada secara cepat dan tepat adalah menggunakan *information retrieval*. *VSM* merupakan salah satu model dari *informaion retrieval* yang merepresentasikan dokumen dan *query* sebagai *vector* multidimensi (Aziz, Saptono, & Suryajaya, 2016). Pada *vector* multidimesi yang dibentuk, perhitungan kemiripan *query* dengan *vector* dilihat berdasarkan nilai sudut paling kecil dari representasi dokumen dan *query*. *Query* dan dokumen dianggap sebagai *vector - vector* pada ruang *n*-dimensi, dimana *t* adalah jumlah dari seluruh *term* yang ada dalam *leksikon*. *Leksikon* adalah daftar semua *term* yang ada dalam *indeks* (Abdul Azis Abdillah, 2013). Selanjutnya kemiripan antara *vector* dan *query* akan dihitung menggunakan *cosine similarity*.



Gambar 2.5 Representasi dokumen dan query pada ruang vektor (Sumber : <https://commanderx78.wordpress.com/2012/09/24/model-vector-space/>)

Vector space model menerapkan konsep menghitung *TF-IDF* pada saat merepresentasikan dokumen dan *query* sebagai *vector* multidimensi. *TF-IDF* merupakan representasi keseluruhan dokumen kedalam suatu dimensi *vector*. Hal ini dilakukan agar dokumen yang menjadi bahan penelitian sudah digambarkan kedalam bentuk ruang *n-vector* multidimensi. Proses representasi tersebut memerlukan beberapa komponen diantaranya *TF* atau *Term Frequency*. *TF* ini merupakan nilai dari frekuensi kata *kt* didalam dokumen *dk*. Hal tersebut dilakukan untuk mendapatkan bobot *kt* pada masing-masing *dk*. Komponen selanjutnya ialah *DF* atau *Document Frequency* yang dihitung berapa kali kata *kt* muncul pada semua dokumen *sdk* yang digunakan, Nilai nol berarti berarti kata *kt* tidak terdapat dalam dokumen manapun. Setiap dokumen yang mengandung *kt(word)* akan diperiksa pada dokumen lain untuk menentukan bobot kata secara keseluruhan pada dokumen lainnya.

Komponen selanjutnya dari *VSM* yaitu *IDF* atau *Inverse Document Frequency*. *IDF* akan dihitung agar memperkecil nilai bobot kata *kt* berdasarkan kemunculannya pada dokumen lain, sehingga dapat dipastikan ketika kata *kt(word)* muncul dibanyak dokumen maka nilainya akan semakin kecil. Berikut adalah persamaan untuk menghitung nilai *IDF-kt* :

$$idf_i = \log \left(\frac{n}{df_i} \right) \dots \dots \dots (1)$$

Persamaan 2.1 menghitung nilai *IDF-kt*

Keterangan :

idf_i : *Invest Dokumen Frekuensi*

n : Jumlah semua dokumen yang tersedia

df_i : *document frekuensi* pada term ke- i

Komponen terakhir dari *VSM* untuk merepresentasikan dokumen yaitu menggunakan nilai *TF-IDF* atau *Term frequency inverse document*. Nilai ini merupakan nilai bobot akhir pada masing-masing kata dan frekuensinya dalam dokumen yang terlibat. Nilai ini digunakan multak pada *VSM* untuk bahan mencari kemiripan dengan *vector query*. Berikut persamaan untuk menghitung nilai *TF-IDF*

$$tf - idf_{ij} = tf_{ij} \times idf_{ij} \dots \dots \dots (2)$$

Persamaan 2.2 menghitung nilai *TF-IDF*

Keterangan :

$tf - idf_{ij}$: *term frequency-inverse document frequency* masing-masing dokumen

tf_{ij} : *term frequency* pada dokumen yang memiliki *term*

df_{ij} : *document frekuensi* pada term ke- i setiap dokumen

Proses ini telah merepresentasikan dokumen kedalam ruang n -vector, jadi dokumen yang sudah diproses menggunakan metode ini nantinya secara umum akan digambarkan dalam bentuk garis-garis dan dihitung tingkat kemiripannya dengan kata kunci yang masuk menggunakan *cosine similarity*.

2.8 Cosine Similarity

Salah satu teknik untuk menghitung kemiripan kata pada pencarian dokumen adalah *cosine similarity*. Konsep dari *cosine similarity* yaitu menghitung nilai *cosinus* sudut antara beberapa vektor yaitu jika diberikan dokumen yang direpresentasikan oleh vektor d_j dan *query* q dan t term yang diekstrak dari database, maka rumus *cosine similarity* dapat didefinisikan sebagai berikut:

$$\cos \left(\vec{ij} \right) = \frac{\sum_k (d_{ik} d_{jk})}{\sqrt{\sum_k (d_{ik}^2)} \sqrt{\sum_k (d_{jk}^2)}} \dots \dots \dots (3)$$

Persamaan 2.3 rumus *cosine similarity*

Keterangan :

$\text{Cos} \left(\frac{\rightarrow}{ij} \right)$: nilai <i>cosine similarity</i> antara query dengan dokumen berita
$\sum_k (d_{ik} d_{jk})$: total jumlah nilai <i>tf-idf</i> setiap <i>term query</i>
$\sqrt{\sum_k (d_{ik}^2)} \sqrt{\sum_k (d_{jk}^2)}$: total jumlah kudrat <i>tf-idf term query</i> pada setiap dokumen yang mengandung <i>term</i>

Setelah ditemukan nilai similarity antara dokumen dengan dan kata pencarian maka perlu dihitung sejauh mana tingkat akurasi pencarian dokumen yang berhasil ditampilkan. Salah satu pengujian sistem temu kembali menggunakan dengan pengujian *recall* dan *precision* dan *f-measure*, karena dengan menggunakan metode perhitungan ini akan mengetahui performa sistem secara keseluruhan dengan nilai yang dapat berubah-ubah.

2.9 Time frame dokumen berita

Dokumen berita online yang dipublish ke internet akan mempunyai tanggal, hari dan jam berita pertama kali waktu publish. Waktu pertama kali berita publish pada situs berita, akan mewakili kapan kejadian sebenarnya yang sedang terjadi. Oleh karena itu durasi waktu suatu kejadian yang terjadi bisa diketahui atau dihitung dari beberapa berita yang membahas mengenai kejadian tertentu. *Time frame* dokumen berita berarti pada akhirnya akan menampilkan grafik tanggal berita terkait kejadiannya, pada penelitian ini kejadian akan sama arti sebagai *query* pengguna atau kata kunci pencarian. *Time frame* ini juga akan menampilkan *frequency* berapa kali ada suatu kejadian yang terjadi yang diwakilkan dari tanggal berita publish, serta akan menghitung total keseluruhan berapa hari suatu kejadian itu sedang dibicarakan oleh masyarakat.

2.10 Pengujian kualitas *information retrieval*

Salah satu pengujian kualitas sistem temu kembali adalah dengan pengujian *recall*, *precision* dan *f-measure*. *Precision* merupakan suatu acuan ketepatan atau ketelitian, dari hasil akhir yang dihasilkan, sedangkan *recall* itu sendiri merupakan tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. Nilai

precision merupakan komposisi dari dokumen yang berhasil ditampilkan oleh sistem, sementara nilai *recall* merupakan komposisi dari dokumen yang relevan sesuai kata kunci pencarian yang berhasil ditampilkan oleh sistem (Sanjaya, 2018). Nilai *f-measure* digunakan untuk mengetahui keakuratan mengenai penggunaan model *Vector Space Model* yang dikombinasikan dengan teknik perhitungan *TF-IDF* dan *Cosine Similarity*. Sehingga proses akhir dalam pengembangan sistem ini nantinya akan menghitung seberapa besar nilai *recall*, *precision* dan *f-measure* dalam hal mencari dokumen yang relevan sesuai dengan kata pencarian. Berikut adalah rumus untuk mencari nilai *recall*, *precision* dan *f-measure*:

$$R = \frac{\text{jumlah Dokumen relevan yang ditampilkan } |RA|}{\text{total jumlah dokumen relevan } |R|} \dots\dots\dots(4)$$

Persamaan 2.4 Rumus untuk mencari nilai *recall*

$$P = \frac{\text{jumlah Dokumen relevan yang ditampilkan } |RA|}{\text{total jumlah dokumen hasil temu kembali } |A|} \dots\dots\dots(5)$$

Persamaan 2.5 Rumus untuk mencari nilai *precision*

$$F = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \dots\dots\dots(6)$$

Persamaan 2.6 Rumus untuk mencari nilai *f-measure*

BAB 3 METODOLOGI PENELITIAN

Bab ini menjelaskan tentang gambaran tahapan yang sistematis yang dilakukan untuk menganalisa data untuk menjawab perumusan masalah sehingga dapat mencapai tujuan sebenarnya dari penelitian. Pada metodologi penelitian akan dijelaskan tentang tahapan dari penelitian.

3.1 Jenis Penelitian

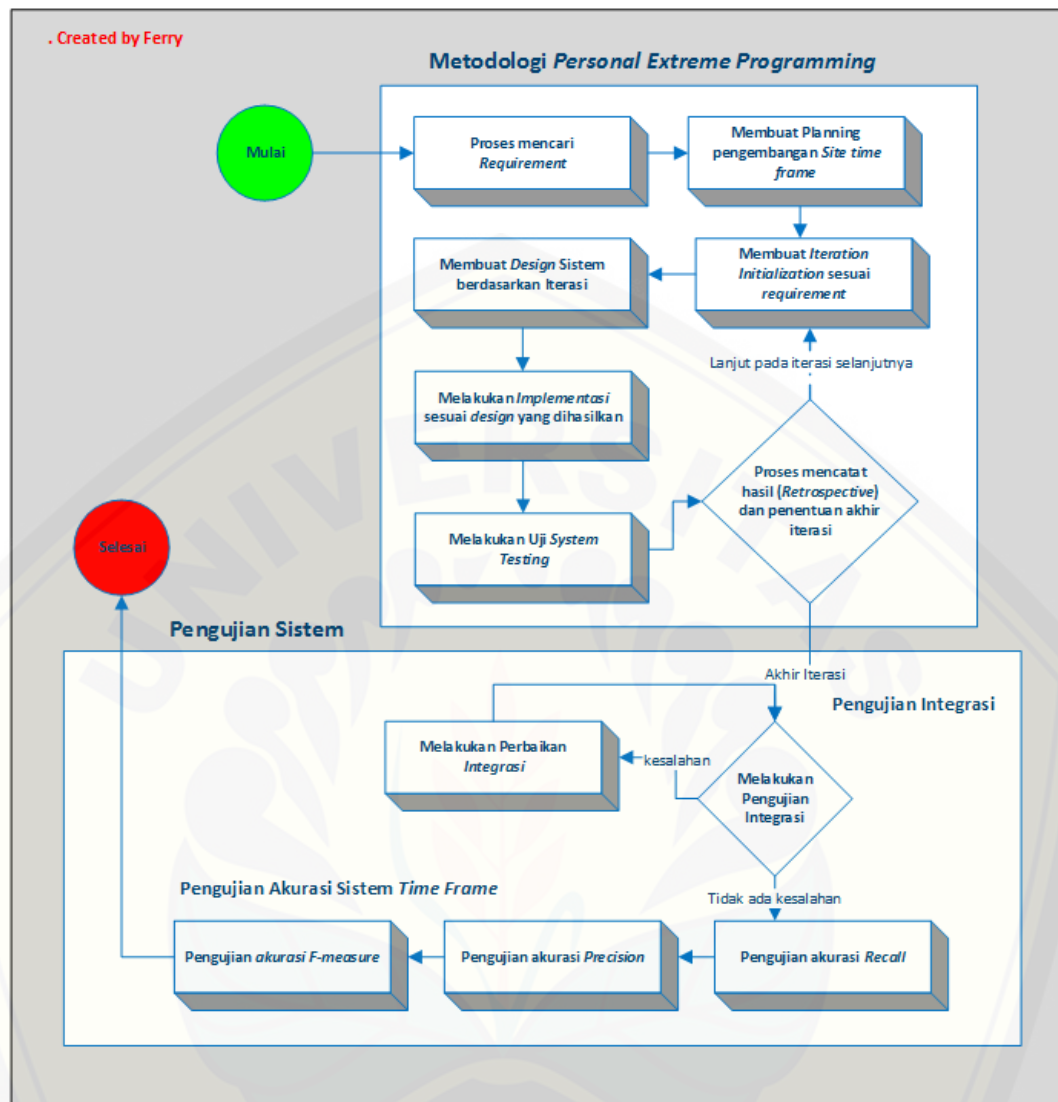
Penelitian yang dilakukan menggunakan jenis penelitian pengembangan. Penelitian ini nantinya diharapkan mampu menciptakan suatu sistem yang mampu memecahkan masalah yang berkaitan dengan upaya inovatif dan penerapan teknologi informasi. Jenis penelitian ini sudah sering digunakan pada dunia pendidikan. Tujuan dari penelitian ini nantinya akan meningkatkan efektivitas para pembaca situs berita online.

3.2 Waktu Penelitian

Waktu penelitian dilakukan selama 4 bulan dimulai pada bulan November 2018 sampai dengan bulan Februari 2019.

3.3 Tahapan Penelitian

Tahapan penelitian merupakan urutan langkah penelitian yang dilakukan. Penelitian ini dilakukan dalam beberapa tahapan. Tahapan dalam penelitian ini digambarkan pada Gambar 3.1.



Gambar 3.1 Diagram Alur Tahapan Penelitian

Gambar 3.1 menggambarkan tahapan penelitian yang akan dilakukan peneliti. Langkah awal yang akan dilakukan yaitu *requirement*, *planning site time frame* kemudian dilanjutkan dengan *iteration initialization*, kemudian mendefinisikan *design* sistem *time frame*, implementasi desain kedalam kode program, kemudian dilanjutkan dengan system testing pada setiap perulangan yang terjadi. Langkah terakhir pada model *personal extreme programming* yaitu *retrospective* untuk memeriksa semua kejadian yang terjadi pada akhir setiap perulangan. Kemudian proses selanjutnya melakukan uji integrasi dengan menggunakan teknik *black box* dan apabila ada yang perlu perbaikan akan diproses

kembali untuk dilakukan perbaikan sampai lolos pada fase pengujian integrasi. Setelah lolos pengujian dilanjutkan dengan proses menguji akurasi luaran sistem *time frame* menggunakan perhitungan *recall*, *precision* dan *f-measure*.

3.4.1 Model *personal extreme programming*

Pada proses ini peneliti melakukan proses pengembangan sistem menggunakan *system development life circle SDLC personal extreme programming (PXP)*. Pada fase ini peneliti membagi beberapa kegiatan kedalam sub kegiatan yaitu:

1. *Requirement*

Pada proses ini peneliti melakukan proses mencari informasi terkait permasalahan yang dihadapi oleh *client* atas nama Achmad Maududie, ST, M.Sc sekaligus mencari informasi tentang kebutuhan aplikasi yang akan dikembangkan. Pemaparan *requirement* ditulis menggunakan model *story board*, sehingga nantinya akan terlihat kebutuhan sistem dan pengguna yang akan terlibat dalam sistem yang akan dikembangkan.

2. *Planning Site Time Frame*

Pada proses ini peneliti melakukan rencana pengembangan sistem yang akan dilakukan. Perencanaan mulai dari fitur apa saja yang mungkin dibutuhkan untuk mengatasi suatu permasalahan yang didapatkan pada saat menemukan *requirement* beserta kerangka pengembangan sistem yang akan dikerjakan. Oleh karena itu, pada proses ini peneliti sudah mulai membuat gambaran sistem yang akan dikembangkan, alur kerja sistem yang akan dikembangkan sekaligus teknik pengembangan yang akan dikerjakan.

3. *Iteration initialization*

Pada proses ini peneliti melakukan serangkaian proses membuat list mengenai apa saja yang akan menjadi hasil akhir atau luaran dari sistem yang akan dikembangkan.

4. *Design*

Pada proses ini peneliti membuat desain sistem *time frame* menggunakan beberapa desain diantaranya seperti *bussinees process*, *use case diaigram*, *CRC (Class Responsible and Collabolator) Card*, *business process model and notation (BPMN)* dan *entitty relationship diaigram (ERD)*.

a. *Diagram Input Output*

Diagram input output adalah diagram yang dapat menggambarkan proses dari sebuah sistem yang meliputi *input*, *output*, dan *goal* yang merupakan tujuan dari sebuah sistem yang dibangun. *Bussines Process* digambarkan dengan diagram *Eriksson-Penker model* karena *Eriksson-Penker model* mampu memvisualisasikan dan mengkomunikasikan proses bisnis dan aliran informasi yang diperlukan dalam suatu organisasi dengan baik (Eriksson & Penker, 2000). *Bussiness process* pada penelitian ini akan dibuat berdasarkan *requirement* yang dihasilkan secara keseluruhan. Sehingga akan terlihat bagaimana arus data *input* dan *ouput* pada sistem.

b. *Usecase Diagram*

Usecase Diagram adalah representasi visual yang mewakili interaksi antara pengguna dan sistem informasi dalam UML. *Usecase* menggambarkan interaksi antara *user* dengan sistem, dan dapat menggambarkan hak akses *user*. *Usecase diagram* pada penelitian ini akan dibuat berdasarkan *bussiness process* yang dihasilkan secara keseluruhan. Setelah arus data digambarkan dari luaran dan masukan data pada sistem, maka dari gambaran tersebut akan direpresentasikan kembali menjadi *usecase diagram* yang akan menggambarkan fitur apa yang tersedia beserta aktor siapa saja yang bisa mengaksesnya.

c. *Businness Process Model and Notation (BPMN)*

Suatu proses representasi yang dikembangkan untuk memodelkan proses bisnis yang ada terkait dengan pengembangan aplikasi yang akan dikembangkan secara keseluruhan kedalam bentuk notasi. BPMN pada penelitian ini akan dikembangkan berdasarkan hasil dari *usecase digram*. Dari gambaran yang ada pada *usecase diagram* akan direpresentasikan kembali ke diagram BPMN berdasarkan keterkaitan antara fitur satu dengan fitur yang lainnya secara keseluruhan. Sehingga dengan demikian akan memudahkan proses pembuatan kerangka sistem.

d. *Class Diagram*

Class Diagram merupakan proses menggambarkan suatu *class*, *package* dan *object* yang ada pada aplikasi yang akan dikembangkan beserta sifat dari setiap elemen yang secara keseluruhan kedalam bentuk diagram. *Class diagram* pada

penelitian ini akan dibuat berdasarkan gambaran dokumen *usecase* dan *bussiness process* yang sudah didefinisikan. *Class diagram* dibuat lebih pada fase panduan untuk melakukan proses eksekusi desain kedalam kode program. Sehingga dengan gambaran desain *usecase* dan *bussiness process* yang sudah didefinisikan akan menghasilkan *class*, *method* apa saja yang mungkin akan digunakan pada fase *coding* nantinya.

e. *CRC card*

Merupakan kumpulan dari kartu indeks standart yang telah dibagi menjadi tiga bagian, yaitu *Class*, *Responsible* dan *Collabulator*. Dokumen desain ini dikembangkan setelah selesai membuat *class diagram* sistem. *CRC* akan menggambarkan *class* apa yang ada pada sistem, *class* apa saja yang terlibat dalam proses tersebut, serta fungsi dari *class* itu sendiri. *CRC* juga menggambarkan kelas-kelas lain ke mana ia mengirim pesan dan di mana dia bergantung untuk melakukan tanggung jawabnya (Suruali et al., 2010). *CRC card* pada penelitian akan dibuat berdasarkan desain *class diagram*. Setelah mendapatkan desain *class* maka langkah berikutnya untuk mempermudah proses panduan saat koding yaitu membuat *CRC card* yang berisi *class* apa saja yang ada dan *class* apa saja yang terlibat pada *class* tersebut serta *class* tersebut berfungsi sebagai apa pada sistem yang dikembangkan.

f. *Entity Relationship Diagram (ERD)*

Entity Relationship diagram digunakan untuk menggambarkan struktur *database* beserta hubungan yang ada pada setiap tabel dalam *database* pada sistem yang akan dikembangkan. *ERD* pada penelitian ini dibuat berdasarkan *class diagram* dan *CRC card*. Berdasarkan gambaran dan definisi *class* yang sudah ada, maka selanjutnya harus membuat desain *ERD* sebagai model penyimpanan data pada sistem.

5. *Implementation*

Proses implementasi merupakan proses membangun program sesuai dengan *design* yang didapatkan mulai dari *diagram input output* sampai *ERD*. Fase ini terdiri dari 3 sub proses yaitu *unit testing*, *code*, dan *refactor* yang dilakukan menjadi satu kesatuan pada proses implementasi. proses ini peneliti mulai membangun sistem secara keseluruhan terkait dengan kebutuhan sistem yang akan dikembangkan.

1. *Unit testing*

Pada fase implementasi sub proses unit testing, peneliti melakukan uji setiap unit fungsionalitas pada perulangan yang sedang berlangsung. Proses ini memastikan bahwasanya setiap unit pada sistem berjalan sesuai yang diinginkan dan untuk meminimalisir terjadinya kesalahan unit.

2. *Code*

Pada fase ini peneliti melakukan eksekusi *code* program untuk membangun sistem yang dikembangkan berdasarkan desain yang ada. Peneliti menggunakan bahasa pemrograman *c#* dengan tipe website *ASP .net* desain arsitektur pemrograman UDC (*user display control*).

3. *Refactor*

Pada fase ini peneliti melakukan proses pemeriksaan *coding*. Teknik *refactoring* dilakukan untuk merampingkan metode, menghapus duplikasi kode, dan membuka jalan untuk perbaikan di masa mendatang agar mudah dilakukan proses perbaikan.

6. *System testing*

Proses ini merupakan proses menguji setiap fitur yang ada pada perulangan yang sedang berlangsung. Fitur yang diuji berdasarkan desain sistem yang sudah dibuat, yaitu *usecase diagram*. Pengujian ini dilakukan agar luaran yang diharapkan pada perulangan tertentu berjalan sesuai dengan *requirement* awal. Peneliti menggunakan teknik pengujian *black box* untuk menguji setiap unit pada fitur yang ada pada sistem. *Black box* dibuat dengan menggunakan tabel yang berisi tentang deskripsi unit pada perulangan tertentu fitur yang diuji beserta status akhir pengujian.

7. *Retrospective*

Proses ini merupakan proses evaluasi kegiatan secara keseluruhan pada setiap perulangan yang ada. Peneliti melakukan pemeriksaan terhadap perulangan pengembangan sistem yang terjadi, dan melakukan pertimbangan mengenai pada saja yang perlu diperbaiki dan dikembangkan kembali pada perulangan selanjutnya.

3.4.2 Pengujian Sistem

Pada proses ini peneliti melakukan pengujian sistem dari segi program apakah semua fungsionalitas dari program sudah baik. Pada proses ini peneliti membagi dua proses pada pengujian sistem yaitu.

1. Pengujian Sistem *integrasi*

Proses ini merupakan proses menguji setiap fitur yang ada pada sistem. Fitur yang diuji berdasarkan desain sistem yang sudah dibuat, yaitu *usecase diagram*. Pengujian ini dilakukan agar luaran yang diharapkan pada setiap perulangan ketika diintegrasikan tidak menimbulkan kesalahan. Peneliti menggunakan tabel yang berisi tentang deskripsi fitur atau komponen apa saja yang terlibat pada semua perulangan yang akan diuji beserta status akhir pengujian.

2. Pengujian Akurasi Sistem *Time Frame*

Pada tahap ini dilakukan uji coba sistem yang telah dibangun. Kegiatan pengujian yang dilakukan bertujuan untuk mengukur tingkat validasi luaran sistem yang telah dibangun. Pengujian yang dilakukan terkait berapa persen akurasi aplikasi yang dikembangkan. Pengujian yang dimaksud didasarkan pada hasil perhitungan *recall* dan *precision* dan *f-measure*.

Tahap pengujian dilakukan dengan memasukkan kata pencarian (*query*) pada masing-masing situs berita yaitu tribunnews.com dengan total berita sebanyak 1.789, detik.com total berita sebanyak 1891 dan liputan6.com sebanyak 1132 berita kemudian diperiksa dokumen yang ditampilkan berdasarkan *query*. Implementasi *vector space model* diletakkan pada semua situs berita yang dituju, yaitu tribunnews.com, detik.com dan liputan6.com sekaligus ditempatkan pada semua situs berita, artinya semua dokumen dari ketiga situs berita dikumpulkan, kemudian diberi akses untuk mencari berita pada semua dokumen situs berita tersebut. *recall* digunakan untuk menguji berapa persen sistem dapat menampilkan dokumen hasil pencarian, *precision* digunakan untuk menguji berapa persen tingkat akurasi kesesuaian dokumen yang ditampilkan sesuai dengan kata pencarian, sedangkan *f-measure* digunakan untuk menguji kesesuaian metode yang digunakan dalam menampilkan dokumen berita relevan.

Proses perhitungan ketiganya dilakukan dengan cara memasukkan 5 *query* yang berbeda pada masing-masing situs berita yang dituju secara acak. Dari masing-masing percobaan tersebut kemudian akan dihitung masing-masing nilai *recall*, *precision* & *f-measure* pada masing-masing *query* yang dimasukkan, sehingga pada proses akhir akan menghasilkan 5 nilai *recall*, 5 nilai *precision* dan nilai *f-measure* akhir. Akurasi yang diuji pada sistem ini mengacu pada masing-masing nilai yang didapatkan tersebut. Peneliti akan menentukan nilai *recall* akhir dari rata-rata 5 hasil percobaan *recall* sebelumnya yang telah dilakukan termasuk juga untuk nilai *precision* akhir yang dihitung dari rata-rata 5 hasil percobaan *precision* sebelumnya dan *f-measure* akhir dihitung dari rata-rata *recall* akhir dan *precision* akhir hasil percobaan sebelumnya.

BAB 4 PENGEMBANGAN SISTEM

Bab ini akan membahas tentang pengembangan sistem *site time frame* dengan metode *vector space model*. Tahap pengembangan dilaksanakan berdasarkan model *personal extreme programming*, dimulai dari *requirement*, *planning*, *iteration initialization*, *design*, *implementation*, *system testing*, *retrospective* dan terakhir dilakukan uji integrasi akhir dari semua modul sistem yang sudah dikerjakan.

4.1 Requirement

Tahap *requirement* merupakan tahapan yang penting dalam pengembangan sebuah sistem informasi. *Requirement* akan dianggap suatu masalah yang harus diselesaikan pada pengembangan sistem yang akan dilakukan. Beberapa *requirement* yang dihasilkan oleh peneliti ditulis menggunakan dokumen *user story*. Menurut Kent Beck pada penelitiannya tahun 1999 memunculkan istilah dengan *User Stories* untuk fitur produk. Dia menggambarkan bahwa *User Story* diceritakan dari perspektif pengguna mengenai apa yang dia inginkan agar lebih seperti yang dapat dilakukan pada sistem. Berikut hasil dokumen *user story* untuk aktor admin website dapat dilihat pada lampiran F.

4.2 Planning

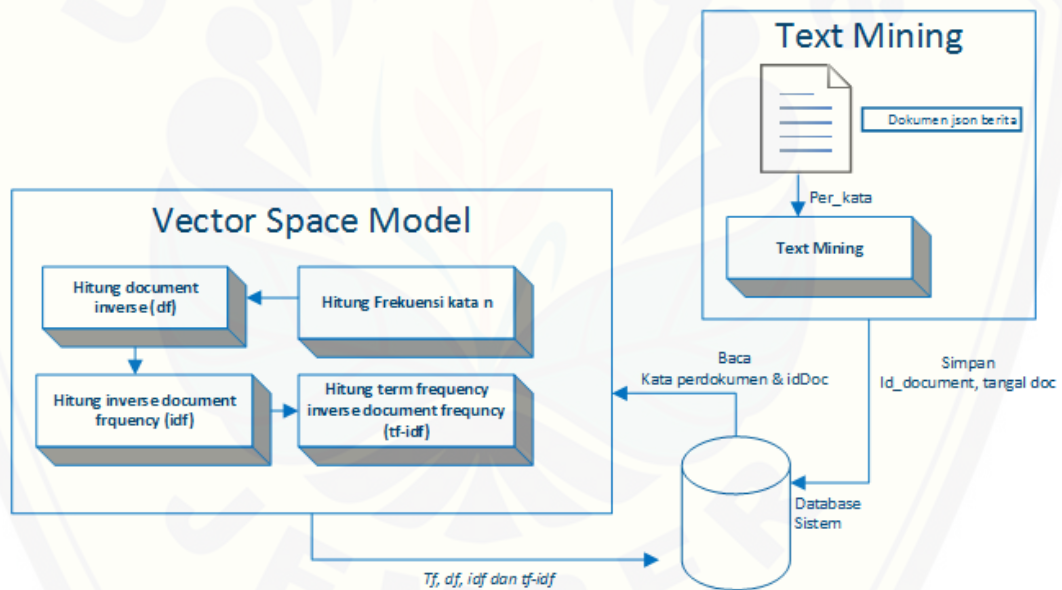
Tahap *planning* merupakan tahapan mulai merencanakan suatu kebutuhan sistem dari aspek pengembangan berdasarkan *requirement* yang didapatkan. Beberapa *planning* dalam penelitian ini diantaranya. Aplikasi yang akan dikembangkan dengan menggunakan metode *Text Mining* dengan teknik *stemming* *TALA & Vector Space Model* untuk menampilkan dokumen berita sesuai dengan pencarian yang dilakukan oleh pengguna serta menampilkan durasi waktu berita dengan cara mengambil tanggal publikasi berita yang ditampilkan dalam bentuk grafik *time series*.

Proses pengambilan data dokumen berita dilakukan dengan menyimpan dokumen berita berbasis *Json* yang didapatkan dari penelitian terdahulu terkait

Scrapping dokumen berita oleh Rohim, 2018. Sistem yang dikembangkan terbagi menjadi dua sub sistem, yaitu *consoleAps* dan *site time frame*. Pengembangan kedua sistem diharapkan mampu membagi kinerja berdasarkan fungsional dari kedua sub sistem.

4.2.1 ConsoleAps

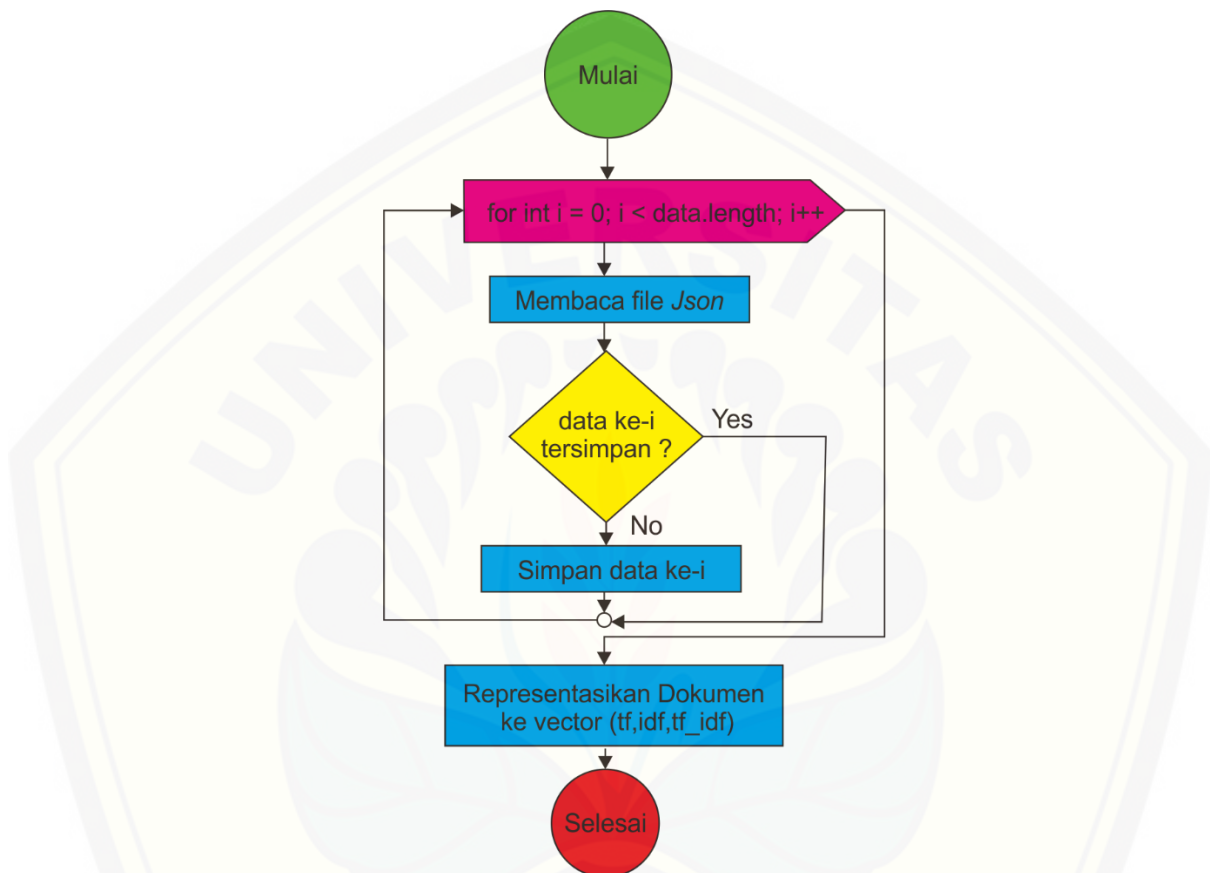
Sub sistem ini melakukan serangkaian proses seperti *text mining* atau mengolah kata pada setiap dokumen, kemudian setelah kata diolah dilanjutkan dengan merepresentasikan dokumen hasil *text mining* kedalam ruang dimensi *vector* berdasarkan perhitungan *TF-IDF*. Serangkaian proses berjalan berurutan mulai dari proses *text mining* sampai menghitung *tf-idf* dokumen. Berikut gambaran alur kerja dari sub sistem *consoleAps* dapat dilihat pada Gambar 3.1 dan *flowchart* alur kerja sub sistem *consoleAps* dapat dilihat pada Gambar 3.2.



Gambar 4.1 Alur kerja sub sistem *consoleAps*

Alur kerja pada sistem *consoleAps* dibagi menjadi 2 bagian penting yaitu *text mining* dan *vector space model*. Bagian pertama dari alur kerja sistem ini untuk melakukan pembacaan berita pada sumber yang dituju sekaligus melakukan teknik *teks mining* untuk mendapatkan dokumen berita dan kata pada dokumen tersebut. Sementara untuk bagian kedua yaitu *vector space model* digunakan untuk membaca hasil data yang telah dilakukan oleh bagian *text mining*, jadi dokumen berita dan

kata pada dokumen berita akan diolah menggunakan model *vector space model* dengan menggunakan teknik TF-IDF untuk menghasilkan representasi dokumen berita dan kata pada dokumen berita kedalam ruang vektor dan kemudian akan disimpan pada database sistem.



Gambar 4.2 Diagram Alir Tahapan Representasi Dokumen kedalam ruang n -dimensi

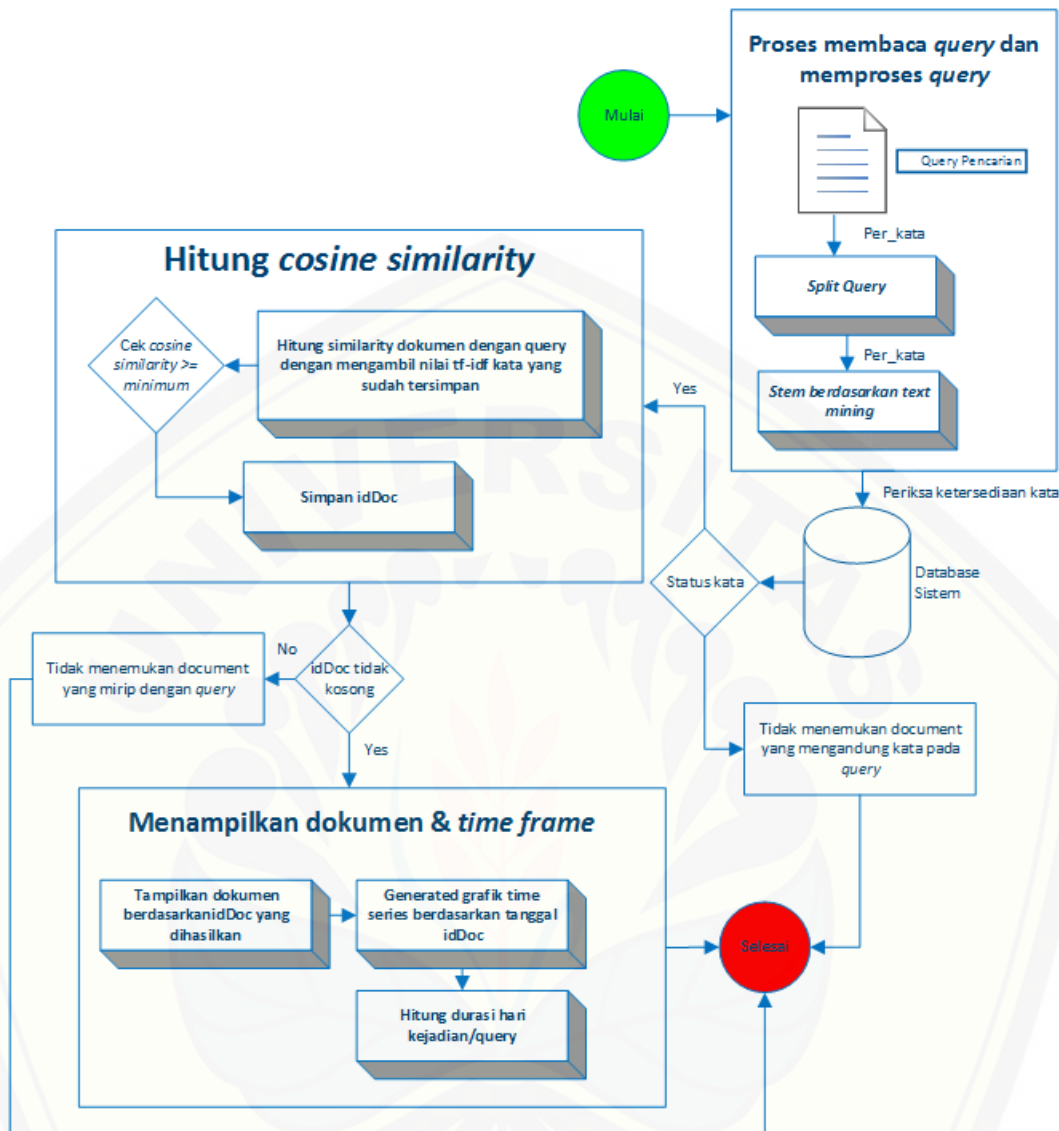
Proses *flowchart* diatas menjabarkan proses yang ada pada kedua bagian yaitu bagian *text mining* dan *vector space model*. Proses tersebut diawali dengan mendefinisikan berapa kali proses membaca dokumen berita yang ada. Kemudian dilanjutkan dengan proses membaca dokumen berita berbasis *Json*, kemudian akan diperiksa apakah dokumen tersebut pada perulangan ke- n sudah disimpan atau belum, jika sudah tersimpan maka akan dilanjutkan untuk membaca dokumen pada perulangan selanjutnya. Ketika dokumen belum tersimpan pada database sistem, maka simpan dokumen pada perulangan ke- n ke database sistem dan kemudian

dilanjutkan dengan representasi dokumen berita yang didapatkan pada perulangan ke- n kedalam ruang vektor menggunakan teknik perhitungan TF-IDF.

4.2.2 *Site time frame*

Sub sistem ini melakukan serangkaian kerja seperti membaca *query* dari pengguna, memeriksa dokumen mana yang mengandung kata pada *query*, menghitung berapa nilai kemiripan dari dokumen yang mengandung *query* serta membatasi nilai minimal *cosine similarity* antara dokumen dengan *query* untuk ditampilkan atau dikembalikan kepada pengguna beserta *grafik time series* dan durasi kejadiannya. Penentuan nilai minimal *cosine similarity* antara dokumen dengan *query* ditentukan dengan cara mengambil sample satu dokumen berita yang dianggap minimal mirip dengan kata kunci pencarian, dan diperiksa berapa nilai *cosine similarity* yang didapatkan. Percobaan tersebut dilakukan sebanyak 5 kali untuk menghasilkan batas nilai kemiripan yang lebih baik, kemudian akan dirata-rata untuk menentukan hasil akhir minimal nilai *cosine similarity*. Nilai yang didapatkan dari percobaan tersebut nantinya akan menjadi batas minimal suatu dokumen dikatakan mirip dengan *query* (*minimum cosine similarity*).

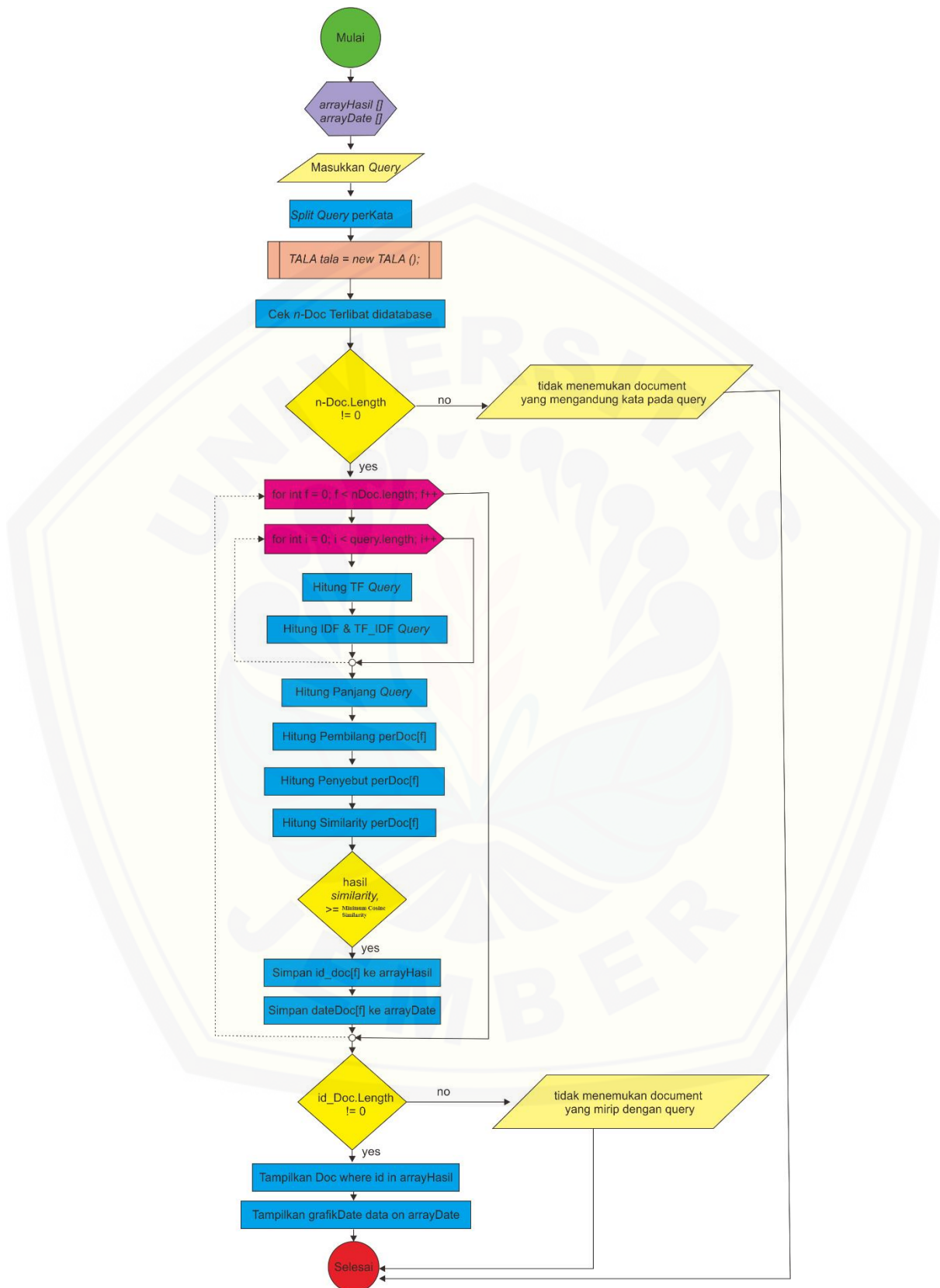
. Rangkaian proses pada sub sistem ini berurutan mulai dari membaca *query*, periksa *query* pada database, hitung *cosine similarity* dan terakhir periksa nilai *cosine similarity* untuk syarat minimal nilai kemiripan antara dokumen dan *query*. Kemudian untuk fase akhir dilakukan proses menghitung akurasi dengan menggunakan teknik perhitungan *recall, precision & f-measure*. Berikut proses pada sub sistem *site time frame* dapat dilihat pada Gambar 4.3 dan *flowchart* sub sistem *site time frame* pada Gambar 4.4



Gambar 4.3 Alur kerja sub sistem *site time frame*

Alur kerja pada *site time frame* diawali dengan bagian membaca *query* dan memproses *query* yang masuk. Pemrosesan *query* yang dimaksud yaitu untuk mengembalikan *query* kedalam bentuk kata pembangunnya agar mudah diproses oleh sistem. Setelah *query* diproses, selanjutnya akan diperiksa ketersediaan kata pada database sistem, ketika kata tersebut tidak tersedia maka proses dihentikan dan ketika kata tersedia maka proses dilanjutkan dengan memproses kata dengan menghitung kemiripan dokumen berita dengan *query* yang dimasukkan.

Teknik perhitungan kemiripan menggunakan *cosine similarity* yang terdiri dari bagian proses yaitu menghitung kemiripan kemudian memeriksa nilai kemiripan sebagai syarat minimal untuk ditampilkan kepada pengguna. Ketika tidak ada pemeriksaan nilai kemiripan minimal antara dokumen dan *query*, maka hasil akhir dokumen yang ditampilkan akan kurang relevan sesuai masukan *query*, oleh karena itu perlu adanya pembatas nilai minimal tersebut untuk mengurangi dokumen yang kurang relevan untuk ditampilkan. Setelah proses pemeriksaan selesai, maka id_dokumen berita yang lebih dari sama dengan nilai minimum akan disimpan terlebih dahulu. Ketika tidak ada id_dokumen yang disimpan maka dapat dipastikan tidak ada berita yang ditampilkan kepada pengguna, namun jika ada id_dokumen yang tersimpan maka dokumen tersebutlah yang akan ditampilkan kepada penggunanya. Proses menampilkan dokumen dan *time frame* dokumen berita bergantung pada id_dokumen yang berhasil disimpan, sehingga *Time frame* berita akan dihitung berdasarkan tanggal publikasi yang ada pada id_dokumen.



Gambar 4.4 Flowchart Tahapan Pencarian Dokumen dan Durasi Berita

4.3 Iteration Initialization sistem

Tahapan ini merupakan proses pembagian kegiatan atau modul berdasarkan tugas dan proses pembagian ini dibagi berdasarkan tingkat kemiripan pengerjaan. Jumlah perulangan diambil dari *requirement* terkait fitur apa saja yang digunakan untuk menyelesaikan masalah. Berikut tabel perulangan sistem yang dikembangkan dapat dilihat pada Table 4.1

Tabel 4.1 Tabel perulangan pengembangan sistem

No	Kode iterasi	Deskripsi	Estimasi/hari
PERULANGAN KD:01			
1.	KD:01:01	<i>Convert object file Json ke Datatable</i>	2
2.	KD:01:02	Perancangan database	1
3.	KD:01:03	Melakukan filter dokumen permasing-masing situs berita pada <i>datatable</i>	2
4.	KD:01:04	Menampilkan <i>datatable</i> berita kedalam data <i>gridview c#</i> permasing-masing situs berita	
5.	KD:01:05	Menghitung data masing-masing jumlah konten berita permasing-masing berita	
6.	KD:01:06	Menampilkan detail berita yang dibaca	
7.	KD:01:07	Melakukan penyimpanan dokumen berita ke <i>database</i> dengan atribut file <i>json</i> yang disimpan (<i>idDoc & date</i>), melakukan <i>text mining stemming</i> TALA untuk menyimpan kata penting permasing-masing <i>idDoc</i> serta menghitung frekuensi kata dimasing-masing dokumen	7
8.	KD:01:08	Akses masuk ke sistem <i>time frame</i> sesuai dengan level pengguna	1
Total estimasi			13
PERULANGAN KD:02			
1.	KD:02:01	<i>Split query</i> perkata dan <i>stemming</i> jika diperlukan dengan TALA, untuk mengembalikan query ke kata pembentuknya	1

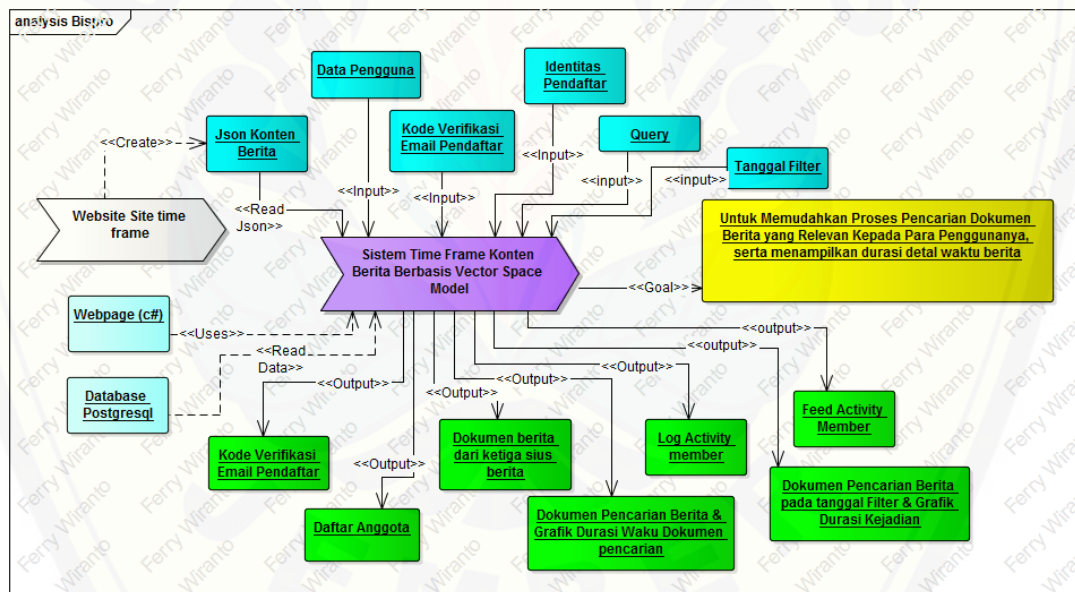
2.	KD:02:02	Implementasi metode vector space model dengan teknik perhitungan <i>TF-IDF</i> dan <i>cosine similarity</i>	10
3.	KD:02:02	Menampilkan dokumen relevan ke <i>datatable</i>	3
Total estimasi			14
PERULANGAN KD:03			
1.	KD:03:01	Mengambil data dokumen terkait untuk ditampilkan kedalam <i>grafik time series</i> serta menghitung durasi kejadiannya	3
Total estimasi			3
PERULANGAN KD:04			
1.	KD:04:01	Mengambil riwayat berita idDoc dan judul berita yang pernah dibaca member	1
2.	KD:04:01	Menampilkan detail riwayat member keseluruhan	
Total estimasi			1
PERULANGAN KD:05			
1.	KD:05:01	Daftar Member	1
2.	KD:05:01	Verifikasi Akun melalui <i>EMAIL</i>	2
Total estimasi			3

4.4 Desain Sistem

Desain sistem yang dibuat meliputi *business process*, *usecase diagram*, *Class Diagram*, *CRC Card*, *BPMN* dan *ERD*. Desain sistem yang dibangun menggunakan pemodelan UML (*Unified Modelling Language*) yakni pemodelan *visual* dengan basis berorientasi objek.

4.4.1 Diagram input output

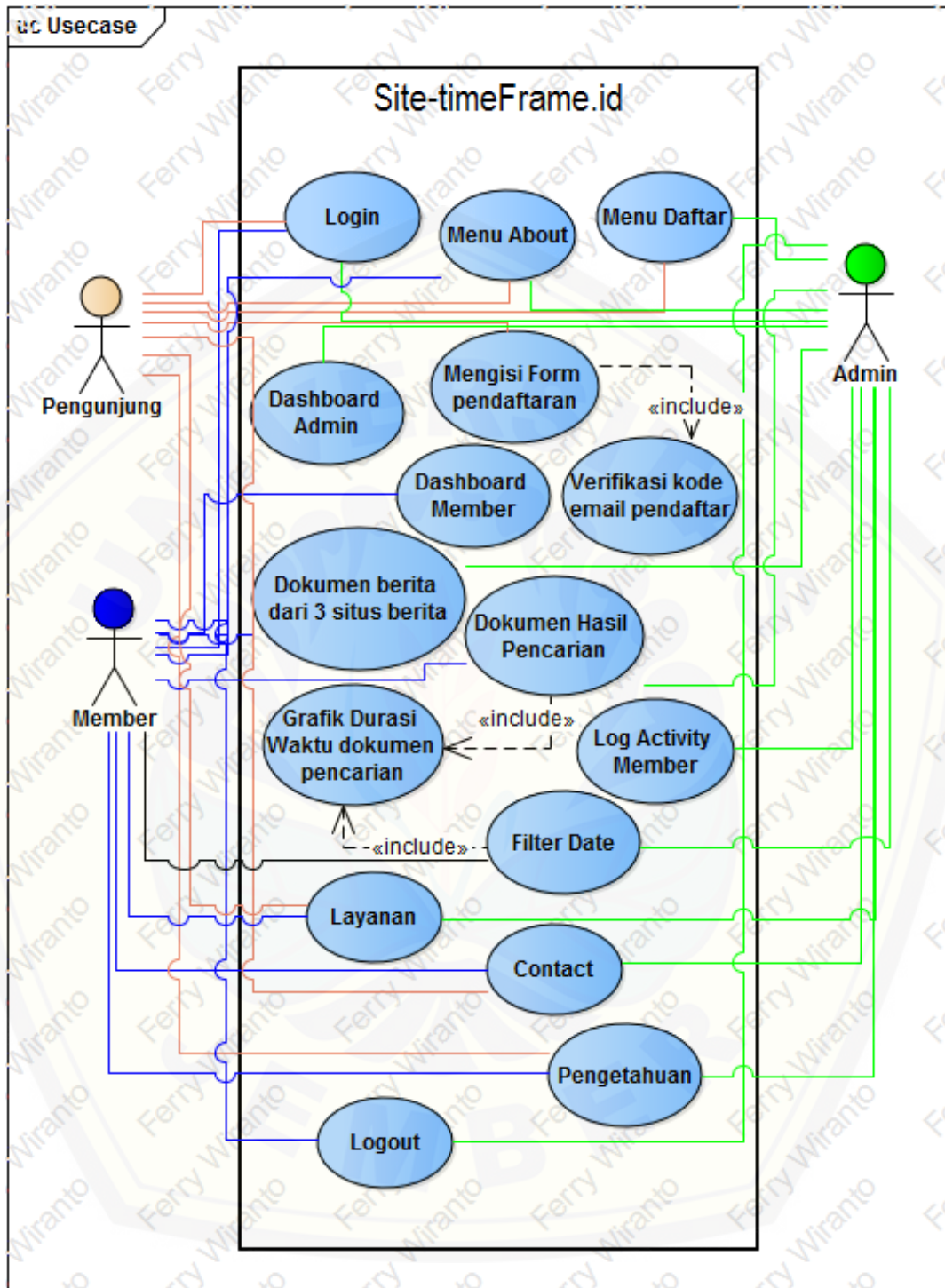
Diagram input output adalah suatu kumpulan aktivitas yang terstruktur untuk mencapai suatu tujuan tertentu atau untuk menghasilkan sebuah produk. Ada beberapa komponen di dalamnya, meliputi data yang menjadi masukan (*input*), data masukan yang kemudian diolah menjadi data keluaran (*output*), media yang digunakan (*uses*), tujuan yang ingin dicapai (*goal*). *Business Process* sistem ini dapat pada Gambar 4.5



Gambar 4.5 Diagram input output

4.4.2 Use Case Diagram

Use case diagram merupakan pemodelan yang dibuat untuk dapat menggambarkan interaksi antara aktor dengan sistem informasi yang akan dibangun. Melalui *use case diagram* dapat diketahui interaksi yang dapat dilakukan aktor terhadap sistem sesuai dengan hak akses yang dimiliki oleh masing-masing aktor atau pengguna. *Use case diagram* sistem ini dapat dilihat pada Gambar 4.6



Gambar 4.6 Use Case Diagram

Penjelasan mengenai definisi aktor dan definisi *use case* dalam *use case* diagram terdapat pada Tabel 4.2

a. Definisi Aktor

Definisi aktor merupakan penjelasan tentang aktor-aktor sebagai pengguna dari sistem yang dibangun. Terdapat 4 aktor seperti yang dijelaskan pada tabel 4.2.

Tabel 4.2 *Definisi Aktor*

Aktor	Deskripsi
Admin	Aktor yang memiliki akses penuh terhadap segala aktivitas yang ada pada aplikasi <i>site time frame</i>
Member	Aktor yang memiliki akses terhadap proses membaca situs berita yang ada pada aplikasi yang dikembangkan.
Pengunjung	Aktor yang memiliki akses hanya melihat halaman awal <i>site time frame</i> . Namun aktor ini bisa menjadi member dengan syarat melakukan pendaftaran akun.

b. Definisi *Use Case*

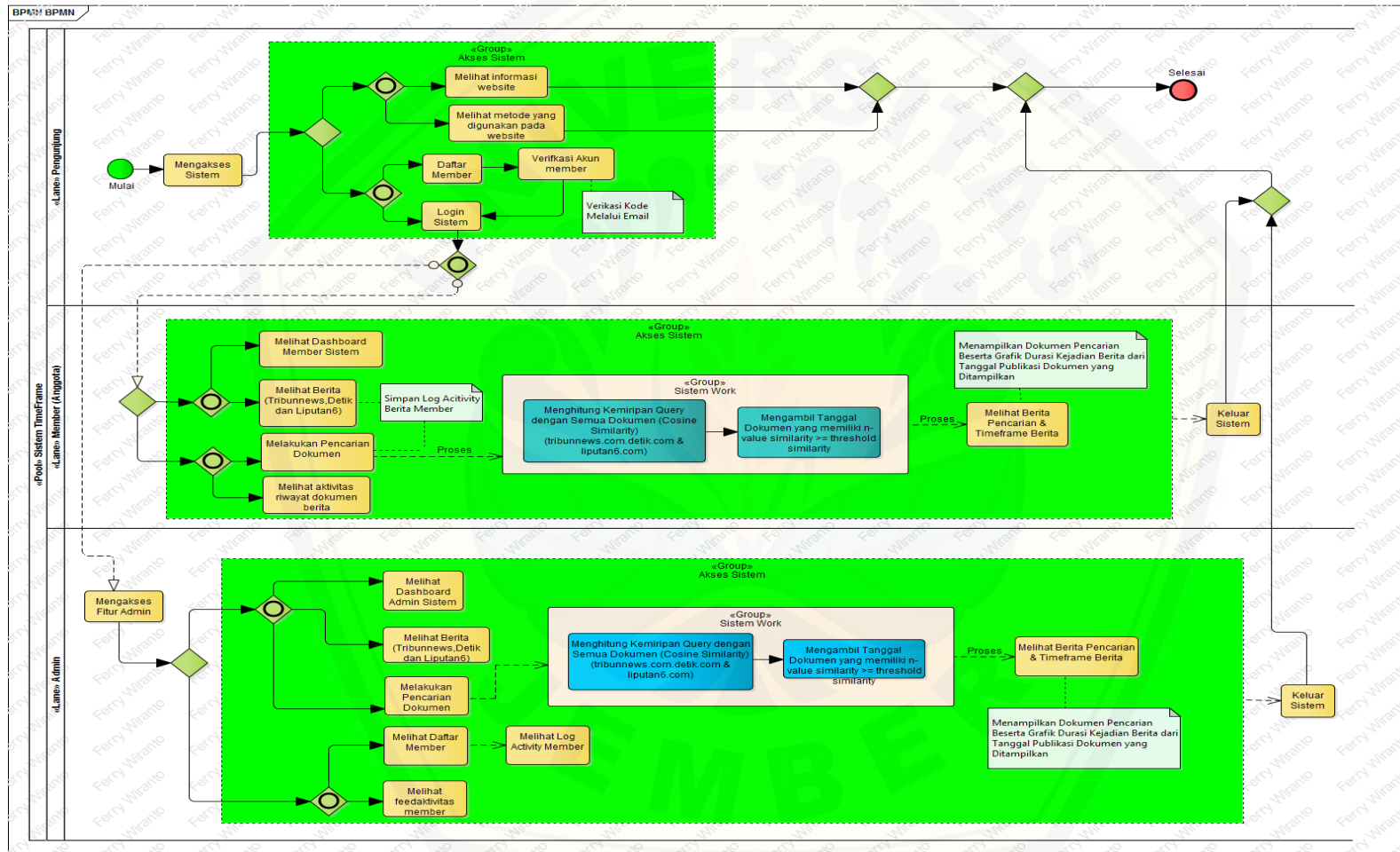
Definisi *Use Case* merupakan penjelasan dari masing-masing *Use Case* atau fitur-fitur dari sistem yang akan dibangun. Definisi masing-masing *Use Case* terdapat pada Tabel 4.3

Tabel 4.3 *Definisi Use Case*

No.	Aktor	Deskripsi
1.	<i>Login</i>	Menggambarkan proses autentifikasi untuk masuk ke system.
2.	Menu About	Menggambarkan proses melihat yang didalamnya berisi tentang <i>information</i> pengembang sistem.
3.	Menu Daftar	Menggambarkan proses melakukan pendaftaran untuk menjadi member aplikasi.
4.	<i>Dashboard Admin</i>	Menggambarkan proses melihat lama akses awal admin, yang berisi tentang informasi situs berita yang ada.

5.	Mengisi Form Pendaftaran	Menggambarkan proses melakukan pengisian form pendaftaran untuk menjadi member aplikasi.
6.	Verifikasi kode email pendaftar	Menggambarkan proses kelanjutan dari mengisi form pendaftaran, ketika proses pengisian semua sudah lengkap sesuai dengan ketentuan, maka pengguna akan mendapatkan kode verifikasi pendaftar melalui email pengguna yang dimasukkan pada saat melakukan pendaftaran. Proses verifikasi ini digunakan untuk proses aktivasi akun pengunjung langsung untuk ingin menjadi member aplikasi.
7.	Dashboard Member	Menggambarkan proses melihat lama akses awal member, yang berisi tentang informasi situs berita yang ada
8.	Dokumen berita dari 3 situs berita	Menggambarkan proses melihat ketiga dokumen berita dari sumber berita yang dituju.
9.	Dokumen hasil pencarian	Menggambarkan proses melihat dokumen hasil pencarian dokumen berita yang telah dilakukan.
10.	Grafik durasi waktu dokumen pencarian	Menggambarkan proses kelanjutan dari dokumen hasil pencarian, juga akan melihat grafik durasi waktu berita yang dicari dalam bentuk grafik harian berdasarkan tanggal publikasi dokumen yang ditampilkan
11.	Filter Date	Menggambarkan Proses pencarian dokumen berdasarkan tanggal berita pada dokumen hasil pencarian.
12.	<i>Log Activity Member</i>	Menggambarkan proses melihat log aktivitas apa saja yang dibaca oleh member dengan batas 10 dokumen berita yang terakhir kali dibaca.
13.	Layanan	Menggambarkan proses melihat layanan apa saja yang tersedia pada sistem <i>site time frame</i>
14.	<i>Contact</i>	Menggambarkan proses melihat informasi <i>contact</i> pengembang aplikasi untuk media keluhan atau lain sebagainya.
15.	Pengetahuan	Menggambarkan melihat informasi ilmu yang digunakan dalam membangun sistem
16.	<i>Logout</i>	Menggambarkan proses keluar dari sistem.

4.4.3 Business Process Model and Notation (BPMN)

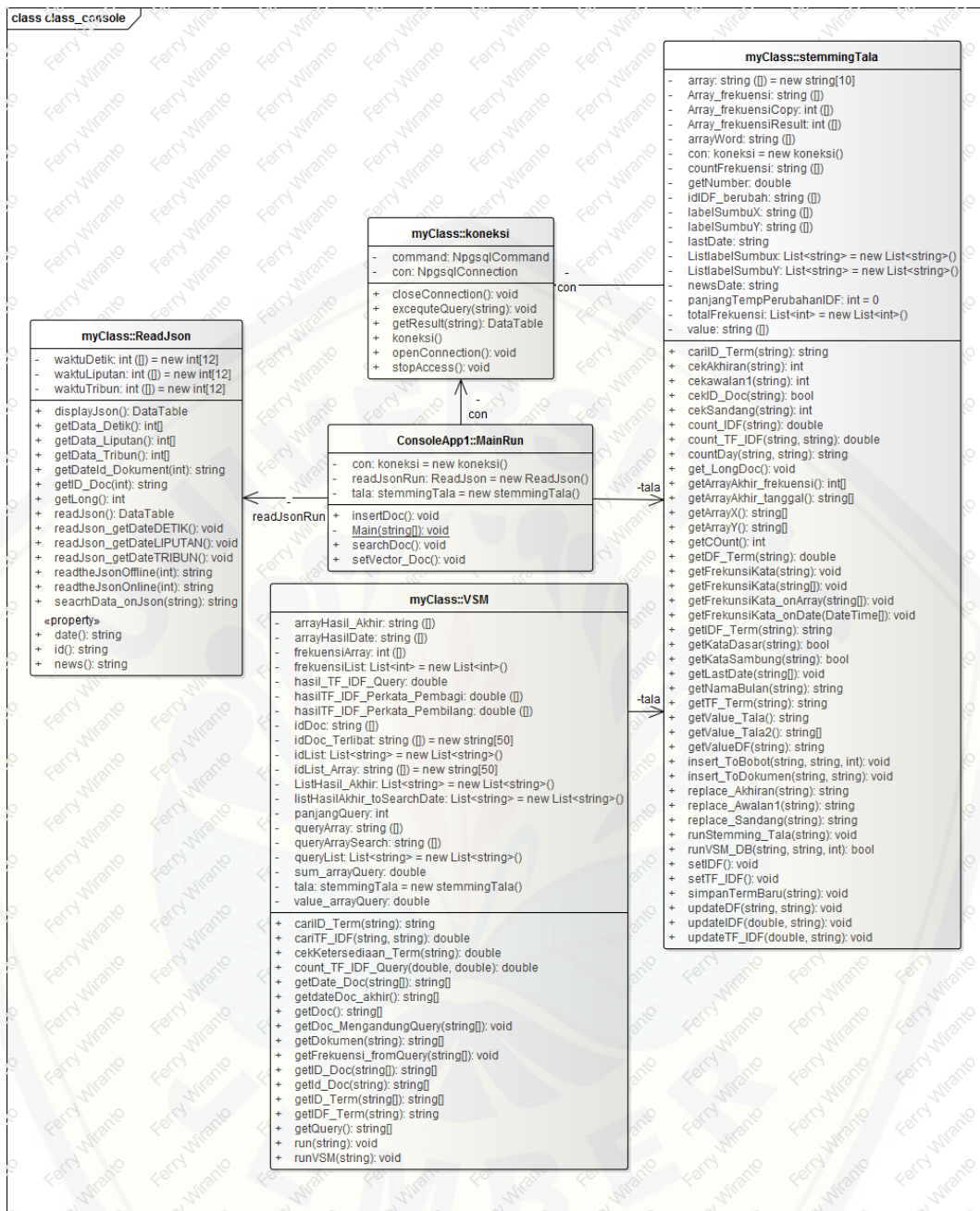


Gambar 4.7 Business Process Model and Notation

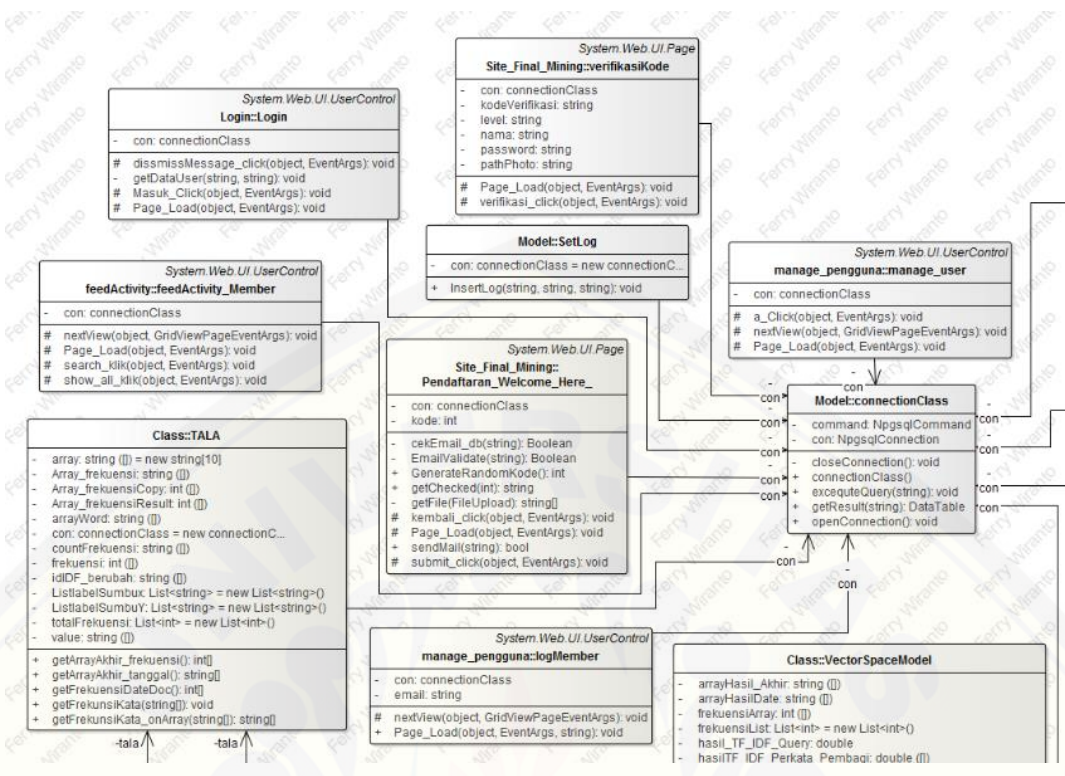
4.4.4 Class diagram

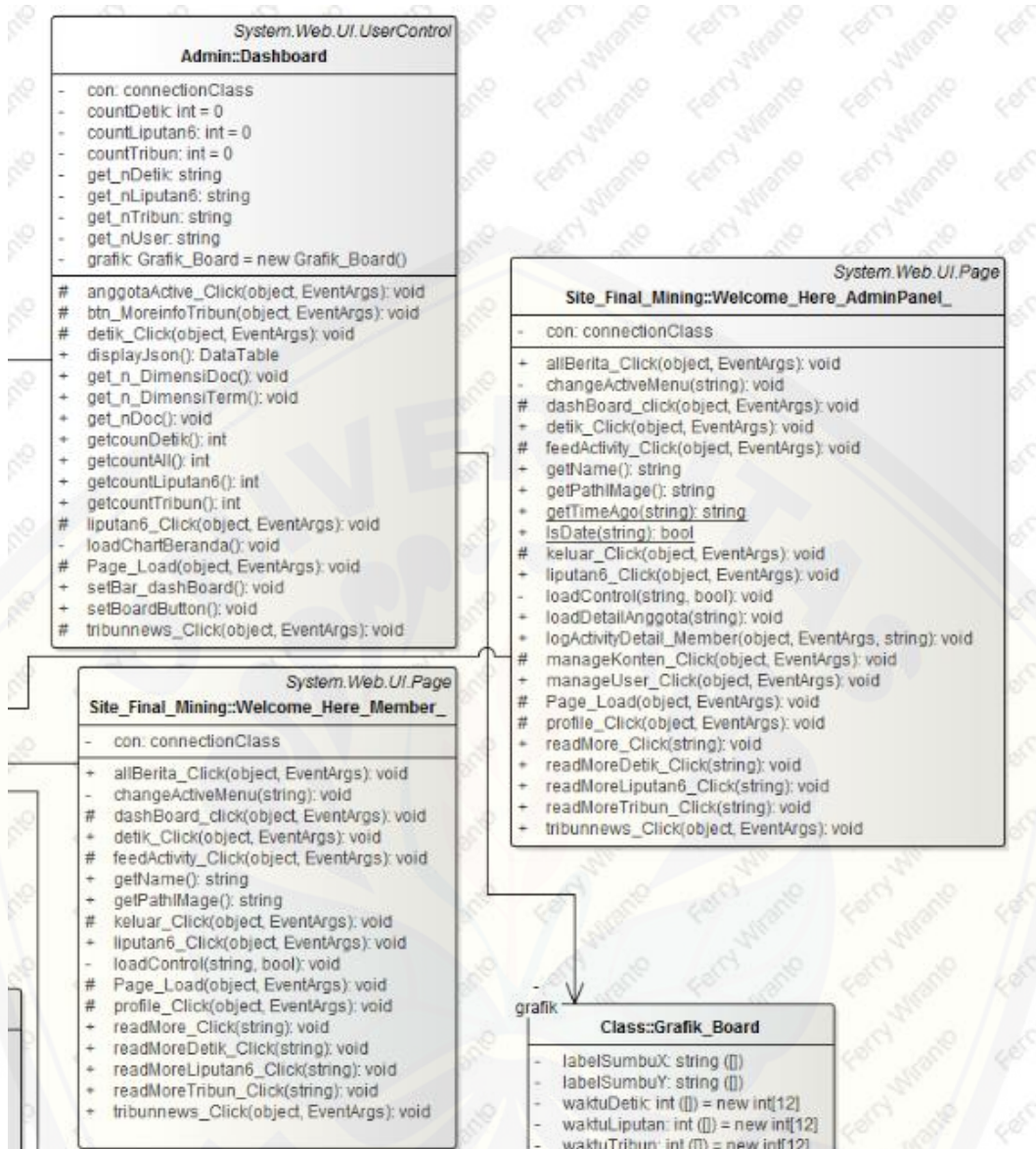
Salah satu model yang menggambarkan struktur data, deskripsi *class*, *package* dan objek serta hubungan antar *class* adalah *class diagram*. *Class* menggambarkan keadaan atribut atau *properti* pada *sistem time frame*. Keseluruhan komponen *class diagram* yaitu, (*class/objek/package*), atribut, nama *method/operations* dan *relationships* atau hubungan antar *class*. Setiap *class* akan mempunyai suatu nama *method* yang akan digunakan pada kasus tertentu sesuai kebutuhan. Atribut akan mewakili apa saja yang diperlukan suatu *method* untuk melakukan komputasi yang diinginkan, relasi dalam *class diagram* untuk membagi suatu proses komputasi atau pengerjaan pada *class* lain dengan relasi tertentu. Relasi *class diagram* bermacam-macam bisa digunakan sesuai dengan kebutuhannya.

Proses pengembangan *class diagram* pada sistem ini dibagi menjadi 2, karena peneliti mengembangkan dua program yang mempunyai fungsional yang berbeda yaitu *consoleAps* untuk melakukan proses *text mining* dan *vector space model*, sedangkan *site time frame* digunakan untuk proses pencarian data, menghitung *cosine similarity* dan menampilkan *time frame* dokumen berita. berikut gambar *class diagram consoleAps* bisa dilihat pada Gambar 4.8 dan *class diagram* untuk *site time frame* pada Gambar 4.9



Gambar 4.8 Class Diagram ConsoleApps





4.4.5 CRC Card

Nama Kelas: Admin	ID: CRC.01
Deskripsi : Aktivitas yang dilakukan dalam keseluruhan pada sistem, mulai dari melihat berita, melihat grafik <i>time frame</i> , melihat durasi kejadian, melihat aktivitas <i>member</i> dan melihat data <i>member</i> .	
Tugas : Melihat aktivitas para pembaca berita Melihat data <i>member</i> Melihat dokumen berita dari ketiga situs berita (Tribunnews.com, Detik.com dan Liputan6.com) Melihat grafik <i>time frame</i> kejadian (<i>query</i>) Melihat durasi kejadian (<i>query</i>).	Kolabolator : Dokumen berita <i>Member</i> <i>Connection</i> <i>Dashboard</i> <i>Grafik_Board</i> <i>Vector Space Model</i> <i>Manage User</i> <i>Set_log</i> TALA <i>Log_Member</i>
Nama Kelas: Member	ID: CRC.02
Deskripsi : Aktivitas yang dilakukan oleh <i>member</i> dalam keseluruhan pada sistem, mulai dari melihat berita, melihat grafik <i>time frame</i> dan melihat durasi kejadian.	
Tugas : 1. Melihat dokumen berita dari ketiga situs berita (Tribunnews.com, Detik.com dan Liputan6.com) 2. Melihat grafik <i>time frame</i> kejadian (<i>query</i>) 3. Melihat durasi kejadian (<i>query</i>).	Kolabolator : 1. Dokumen berita 2. <i>Connection</i> 3. <i>Dashboard</i> 4. <i>Grafik_Board</i> 5. <i>Vector Space Model</i> 6. <i>Set_log</i> 7. TALA
Nama Kelas: Connection Class	ID: CRC.03

Deskripsi : Aktivitas untuk melakukan proses menghubungkan antara sistem yang dikembangkan dengan database.

Tugas :

1. Melakukan transfer proses *request* dan *eksekusi* permintaan dari sistem ke database maupun sebaliknya.

Kolabolator :

1. Dokumen berita
2. *Member*
3. *Admin*
4. *Connection*
5. *Dashboard*
6. *Grafik_Board*
7. *Vector Space Model*
8. *Manage User*
9. *Set_log*
10. *TALA*
11. *Log_member*

Nama Kelas: <i>Vector Space Model</i>	ID: CRC.04
Deskripsi : Aktivitas untuk melakukan proses representasi dokumen berita kedalam ruang <i>vector</i> dan menggunakan teknik perhitungan <i>TF-IDF</i> dan <i>Cosine similarity</i> .	
Tugas : <ol style="list-style-type: none">1. Melakukan proses representasi dokumen kedalam vector.2. Menghitung <i>TF-IDF</i> setiap dokumen berita3. Menghitung kemiripan antara dokumen dengan <i>query</i>4. Mengembalikan dokumen relevan sesuai dengan ketentuan5. Mengembalikan data nilai tanggal kejadian berdasarkan dokumen relevan yang ditampilkan.	Kolaborator : <ol style="list-style-type: none">1. Dokumen berita2. <i>Member</i>3. <i>Admin</i>4. <i>Connection_Class</i>5. <i>Dashboard</i>6. <i>Grafik_Board</i>7. <i>Vector Space Model</i>8. <i>TALA</i>

Nama Kelas: <i>TALA</i>	ID: CRC.05
Deskripsi : Aktivitas untuk melakukan proses <i>text mining</i> pada dokumen berita dan menggunakan metode <i>TALA</i> pada proses <i>stemming text mining</i> .	
Tugas : 1. Melakukan proses <i>text mining</i> setiap dokumen, untuk menyimpan kata penting dari setiap dokumen berita. 2. Melakukan proses <i>stemming</i> kata menggunakan metode <i>TALA</i> , untuk mengembalikan kata menjadi kata pembangunnya (kata sambung).	Kolabolator : 1. Dokumen berita 2. <i>Member</i> 3. <i>Admin</i> 4. <i>Connection_Class</i>

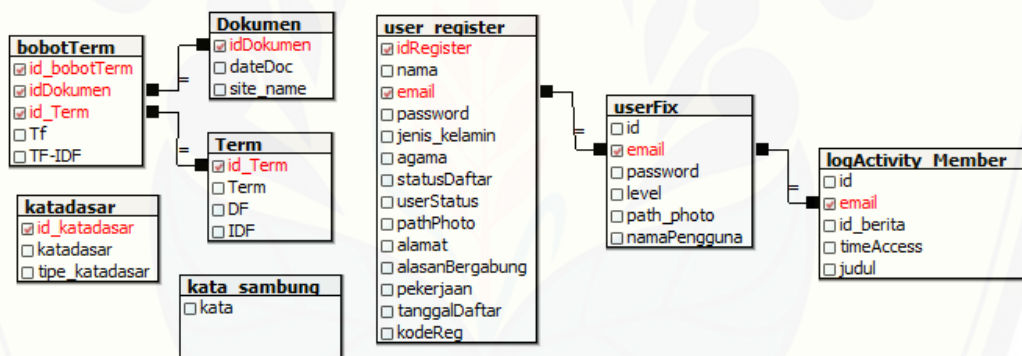
Nama Kelas: <i>Set_Log</i>	ID: CRC.06
Deskripsi : Aktivitas untuk menyimpan riwayat membaca berita oleh <i>member</i> .	
Tugas : 1. Melakukan proses membaca dan menyimpan riwayat terakhir dokumen berita oleh <i>member</i> kedalam database.	Kolabolator : 1. Dokumen berita 2. <i>Admin</i> 3. <i>Connection_Class</i>

Nama Kelas: Dokumen Berita	ID: CRC.07
Deskripsi : Aktivitas untuk membaca file dokumen berita <i>Json</i> kedalam <i>data table</i>	
Tugas : 1. Melakukan proses membaca dokumen berita yang format <i>Json</i> untuk dikonversikan kedalam tipe <i>data table</i> untuk memudahkan proses <i>text mining</i> .	Kolabolator : 1. <i>TALA</i> 2. <i>Vector space model</i>

Nama Kelas: Log_Member	ID: CRC.08
Deskripsi : Aktivitas untuk membaca membaca data member sekaligus semua riwayat membaca berita oleh <i>member</i> .	
Tugas : 1. Melakukan proses menampilkan semua data <i>member</i> yang sudah terdaftar. 2. Melakukan proses menampilkan aktivitas membaca yang dilakukan oleh <i>member</i> .	Kolabolator : 1. <i>Connection_Class</i> 2. <i>Admin</i>

4.4.6 Entity Relationship Diagram (ERD)

ERD merupakan gambaran komponen dan struktur *database* yang digunakan dalam pembangunan sistem. ERD memiliki beberapa komponen, yaitu tabel, atribut dan relasi antar tabel. Tabel menggambarkan suatu media penyimpanan data yang diperlukan oleh sistem *time frame*. Atribut menggambarkan identitas apa saja yang akan disimpan kedalam tabel pada sistem *time frame*. Relasi menggambarkan hubungan antara tabel dengan tabel lainnya yang memiliki hubungan dengan cara menggunakan *primary key* atau kunci tabel. Beberapa data yang digunakan dalam sistem *time frame* bisa dijadikan dalam satu tabel. Namun hal tersebut akan memlambat proses membaca data. Oleh karena itu relasi dalam ERD sangat penting agar data mudah diakses dan mudah dalam proses komputasinya. ERD pada sistem deteksi *timeframe* dokumen berita dapat dilihat pada Gambar 4.10



Gambar 4.10 Entity Relationship Diagram (ERD)

4.5 Implementasi

Implementasi yang dilakukan berdasarkan iterasi yang sudah didefinisikan di awal.

Implementasi dibagi menjadi 3 bagian sub proses yaitu:

1. **Unit Testing.** Proses ini melakukan pengujian pada setiap unit fungsi atau prosedur dalam suatu program. Satu perulangan memungkinkan terdiri dari banyak fungsi dan prosedur didalamnya, agar semua unit tersebut berjalan sesuai yang diharapkan dan tidak terjadi kesalahan, maka perlu dilakukan *unit testing* pada setiap perulangan yang ada. Berikut dokumen uji *unit testing* sistem dapat dilihat pada Lampiran A.
2. **Code.** Proses ini melakukan proses eksekusi desain sistem kedalam kode program yang dipakai untuk mendefinisikan suatu program yang akan dikembangkan. Berikut dokumentasi beberapa code program yang telah dikembangkan dapat dilihat dilampiran G.
3. **Refactor.** Proses ini melakukan pemeliharaan dan pemeriksaan kembali kode program untuk memeriksa fungsionalitas kode yang dibangun apakah sudah efektif dan efisien serta meminimalisir peluang munculnya *bug* pada *site time frame*.

4.6 System Testing

Proses ini merupakan validasi akhir apakah permasalahan yang diselesaikan pada setiap iterasi sesuai dengan *requirement* atau tidak. Berikut tabel *system testing* pada *site time frame* pada setiap *iterasi* dapat dilihat pada Lampiran B.

4.7 Retrospective

Proses ini merupakan akhir setiap perulangan yang ada pada pengembangan sistem. Proses ini menjadi penanda dari setiap perulangan yang ada. Pada proses ini akan didokumentasikan segala kegiatan yang sudah dilaksanakan pada setiap *iterasi* yang ada.

Proses percobaan tersebut mengacu pada judul berita dengan *query* yang akan dimasukkan. Peneliti menganggap bahwa judul berita “Ditlantas polda metro jaya tolak jadi saksi meringankan Fredrich Yunadi” yang isi dokumen beritanya menggambarkan cuplikan dari makna kata “kecelakaan jalan raya” yaitu mempunyai makna minimal relevan dengan *query* yaitu “kecelakaan lalu lintas”. Dokumen tersebut kemudian diproses pada sistem yang dikembangkan untuk melihat berapa nilai *similarity* yang dihasilkan dari dokumen dengan *query* tersebut.

Berdasarkan hasil pengujian yang dilakukan menunjukkan bahwa nilai *cosine similarity* antara dokumen berita yang berjudul “Ditlantas polda metro jaya tolak jadi saksi meringankan Fredrich Yunadi” dengan *query* “kecelakaan jalan raya” menghasilkan nilai akhir sebesar 0,79738. Nilai tersebut dinaikkan menjadi 0,80 oleh peneliti. Oleh karena itu dapat disimpulkan bahwa ketika pengguna memasukkan *query* “kecelakaan jalan raya” maka dokumen berita yang menjadi nilai paling kecil dengan toleransi untuk ditampilkan sebesar 0,80 pada percobaan pertama. Berikut hasil pengujian dokumen berita dengan masukan *query* “kecelakaan jalan raya” dengan id berita 94338f8a-2fc8-4c51-a4d7-bcf2677c6881 dapat dilihat pada Gambar 4.12

```

hasil total pembilang id Doc '8d82f59b-9833-4967-9b87-3e26b8d6c8a7' = 1731856
hasil total penyebut id Doc '8d82f59b-9833-4967-9b87-3e26b8d6c8a7' = 1316
Similarity Doc id '8d82f59b-9833-4967-9b87-3e26b8d6c8a7' dengan Query = 0,39113
=====
'5dafbe2b-009b-49f4-ac6f-5c16c6319c59', '4a5372e6-e1e7-4122-b77e-199000650520' :
2727
'5dafbe2b-009b-49f4-ac6f-5c16c6319c59', '41b4f9c5-50c6-411d-a071-2919d0fb396b' :
0
'5dafbe2b-009b-49f4-ac6f-5c16c6319c59', 'e83167eb-afe0-422f-a297-61b8ce73a828' :
0
Panjang Q = 3364,59121
hasil total pembilang id Doc '5dafbe2b-009b-49f4-ac6f-5c16c6319c59' = 4000509
hasil total penyebut id Doc '5dafbe2b-009b-49f4-ac6f-5c16c6319c59' = 2727
Similarity Doc id '5dafbe2b-009b-49f4-ac6f-5c16c6319c59' dengan Query = 0,43601
=====
'c2ec1d29-ed10-46fc-a563-ac4640006ecc', '4a5372e6-e1e7-4122-b77e-199000650520' :
0
'c2ec1d29-ed10-46fc-a563-ac4640006ecc', '41b4f9c5-50c6-411d-a071-2919d0fb396b' :
8802
'c2ec1d29-ed10-46fc-a563-ac4640006ecc', 'e83167eb-afe0-422f-a297-61b8ce73a828' :
0
Panjang Q = 3364,59121
hasil total pembilang id Doc 'c2ec1d29-ed10-46fc-a563-ac4640006ecc' = 24003054
hasil total penyebut id Doc 'c2ec1d29-ed10-46fc-a563-ac4640006ecc' = 8802
Similarity Doc id 'c2ec1d29-ed10-46fc-a563-ac4640006ecc' dengan Query = 0,81805
=====
'e0a48b28-c961-48d3-8db1-ac0ea8cb2cea', '4a5372e6-e1e7-4122-b77e-199000650520' :
0
'e0a48b28-c961-48d3-8db1-ac0ea8cb2cea', '41b4f9c5-50c6-411d-a071-2919d0fb396b' :
0
'e0a48b28-c961-48d3-8db1-ac0ea8cb2cea', 'e83167eb-afe0-422f-a297-61b8ce73a828' :
1316
Panjang Q = 3364,59121
hasil total pembilang id Doc 'e0a48b28-c961-48d3-8db1-ac0ea8cb2cea' = 1731856
hasil total penyebut id Doc 'e0a48b28-c961-48d3-8db1-ac0ea8cb2cea' = 1316
Similarity Doc id 'e0a48b28-c961-48d3-8db1-ac0ea8cb2cea' dengan Query = 0,39113
=====
'94338f8a-2fc8-4c51-a4d7-bcf2677c6881', '4a5372e6-e1e7-4122-b77e-199000650520' :
0
'94338f8a-2fc8-4c51-a4d7-bcf2677c6881', '41b4f9c5-50c6-411d-a071-2919d0fb396b' :
2934
'94338f8a-2fc8-4c51-a4d7-bcf2677c6881', 'e83167eb-afe0-422f-a297-61b8ce73a828' :
3948
Panjang Q = 3364,59121
hasil total pembilang id Doc '94338f8a-2fc8-4c51-a4d7-bcf2677c6881' = 13196586
hasil total penyebut id Doc '94338f8a-2fc8-4c51-a4d7-bcf2677c6881' = 4918,84743
Similarity Doc id '94338f8a-2fc8-4c51-a4d7-bcf2677c6881' dengan Query = 0,79738
=====
'd58dd220-1630-4c55-80be-7a750e05d41c', '4a5372e6-e1e7-4122-b77e-199000650520' :
0
'd58dd220-1630-4c55-80be-7a750e05d41c', '41b4f9c5-50c6-411d-a071-2919d0fb396b' :
0
'd58dd220-1630-4c55-80be-7a750e05d41c', 'e83167eb-afe0-422f-a297-61b8ce73a828' :
1316

```

Gambar 4.12 hasil pengujian nilai similarity dokumen berita dengan query

Pada penelitian ini, pengujian menggunakan 5 kali pengujian yang sama dengan menggunakan *query* yang berbeda pada situs berita yang juga berbeda. Oleh karena itu nilai *minimum similarity* nantiya akan ditentukan dari rata-rata 5 nilai hasil percobaan. Nilai yang dihasilkan akan digunakan pada sistem *site time frame* dokumen berita untuk menampilkan dokumen berita relevan dan durasi kejadian berdasarkan *query* yang dimasukkan. Berikut rangkuman perhitungan kelima percobaan dalam menghitung *minimal cosine similarity* dapat dilihat pada Tabel 4.4.

Tabel 4.4 perhitungan minimal *cosine similarity* dengan 5 kali percobaan

Id_dokumen	Judul berita	Query Masukan	Sumber	Hasil
94338f8a-2fc8-4c51-a4d7-bcf2677c6881	Ditlantas polda metro jaya tolak jadi saksi meringankan Fredrich Yunadi	kecelakaan jalan raya	Tribunnews.com	0.80
9b8b77c2-c53c-49a0-8cae-33d83cc0dda7	BNPB: Dalam Setahun, Indonesia Diguncang 6.000 Kali Gempa	tsunami papua	Liputan6.com	0,77
b7caa751-d687-4908-8daf-0efd990729b1	Pemerintah Beri Beasiswa Anak Pelaku dan Korban Teroris	bencana alam banjir	detik.com	0,80
df6693cb-fcfa-4fe1-9e01-1010931541b6	Pimpinan KPK Ingatkan Calon Kepala Daerah Tak Main Politik Uangyang Diincarnya	korupsi uang negara	Detik.com	0,82

ddf3d217-60a3-42d0-909f-84034a97aba8	Kebakaran Swalayan Sinar Fresh dan Gudang Diduga Korsleting Listrik	hari raya idul fitri	liputan6.com	0,75
--------------------------------------	---	----------------------	--------------	------

Berdasarkan hasil percobaan yang dilakukan berdasarkan tabel 4.4 maka dapat dihasilkan nilai minimal *cosine similarity* akhir yang akan digunakan yaitu dengan cara mengambil rata-rata nilai dari kelima percobaan tersebut. Berikut perhitungan minimal nilai *cosine similarity* akhir yang akan digunakan oleh peneliti.

$$\text{Rata - rata (hasil)} = \frac{(0,80 + 0,77 + 0,80 + 0,82 + 0,75)}{5} = \frac{3,94}{5} = 0,788$$

BAB 6 PENUTUP

Bab ini berisi tentang kesimpulan dan saran dari peneliti tentang penelitian yang telah dilakukan. Kesimpulan dan saran yang diberikan dapat digunakan sebagai acuan dalam penelitian selanjutnya.

6.1 Kesimpulan

Hasil dari penelitian yang dilakukan, dapat diambil kesimpulan sebagai berikut :

1. Berdasarkan hasil analisis kebutuhan pengguna, sistem deteksi sistem *time frame* konten berita berbasis *vector space model* untuk menampilkan *time frame* dokumen berita berdasarkan pencarian berita pengguna membutuhkan 15 proses dan melibatkan 3 aktor yaitu pengunjung, *member* dan admin. Proses implementasi sistem yang dihasilkan, diwujudkan kedalam beberapa *class*, dimana 5 kelas digunakan dalam aplikasi *consoleAps* digunakan untuk merepresentasikan dokumen berita kedalam ruang vektor berbasis *vector space model*, yaitu kelas dokumen berita, *connection*, *vector space model*, TALA dan *teks mining* dan 8 kelas lain yaitu dokumen berita, *connection*, *vector space model*, TALA dan *text mining*, digunakan dalam aplikasi *site time frame* untuk melakukan proses pencarian dokumen berita dan menampilkan durasi kejadiannya berdasarkan dokumen yang ditampilkan.
2. Berdasarkan hasil percobaan yang telah dilakukan, nilai *cosine similarity* minimal (*threshold*) yang dibutuhkan untuk mendapatkan dokumen yang relevan adalah 0,788. Artinya ketika akan menampilkan dokumen berita yang relevan, diperlukan sesuatu nilai *threshold* yang digunakan sebagai acuan nilai yang akan ditampilkan berdasarkan nilai hasil kemiripan antara dokumen tertentu dengan *query* yang dimasukkan.
3. Berdasarkan penggunaan nilai *threshold* 0,788 sistem deteksi *time frame* konten berita berbasis *vector space model* yang telah dibangun mendapatkan nilai rerata *recall* 100%, *precision* 90% dan *f-measure* 94,7%. Artinya kualitas sistem yang sudah dikembangkan mampu menampilkan berita sebesar 100%, keakuratan menampilkan dokumen berita sesuai pencarian

sebesar 90% keakuratan mengenai penggunaan model vector space model yang dikombinasikan dengan teknik perhitungan *TF-IDF* dan *cosine similarity* menghasilkan nilai sebesar 94,7%. Namun nilai tersebut akan berubah tergantung dengan jumlah dokumen yang digunakan.

6.2 Saran

Adapun saran yang diberikan untuk menjadi masukan sebagai bahan pertimbangan dalam rangka perbaikan sistem maupun penelitian selanjutnya adalah sebagai berikut:

1. Proses pencarian sistem ini masih membutuhkan proses yang relatif lama, sebaiknya dilakukan penelitian untuk meningkatkan kecepatan prosesnya. Hasil tingkat akurasi yang sudah cukup baik, namun sebaiknya pada penelitian berikutnya jumlah dokumen berita yang digunakan ditingkatkan dengan jumlah jauh lebih besar untuk memastikan tingkat akurasi yang telah diperoleh.
2. Dalam penelitian ini, sumber dokumen berita diambil dari ketiga situs berita terpopuler di Indonesia, yaitu: www.detik.com, www.tribunnews.com dan www.liputan6.com Untuk penelitian selanjutnya sebaiknya diperluas cakupan situs berita, untuk mendapatkan durasi kejadian yang lebih akurat.
3. Dalam penelitian ini, cara melakukan proses penyimpanan data dokumen berita secara permanen di *library* bahasa pemrograman yang digunakan, maka harapannya pada penelitian selanjutnya perlu penerapan penyimpanan dokumen secara online untuk mendapatkan jumlah dokumen berita secara bertahap yang akan terus bertambah sesuai dengan berita pada situs berita yang dituju.

DAFTAR PUSTAKA

- Abdul Azis Abdillah, I. B. M. (2013). Implementasi Vector Space Model Untuk Pencarian Dokumen. *Prosiding Seminar Nasional Matematika Dan Pendidikan Matematika 2013*, (May 2013), 1–7.
- Anna, & Hendini, A. (2018). Implementasi Vector Space Model pada Sistem Pencarian Mesin Karaoke. *Jurnal Evolusi*, 6(1), 1–6.
- Aziz, A., Saptono, R., & Suryajaya, K. P. (2016). Implementasi Vector Space Model dalam Pembangkitan Frequently Asked Questions Otomatis dan Solusi yang Relevan untuk Keluhan Pelanggan. *Scientific Journal of Informatics*, 2(2), 111. <https://doi.org/10.15294/sji.v2i2.5076>
- Dasar, P., Wicaksono, V. B., & Sihwi, S. W. (2015). Analisis Perbandingan Metode Vector Space Model dan Weighted Tree Similarity dengan Cosine Similarity pada kasus Pencarian Informasi Pedoman Analisis Perbandingan Metode Vector Space Model dan Weighted Tree Similarity dengan Cosine Similarity pada kasus Pencarian Informasi Pedoman Pengobatan Dasar di Puskesmas, (March 2017). <https://doi.org/10.20961/its.v4i2.1768>
- Dzhurov, Y., Krasteva, I., & Ilieva, S. (2014). Personal Extreme Programming – An Agile Process for Autonomous Developers Personal Extreme Programming – An Agile Process for Autonomous Developers, (May).
- Pressman, R. S. (2015). *Software Engineering : a practitioner's approach*. New York: McGraw-Hill.
- Rohim, M. A. (2018). IMPLEMENTASI EKSTRAKSI WEB (WEB SCRAPING) PADA SITUS BERITA MENGGUNAKAN METODE EKSPRESI REGULER.
- Sanjaya, F. (2018). Pemanfaatan Sistem Temu Kembali Informasi dalam Pencarian Dokumen Menggunakan Metode Vector Space Model. *Journal of Information and Technology*, 5(2), 147–153.
- Siregar, R. R. A., Sinaga, F. A., Arianto, R., Studi, P., & Teknik, S. (2017). APLIKASI PENENTUAN DOSEN PENGUJI SKRIPSI MENGGUNAKAN METODE TF-IDF DAN, 2, 171–186.

- Suruali, N., Program, D., Teknik, S., Teknik, F., Ambon, U. P., & Paillin, D. B. (2010). ANALISIS DAN DESAIN SISTIM INFORMASI PERPUSTAKAAN, *04*(2).
- Utomo, M. S. (2013). Implementasi Stemmer Tala pada Aplikasi Berbasis Web. *Jurnal Teknologi Informasi DINAMIK Volume, 18*(1), 41–45.
- Eriksson, H.-E. & Penker, M., 2000. *Business Modeling with UML: Business Patterns at Work*. 1st penyunt. Canada: John Wiley & Sons, Inc.
- ligiaprapta17. (2018, September Rabu). Wordpress. Diambil kembali dari pengertian-information-retrieval-ir-peranan-ir-dan-contoh-contoh-ir/: <https://ligiaprapta17.wordpress.com/2015/03/03/pengertian-information-retrieval-ir-peranan-ir-dan-contoh-contoh-ir/>
- liyantanto. (2018, September Rabu). wordpress.com. Diambil kembali dari pencarian-dengan-metode-vektor-space-model-vsm/: <https://liyantanto.wordpress.com/2011/06/28/pencarian-dengan-metode-vektor-space-model-vsm/>
- Alexa. (2018, November Senin). 10 situs teratas di Indonesia dilihat dari aksesibilitasnya/: alexa.com

LAMPIRAN

A. UNIT TESTING

Keterangan :

- ✓ : Selesai di testing
- X : Belum selesai
- N : Tidak ada
- Y : Terdapat BUG
- BM : Belum ada perubahan
- TP : Sudah Diperbaiki
- TPP : Tidak Perlu Perbaikan

Tabel A.1 *Unit testing Iterasi 01*

Kode Requirement : KD:01			Kode Iterasi: KD:01:01	
Nama Unit	Status	Componen Unit	BUG	Perbaikan
<i>Convert object file Json ke Datatable</i>	✓	Baca file <i>Json</i>	✓	TPP
	✓	Convert menggunakan library <i>Newtonsoft.Json</i> , dan baca setiap data pada file <i>json</i>	✓	TPP

Tabel A.2 *Unit testing Iterasi 03*

Kode Requirement : KD:01			Kode Iterasi: KD:01:03	
Nama Unit	Status	Componen Unit	BUG	Perbaikan
Melakukan filter dokumen permasing-masing situs berita pada <i>datatable</i>	✓	Melakukan filter dokumen permasing-masing sumber berita	✗	TP

Tabel A.3 Unit testing Iterasi 04

Kode Requirement : KD:01			Kode Iterasi: KD:01:04	
Nama Unit	Status	Componen Unit	BUG	Perbaikan
Menampilkan <i>datatable</i> berita kedalam data <i>gridview</i> c# permasing-masing situs berita	✓	Menampilkan dokumen berita kedalam <i>gridview</i>	✘	TPP

Tabel A.4 Unit testing Iterasi 05

Kode Requirement : KD:01			Kode Iterasi: KD:01:05	
Nama Unit	Status	Componen Unit	BUG	Perbaikan
Menghitung data masing-masing jumlah konten berita permasing-masing berita	✓	Hitung jumlah detik.com	✘	TPP
	✓	Hitung jumlah tribunnews.com	✘	TPP
	✓	Hitung jumlah liputan6.com	✘	TPP

Tabel A.5 Unit testing Iterasi 06

Kode Requirement : KD:01			Kode Iterasi: KD:01:06	
Nama Unit	Status	Componen Unit	BUG	Perbaikan
Menampilkan detail berita yang dibaca	✓	Tampilkan berita	✘	TPP
	✓	<i>Readmore</i> berita	✓	TP

Tabel A.6 Unit testing Iterasi 07

Kode Requirement : KD:01			Kode Iterasi: KD:01:07	
Nama Unit	Status	Componen Unit	BUG	Perbaikan
Melakukan penyimpanan dokumen berita ke <i>database</i> dengan atribut file <i>json</i> yang disimpan (<i>idDoc</i>)	✓	Implementasi <i>text mining</i>	✓	TP
	✓	Baca frekuensi kata pada <i>array</i>	✓	TP

& date), melakukan <i>text mining stemming</i> TALA untuk menyimpan kata penting permasing- masing <i>idDoc</i> serta menghitung frekuensi kata dimasing-masing dokumen	✓	Implementasi <i>stemming</i> TALA	✓	TP
	✓	Simpan <i>idDoc</i> dan <i>date</i> dokumen	✓	TP

Tabel A.7 Unit testing Iterasi 01

Kode Requirement : KD:02			Kode Iterasi: KD:02:01	
Nama Unit	Status	Componen Unit	BUG	Perbaikan
<i>Split query</i> perkata dan <i>stemming</i> jika diperlukan dengan TALA, untuk mengembalikan query ke kata pembentuknya	✓	Integrasikan dengan <i>class</i> TALA untuk <i>request</i> kata pembentuk pada <i>array</i>	✓	TP
	✓	Integrasikan dengan <i>class</i> TALA untuk <i>request</i> data frekuensi pada <i>array</i>	✓	TP

Tabel A.8 Unit testing Iterasi 02

Kode Requirement : KD:02			Kode Iterasi: KD:02:02	
Nama Unit	Status	Componen Unit	BUG	Perbaikan
Implementasi metode vector space model dengan teknik perhitungan <i>TF-IDF</i> dan <i>cosine</i> <i>similarity</i>	✓	Simpan frekuensi kata pada array dokumen	✓	TP
	✓	Baca <i>id_dokumen</i> tertentu	✓	TPP
	✓	Hitung <i>df</i>	✓	TP
	✓	Hitung <i>IDF</i>	✓	TP
	✓	Hitung <i>TF-IDF</i>	✓	TP
	✓	Implementasi metode <i>cosine similarity</i>	✓	TP

Tabel A.9 Unit testing Iterasi 03

Kode Requirement : KD:02			Kode Iterasi: KD:02:03	
Nama Unit	Status	Componen Unit	BUG	Perbaikan
Menampilkan dokumen relevan ke datatable	✓	Request idDoc ke kelas VSM	✓	TP
	✓	Tampilkan doc sesuai dengan idDoc yang didapat	✓	TPP
	✓	Refresh table dan tombol detail ketika melakukan pencarian kembali	✓	TP

Tabel A.10 Unit testing Iterasi 01

Kode Requirement : KD:03			Kode Iterasi: KD:03:01	
Nama Unit	Status	Componen Unit	BUG	Perbaikan
Mengambil date dokumen terkait untuk ditampilkan kedalam grafik <i>time series</i> serta menghitung durasi kejadiannya	✓	Request idDoc ke kelas VSM	✓	TP
	✓	Tampilkan date sesuai dengan idDoc yang didapat	✓	TPP
	✓	Request frekuensi data pada array ke class TALA	✓	TP
	✓	SetGrafik dengan data pada array	✓	TP
	✓	Hitung panjang array hasil return frekuensi data pada array ke class TALA	✓	TP
	✓	Refresh grafik ketika melakukan pencarian kembali	✓	TP

Tabel A.11 *Unit testing Iterasi 01*

Kode Requirement : KD:04			Kode Iterasi: KD:04:01	
Nama Unit	Status	Componen Unit	BUG	Perbaikan
Mengambil riwayat berita idDoc dan judul berita yang pernah dibaca member	✓	Tampilkan semua data member	✓	TP
	✓	Tampilkan data aktivitas member	✓	TP
	✓	Menyimpan data riwayat member setelah membaca detail berita	✓	TP

Tabel A.12 *Unit testing Iterasi 01*

Kode Requirement : KD:05			Kode Iterasi: KD:05:01	
Nama Unit	Status	Componen Unit	BUG	Perbaikan
Daftar Member	✓	Periksa email yang sudah digunakan	✓	TP
	✓	Periksa form yang tidak valid	✓	TP

Tabel A.13 *Unit testing Iterasi 02*

Kode Requirement : KD:05			Kode Iterasi: KD:05:02	
Nama Unit	Status	Componen Unit	BUG	Perbaikan
Verifikasi Akun melalui EMAIL	✓	<i>Generate kode verifikasi</i>	✓	TP
	✓	Kirim kode melalui <i>email</i> pendaftar	✓	TP
	✓	Simpan kode ke <i>database</i>	✓	TP
	✓	Direct <i>page</i> ke <i>login page</i>	✓	TP

B. DOKUMEN SYSTEM TESTING

Tabel B.1 Black Box System testing 01

Kode Iterasi : KD:01					
Tujuan/luaran Iterasi	Input	Spesifikasi Fitur & kode fitur	Output	Status Perbaikan	Keterangan perbaikan
Mampu menampilkan semua dokumen berita dari situs berita yang dituju yaitu detik.com, tribunnews.com dan liputan6.com sekaligus hak akses ke sistem	Dokumen berita hasil <i>convert json to datatable</i>	Menampilkan semua dokumen berita tribunnews.com pada <i>gridview</i> beserta tombol <i>readmore</i> , dan menyimpan riwayat berita yang dibaca. F:101	Berhasil Menampilkan semua dokumen berita tribunnews.com pada <i>gridview</i> beserta tombol <i>readmore</i> , dan menyimpan riwayat berita yang dibaca.	x	
	Dokumen berita hasil <i>convert json to datatable</i>	Menampilkan semua dokumen berita detik.com pada <i>gridview</i> beserta tombol <i>readmore</i> , dan menyimpan riwayat berita yang dibaca. F:102	Berhasil Menampilkan semua dokumen berita detik.com pada <i>gridview</i> beserta tombol <i>readmore</i> , dan menyimpan riwayat berita yang dibaca.	x	
	Dokumen berita hasil <i>convert json to datatable</i>	Menampilkan semua dokumen berita liputan6.com pada <i>gridview</i> beserta tombol <i>readmore</i> , dan menyimpan riwayat berita yang dibaca. F:103	Berhasil Menampilkan semua dokumen berita liputan6.com pada <i>gridview</i> beserta tombol <i>readmore</i> , dan menyimpan riwayat berita yang dibaca.	x	

	<i>Email & Password</i>	Masuk kehalaman dashboard masing-masing level pengguna pada sistem (<i>Login</i>) F:104	Berhasil masuk ke tampilan <i>dashboard</i> masing-masing aktor sesuai dengan level penggunaannya.	x	-
	<i>Email</i>	Masuk kehalaman dashboard masing-masing level pengguna pada sistem (<i>Login</i>) F:104	Berhasil Melakukan peringatan terkait perlunnya memasukkan semua data yang diminta pada saat ingin masuk sistem yaitu mengisi email dan password dengan menampilkan pesan (“Peringatan, Email tidak boleh kosong”)	x	-
	<i>Password</i>	Masuk kehalaman dashboard masing-masing level pengguna pada sistem (<i>Login</i>) F:104	Berhasil Melakukan peringatan terkait perlunnya memasukkan semua data yang diminta pada saat ingin masuk sistem yaitu mengisi email dan password dengan menampilkan pesan (“Peringatan, password tidak boleh kosong”)	x	-

	<i>Email tidak valid contain (@)</i>	Masuk kehalaman dashboard masing-masing level pengguna pada sistem (<i>Login</i>) F:104	Berhasil Melakukan peringatan terkait perlunya memasukkan semua data yang diminta pada saat ingin masuk sistem yaitu mengisi email dan password dengan menampilkan pesan (“menampilkan pesan tepat di area <i>field</i> email yang menunjukkan bahwa email harus mengandung @”)	x	-
--	--------------------------------------	---	--	---	---

Tabel B.2 *Black Box System testing* 02

Kode Iterasi : KD:02					
Tujuan/luaran Iterasi	Input	Spesifikasi Fitur & Kode Fitur	Output	Status Perbaikan	Keterangan perbaikan
Sistem mampu menampilkan dokumen berita yang relevan sesuai dengan kata kunci pencarian	<i>Query</i> (kata pencarian)	Menampilkan semua dokumen berita relevan hasil pencarian tribunnews.com pada <i>gridview</i> beserta tombol <i>readmore</i> , dan menyimpan riwayat berita yang dibaca. F:201	Berhasil Menampilkan semua dokumen berita relevan hasil pencarian tribunnews.com pada <i>gridview</i> beserta tombol <i>readmore</i> , dan menyimpan riwayat berita yang dibaca.	x	-

Query (kata pencarian)	Menampilkan semua dokumen berita relevan hasil pencarian berita detik.com pada <i>gridview</i> beserta tombol <i>readmore</i> , dan menyimpan riwayat berita yang dibaca. F:202	Berhasil Menampilkan semua dokumen berita relevan hasil pencarian detik.com pada <i>gridview</i> beserta tombol <i>readmore</i> , dan menyimpan riwayat berita yang dibaca.	x
Query (kata pencarian)	Menampilkan semua dokumen berita relevan hasil pencarian liputan6.com pada <i>gridview</i> beserta tombol <i>readmore</i> , dan menyimpan riwayat berita yang dibaca. F:203	Berhasil Menampilkan semua dokumen berita relevan hasil pencarian liputan6.com pada <i>gridview</i> beserta tombol <i>readmore</i> , dan menyimpan riwayat berita yang dibaca.	x
Query (kata pencarian)	Menampilkan semua dokumen berita hasil pencarian pada <i>gridview</i> beserta tombol <i>readmore</i> , dan menyimpan riwayat berita yang dibaca. F:204	Berhasil Menampilkan semua dokumen berita relevan hasil pencarian pada semua situs berita ke <i>gridview</i> beserta tombol <i>readmore</i> , dan menyimpan riwayat berita yang dibaca.	x

	<i>Query</i> (kata pencarian) “kosong”	Pada masing fitur F:201, F:202, F:203 F:204, F:205 dilengkapi dengan peringatan bahwa <i>query</i> tidak boleh kosong	Berhasil Menghentikan proses pencarian dokumen apabila <i>query</i> kosong tepat pada area input <i>query</i> akan muncul pemberitahuan bahwa <i>field</i> tersebut tidak boleh kosong apabila akan melakukan proses pencarian dokumen berita.	x	
--	--	---	--	----------	--



Tabel B. 3 *Black Box System testing* 03

Kode Iterasi : KD:03					
Tujuan/luaran Iterasi	Input	Spesifikasi Fitur & Kode Fitur	Output	Status Perbaikan	Keterangan perbaikan
Sistem mampu menampilkan grafik durasi berita sesuai dengan dokumen relevan yang ditampilkan, sekaligus mampu menampilkan durasi kejadian berdasarkan dokumen pencarian	Tanggal publikasi berita	Menampilkan grafik durasi kejadian berita F:301	Berhasil Menampilkan grafik kejadian berita berdasarkan tanggal publikasi berita yang ditampilkan setelah proses pencarian berita.	x	
	Dokumen hasil pencarian berita	<i>Refresh</i> tabel ketika menampilkan dokumen berita hasil pencarian. F:302	Berhasil <i>Refresh</i> tabel ketika menampilkan dokumen berita hasil pencarian.	x	
	Dokumen hasil pencarian berita	<i>Refresh</i> grafik <i>time series</i> ketika melakukan pencarian kembali F:303	Berhasil <i>Refresh</i> grafik <i>time series</i> ketika melakukan pencarian kembali	x	
	Dokumen hasil pencarian berita	Menampilkan durasi kejadian pencarian berdasarkan dari dokumen hasil pencarian F:304	Berhasil Menampilkan durasi kejadian pencarian berdasarkan dari dokumen hasil pencarian	x	

	Dokumen hasil pencarian berita	Melakukan proses filter diokumen berita berdasarkan tanggal pada dokumen pencarian F:305	Berhasil Melakukan proses filter diokumen berita berdasarkan tanggal pada dokumen pencarian, dan menampilkan dokumen berita pada tanggal yang dipilih	x	
--	--------------------------------	--	---	---	--

Tabel B.4 *Black Box System testing* 04

Kode Iterasi : KD:04					
Tujuan/luaran Iterasi	Input	Spesifikasi Fitur & Kode Fitur	Output	Status Perbaikan	Keterangan perbaikan
Sistem mampu melihat <i>log activity member</i>	Dokumen berita	Menyimpan riwayat dokumen berita <i>member</i> yaitu judul dan id_doc F:401	Berhasil Menyimpan riwayat dokumen berita yang dibaca oleh <i>member</i>	x	
	Data member hasil registrasi sistem	Menampilkan detail data member F:402	Berhasil Menampilkan detail data <i>member</i>	x	-
	Data member yang dipilih	Menampilkan semua <i>log</i> aktivitas permasing-masing <i>member</i> F:403	Berhasil Menampilkan semua <i>log</i> aktivitas permasing-masing <i>member</i>	x	

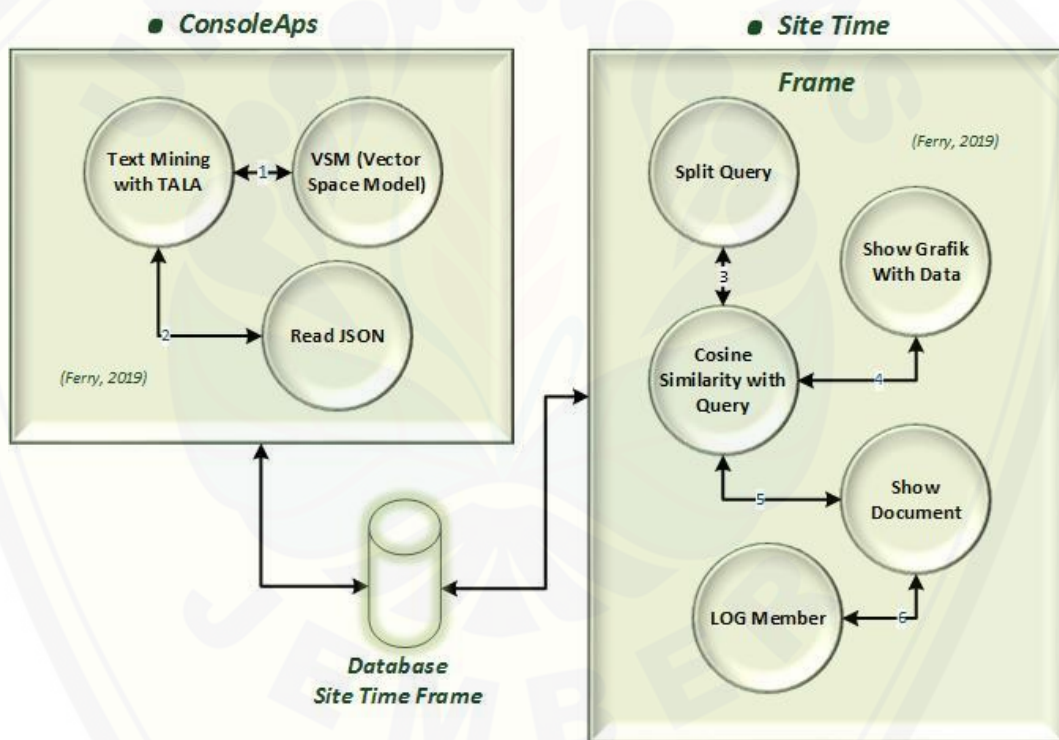
	-	Menampilkan log aktivitas permasing-masing <i>member</i> yang terbaru F:404	Berhasil Menampilkan log aktivitas permasing-masing <i>member</i> yang terbaru	x	
--	---	---	--	---	--

Tabel B.5 *Black Box System testing 05*Kode Iterasi : **KD:05**

Tujuan/luaran Iterasi	Input	Spesifikasi Fitur & Kode Fitur	Output	Status Perbaikan	Keterangan perbaikan
Sistem mampu melayani pendaftaran online melalui verifikasi email	Data diri <i>pendaftar</i>	Validasi form pendaftaran permasing-masing atribut (mulai dari kosong dan tidak sesuai) F:501	Berhasil melakukan Validasi <i>form</i> pendaftaran permasing-masing atribut (mulai dari kosong dan tidak sesuai) sehingga masing-masing atribut data yang tersimpan sesuai dengan kriteria yang ditentukan	x	-

C. DOKUMEN PENGUJIAN INTEGRASI AKHIR

Site time frame dikembangkan berdasarkan beberapa modul. Pemilihan beberapa modul pada sistem ditentukan dari beberapa fungsional utama yang ada pada sistem, kemudian setiap modul yang dikembangkan terdiri dari beberapa fitur yang ada pada sistem. Peneliti melakukan setiap modul yang ada sesuai dengan keterkaitan antara modul dengan modul lain. Hal tersebut dilakukan karena peneliti melakukan pengembangan untuk setiap masing-masing modul. Beberapa modul yang akan diuji dengan integrasi akhir dapat dilihat pada gambar C.1.



Gambar C.1 Daftar Modul *site time frame* yang akan diuji integrasi akhir

Table C. 1 Integrasi modul *Text mining with TALA & VSM*

Integrasi 1					
Modul 1	Modul 2	Proses Transfer Data	BUG	Perbaikan	Keterangan
<i>Text mining with TALA</i>	VSM	Mining Kata pada masing-masing dokumen berita	✓	✓	Kata yang seharusnya perlu <i>stemming</i> masing tidak dilakukan proses <i>stemming</i>
		Frekuensi kata pada masing-masing dokumen berita	x	x	-

Tabel C. 2 Integrasi modul *Text mining with TALA & Read Json*

Integrasi 2					
Modul 1	Modul 2	Proses Transfer Data	BUG	Perbaikan	Keterangan
<i>Text mining with TALA</i>	Read Json	Baca Semua atribut dokumen berita pada json	x	x	-
		Mengembalikan data id dokumen, tanggal publikasi dan isi konten berita	x	x	-

Tabel C. 3 Integrasi modul *Split Query & Cosine Similarity With Query*

Integrasi 3					
Modul 1	Modul 2	Proses Transfer Data	BUG	Perbaikan	Keterangan
<i>Split Query</i>	<i>Cosine Similarity With Query</i>	Potong <i>query</i> masuk sesuai aturan <i>text mining</i>	x	x	-
		Baca <i>query</i> hasil <i>split</i> dan hitung kemiripannya dengan setiap dokumen berita	x	x	-

Tabel C. 4 Integrasi modul *Show Grafik With Data & Cosine Similarity With Query*

Integrasi 4					
Modul 1	Modul 2	Proses Transfer Data	BUG	Perbaikan	Keterangan
<i>Show Grafik With Data</i>	<i>Cosine Similarity With Query</i>	Menghitung kemiripan <i>query</i> dengan setiap dokumen yang terlibat	x	x	-
		Mengembalikan nilai kemiripan <i>query</i> dengan dokumen berita sekaligus filter nilai diatas 0,788 kemiripan yang akan ditampilkan	✓	✓	Menghasilkan nilai kemiripan diluar jangkauan nilai cosine similarity antara 0-1

		Mengambil data tanggal publikasi berita berdasarkan dokumen yang memiliki nilai diatas 0,8 untuk dihitung durasinya dan diampikan dalam grafik <i>time series</i>	✓	✓	Kesalahan Proses mengumpulkan tanggal yang sama mengakitbatkan beberapa tanggal tidak ditampilkan
--	--	---	---	---	---

Tabel C.5 Integrasi modul *Show Document & Cosine Similarity With Query*

Integrasi 5					
Modul 1	Modul 2	Proses Transfer Data	BUG	Perbaikan	Keterangan
<i>Show Document</i>	<i>Cosine Similarity With Query</i>	Tampilkan detail berita berdasarkan masing-masing id berita yang memiliki nilai kemiripan diatas 0,788	x	x	-

Tabel C.6 Integrasi modul *Show Document* & *Log Member*

Integrasi 6					
Modul 1	Modul 2	Proses Transfer Data	BUG	Perbaikan	Keterangan
<i>Show Document</i>	<i>Log Member</i>	Simpan judul dan id dokumen yang dibaca oleh pengguna	✓	✓	Dokumen berita yang dibara secara full dalam satu judul berita tidak sesuai dengan yang dipilih, oleh karena itu kesesuaian dokumen yang disimpan menjadi salah

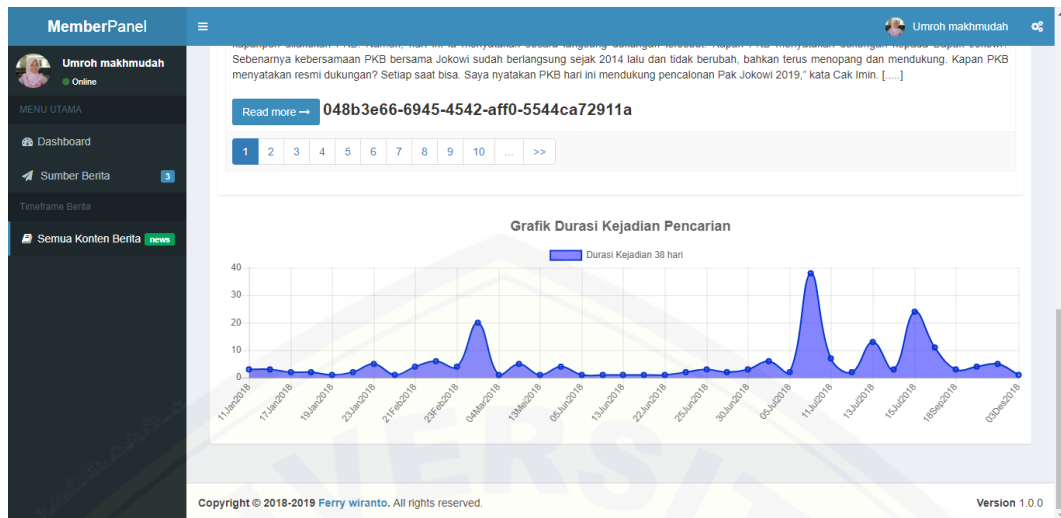
D. TAMPILAN SISTEM



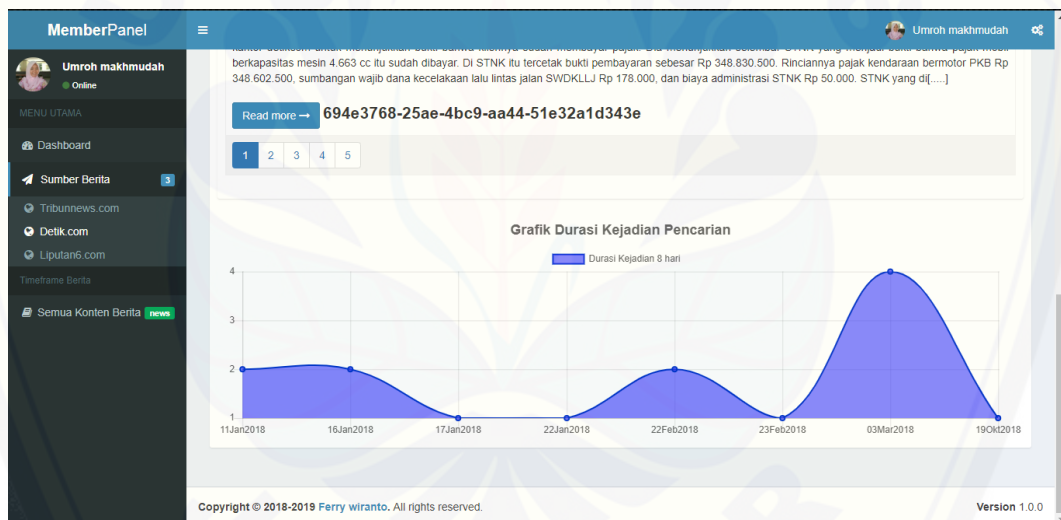
Gambar D.1 Dashboard Member



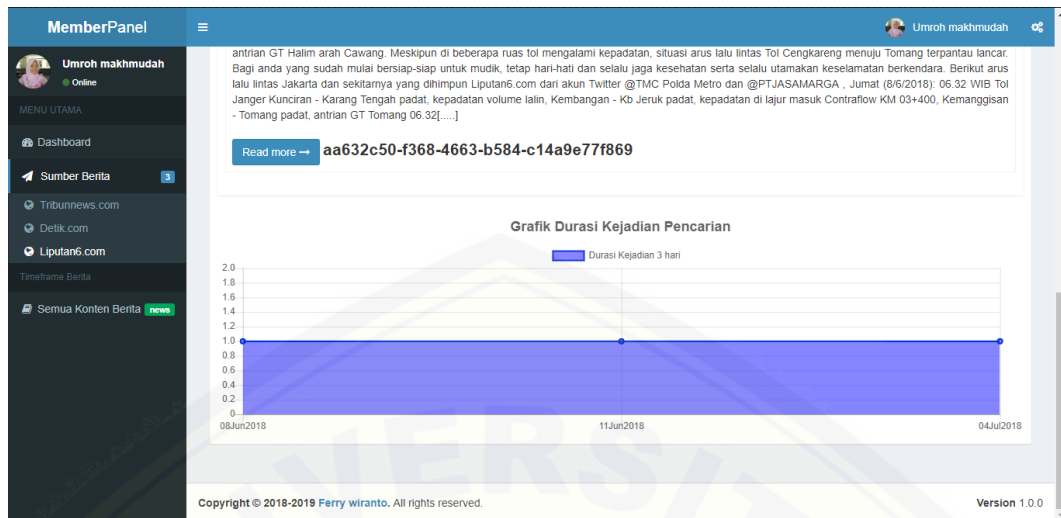
Gambar D.2 Dokumen Berita member pada Sumber Tribunnews.com Member



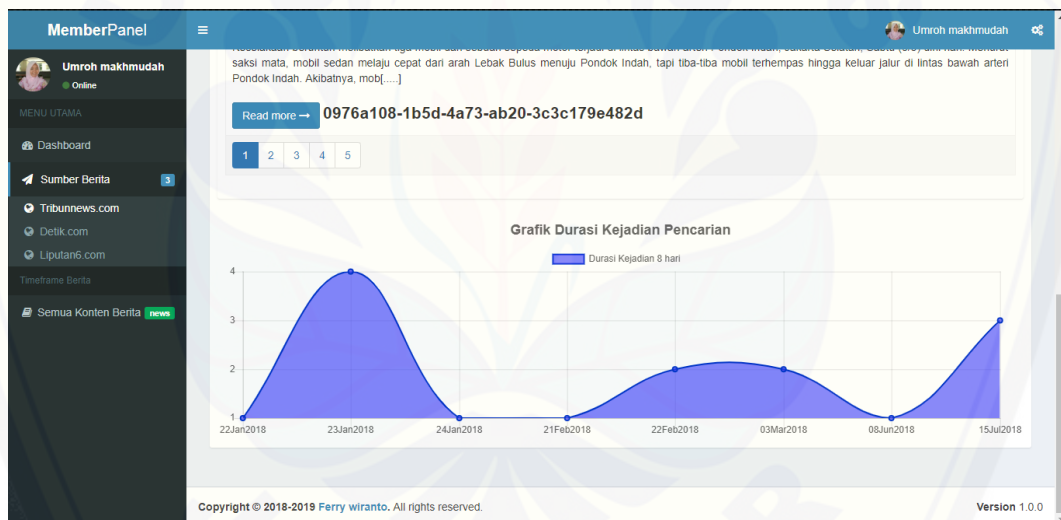
Gambar D.3 Dokumen pencarian berita member pada semua situs berita, beserta grafik *time frame* dan durasi kejadian.



Gambar D.4 Dokumen pencarian berita member pada situs berita detik.com, beserta grafik *time frame* dan durasi kejadian



Gambar D.5 Dokumen pencarian berita member pada situs berita liputan6.com, beserta grafik *time frame* dan durasi kejadian



Gambar D.6 Dokumen pencarian berita member pada situs berita tribunnews.com, beserta grafik *time frame* dan durasi kejadian

E. Perhitungan *Similarity* Antara Dokumen Berita dengan *Query*

Dokumen berita yang digunakan dalam penelitian ini berjumlah 4.812, semua dokumen tersebut sudah direpresentasikan kedalam ruang n -dimensi vector. Semisal diketahui terdapat 5 Dokumen berita sample dari 4.812, serta ringkasan isi dokumen semisal sebagai berikut:

1. Doc1. [Seolah tak terpengaruh dengan adanya rencana pemindahan Kedutaan Besar Australia dari Tel Aviv ke Yerusalem, Indonesia dan Australia akan menandatangani Kerja Sama Ekonomi Komprehensif Indonesia-Australia (Indonesia-Australia Comprehensive Economic Partnership Agreement/IA CEPA) pada akhir tahun mendatang.]
2. Doc2. [Partai Golkar mendukung program Jokowi dan menepis adanya unsur politis di balik Dana Kelurahan. Kebijakan Dana Kelurahan merupakan bagian dari upaya pemerataan keadilan anggaran yang selama ini telah ditunjukkan dengan kebijakan Dana Desa semenjak diberlakukannya UU Desa sejak tahun 2015 lalu, ujar Ketua DPP Golkar Ace Hasan Syadzily, Jumat (19/10/2018).]
3. Doc3. [Presiden Joko Widodo (Jokowi) mengaku banyak mendapat keluhan terkait dana untuk tingkat kelurahan. Untuk itu, tahun depan dia akan mengeluarkan kebijakan Dana Kelurahan dan Dana Operasional Desa. Jokowi mengatakan, saat ini payung hukum untuk regulasi itu tengah dikaji. Diperkirakan, dana operasional desa akan diambil dari Dana Desa sebanyak 5 persen.]
4. Doc4. [Mardani enggan mengomentari soal apakah kebijakan Jokowi tersebut terkesan politik jelang Pilpres 2019. Dia menyerahkan ke masyarakat yang akan menilai. \ "Biar masyarakat yang menilai. Tapi keputusan apapun yang dibuat Presiden akan dinilai publik]
5. Doc5. [Cawapres Ma'ruf Amin melakukan kunjungan ke Madura. Ia meminta dukungan dari masyarakat agar menang pada Pilpres 2019.]

Kasus. Dokumen yang dicari, Apabila terdapat kata kunci pencarian atau Query (Q) "Pilpres 2019", carilah dokumen yang relevan dengan Q tersebut. Maka langkah 1. Lakukan proses *Text Mining* (*Case Folding* – *Tokenizing* – *Filtering* –

Stemming(TALA) terhadap semua dokumen tersebut ($n=5$) Hasil Akhir proses *Text Mining* Dokumen sebagai berikut.

Tabel E.1 Contoh beberapa dokumen beserta *term* kata yang dikandung

Query	Dokument	Term akhir setiap dokumen
"Pilpres 2019"	Doc1	Rencana pindah kudeta datang duta Besar Australia Yerusalem Indonesia Australia Kerja Ekonomi Comprehensive Economic Partnership akhir tahun 2019
	Doc2	Partai Golkar dukung program Jokowi unsur politik balik dana lurah tahun keluar kebijakan adil anggaran rata UU desa tahun 2015 DPP Golkar Ace Hasan Syadzily Jumat (19/10/2018).
	Doc3	Presiden Joko Widodo Jokowi ngaku keluh dana tingkat pilpres lurah tahun depan keluar kebijakan Dana Kelurahan Dana Operasional Desa Jokowi payung hukum regulasi tengah kaji dana operasional desa ambil Dana Desa 5 persen
	Doc4	Mardani enggan kebijakan Jokowi kesan politik jelang pilpres 2019 serah masyarakat nilai masyarakat putus presiden publik
	Doc5	Cawapres Ma'ruf Amin kunjung Madura minta dukung masyarakat menang pilpres 2019

Langkah 2.

Tabel E.2 Proses Penentuan bobot setiap term dari semua dokumen tersebut.

Term	tf						df	idf (log(n/df))	tf_idf						
	Q	D1	D2	D3	D4	D5			Q	D1	D2	D3	D4	D5	
Rencana		1					1	0,6990	0,0000	0,6990	0,0000	0,0000	0,0000	0,0000	0,0000
pindah		1					1	0,6990	0,0000	0,6990	0,0000	0,0000	0,0000	0,0000	0,0000
kudeta		1					1	0,6990	0,0000	0,6990	0,0000	0,0000	0,0000	0,0000	0,0000
datang		1					1	0,6990	0,0000	0,6990	0,0000	0,0000	0,0000	0,0000	0,0000
duta		1					1	0,6990	0,0000	0,6990	0,0000	0,0000	0,0000	0,0000	0,0000
Besar		1					1	0,6990	0,0000	0,6990	0,0000	0,0000	0,0000	0,0000	0,0000
Australia		1					1	0,6990	0,0000	0,6990	0,0000	0,0000	0,0000	0,0000	0,0000
Yerusalem		1					1	0,6990	0,0000	0,6990	0,0000	0,0000	0,0000	0,0000	0,0000
Indonesia		1					1	0,6990	0,0000	0,6990	0,0000	0,0000	0,0000	0,0000	0,0000
Kerja		1					1	0,6990	0,0000	0,6990	0,0000	0,0000	0,0000	0,0000	0,0000
Ekonomi		1					1	0,6990	0,0000	0,6990	0,0000	0,0000	0,0000	0,0000	0,0000
Comprehensive		1					1	0,6990	0,0000	0,6990	0,0000	0,0000	0,0000	0,0000	0,0000
Economic		1					1	0,6990	0,0000	0,6990	0,0000	0,0000	0,0000	0,0000	0,0000
Partnership		1					1	0,6990	0,0000	0,6990	0,0000	0,0000	0,0000	0,0000	0,0000
Akhir		1					1	0,6990	0,0000	0,6990	0,0000	0,0000	0,0000	0,0000	0,0000
Tahun		1	1	1			3	0,2218	0,0000	0,2218	0,2218	0,2218	0,0000	0,0000	0,0000
Partai			1				1	0,6990	0,0000	0,0000	0,6990	0,0000	0,0000	0,0000	0,0000
Golkar			1				1	0,6990	0,0000	0,0000	0,6990	0,0000	0,0000	0,0000	0,0000
Dukung			1			1	2	0,3979	0,0000	0,0000	0,3979	0,0000	0,0000	0,0000	0,3979
Program			1				1	0,6990	0,0000	0,0000	0,6990	0,0000	0,0000	0,0000	0,0000
Jokowi			1				1	0,6990	0,0000	0,0000	0,6990	0,0000	0,0000	0,0000	0,0000
Unsur			1				1	0,6990	0,0000	0,0000	0,6990	0,0000	0,0000	0,0000	0,0000
Politik			1				1	0,6990	0,0000	0,0000	0,6990	0,0000	0,0000	0,0000	0,0000
Balik			1				1	0,6990	0,0000	0,0000	0,6990	0,0000	0,0000	0,0000	0,0000
Dana			1	5			2	0,3979	0,0000	0,0000	0,3979	1,9897	0,0000	0,0000	0,0000
Lurah			1				1	0,6990	0,0000	0,0000	0,6990	0,0000	0,0000	0,0000	0,0000
Keluar			1				1	0,6990	0,0000	0,0000	0,6990	0,0000	0,0000	0,0000	0,0000
Kebijakan			1	1	1		3	0,2218	0,0000	0,0000	0,2218	0,2218	0,2218	0,0000	0,0000
Adil			1				1	0,6990	0,0000	0,0000	0,6990	0,0000	0,0000	0,0000	0,0000

F. *User Story Site Time Frame*Table F.1 *User Story as Admin*

<i>User Story ID</i>	<i>I want to</i>	<i>So That</i>
1.	Melihat informasi tentang website <i>site time frame</i> pada <i>frontpage</i> website	Semua pengguna website tidak langsung dihadapkan dengan halaman login, tetapi perlu keterangan terkait informasi website terlebih dahulu, seperti tujuan pembangunan website, informasi ilmu atau teori yang digunakan sekaligus kontak dari pemilik atau admin website.
2.	Melihat informasi seluruh data pada website di <i>dashboard</i> .	Dapat melihat informasi dari sumber berita yang dituju, mulai dari berapa total berita permasing-masing sumber berita, melihat informasi <i>member</i> yang sedang bergabung dengan batas 3 member terbaru. Dapat melihat informasi dalam bentuk grafik yang mampu menyajikan data jumlah dokumen berita pada masing-masing sumber berita berdasarkan bulan. Menampilkan jumlah total dimensi kata yang dimiliki oleh website dan jumlah

		total kata yang tersimpan pada <i>database</i> .
3.	Melihat daftar member yang sedang bergabung pada website	Melihat informasi member yang sedang bergabung beserta dengan atribut masing-masing, seperti foto <i>member</i> , nama, alamat, dll.
4.	Melihat aktivitas member pada website	Melihat aktivitas terkait dokumen berita apa saja yang pernah dilihat oleh <i>member</i> .
5.	Melihat dokumen berita pada masing-masing sumber situs berita.	Melihat semua dokumen berita yang dimiliki pada situs berita yang dipilih, dan menyediakan fitur untuk melakukan proses pencarian dokumen berita agar mendapatkan dokumen yang sesuai dengan kata pencarian, serta dapat menghitung durasi kejadian berdasarkan hari sekaligus durasi kejadian pada situs berita yang dipilih dalam bentuk grafik <i>time series</i>
6.	Melihat dokumen berita pada semua sumber situs berita yang dituju yaitu tribunnews.com , detik.com dan liputan6.com .	Melihat semua dokumen berita yang dimiliki pada semua situs berita, dan menyediakan fitur untuk melakukan proses pencarian dokumen berita agar mendapatkan

		dokumen yang sesuai dengan kata pencarian, serta dapat menghitung durasi kejadian berdasarkan hari sekaligus durasi kejadian pada semua situs berita dalam bentuk grafik <i>time series</i>
--	--	---

Tabel F.2 *User Story as Member (Anggota)*

<i>User Story ID</i>	<i>I want to</i>	<i>So That</i>
7.	Melihat informasi tentang website <i>site time frame</i> pada <i>frontpage</i> website	Pengguna website tidak langsung dihadapkan dengan halaman login, tetapi perlu keterangan terkait informasi website terlebih dahulu, seperti tujuan pembangunan website, informasi ilmu atau teori yang digunakan sekaligus kontak dari pemilik atau admin website.
8.	Melihat informasi seluruh data pada website di <i>dashboard</i> .	Dapat melihat informasi dari sumber berita yang dituju, mulai dari berapa total berita permasing-masing sumber berita, Dapat melihat informasi dalam bentuk grafik yang mampu menyajikan data jumlah dokumen berita pada masing-masing

		sumber berita berdasarkan bulan. Menampilkan jumlah total dimensi kata yang dimiliki oleh website dan jumlah total kata yang tersimpan pada <i>database</i> .
9.	Melihat dokumen berita pada masing-masing sumber situs berita.	Melihat semua dokumen berita yang dimiliki pada situs berita yang dipilih, dan menyediakan fitur untuk melakukan proses pencarian dokumen berita agar mendapatkan dokumen yang sesuai dengan kata pencarian, serta dapat menghitung durasi kejadian berdasarkan hari sekaligus durasi kejadian pada situs berita yang dipilih dalam bentuk grafik <i>time series</i>
10.	Melihat dokumen berita pada semua sumber situs berita yang dituju yaitu tribunnews.com , detik.com dan liputan6.com .	Melihat semua dokumen berita yang dimiliki pada semua situs berita, dan menyediakan fitur untuk melakukan proses pencarian dokumen berita agar mendapatkan dokumen yang sesuai dengan kata pencarian, serta dapat menghitung durasi kejadian berdasarkan hari sekaligus

		durasi kejadian pada semua situs berita dalam bentuk grafik <i>time series</i>
--	--	--

Tabel F. 3 *User Story as Guest* (Pengunjung)

<i>User Story ID</i>	<i>I want to</i>	<i>So That</i>
11.	Melihat informasi tentang website <i>site time frame</i> pada <i>frontpage</i> website	Pengguna website tidak langsung dihadapkan dengan halaman login, tetapi perlu keterangan terkait informasi website terlebih dahulu, seperti tujuan pembangunan website, informasi ilmu atau teori yang digunakan sekaligus kontak dari pemilik atau admin website.
12.	Mendaftar sebagai member <i>site time frame</i>	Melakukan pendaftaran sebagai member dan aktivasi akun menggunakan verifikasi <i>Email</i> .


```

355 private void loadChartSearch(string[] data)
356 {
357     int[] frekuensi = new int[data.Length];
358     int i, j, ctr;
359     List<string> sumbuX = new List<string>();
360     List<int> sumbuY = new List<int>();
361     string[] sumbuXX = new string[data.Length];
362     result_sumbuXX = new string[data.Length];
363     int[] sumbuYY = new int[data.Length];
364     //inisialisasi awal chart
365     for (int fer = 0; fer < result_sumbuXX.Length; fer++)
366     {
367         result_sumbuXX[fer] = "";
368         sumbuYY[fer] = 0;
369     }
370     //baca data dan tampilkan ke chart
371     for (i = 0; i < data.Length; i++)
372     {
373         frekuensi[i] = -1;
374     }
375     for (i = 0; i < data.Length; i++)
376     {
377         ctr = 1;
378         for (j = i + 1; j < data.Length; j++)
379         {
380             if (data[i].Equals(data[j]))
381             {
382                 ctr++;
383                 frekuensi[j] = 0;
384             }
385         }
386         if (frekuensi[i] != 0)
387         {
388             frekuensi[i] = ctr;
389         }
390     }
391     for (i = 0; i < data.Length; i++)

```

Gambar G. 3 Dokumentasi *source code* untuk menampilkan data pada grafik *time series*

```

411 var DataJS =
412     "<script>" +
413     "let myChart = document.getElementById('barChart').getContext('2d');" +
414     "Chart.defaults.global.defaultFontSize = 12;" +
415     "let ChartPopulation = new Chart(myChart, {" +
416     "type: 'line'," +
417     "data:" +
418     "{" +
419     "labels: [" + string.Join(", ", result_sumbuXX) + "]," +
420     "datasets: [{" +
421     "label: 'Durasi Kejadian " + sumbuXX.Length + " hari' ," +
422     "data: [" +
423     "    " + string.Join(", ", sumbuYY) + " " +
424     "]," +
425     "backgroundColor: [" +
426     "    'rgba(51,51,255,0.6)'" +
427     "]," +
428     "borderWidth: 2," +
429     "borderColor: '#0033cc'," +
430     "hoverBorderWidth: 3," +
431     "hoverBorderColor: 'red'" +
432     "}]}" +
433     "}," +
434     "options:" +
435     "{" +
436     "title:" +
437     "{" +
438     "display: true," +
439     "text: 'Grafik Durasi Kejadian Pencarian'," +
440     "fontSize: 18" +
441     "}," +
442     "}" +
443     "});" +
444     "</script>";
445     ScriptManager.RegisterStartupScript(this, this.GetType(), Guid.NewGuid().ToString(),
446     DataJS, false);
447 }

```

G. 4 Dokumentasi *source code* lanjutan untuk menampilkan data pada grafik *time series*

```
35
36 public string getIdf_Term(string term)
37 {
38     connectionClass con = new connectionClass();
39     con.openConnection();
40     double getIdf_Term = 0;
41     string QgetIdf_Term = "SELECT \"IDF\" FROM public.\"Term\" where \"Term\"='\" + term + \"'";
42     DataTable result = con.getResult(QgetIdf_Term);
43     getIdf_Term = Convert.ToDouble(result.Rows[0][\"IDF\"]);
44     return getIdf_Term.ToString();
45 }
46
47
48 public double cekKetersediaan_Term(string kata)
49 {
50     connectionClass con = new connectionClass();
51     con.openConnection();
52     string query = "SELECT \"Term\" FROM public.\"Term\" where \"Term\"='\" + kata + \"'";
53     DataTable result = con.getResult(query);
54     if (result.Rows.Count >= 1)
55     {
56         con.openConnection();
57         double getIdf_Term = 0;
58         string QgetIdf_Term = "SELECT \"IDF\" FROM public.\"Term\" where \"Term\"='\" + kata + \"'";
59         DataTable result2 = con.getResult(QgetIdf_Term);
60         getIdf_Term = Convert.ToDouble(result2.Rows[0][\"IDF\"]);
61         return getIdf_Term;
62     }
63     else
64     {
65         return 0;
66     }
67 }
68
69 }
```

Gambar G. 5 Dokumentasi *source code* proses representasi kata pada dokumen kedatabase

```
79 public string cariID_Term(string data)
80 {
81     koneksi con = new koneksi();
82     con.openConnection();
83     string termCari = null;
84     string termMasuk = "SELECT * FROM public.\"Term\" where \"Term\"='\" + data + \"'";
85     DataTable result2 = con.getResult(termMasuk);
86     idDoc = new string[result2.Rows.Count];
87     if (result2.Rows.Count >= 1)
88     {
89         termCari = result2.Rows[0][\"id_Term\"].ToString();
90         con.stopAccess();
91         return termCari;
92     }
93     else
94     {
95         return \"kosong\";
96     }
97 }
98
99
100 public double cariTF_IDF(string id_doc, string idTerm)
101 {
102     koneksi con = new koneksi();
103     con.openConnection();
104     double termCari;
105     string termMasuk = "SELECT \"TF-IDF\" FROM public.\"bobotTerm\" where \"
106     \"idDokumen\"='\" + id_doc + \"\" and \"id_Term\"='\" + idTerm + \"'";
107     DataTable result2 = con.getResult(termMasuk);
108     if (result2.Rows.Count >= 1)
109     {
110         termCari = Convert.ToDouble(result2.Rows[0][\"TF-IDF\"]);
111         con.stopAccess();
112         return termCari;
113     }
114     else
115     {
```

Gambar G. 6 Dokumentasi *source code* lanjutan proses representasi kata pada dokumen kedatabase


```

47     }
48     public void insertDoc()
49     {
50         string idDokument, dateDoc;
51         for (int i = 4001; i <= 4812; i++)
52         {
53             con.openConnection();
54             string textJson = null;
55             int run = i;
56             textJson = readJsonRun.readtheJsonOffline(run);
57             idDokument = readJsonRun.getID_Doc(run);
58             dateDoc = readJsonRun.getDateId_Dokument(run);
59             tala.runStemming_Tala(textJson);
60             tala.getFrekuensiKata(idDokument);
61             tala.insert_ToDokumen(idDokument, dateDoc);
62             con.stopAccess();
63         }
64     }
65     public void setVector_Doc()
66     {
67         Console.WriteLine("Proses Menghitung Panjang doc.....");
68         tala.getLongDoc();
69         Console.WriteLine("Menghitung Panjang doc Selesai.....");
70         Console.WriteLine(".....");
71         Console.WriteLine("Proses Menghitung IDF.....");
72         tala.setIDF();
73         Console.WriteLine("Proses Menghitung IDF SELESAI.....");
74         Console.WriteLine(".....");
75         Console.WriteLine("Proses Menghitung TF-IDF.....");
76         tala.setTF_IDF();
77         Console.WriteLine("Proses Menghitung TF-IDF SELESAI.....");
78     }

```

Gambar G. 7 Dokumentasi *source code consoleAps* proses representasi dokumen kedalam vector dengan menggunakan teknik *TF-IDF* yang disimpan ke database

```

521     for (int i = 0; i < words.Length; i++)
522     {
523         arrayWord[i] = text.Split(delimiterChars)[i].ToString().ToLower();
524         Console.WriteLine("[ " + arrayWord[i] + " ]");
525         //Proses filtering, atau menghilangkan kata sambung//
526
527         if (count.getKataSambung(arrayWord[i]) == true)
528         {
529             //proses menghilangkan wordList//
530             Console.WriteLine("[ " + arrayWord[i] + " ]" + "=> masuk kata sambung, kata dalam index ini");
531             arrayWord[i] = "";
532         }
533         //memeriksa apakah sudah kata dasar//
534         else if (count.getKataDasar(arrayWord[i]) == true)
535         {
536             arrayWord[i] = arrayWord[i];
537             this.con = new koneksi();
538             Console.WriteLine("[ " + arrayWord[i] + " ]" + "[ ini kata dasar_ so Fix ]");
539         }
540         //proses tala ke-1// hilangkan sandang kalau ada
541         else if (count.ccekSandang(arrayWord[i]) > 0)//cek apakah punya sandang
542         {
543             Console.WriteLine("[ " + arrayWord[i] + " ]" + " Masuk Proses Tala 1");
544             arrayWord[i] = count.replace_Sandang(arrayWord[i]); //proses hapus sandang
545
546             if (count.getKataDasar(arrayWord[i]) == true)
547             {
548                 Console.WriteLine("sudah kata dasar jadi " + "[ " + arrayWord[i] + " ]");
549             }
550             else if (count.ccekAkhiran(arrayWord[i]) > 0)//cek akhirnya
551             {
552                 Console.WriteLine("[ " + arrayWord[i] + " ]" + " Masuk Proses Tala 1");
553                 arrayWord[i] = count.replace_Akhiran(arrayWord[i]); //proses hapus akhiran
554                 if (count.getKataDasar(arrayWord[i]) == true)
555                 {
556                     Console.WriteLine("sudah kata dasar jadi " + "[ " + arrayWord[i] + " ]");
557                 }
558             }
559         }
560     }

```

Gambar G. 8 Dokumentasi *source code consoleAps* pada teknik *text mining* dengan menggunakan metode stemming TALA

H. Dokumen *Retrospective*

The screenshot displays the FunRetrospective web application interface. At the top, there is a navigation bar with the logo "FunRetros", a search bar "Filter your cards", a "Sort: order" dropdown, and links for "Dashboard" and "Layout". On the right side of the navigation bar, there is a "Prime Directive" link and a settings icon. Below the navigation bar, there is a "Skripsi" section with the text "Set the context of the retrospective here...". To the right of this section, there are links for "vote stats", "New column", and "settings". A green banner below the navigation bar states "Votes are disabled" with an "Enable votes" button. Below the banner, there is a "Went well" section with a plus sign. The main content area contains five retrospective cards, each with a title, a user name, a comment input field, and a list of action items. The cards are:

- Retrospective 1** (User: ferry, 6 comments):
 - Pengerjaan 10 Hari (under time)
 - Kembangkan lagi hasil text mining terutama pada teknik TALA
 - Database Sambil lalu diperiksa sesuai dengan kebutuhan data nanti
 - Pelajari cara penggunaan Library convert json
 - Beberapa fitur dikembangkan lagi perulangan selanjutnya
 - Berhasil Menjalankan Semua unit fungsi yang ada pada perulangan 1
- Retrospective 2** (User: ferry, 6 comments):
 - Total Pengerjaan 14 Hari (on time)
 - Metode VSM yang sudah dibangun menampilkan dokumen yang relevan, tapi menghasilkan nilai similarity yang salah
 - Perbaiki lanjutan Database dari dospem
 - Kembangkan lagi site time frame diberanda dan halaman berita
 - kembangkan beberapa view dan fitur yang ada
 - Masih gagal menjalankan fitur VSM, harapkan dikembangkan lagi pada perulangan selanjutnya
- Retrospective 3** (User: ferry, 5 comments):
 - Lama pengerjaan 5 hari (over time) karena keterbatasan peneliti terkait bahasa javascript pada c#
 - Grafik perbarui sesuai saran dospem
 - Tambahkan fitur untuk filter dokumen berita hasil pencarian dari tanggal yang ada
 - Kembangkan beberapa view dan fitur yang ada
 - Masih Mengalami kegagalan di similarity dan VSM terkait integrasi data grafik dan durasi waktu
- Retrospective 4** (User: ferry, 4 comments):
 - Lama pengerjaan 1 hari (on time)
 - Berhasil menjalankan fungsionalitas pada perulangan ini yaitu Mengambil riwayat berita idDoc dan judul berita yang pernah dibaca member
 - Kembangkan beberapa view dan fitur yang ada
 - Pengembangan VMS dan Similarity selesai
- Retrospective 5** (User: ferry, 3 comments):
 - Verifikasi akun menggunakan email berhasil setelah konsul ke dospem
 - Lama pengerjaan 3 hari (on time)
 - Finishing Semua fitur dan view

Gambar H.1 Catatan Hasil *Retrospective*