

**PENGEMBANGAN SISTEM UNTUK MEMONITOR AKTIVITAS
MEDIA SOSIAL BERBASIS WEB (STUDI KASUS MEDIA SOSIAL
TWITTER CALON LEGISLATIF PADA PEMILIHAN LEGISLATIF
2019)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Vasha Farisi Sarwan Halim
NIM: 155150200111053



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019**

PENGESAHAN

PENGEMBANGAN SISTEM UNTUK MEMONITOR AKTIVITAS MEDIA SOSIAL BERBASIS WEB (STUDI KASUS MEDIA SOSIAL TWITTER CALON LEGISLATIF PADA PEMILIHAN LEGISLATIF 2019)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer

Disusun Oleh :
Vasha Farisi Sarwan Halim
NIM: 155150200111053

Skripsi ini telah diuji dan dinyatakan lulus pada
17 Juni 2019

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Fajar Pradana, S.ST., M.Eng
NIP: 19871121 201504 1 004

Dosen Pembimbing 2

Dr. Eng. Fitra Abdurrachman Bachtiar,
S.T, M.Eng
NIK: 201201 840628 1 001

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 17 Juni 2019



Vasha Farisi Sarwan Halim
NIM: 155150200111053

PRAKATA

Puji serta syukur penulis panjatkan atas kehadiran Allah SWT karena anugerah dan rahmat-Nya penulis dapat menyelesaikan skripsi dengan judul “Pengembangan Sistem untuk Memonitor Aktivitas Media Sosial Berbasis Web (Studi Kasus Media Sosial Twitter Calon Legislatif pada Pemilihan Legislatif 2019)”. Skripsi ini disusun sebagai syarat dalam memperoleh gelar sarjana pada Program Studi Teknik Informatika, Jurusan Teknik Informatika. Fakultas Ilmu Komputer. Universitas Brawijaya.

Dalam menyelesaikan skripsi ini penulis ingin mengucapkan banyak terima kasih kepada pihak yang telah memberikan kontribusi baik berupa moral maupun material. Rasa terima kasih tersebut penulis sampaikan kepada:

1. Bapak Halimurrahman dan Ibu Umi Maslikah selaku kedua orang tua penulis serta Salsabila Azka Nadya Halim selaku kakak penulis yang selalu memberikan motivasi, semangat dan doa.
2. Fajar Pradana, S.ST, M.Eng selaku dosen pembimbing 1 yang telah membimbing penulis selama penyusunan skripsi.
3. Dr.Eng Fitra Abdurrachman Bachtiar, S.T, M.Eng selaku dosen pembimbing 2 yang telah membimbing penulis selama penyusunan skripsi.
4. Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku ketua jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
5. Agus Wahyu Widodo, S.T, M.Cs., selaku ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
6. Seluruh warga Himpunan Mahasiswa Teknik Informatika dan Eksekutif Mahasiswa Informatika yang telah menyediakan wadah agar penulis dapat belajar dan berkembang menjadi pribadi yang lebih baik.
7. Seluruh keluarga intelek Departemen Pengembangan Ilmu dan Profesi Eksekutif Mahasiswa Informatika 2018/2019 yang selalu memberikan motivasi, kritik serta saran.

Dengan segala kerendahan hati, penulis menyadari bahwa dalam penulisan serta penyusunan skripsi ini masih jauh dari kata sempurna. Akhir kata penulis berharap semoga skripsi ini dapat bermanfaat dan dapat membantu pembaca.

Malang, 17 Juni 2019

Penulis
vashafarisi@student.ub.ac.id

ABSTRAK

Vasha Farisi Sarwan Halim, Pengembangan Sistem untuk Memonitor Aktivitas Media Sosial Berbasis Web (Studi Kasus Media Sosial Twitter Calon Legislatif pada Pemilihan Legislatif 2019)

Pembimbing: Fajar Pradana, S.ST., M.Eng dan Dr.Eng. Fitra Abdurrachman Bachtiar, S.T, M.Eng

Media sosial dinilai menjadi media yang efektif dalam melakukan kampanye karena terbukti penggunaannya dapat meningkatkan elektabilitas seorang kandidat politik. Namun karena terdapat keterbatasan waktu untuk terus memantau kondisi media sosial, sering kali media sosial seorang kandidat politik menjadi tidak termonitor dan dikelola dengan baik. Untuk mengatasi permasalahan tersebut, dibangun sistem yang dapat memonitor aktivitas media sosial dengan menerapkan model pengembangan *waterfall* dan pendekatan *object oriented*. Sistem ini mampu menampilkan aktivitas media sosial, analisis sentimen masyarakat, rekomendasi aktivitas dalam pengelolaan akun media sosial, dan tren topik Indonesia. Sistem diimplementasikan menggunakan konsep *Model View Controller* (MVC) dengan bantuan *framework* Laravel dan bahasa pemrograman PHP, HTML, dan Javascript. Setelah tahap implementasi selesai, pengujian *white box* dan *black box* dilakukan dan menghasilkan kesimpulan bahwa sistem telah 100% valid sesuai kebutuhan yang telah didefinisikan pada awal fase pengembangan. Pengujian *usability* menggunakan metode *heuristic evaluation* juga dilakukan agar memastikan kemudahan sistem untuk digunakan oleh pengguna. Hasil dari pengujian *usability* ditemukan kejanggalan sistem pada aspek *visibility of system status* dan *user control and freedom* dengan tingkat *severity ratings* bernilai 1 yang berarti tidak perlu ada perbaikan pada sistem dan sistem telah baik digunakan oleh pengguna.

Kata kunci: *waterfall, laravel, monitor, media sosial*

ABSTRACT

Vasha Farisi Sarwan Halim, Pengembangan Sistem Untuk Memonitor Aktivitas Media Sosial Berbasis Web (Studi Kasus Media Sosial Twitter Calon Legislatif pada Pemilihan Legislatif 2019)

Supervisors: Fajar Pradana, S.ST., M.Eng and Dr.Eng. Fitra Abdurrachman Bachtiar, S.T, M.Eng

Social media has been assessed to be an effective media for campaign and the use of social media is evidently could develop electability of political candidate. However, because there is limitation of time to keep monitor the condition of social media, usually political candidate's social media end up unmonitored and also not managed very well. To solve the problem, this study will build a system to monitor social media activity by applying waterfall development model with object oriented approach. This system will provide features such as displaying social media activity, people's sentiment analysis, recommendation activity to manage social media, and Indonesian trending topic. System will be developed with Model View Controller (MVC) concept with Laravel framework and PHP, HTML, Javascript as the programming languages. When implementation phase is finished, white box and black box testing is done with result the system has been 100% valid. Usability testing by heuristic evaluation method also done to make sure that system is already made easy for users. The result of usability testing is system has visibility of system status and user control and freedom problem with severity ratings valued 1 which means there is no need to change the system and the system is already made easy for users.

Keywords: *waterfall, laravel, monitor, social media*

DAFTAR ISI

PENGESAHAN	Error! Bookmark not defined.
PERNYATAAN ORISINALITAS	iii
PRAKATA.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xvi
DAFTAR LAMPIRAN	xviii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	4
1.6 Sistematika Pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Kajian Pustaka	6
2.2 Rekayasa Perangkat Lunak	7
2.3 Pengembangan Perangkat Lunak	8
2.3.1 Model <i>Waterfall</i>	8
2.3.2 Pendekatan Berorientasi Objek	9
2.4 Kampanye	11
2.5 Media Sosial.....	11
2.6 Konsep <i>Model View Controller</i>	12
2.7 Laravel.....	12
2.8 Twitter API	13
2.8.1 Tweepy	13
2.9 <i>Web Scraping</i>	14
2.10 <i>Lexicon Based Sentiment Analysis</i>	14

2.11 <i>Weighted Product</i>	14
2.12 Pengujian Perangkat Lunak	15
2.12.1 Definisi Pengujian Perangkat Lunak.....	15
2.12.2 Pengujian <i>White Box</i>	16
2.12.3 Pengujian <i>Black-Box</i>	16
2.12.4 Pengujian <i>Usability</i>	16
BAB 3 METODOLOGI	19
3.1 Studi Literatur	19
3.2 Pengumpulan Data	20
3.3 Analisis Kebutuhan	20
3.4 Perancangan dan Implementasi	20
3.5 Pengujian	21
3.6 Penarikan Kesimpulan dan Saran	21
BAB 4 ANALISIS KEBUTUHAN	22
4.1 Deskripsi Umum Sistem	22
4.2 Elisitasi Kebutuhan.....	22
4.3 Identifikasi Aktor.....	23
4.4 Kebutuhan Fungsional	24
4.5 Kebutuhan Non-Fungsional	34
4.6 <i>Use Case Diagram</i>	35
4.7 <i>Use Case Scenario</i>	36
4.7.1 <i>Use Case Scenario Login</i>	36
4.7.2 <i>Use Case Scenario</i> Daftar Akun	36
4.7.3 <i>Use Case Scenario</i> Menetapkan Akun Twitter	37
4.7.4 <i>Use Case Scenario</i> Lihat Profil Akun Twitter	38
4.7.5 <i>Use Case Scenario</i> Lihat Penilaian Aktivitas Selama Seminggu Terakhir	39
4.7.6 <i>Use Case Scenario</i> Lihat Rekomendasi Dalam Pengelolaan Akun Twitter.....	39
4.7.7 <i>Use Case Scenario</i> Lihat Rata-Rata Peningkatan <i>Followers</i> Pengguna dan Kompetitor	40
4.7.8 <i>Use Case Scenario</i> Lihat Rata-Rata <i>Posting Tweet</i> Pengguna dan Kompetitor	41

4.7.9 Use Case Scenario Lihat Rata-Rata <i>Retweet</i> Pengguna dan Kompetitor	42
4.7.10 Use Case Scenario Lihat Rata-Rata <i>Likes</i> Pengguna dan Kompetitor	43
4.7.11 Use Case Scenario Lihat Rata-Rata <i>Replies</i> Pengguna dan Kompetitor	43
4.7.12 Use Case Scenario Lihat Hasil Sentimen Analisis	44
4.7.13 Use Case Scenario Lihat <i>Tweet</i> Terbaik.....	45
4.7.14 Use Case Scenario Mengelola Daftar Kompetitor.....	45
4.7.15 Use Case Scenario Tambah Kompetitor.....	46
4.7.16 Use Case Scenario Hapus Kompetitor	47
4.7.17 Use Case Scenario Lihat Detail Perbandingan Aktivitas Pengguna dan Kompetitor	47
4.7.18 Use Case Scenario <i>Posting Tweet</i>	48
4.7.19 Use Case Scenario Lihat Tren Topik Indonesia.....	49
4.7.20 Use Case Scenario Lihat Contoh <i>Tweet</i> Berdasarkan Tren	49
4.7.21 Use Case Scenario Cari Akun Twitter	50
4.7.22 Use Case Scenario <i>Update Data</i>	50
4.7.23 Use Case Scenario <i>Logout</i>	51
BAB 5 PERANCANGAN DAN IMPLEMENTASI	52
5.1 Perancangan Sistem.....	52
5.1.1 Perancangan Arsitektur Sistem.....	52
5.1.2 Perancangan Data	56
5.1.3 Perancangan Komponen	57
5.1.4 Perancangan Antarmuka.....	59
5.2 Implementasi Sistem	64
5.2.1 Spesifikasi Sistem	64
5.2.2 Implementasi Kode Program	65
5.2.3 Implementasi Basis Data	68
5.2.4 Implementasi Antarmuka	71
BAB 6 Pengujian	76
6.1 Pengujian Unit.....	76



6.1.1 Pengujian Unit Algoritme Lihat Rekomendasi Dalam Pengelolaan Akun Media Sosial	76
6.1.2 Pengujian Unit Algoritme Lihat Hasil Sentimen Analisis.....	81
6.1.3 Pengujian Unit Algoritme Lihat Tren Topik Indonesia	83
6.2 Pengujian Integrasi	84
6.3 Pengujian Validasi	87
6.3.1 <i>Login</i>	87
6.3.2 Daftar Akun	88
6.3.3 Menetapkan Akun Twitter	91
6.3.4 Lihat Profil Akun Twitter	92
6.3.5 Lihat Penilaian Aktivitas Selama Seminggu Terakhir	92
6.3.6 Lihat Rekomendasi Dalam Pengelolaan Akun Twitter	93
6.3.7 Lihat Rata-Rata Peningkatan <i>Followers</i> Pengguna dan Kompetitor	96
6.3.8 Lihat Rata-Rata <i>Posting Tweet</i> Pengguna dan Kompetitor.....	97
6.3.9 Lihat Rata-Rata <i>Retweet</i> Pengguna dan Kompetitor	99
6.3.10 Lihat Rata-Rata <i>Likes</i> Pengguna dan Kompetitor.....	100
6.3.11 Lihat Rata-Rata <i>Replies</i> Pengguna dan Kompetitor	101
6.3.12 Lihat Hasil Sentimen Analisis.....	102
6.3.13 Lihat <i>Tweet</i> Terbaik.....	103
6.3.14 Mengelola Daftar Kompetitor	104
6.3.15 Tambah Kompetitor	105
6.3.16 Hapus Kompetitor	106
6.3.17 Lihat Detail Perbandingan Aktivitas Pengguna dan Kompetitor	107
6.3.18 <i>Posting Tweet</i>	108
6.3.19 Lihat Tren Topik Indonesia.....	108
6.3.20 Lihat Contoh <i>Tweet</i> Berdasarkan Tren.....	109
6.3.21 Cari Akun Twitter	110
6.3.22 <i>Update Data</i>	111
6.3.23 <i>Logout</i>	111
6.4 Pengujian <i>Usability</i>	112
6.4.1 Analisis Hasil Pengujian <i>Usability</i>	112

BAB 7 PENUTUP	113
7.1 Kesimpulan.....	113
7.2 Saran	113
DAFTAR REFERENSI	114
LAMPIRAN A Keterangan Melakukan Penelitian	116
LAMPIRAN B Hasil <i>Brainstroming</i>	117
LAMPIRAN C Manualisasi Metode <i>Lexicon Based Sentiment Analysis</i>	118
LAMPIRAN D Manualisasi Metode <i>Weighted Product</i>	121



DAFTAR TABEL

Tabel 2.1 Aspek <i>Heuristic Evaluation</i>	17
Tabel 2.2 Keterangan <i>Severity Ratings</i>	18
Tabel 4.1 Peserta <i>Brainstorming</i>	23
Tabel 4.2 Identifikasi aktor.....	23
Tabel 4.3 Kebutuhan fungsional	24
Tabel 4.4 Kebutuhan Non-Fungsional.....	34
Tabel 4.5 <i>Use Case Scenario Login</i>	36
Tabel 4.6 <i>Use Case Scenario</i> Daftar Akun	37
Tabel 4.7 <i>Use Case Scenario</i> Menetapkan Akun Twitter	37
Tabel 4.8 <i>Use Case Scenario</i> Lihat Profil Akun Twitter	38
Tabel 4.9 <i>Use Case Scenario</i> Lihat Penilaian Aktivitas Selama Seminggu Terakhir	39
Tabel 4.10 <i>Use Case Scenario</i> Lihat Rekomendasi Dalam Pengelolaan Akun Twitter	40
Tabel 4.11 <i>Use Case Scenario</i> Lihat Rata-Rata Peningkatan <i>Followers</i> Pengguna dan Kompetitor	41
Tabel 4.12 <i>Use Case Scenario</i> Lihat Rata-Rata <i>Posting Tweet</i> Pengguna dan Kompetitor	41
Tabel 4.13 <i>Use Case Scenario</i> Lihat Rata-Rata <i>Retweet</i> Pengguna dan Kompetitor	42
Tabel 4.14 <i>Use Case Scenario</i> Lihat Rata-Rata <i>Likes</i> Pengguna dan Kompetitor..	43
Tabel 4.15 <i>Use Case Scenario</i> Lihat Rata-Rata <i>Replies</i> Pengguna dan Kompetitor	44
Tabel 4.16 <i>Use Case Scenario</i> Lihat Hasil Sentimen Analisis.....	44
Tabel 4.17 <i>Use Case Scenario</i> Lihat <i>Tweet</i> Terbaik.....	45
Tabel 4.18 <i>Use Case Scenario</i> Mengelola Daftar Kompetitor	45
Tabel 4.19 <i>Use Case Scenario</i> Tambah Kompetitor	46
Tabel 4.20 <i>Use Case Scenario</i> Hapus Kompetitor	47
Tabel 4.21 <i>Use Case Scenario</i> Lihat Detail Perbandingan Aktivitas Pengguna dan Kompetitor	47
Tabel 4.22 <i>Use Case Scenario</i> <i>Posting Tweet</i>	48
Tabel 4.23 <i>Use Case Scenario</i> Lihat Tren Topik Indonesia	49

Tabel 4.24 <i>Use Case Scenario</i> Lihat Contoh <i>Tweet</i> Berdasarkan Tren.....	49
Tabel 4.25 <i>Use Case Scenario</i> Cari Akun Twitter	50
Tabel 4.26 <i>Use Case Scenario</i> Update Data	50
Tabel 4.27 <i>Use Case Scenario</i> Logout	51
Tabel 5.1 Perancangan Algoritme Lihat Rekomendasi Dalam Pengelolaan Akun Media Sosial	58
Tabel 5.2 Perancangan Algoritme Lihat Hasil Sentimen Analisis.....	59
Tabel 5.3 Perancangan Algoritme Lihat Tren Topik Indonesia	59
Tabel 5.4 Keterangan Perancangan Antarmuka Halaman <i>Dashboard</i>	60
Tabel 5.5 Keterangan Perancangan Antarmuka Halaman <i>Competitor</i>	62
Tabel 5.6 Keterangan Perancangan Antarmuka Halaman <i>Trend</i>	63
Tabel 5.7 Spesifikasi Perangkat Keras	65
Tabel 5.8 Spesifikasi Perangkat Lunak	65
Tabel 5.9 Kode Program Lihat Rekomendasi Dalam Pengelolaan Akun Media Sosial	65
Tabel 5.10 Kode Program Lihat Hasil Sentimen Analisis.....	67
Tabel 5.11 Kode Program Lihat Tren Topik Indonesia	68
Tabel 5.12 Implementasi Tabel <i>Competitor</i>	69
Tabel 5.13 Implementasi Tabel <i>Twitter_Accounts</i>	69
Tabel 5.14 Implementasi Tabel <i>Twitter_Accounts_Log</i>	69
Tabel 5.15 Implementasi Tabel <i>Twitter_Replies</i>	70
Tabel 5.16 Implementasi Tabel <i>Twitter_Trend_Detail</i>	70
Tabel 5.17 Implementasi Tabel <i>Twitter_Trends</i>	70
Tabel 5.18 Implementasi Tabel <i>Twitter_Tweets</i>	71
Tabel 5.19 Implementasi Tabel <i>Users</i>	71
Tabel 6.1 Perancangan Algoritme showRecommendation	76
Tabel 6.2 Hasil Pengujian Unit Lihat Rekomendasi Dalam Pengelolaan Akun Media Sosial	78
Tabel 6.3 Perancangan Algoritme Lihat Hasil Sentimen Analisis.....	82
Tabel 6.4 Hasil Pengujian Unit Lihat Hasil Sentimen Analisis	82
Tabel 6.5 Perancangan Algoritme Lihat Tren Topik Indonesia	83
Tabel 6.6 Hasil Pengujian Unit Lihat Tren Topik Indonesia.....	84
Tabel 6.7 Perancangan Algoritme Lihat Hasil Sentimen Analisis.....	85

Tabel 6.8 Hasil Pengujian Integrasi	86
Tabel 6.9 Pengujian Validasi <i>Login</i>	87
Tabel 6.10 Pengujian Validasi Login – Alternatif Pertama	88
Tabel 6.11 Pengujian Validasi <i>Login</i> – Alternatif Kedua	88
Tabel 6.12 Pengujian Validasi Daftar Akun	89
Tabel 6.13 Pengujian Validasi Daftar Akun – Alternatif Pertama	89
Tabel 6.14 Pengujian Validasi Daftar Akun – Alternatif Kedua.....	90
Tabel 6.15 Pengujian Validasi Daftar Akun – Alternatif Ketiga.....	90
Tabel 6.16 Pengujian Validasi Menetapkan Akun Twitter	91
Tabel 6.17 Pengujian Validasi Menetapkan Akun Twitter – Alternatif Pertama ..	91
Tabel 6.18 Pengujian Validasi Lihat Profil Akun Twitter	92
Tabel 6.19 Pengujian Validasi Lihat Penilaian Aktivitas Selama Seminggu Terakhir	93
Tabel 6.20 Pengujian Validasi Lihat Penilaian Aktivitas Selama Seminggu Terakhir – Alternatif Pertama.....	93
Tabel 6.21 Pengujian Validasi Lihat Rekomendasi Dalam Pengelolaan Akun Twitter	94
Tabel 6.22 Pengujian Validasi Lihat Rekomendasi Dalam Pengelolaan Akun Twitter – Alternatif Pertama.....	94
Tabel 6.23 Pengujian Validasi Lihat Rekomendasi Dalam Pengelolaan Akun Twitter – Alternatif Kedua	95
Tabel 6.24 Pengujian Validasi Lihat Rekomendasi Dalam Pengelolaan Akun Twitter – Alternatif Ketiga	96
Tabel 6.25 Pengujian Validasi Lihat Rata-Rata Peningkatan Followers Pengguna dan Kompetitor	96
Tabel 6.26 Pengujian Validasi Lihat Rata-Rata Peningkatan <i>Followers</i> Pengguna dan Kompetitor – Alternatif Pertama	97
Tabel 6.27 Pengujian Validasi Lihat Rata-Rata <i>Posting Tweet</i> Pengguna dan Kompetitor	98
Tabel 6.28 Pengujian Validasi Lihat Rata-Rata <i>Posting Tweet</i> Pengguna dan Kompetitor – Alternatif Pertama	98
Tabel 6.29 Pengujian Validasi Lihat Rata-Rata <i>Retweet</i> Pengguna dan Kompetitor	99
Tabel 6.30 Pengujian Validasi Lihat Rata-Rata <i>Retweet</i> Pengguna dan Kompetitor – Alternatif Pertama.....	99

Tabel 6.31 Pengujian Validasi Lihat <i>Likes</i> Pengguna dan Kompetitor	100
Tabel 6.32 Pengujian Validasi Lihat Rata-Rata <i>Likes</i> Pengguna dn Kompetitor – Alternatif Pertama.....	101
Tabel 6.33 Pengujian Validasi Lihat <i>Replies</i> Pengguna dan Kompetitor	101
Tabel 6.34 Pengujian Validasi Lihat Rata-Rata <i>Replies</i> Pengguna dan Kompetitor – Alternatif Pertama.....	102
Tabel 6.35 Pengujian Validasi Lihat Hasil Sentimen Analisis	103
Tabel 6.36 Pengujian Validasi Lihat Hasil Sentimen Analisis – Alternatif Pertama	103
Tabel 6.37 Pengujian Validasi Lihat Tweet Terbaik.....	104
Tabel 6.38 Pengujian Validasi Lihat Tweet Terbaik – Alternatif Pertama.....	104
Tabel 6.39 Pengujian Validasi Mengelola Daftar Kompetitor.....	105
Tabel 6.40 Pengujian Validasi Mengelola Daftar Kompetitor – Alternatif Pertama	105
Tabel 6.41 Pengujian Validasi Tambah Kompetitor	106
Tabel 6.42 Pengujian Validasi Tambah Kompetitor – Alternatif Pertama.....	106
Tabel 6.43 Pengujian Validasi Hapus Kompetitor	107
Tabel 6.44 Pengujian validasi lihat detail perbandingan aktivitas pengguna dan kompetitor	107
Tabel 6.45 Pengujian Validasi <i>Posting Tweet</i>	108
Tabel 6.46 Pengujian Validasi Lihat Tren Topik Indonesia.....	109
Tabel 6.47 Pengujian Validasi Lihat Contoh <i>Tweet</i> Berdasarkan Tren	109
Tabel 6.48 Pengujian Validasi Cari Akun Twitter	110
Tabel 6.49 Pengujian Validasi Cari Akun Twitter – Alternatif Pertama	110
Tabel 6.50 Pengujian Validasi <i>Update Data</i>	111
Tabel 6.51 Pengujian Validasi Logout	111
Tabel 6.52 Hasil Pengujian <i>Heuristic Evaluation</i>	112



DAFTAR GAMBAR

Gambar 2.1 Model <i>Waterfall</i>	8
Gambar 2.2 Contoh <i>Class Diagram</i>	10
Gambar 2.3 Contoh <i>Use Case Diagram</i>	10
Gambar 2.4 Contoh <i>Sequence Diagram</i>	11
Gambar 2.5 Struktur MVC.....	12
Gambar 2.6 Struktur Laravel	13
Gambar 2.7 Contoh Penggunaan Tweepy	13
Gambar 3.1 Diagram Alir Metodologi Penelitian.....	19
Gambar 4.1 Gambaran Umum Sistem	22
Gambar 4.2 <i>Use Case Diagram</i>	35
Gambar 5.1 <i>Sequence Diagram</i> Lihat Rekomendasi Dalam Pengelolaan Akun Twitter.....	53
Gambar 5.2 <i>Sequence Diagram</i> Lihat Hasil Sentimen Analisis	54
Gambar 5.3 <i>Sequence Diagram</i> Lihat Tren Topik Indonesia.....	54
Gambar 5.4 <i>Class Diagram</i> Sistem.....	55
Gambar 5.5 <i>Entity Relationship Diagram</i>	56
Gambar 5.6 <i>Physical Data Model</i>	57
Gambar 5.7 Perancangan Antarmuka Halaman <i>Dashboard</i>	60
Gambar 5.8 Perancangan Antarmuka Halaman <i>Competitor</i>	62
Gambar 5.9 Perancangan Antarmuka Halaman <i>Trend</i>	63
Gambar 5.10 Implementasi Antarmuka Halaman <i>Dashboard</i>	72
Gambar 5.11 Implementasi Antarmuka Halaman <i>Dashboard</i> (Lanjutan)	72
Gambar 5.12 Implementasi Antarmuka Halaman <i>Dashboard</i> (Lanjutan)	73
Gambar 5.13 Implementasi Antarmuka Halaman <i>Dashboard</i> (Lanjutan)	73
Gambar 5.14 Implementasi Antarmuka Halaman <i>Dashboard</i> (Lanjutan)	74
Gambar 5.15 Implementasi Antarmuka Halaman <i>Competitor</i>	74
Gambar 5.16 Implementasi Antarmuka Halaman <i>Trend</i>	75
Gambar 6.1 <i>Flow Graph</i> Fungsi <i>showRecommendation()</i>	77
Gambar 6.2 <i>Flow Graph</i> Fungsi <i>showSentiment()</i>	82
Gambar 6.3 <i>Flow Graph</i> Fungsi <i>Index()</i>	84



Gambar 6.4 *Flow Graph* Fungsi *showSentiment()* 85



DAFTAR LAMPIRAN

LAMPIRAN A Keterangan Melakukan Penelitian	116
LAMPIRAN B Hasil <i>Brainstroming</i>	117
LAMPIRAN C Manualisasi Metode <i>Lexicon Based Sentiment Analysis</i>	118
LAMPIRAN D Manualisasi Metode <i>Weighted Product</i>	121



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Tahun 2019 merupakan tahun demokrasi bagi Indonesia. Agenda demokrasi yang diselenggarakan lima tahun sekali yaitu pemilihan umum (Pemilu) kembali dilaksanakan. Pemilu merupakan serangkaian proses kompetitif untuk memilih wakil rakyat melalui suara rakyat. Wakil rakyat yang dimaksud meliputi eksekutif dan legislatif di berbagai tingkat pemerintahan. Wakil rakyat yang terpilih berhak atas jabatan politik dan memiliki peran terhadap pengambilan kebijakan sesuai dengan masa jabatannya. Oleh karena itu setiap kandidat politik yang ikut serta dalam Pemilu saling berkompetisi untuk mengambil suara rakyat. Kontestasi politik dinilai menjadi salah satu tantangan dalam proses penyelenggaraan Pemilu (Fitri, 2019). Dalam menghadapi kontestasi politik tersebut, kampanye menjadi kunci untuk mengumpulkan suara rakyat dalam upaya kemenangan Pemilu.

Pada dasarnya kampanye merupakan bentuk penyampaian pesan dari pengirim kepada khalayak (Ardha, 2014). Sedangkan kampanye politik dipahami sebagai upaya komunikasi yang dikerjakan secara intensif pada kurun waktu yang telah ditentukan untuk memengaruhi pengambilan keputusan suatu kelompok dengan tujuan memenangkan kandidat tertentu (Fatimah, 2018). Komunikasi yang disampaikan kandidat politik menjadi hal yang perlu diperhatikan saat melakukan kampanye. Pada umumnya komunikasi yang dilakukan oleh kandidat politik adalah menyampaikan visi, misi, dan program kerja yang ditawarkan. Namun disamping itu seringkali kandidat politik melakukan komunikasi dengan cara menanggapi isu atau topik yang sedang berkembang dalam masyarakat. Hal ini dilakukan untuk menarik perhatian masyarakat terhadap kandidat tersebut.

Media sosial dinilai menjadi salah satu media yang efektif dalam melakukan kampanye (Vonderschmitt, 2012). Kampanye pada tahun 2008 yang dilakukan oleh mantan presiden Amerika Serikat Barack Obama menjadi bukti peran pentingnya media sosial dalam proses kemenangan Pemilu. Di Indonesia sendiri *traffic* penggunaan media sosial semakin meningkat. Menjelang Pemilu partai politik mulai gencar membuat akun media sosial untuk melakukan kampanye terhadap partai dan kandidat mereka (Bright, 2017). Hal yang perlu digaris bawahi mengapa kampanye melalui media sosial menjadi penting adalah karena media sosial dapat digunakan sebagai sarana yang ideal untuk mengetahui opini publik terhadap kandidat politik (Ardha, 2014). Selain itu menurut penelitian sebelumnya mengenai pengaruh media sosial kandidat politik terhadap persepsi pemilih menunjukkan adanya hubungan antara bagaimana media sosial yang dikelola dengan pendekatan informal dan lebih personal dapat meningkatkan elektabilitas dari kandidat politik tersebut (Hellweg, 2011).

Adapun permasalahan dalam melakukan kampanye atau komunikasi melalui media sosial yaitu terdapat keterbatasan waktu untuk terus memantau kondisi media sosial itu sendiri. Padahal kondisi media sosial terus berubah mengikuti tren topik yang sedang ramai diperbincangkan masyarakat. Tren topik dapat

dipandang sebagai cerminan dari keresahan masyarakat, dengan memahami tren topik maka dapat membantu dalam memahami kondisi dari suatu kelompok masyarakat (Dang-xuan, 2014). Adanya keterbatasan waktu untuk terus memantau kondisi media sosial dan tren topik yang terus berubah berdampak pada informasi dan komunikasi yang disampaikan kandidat politik menjadi tidak relevan (Vonderschmitt, 2012). Selain itu kondisi media sosial yang tidak termonitor dengan baik mengakibatkan sulitnya mengukur keberhasilan dalam penyampaian pesan melalui media sosial. Maka dari itu dibutuhkan pengembangan sistem untuk membantu memonitor kondisi media sosial.

Terdapat penelitian sebelumnya terkait pengembangan sistem untuk memonitor kondisi media sosial oleh Rosmala dan Syafei (2012). Penelitian tersebut membangun sistem yang dapat menggali data Twitter menggunakan metode *web crawler* untuk mengetahui opini masyarakat terhadap sebuah produk dengan model pengembangan *prototype*. Selain digunakan dalam memonitor kondisi media sosial, data Twitter yang telah diambil diharapkan dapat menjadi acuan untuk menciptakan strategi pemasaran produk kedepannya. Terdapat kekurangan pada penelitian tersebut yaitu belum diterapkan algoritme pemrosesan data untuk membantu pengguna dalam memahami konteks dari data yang diperoleh. Selain itu terdapat juga penelitian sebelumnya oleh Alshamrani dan Abdullah (2015) mengenai perbandingan pengembangan sistem model *waterfall*, *spiral*, dan *incremental/iterative*. Hasil dari penelitian tersebut menyebutkan bahwa pengembangan model *waterfall* baik digunakan pada pengembangan sistem yang kebutuhan dan spesifikasinya telah jelas didefinisikan pada awal fase pengembangan, tidak membutuhkan biaya besar, dan cenderung mudah untuk diimplementasikan.

Penelitian ini akan membangun sistem untuk memonitor aktivitas media sosial berbasis web yang dapat membantu pengguna yaitu seorang kandidat politik dalam mengetahui kondisi media sosial dengan lebih mudah. Model pengembangan yang digunakan adalah *waterfall* karena seluruh kebutuhan sistem telah didefinisikan dengan jelas pada awal fase pengembangan. Selanjutnya pemodelan dan perancangan dilakukan dengan menggunakan pendekatan *object oriented* (OO) sehingga menghasilkan beberapa diagram antara lain *use case*, *sequence*, dan *class diagram*. Sistem diimplementasikan menggunakan konsep *model view controller* (MVC) dengan bantuan *framework* Laravel dan bahasa pemrograman HTML, PHP, dan Javascript. Adapun algoritme yang diterapkan pada penelitian ini yaitu *lexicon based sentiment analysis* untuk mengetahui polaritas opini masyarakat terhadap pengguna dan *weighted product* untuk melakukan pemeringkatan terhadap *tweet* terbaik pengguna berdasarkan *tweet* yang paling banyak menarik perhatian masyarakat dan pemeringkatan terhadap contoh *tweet* yang membahas tren topik tertentu dengan proses manualisasi terlampir dan dapat dilihat pada Lampiran C dan Lampiran D. Pengujian akan dilakukan setelah sistem selesai diimplementasikan, tujuannya untuk memastikan sistem dapat bekerja dengan baik dan sesuai dengan kebutuhan yang telah didefinisikan pada awal fase pengembangan.

Berdasarkan uraian yang telah disampaikan dapat diketahui saat ini proses kampanye melalui media sosial masih memiliki kendala. Oleh karena itu penelitian ini yang berjudul “Pengembangan Sistem untuk Memonitor Aktivitas Media Sosial Berbasis Web (Studi Kasus Media Sosial Twitter Calon Legislatif pada Pemilihan Legislatif 2019)” akan mengembangkan sebuah sistem yang dapat memonitor aktivitas media sosial pengguna dan kompetitornya, menampilkan analisis sentimen masyarakat terhadap pengguna, menampilkan *tweet* pengguna yang paling banyak mendapat perhatian masyarakat, dan dapat menampilkan tren topik Indonesia beserta contoh *tweet* yang membahas mengenai tren tersebut. Diharapkan dengan dikembangkannya sistem ini dapat membantu kandidat politik dalam meningkatkan efektivitas kampanye melalui media sosial.

1.2 Rumusan Masalah

Rumusan masalah pada penelitian ini antara lain sebagai berikut:

1. Bagaimanakah hasil analisis kebutuhan sistem untuk memonitor aktivitas media sosial?
2. Bagaimanakah hasil perancangan dan implementasi sistem untuk memonitor aktivitas media sosial yang sesuai dengan fase analisis kebutuhan?
3. Bagaimanakah hasil pengujian sistem untuk memonitor aktivitas media sosial?

1.3 Tujuan

Tujuan dari penelitian ini antara lain sebagai berikut:

1. Melakukan analisis kebutuhan pada sistem untuk memonitor aktivitas media sosial twitter.
2. Melakukan perancangan dan implementasi sistem untuk memonitor aktivitas media sosial yang sesuai dengan fase analisis kebutuhan.
3. Melakukan pengujian terhadap sistem untuk memonitor aktivitas media sosial.

1.4 Manfaat

Manfaat dari penelitian ini antara lain sebagai berikut:

1. Dapat memonitor aktivitas media sosial pengguna dan kompetitornya agar pengguna dapat mengetahui kondisi media sosial dengan lebih mudah.
2. Dapat memberikan rekomendasi aksi sesuai dengan hasil monitor aktivitas pengguna dan kompetitor agar komunikasi yang dilakukan pengguna lebih baik dari kompetitor.

3. Dapat menampilkan tren topik Indonesia selama 24 jam terakhir beserta contoh tweet yang membahas tren tersebut agar pengguna dapat mengetahui topik yang sedang ramai diperbincangkan masyarakat.

1.5 Batasan Masalah

Batasan masalah pada penelitian ini antara lain sebagai berikut:

1. Penelitian menggunakan media sosial Twitter.
2. Data Twitter diambil menggunakan Twitter API dan *scraping* melalui *web* Twitter.
3. Data Twitter yang diambil merupakan data media sosial calon legislatif dewan perwakilan rakyat republik Indonesia pada pemilihan legislatif tahun 2019.
4. Data tren yang digunakan pada penelitian ini berasal dari *web* penyedia daftar tren topik Indonesia selama 24 jam terakhir.

1.6 Sistematika Pembahasan

Sistematika pembahasan pada penelitian ini adalah sebagai berikut:

1. BAB I PENDAHULUAN

Pada bab ini berisi latar belakang penelitian, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan penelitian, dan sistematika pembahasan yang akan digunakan yang akan digunakan sebagai dasar dalam melakukan penelitian terkait pengembangan sistem untuk memonitor aktivitas media sosial.

2. BAB II LANDASAN KEPUSTAKAAN

Pada bab ini terdapat dasar teori yang relevan dengan pengembangan sistem untuk memonitor aktivitas media sosial seperti rekayasa perangkat lunak, pendekatan berorientasi objek, pengujian perangkat lunak, kampanye, media sosial, maupun penelitian-penelitian sebelumnya yang terkait dengan penelitian ini.

3. BAB III METODOLOGI

Pada bab ini membahas tentang alur metode yang digunakan dalam pengembangan sistem untuk memonitor aktivitas media sosial diantaranya meliputi studi literatur, analisis kebutuhan, perancangan dan implementasi, pengujian, lalu terakhir penarikan kesimpulan dan saran.

4. BAB IV ANALISIS KEBUTUHAN

Pada bab ini membahas tentang elisitasi kebutuhan, identifikasi aktor, penentuan kebutuhan fungsional dan non-fungsional serta pemodelan dari hasil analisis kebutuhan dalam pengembangan sistem untuk memonitor aktivitas media sosial.

5. BAB V PERANCANGAN DAN IMPLEMENTASI

Pada bab ini membahas tentang proses perancangan sistem untuk memonitor aktivitas media sosial lalu dilanjutkan tentang bagaimana sistem tersebut diimplementasikan.

6. BAB VI PENGUJIAN

Pada bab ini membahas tentang proses pengujian sistem untuk memonitor aktivitas media sosial yang meliputi pengujian unit, integrasi, validasi, dan pengujian *usability* sistem yang telah dibangun. Kemudian melakukan analisis berdasarkan hasil pengujian tersebut.

7. BAB VII PENUTUP

Pada bab ini berisi kesimpulan dari penelitian mengenai pengembangan sistem untuk memonitor aktivitas media sosial yang telah dilakukan serta memberikan saran untuk kepentingan penelitian selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Penelitian sebelumnya oleh Hellweg (2011) menjelaskan keterkaitan antara cara kandidat politik menggunakan media sosialnya dalam melakukan proses kampanye akan berdampak pada elektabilitas kandidat tersebut. Setelah menggunakan beberapa pendekatan metode analisis konten dan hasil survey ditemukan bahwa calon pemilih akan tertarik pada kandidat yang menyediakan konten bersifat informal dan dekat dengan kepribadian kandidat tersebut dibandingkan kandidat yang hanya berfokus pada konten politik profesional. Pada akhirnya penggunaan media sosial seorang kandidat politik secara signifikan mempengaruhi cara publik menilai kandidat tersebut.

Penelitian sebelumnya oleh Rosmala dan Syafei (2012) menjelaskan mengenai pengembangan sistem untuk memonitor kondisi media sosial. Penelitian tersebut membangun sistem yang dapat menggali data Twitter menggunakan metode *web crawler* untuk mengetahui opini masyarakat terhadap sebuah produk dengan model pengembangan *prototype*. Selain digunakan dalam memonitor kondisi media sosial, data Twitter yang telah diambil diharapkan dapat menjadi acuan untuk menciptakan strategi pemasaran produk kedepannya.

Penelitian sebelumnya oleh Alshamrani dan Abdullah (2015) menjelaskan perbandingan antara tiga model pengembangan yaitu model *waterfall*, *spiral*, dan *incremental/iterative*. Penelitian tersebut bertujuan untuk membandingkan kelebihan dan kekurangan dari setiap model pengembangan sehingga kedepannya dapat dijadikan sebagai acuan dalam memilih model pengembangan sesuai dengan kondisi dan kebutuhan. Hasil dari penelitian tersebut menyebutkan model pengembangan *waterfall* baik digunakan pada pengembangan sistem yang kebutuhan dan spesifikasinya telah jelas didefinisikan pada awal fase pengembangan, tidak membutuhkan biaya besar, dan cenderung mudah untuk diimplementasikan.

Penelitian sebelumnya oleh Banjarnahor dan Hartomo (2016) melakukan pengembangan sistem informasi berbasis web yang bertujuan untuk meningkatkan promosi produk unggulan usaha kecil menengah (UKM). Penelitian tersebut diimplementasikan menggunakan konsep *model view controller* (MVC) dengan memanfaatkan *framework* Laravel. Salah satu hasil penelitian tersebut menyatakan bahwa *framework* Laravel memiliki beberapa keunggulan antara lain mudah diimplementasikan, cepat, dan terdapat dokumentasi lengkap untuk memudahkan pengembang dalam menggunakan *framework* tersebut.

Penelitian sebelumnya oleh Kusumawati (2017) menjelaskan tentang penerapan metode analisis sentimen *lexicon based* untuk melihat opini pengguna Twitter terhadap kenaikan harga rokok. Metodologi yang digunakan pada penelitian tersebut meliputi tahap pengumpulan data, *preprocessing data*, *translation*, dan *classification*. Penelitian tersebut menggunakan pendekatan *first sense* dalam memilih kata yang memiliki arti ganda untuk disesuaikan dengan

kamus. Hasil penelitian tersebut menyebutkan bahwa analisis sentimen menggunakan metode *lexicon based* menghasilkan akurasi tertinggi pada tipe opini netral sebesar 81%.

Penelitian sebelumnya oleh Wiyanti (2014) menjelaskan tentang penerapan metode *Weighted Product* (WP) untuk menyelesaikan permasalahan dalam seleksi penerima beasiswa berprestasi. Pada penelitian tersebut metode WP digunakan untuk melakukan sistem pemeringkatan kelayakan siswa mendapatkan beasiswa berdasarkan atribut pendukung seperti nilai rata-rata belajar, jumlah penghasilan orang tua, dan banyaknya tanggungan orang tua. Hasil dari penelitian tersebut menyebutkan bahwa WP merupakan metode yang optimal dalam menyelesaikan sistem pemeringkatan dengan waktu komputasi yang relatif singkat.

Berdasarkan kajian pustaka yang telah dilakukan, terdapat beberapa perbedaan antara penelitian ini dengan penelitian terkait yang telah dilakukan sebelumnya. Pengembangan sistem untuk memonitor aktivitas media sosial pada penelitian ini dilakukan dengan model *waterfall* karena kebutuhan dan spesifikasinya telah didefinisikan dengan jelas pada awal fase pengembangan. Selanjutnya penelitian ini menerapkan algoritme *lexicon based sentiment analysis* karena tidak diperlukan tahapan *labeling* pada setiap proses penggalian data baru sedangkan penerapan algoritme *weighted product* diterapkan karena algoritme tersebut dinilai optimal dalam menyelesaikan sistem pemeringkatan dengan waktu komputasi yang relatif singkat. Penelitian ini diharapkan dapat membantu kandidat politik dalam melakukan kampanye melalui media sosial.

2.2 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah sebuah ilmu rekayasa yang berfokus pada seluruh tahapan dalam melakukan rekayasa perangkat lunak. Dimulai dari tahapan awal yang dilakukan yaitu spesifikasi sistem sampai pada tahapan akhirnya yaitu pemeliharaan sistem. Proses rekayasa merupakan tantangan untuk mendapatkan hasil dengan kualitas yang direncanakan sesuai dengan batasan waktu dan biaya. Secara umum pengembang perangkat lunak cenderung menggunakan pendekatan yang terstruktur dan terorganisir dalam bekerja, sehingga harapannya dapat menghasilkan perangkat lunak yang memiliki kualitas tinggi. Pendekatan terstruktur yang digunakan pada rekayasa perangkat lunak sering kali dinyatakan sebagai sebuah proses perangkat lunak. Proses perangkat lunak merupakan tahapan-tahapan kegiatan dalam memproduksi produk perangkat lunak (Sommerville, 2011). Terdapat empat aktivitas pada proses ini, antara lain:

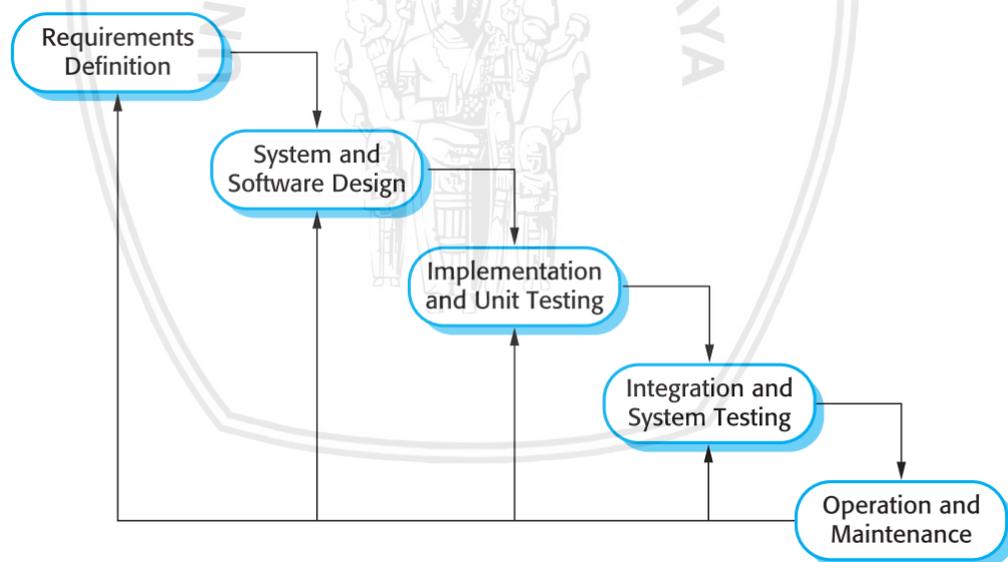
1. *Software Specification*, kebutuhan perangkat lunak didefinisikan secara bersama oleh klien dan pengembang sehingga akan menghasilkan batasan-batasan yang disetujui dalam proses pengembangannya.
2. *Software Development*, perangkat lunak dirancang dan diprogram.

3. *Software Validation*, perangkat lunak diuji untuk memastikan bahwa perangkat lunak telah memenuhi kebutuhan klien.
4. *Software Evolution*, perangkat lunak disesuaikan dengan keinginan klien dan kebutuhan pasar.

2.3 Pengembangan Perangkat Lunak

2.3.1 Model *Waterfall*

Model *waterfall* memiliki tahapan spesifikasi kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan. Pada konsepnya setiap tahapan yang dilakukan harus selesai terlebih dahulu sebelum dapat dilanjutkan ke tahapan berikutnya, namun pada praktiknya terdapat banyak iterasi yang terjadi antar tahapan (Sommerville, 2011). Alasan penggunaan model *waterfall* pada penelitian ini yaitu seluruh kebutuhan yang akan disediakan oleh sistem telah didefinisikan pada awal tahapan pengembangan sehingga penelitian dapat dilakukan secara bertahap dimulai dari analisis kebutuhan, perancangan, implementasi, dan pengujian. Tahap pemeliharaan tidak dilakukan karena pada penelitian ini sistem yang dikembangkan belum digunakan oleh pengguna akhir secara berkala. Model *waterfall* diilustrasikan pada Gambar 2.1.



Gambar 2.1 Model *Waterfall*

(Sumber: Sommerville, 2011)

1. *Requirements Definition*

Pada tahap ini layanan yang akan diberikan oleh sistem, batasan sistem, dan tujuan sistem didefinisikan dengan mengkonsultasikannya dengan pengguna sistem. Pendefinisian dilakukan dengan detail lalu digunakan sebagai spesifikasi sistem.

2. *System and Software Design*

Tahap ini perancangan sistem dilakukan dengan mengalokasikan kebutuhan perangkat keras dan perangkat lunak dengan membangun keseluruhan arsitektur sistem. Perancangan perangkat lunak melibatkan identifikasi dan menjelaskan abstraksi mendasar sistem perangkat lunak dan hubungannya.

3. *Implementation and Unit Testing*

Tahap ini dilakukan implementasi berdasarkan perancangan menjadi sebuah kode program. Pengujian unit mencakup tahapan verifikasi bahwa setiap unit telah memenuhi spesifikasi sistem.

4. *Integration and System Testing*

Tahap ini integrasi dan pengujian program dilakukan untuk menjadikan suatu kesatuan sistem yang lengkap dan perlu dipastikan kembali bahwa seluruh kebutuhan yang telah didefinisikan pada tahapan sebelumnya telah terpenuhi. Setelah dilakukan pengujian, perangkat lunak dapat diberikan pada klien.

5. *Operation and maintenance*

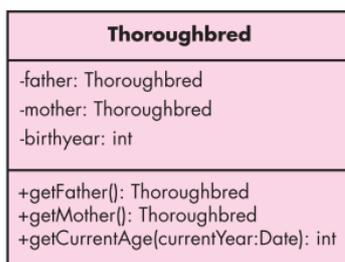
Tahap ini merupakan tahap yang paling panjang karena sistem yang telah diimplementasi harus dipastikan dapat beroperasi dengan baik. Proses pemeliharaan meliputi memperbaiki kode *error* dan memperbaiki kinerja sistem agar semakin baik.

2.3.2 Pendekatan Berorientasi Objek

Pendekatan berorientasi objek merupakan pendekatan yang digunakan dalam melihat suatu permasalahan dengan cara membagi permasalahan tersebut menjadi objek-objek yang memiliki atribut dan tingkah laku spesifik, objek tersebut dimanipulasi sehingga memiliki fungsi yang biasa disebut *methods*, *operations*, atau *services* yang saling berkomunikasi satu dan lainnya (Pressman, 2010). Pendekatan berorientasi objek secara umum menggunakan notasi UML atau *Unified Modeling Language*. Notasi tersebut terdiri dari beberapa jenis diagram diantaranya:

a. *Class Diagram*

Proses pemodelan kelas yang meliputi atribut, operasi, dan relasi antar kelas. *Class diagram* menggambarkan sistem secara terstruktur dan bersifat statik, sehingga tidak menampilkan komunikasi yang dinamis antar objek didalam *class diagram* (Pressman, 2010). Elemen utama pada *class diagram* terdiri dari kotak, kotak ini menjadi representasi *class* dan *interface*. Pada setiap kotak dibagi oleh sebuah garis horizontal, bagian paling atas merupakan nama kelas tersebut diikuti atribut yang merupakan sesuatu yang dapat disediakan oleh objek dari kelas tersebut setiap saat dan pada bagian ketiga terdapat operasi atau perilaku dari sebuah kelas (Pressman, 2010). Contoh *class diagram* dapat dilihat pada Gambar 2.2.

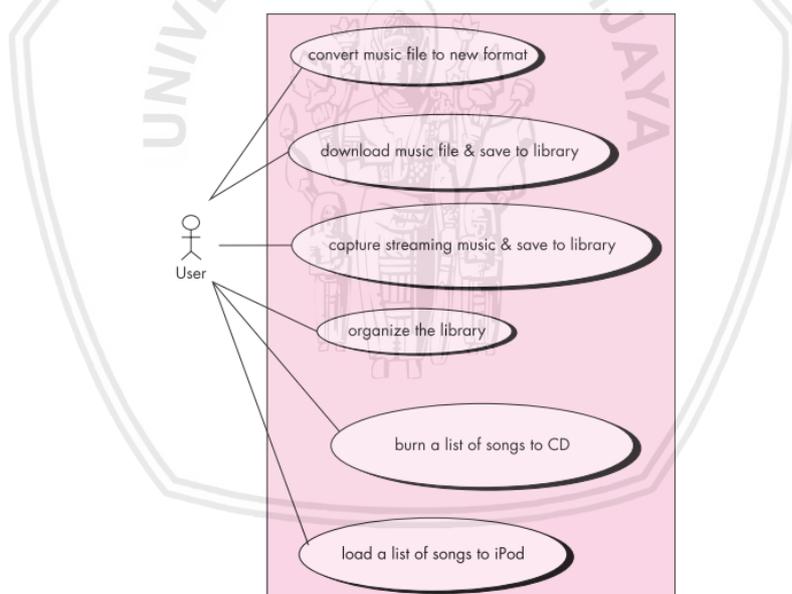


Gambar 2.2 Contoh Class Diagram

(Sumber: Pressman, 2010)

b. *Use Case Diagram*

Use case diagram menggambarkan interaksi pengguna dengan sistem dalam keadaan tertentu. Selain itu *use case diagram* juga dapat membantu dalam menentukan fungsionalitas dan fitur yang akan disediakan oleh sistem (Pressman, 2010). Contoh *use case diagram* dapat dilihat pada Gambar 2.3.



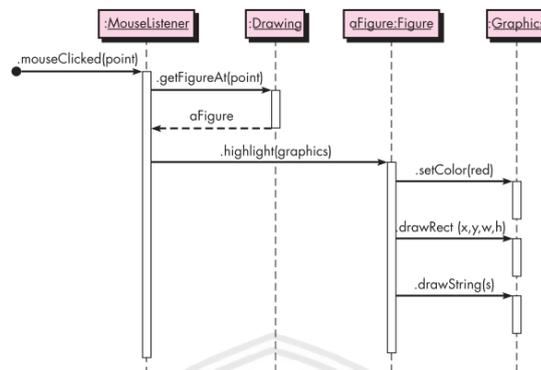
Gambar 2.3 Contoh Use Case Diagram

(Sumber: Pressman, 2010)

c. *Sequence Diagram*

Sequence diagram digunakan untuk menggambarkan interaksi antar objek ketika sebuah fungsi dieksekusi. Selain itu juga menampilkan urutan pesan yang dikirim antar objek agar suatu fungsi dapat dieksekusi dengan baik (Pressman, 2010). Pemanggilan fungsi pada *sequence diagram* digambarkan dengan panah horizontal berawal dari yang memanggil menuju yang dipanggil, selanjutnya dilabeli dengan nama fungsi dan secara

opsional juga diikuti parameter, tipe data, atau tipe kembaliannya (Pressman, 2010). Contoh *sequence diagram* dapat dilihat pada Gambar 2.4.



Gambar 2.4 Contoh Sequence Diagram

(Sumber: Pressman, 2010)

2.4 Kampanye

Kampanye pada dasarnya merupakan bentuk penyampaian pesan dari pengirim kepada khalayak (Ardha, 2014). Sedangkan kampanye politik dipahami sebagai upaya dalam memengaruhi pengambilan keputusan suatu kelompok dengan tujuan memenangkan calon tertentu (Fatimah, 2018). Proses kampanye dapat dilaksanakan oleh perorangan atau sekelompok orang yang terorganisir. kampanye dapat juga bertujuan untuk memengaruhi, menghambat, atau membelokan pencapaian (Otakviani, 2015). Berdasarkan pada uraian tersebut, maka setidaknya kegiatan kampanye mencakup 4 hal yakni:

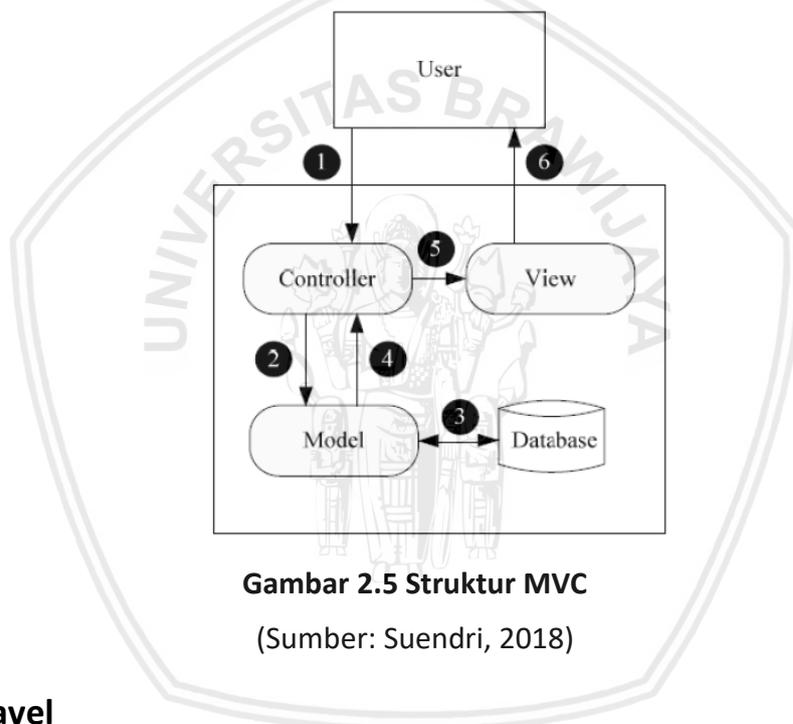
- Tindakan untuk menghasilkan dampak tertentu.
- Target sasaran besar.
- Dilakukan pada waktu tertentu.
- Melalui serangkaian tindakan komunikasi.

2.5 Media Sosial

Media sosial adalah tempat dimana setiap orang dapat memiliki halaman pribadi kemudian terhubung satu sama lain untuk berbagi informasi. Berbeda dengan media tradisional yang secara umum penyebarannya melalui media cetak ataupun *broadcast*, media sosial menggunakan internet sebagai sarana berkomunikasi. Media sosial mampu menampung berbagai informasi secara terbuka sehingga mengundang siapa saja untuk berpartisipasi dalam memberikan komentar ataupun melakukan komunikasi (Lesmana, 2017). Media sosial semakin berkembang seiring dengan pesatnya pertumbuhan teknologi dan internet. Kini media sosial dapat diakses dengan mudah dan cepat sehingga berdampak pada pertumbuhan penggunaannya yang semakin banyak.

2.6 Konsep Model View Controller

Konsep *Model View Controller* (MVC) membagi sebuah sistem menjadi tiga modul yang saling berhubungan yaitu *model*, *view*, dan *controller* (Pastima Simanjuntak & Arwin Kasnady, 2016). *Model* digunakan untuk mengelola data dan perubahan informasi. *Model* juga seringkali merupakan sebuah abstraksi entitas dari dunia nyata. *Model* menjadi jembatan antara data yang tersimpan dalam *database* untuk diambil dan diolah pada *controller*. *View* memiliki tanggung jawab untuk menampilkan data yang telah diolah pada *controller*. Pada umumnya *view* dalam sebuah website diimplementasikan menggunakan bahasa HTML. *Controller* merupakan bagian yang menghubungkan *view* dan *model*. *Controller* menyimpan sekumpulan logika yang digunakan untuk mengolah data sebelum akhirnya ditampilkan pada *view*. Struktur MVC dapat dilihat pada Gambar 2.5.

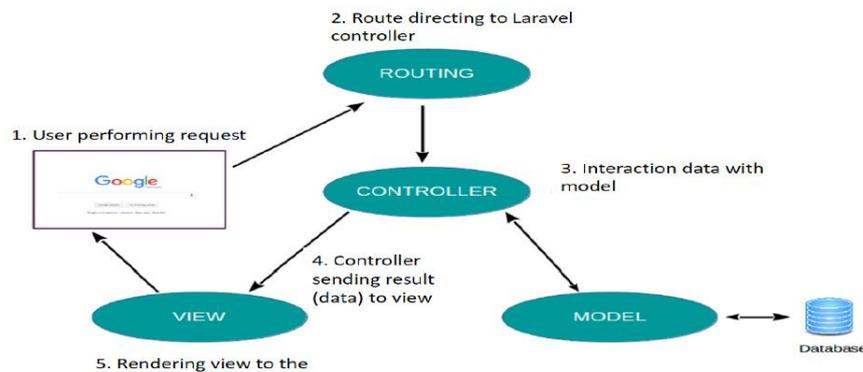


Gambar 2.5 Struktur MVC

(Sumber: Suendri, 2018)

2.7 Laravel

Laravel adalah salah satu dari banyak *framework* pengembangan *website* yang menerapkan konsep *model view controller* (MVC). Laravel diciptakan agar mutu pengembangan perangkat lunak dapat ditingkatkan sekaligus menekan biaya baik pada tahap pengembangan awal maupun telah memasuki tahap pemeliharaan. Laravel menyediakan *library* umum untuk memudahkan proses pengembangan sistem seperti *bundle*, *eloquent ORM*, *reverse routing* dan masih banyak lagi. Selain itu Laravel juga menyesuaikan dengan versi PHP terbaru. Struktur Laravel dapat dilihat pada Gambar 2.6.



Gambar 2.6 Struktur Laravel

(Sumber: Soegoto, 2018)

2.8 Twitter API

Twitter menyediakan API untuk memudahkan pengembang dalam mengakses data Twitter. Data yang dapat diakses diantaranya profil akun Twitter dan *tweet* pengguna. Twitter API terbagi menjadi lima kategori yaitu *standard APIs*, *Premium APIs*, *Enterprise APIs*, *Ads APIs*, dan *Twitter for website*. Kategori tersebut menentukan izin akses data yang diberikan pada kategori pengembang tertentu. Adapun batasan dalam mengakses Twitter API secara umum terbagi dua yaitu terdapat API yang hanya dapat diakses maksimal 15 kali dalam jangka waktu 15 menit dan terdapat pula API yang dapat diakses maksimal 180 kali dalam jangka waktu 15 menit. Pada penelitian ini proses pengaksesan Twitter API menggunakan bantuan *library* pada python bernama Tweepy.

2.8.1 Tweepy

Tweepy merupakan sebuah *library open source* yang digunakan untuk membantu mengakses Twitter API melalui bahasa pemrograman python dengan lebih mudah. Untuk menggunakan Tweepy hanya perlu melakukan *clone* melalui github yang disediakan Tweepy atau dapat juga melalui perintah “pip install tweepy”. Adapun contoh penggunaan tweepy dapat dilihat pada Gambar 2.7.

```
import tweepy

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)

api = tweepy.API(auth)

public_tweets = api.home_timeline()
for tweet in public_tweets:
    print(tweet.text)
```

Gambar 2.7 Contoh Penggunaan Tweepy

2.9 Web Scraping

Web scraping merupakan teknik yang digunakan untuk secara otomatis mengambil informasi tertentu dari sebuah *website*. Tujuan dari *web scraping* sendiri adalah untuk mencari suatu informasi lalu mengekstraknya dan dikumpulkan untuk digunakan kembali. Secara khusus, teknik *web scraping* berfokus untuk mengubah data yang tidak terstruktur lalu menyimpannya pada *database* yang terstruktur (Vargiu dan Urru, 2012). Langkah – langkah dalam melakukan *web scraping* secara umum dimulai dengan mengambil *template* dokumen HTML untuk dipelajari *tag* yang terdapat didalamnya lalu dilanjutkan dengan membuat program untuk mengambil informasi berdasarkan *template* dokumen HTML tersebut. Setelah itu dilakukan proses otomatisasi pengambilan informasi menggunakan program yang telah dibuat sebelumnya (Josi, Abdillah dan Suryayusra, 2014).

2.10 Lexicon Based Sentiment Analysis

Analisis sentimen merupakan sebuah metode pendekatan yang digunakan untuk mengklasifikasi polaritas suatu kalimat atau topik, sedangkan *lexicon based sentiment analysis* merupakan salah satu metode analisis sentimen yang menggunakan kamus dalam mendukung proses klasifikasi tersebut (Kusumawati, 2017). Tahap pertama untuk melakukan sentimen analisis *lexicon based* adalah pengumpulan data lalu dilanjutkan dengan tahapan *preprocessing*. *Preprocessing* merupakan upaya yang dilakukan untuk normalisasi data teks dengan tujuan menghasilkan data terstruktur untuk kebutuhan proses berikutnya (Kartini dan Bachtiar, 2019). Selanjutnya klasifikasi dilakukan dengan mencocokkan kata hasil *preprocessing* dengan data pada kamus lalu mengambil nilai polaritas dari kata yang cocok tersebut. Hasil penambahan nilai polaritas menjadi acuan dalam klasifikasi sentimen. Bernilai positif apabila hasil penambahan polaritas lebih dari nol, netral apabila hasil penambahan polaritas sama dengan nol, dan bernilai negatif apabila hasil penambahan polaritas kurang dari nol.

2.11 Weighted Product

Weighted product (WP) merupakan metode sistem pendukung keputusan multi-kriteria. WP mampu memberikan solusi dalam sistem pemeringkatan dengan optimal (Basri, 2017). Konsep dari metode WP sendiri mengevaluasi sejumlah alternatif terhadap sekumpulan kriteria atau atribut. Alternatif merupakan objek-objek berbeda yang dapat dipilih oleh pengambil keputusan, sedangkan kriteria merupakan nilai pendukung dalam menentukan alternatif yang akan dipilih. Terdapat 2 jenis kriteria yaitu kriteria keuntungan dan kriteria biaya. Kriteria keuntungan merupakan kriteria yang apabila nilainya semakin besar maka hasilnya akan semakin baik sedangkan kriteria biaya merupakan kriteria yang apabila nilainya semakin besar maka hasilnya akan semakin buruk. Tahap pertama yang dilakukan dalam metode WP yaitu menentukan bobot untuk setiap kriteria dengan total bobot bernilai 1. Untuk memastikan total bobot bernilai 1 dapat menggunakan perbaikan bobot dengan Persamaan 2.1.

$$W_j = \frac{w_j}{\sum w_j} \quad (2.1)$$

Tahap berikutnya adalah menentukan nilai normalisasi keputusan (S_i) pada alternatif dengan menggunakan Persamaan 2.2.

$$S_i = \prod_{j=1}^n X_{ij}^{W_j} \quad (2.2)$$

Keterangan:

S_i = Nilai normalisasi keputusan

X_{ij} = Nilai kriteria setiap alternatif

W_j = Bobot kriteria

Tahap terakhir yaitu menghitung nilai preferensi (V_i) yang digunakan untuk menentukan hasil pemeringkatan alternatif dengan menggunakan Persamaan 2.3.

$$V_i = \frac{\prod_{j=1}^n X_{ij}^{W_j}}{\prod_{j=1}^n X_{ij} * W_j} \quad (2.3)$$

2.12 Pengujian Perangkat Lunak

2.12.1 Definisi Pengujian Perangkat Lunak

Pengujian perangkat lunak dilakukan untuk menunjukkan bahwa perangkat lunak dapat melakukan apa yang seharusnya dapat dilakukan juga untuk mencari *bug* sehingga dapat diatasi sebelum perangkat lunak digunakan (Sommerville, 2011). Terdapat dua tujuan dalam melakukan pengujian, diantaranya:

1. Mendemonstrasikan kepada pengembang dan klien bahwa perangkat lunak telah sesuai dengan kebutuhannya. Pada *custom software* perlu adanya minimal sekali pengujian untuk setiap kebutuhan yang tercantum pada dokumen kebutuhan, sedangkan pada *generic software* perlu dilakukan pengujian terhadap keseluruhan fitur sistem, ditambah kombinasi dari fitur – fitur tersebut.
2. Mendeteksi situasi yang tidak terduga dari spesifikasi yang telah didefinisikan sebelumnya. Ini merupakan konsekuensi dari kekurangan sistem.

2.12.2 Pengujian *White Box*

Pengujian *white box* merupakan pengujian yang menggunakan struktur kontrol pada *component-level design* untuk memperoleh kasus uji (Pressman, 2010). Dengan menggunakan pengujian *white box* maka dapat memperoleh kasus uji sebagai berikut:

1. Memastikan bahwa setiap jalur independen yang terdapat pada modul telah diuji minimal satu kali.
2. Menguji setiap keputusan logika dengan kondisi *true* maupun *false*.
3. Mengeksekusi setiap perulangan sesuai batasannya.
4. Memastikan validnya data pada struktur data internal.

2.12.3 Pengujian *Black-Box*

Pengujian *black box* akan berfokus untuk melakukan pengujian kebutuhan fungsional perangkat lunak. Pada pengujian *black box* pengujian dilakukan untuk memperoleh kondisi masukan yang dapat memenuhi setiap kebutuhan fungsional sistem. Pengujian *black box* tidak dapat diasumsikan sebagai pengganti dari pengujian *white box* melainkan pengujian *black box* merupakan pelengkap dari pengujian *white box*. Pengujian *black box* dilakukan ketika memasuki tahap akhir pengujian. Hal tersebut dikarenakan pengujian *black box* lebih berfokus pada domain informasi dibandingkan pada struktur kontrol (Pressman, 2010).

2.12.4 Pengujian *Usability*

Pengujian *usability* bertujuan untuk mengevaluasi kemudahan pengguna saat berinteraksi dengan sistem (Pressman, 2010). Pengujian ini dilakukan ketika sistem telah selesai diimplementasikan dan siap digunakan oleh pengguna. Pengujian *usability* melibatkan responden untuk memberikan penilaian terhadap pengalaman dalam berinteraksi dengan sistem. Pada penelitian ini metode pengujian *usability* yang digunakan adalah *heuristic evaluation*.

2.12.4.1 *Heuristic Evaluation*

Heuristic evaluation melibatkan pakar dalam melakukan pengujian. Pakar tersebut akan mengevaluasi sistem secara langsung dengan mencari kejanggalan pada 10 aspek usabilitas yang telah dikembangkan oleh Jacob Nielsen (Kurniawan, Rokhmawati dan Rachmadi, 2018). Hasil temuan kejanggalan akan diberi nilai *severity ratings* dengan rentang nilai 0 sampai 4 sesuai dengan tingkat keparahan yang ditemukan. Nilai tersebut menjadi pertimbangan dalam urgensi melakukan perbaikan pada sistem. Tabel 2.1 akan menjelaskan mengenai 10 aspek *heuristic evaluation* dan Tabel 2.2 akan menjelaskan mengenai nilai *severity ratings*.

Tabel 2.1 Aspek *Heuristic Evaluation*

No	Aspek <i>Heuristic Evaluation</i>	Penjelasan
1	<i>Visibility of system status</i>	Sistem mampu menyediakan informasi terkait kondisi yang sedang terjadi, dapat melalui pesan <i>feedback</i> dengan waktu yang sewajarnya.
2	<i>Match between system and the real world</i>	Sistem menggunakan kata dan kalimat yang familiar dengan bahasa pengguna. Informasi ditampilkan pada sistem dapat dibaca secara natural dan logis.
3	<i>User control and freedom</i>	Sistem menyediakan verifikasi aksi dikarenakan pengguna sering kali melakukan kesalahan dalam menggunakan fungsional sistem.
4	<i>Consistency and standards</i>	Penggunaan kata, aksi dan situasi yang diterapkan dalam sistem mengikuti sistem sejenis.
5	<i>Error prevention</i>	Sistem menyediakan fungsi pencegahan sebelum terjadinya <i>error</i> .
6	<i>Recognition rather than recall</i>	Pengguna tidak perlu mengingat informasi dalam menjalankan suatu fungsional.
7	<i>Flexibility and efficiency of use</i>	Sistem fleksibel dan efisien untuk digunakan pengguna baru maupun pengguna yang telah familiar dengan sistem sejenis.
8	<i>Aesthetic and minimalist design</i>	Sistem menyampaikan pesan yang relevan dan dibutuhkan oleh pengguna.
9	<i>Help users recognize, diagnose, and recover from errors</i>	Sistem menggunakan bahasa natural dan bukan dengan bahasa pemrograman dalam menyampaikan pesan <i>error</i> .
10	<i>Help and documentation</i>	Terdapat dokumentasi yang berfokus pada keterangan aktivitas yang dapat dilakukan oleh pengguna.

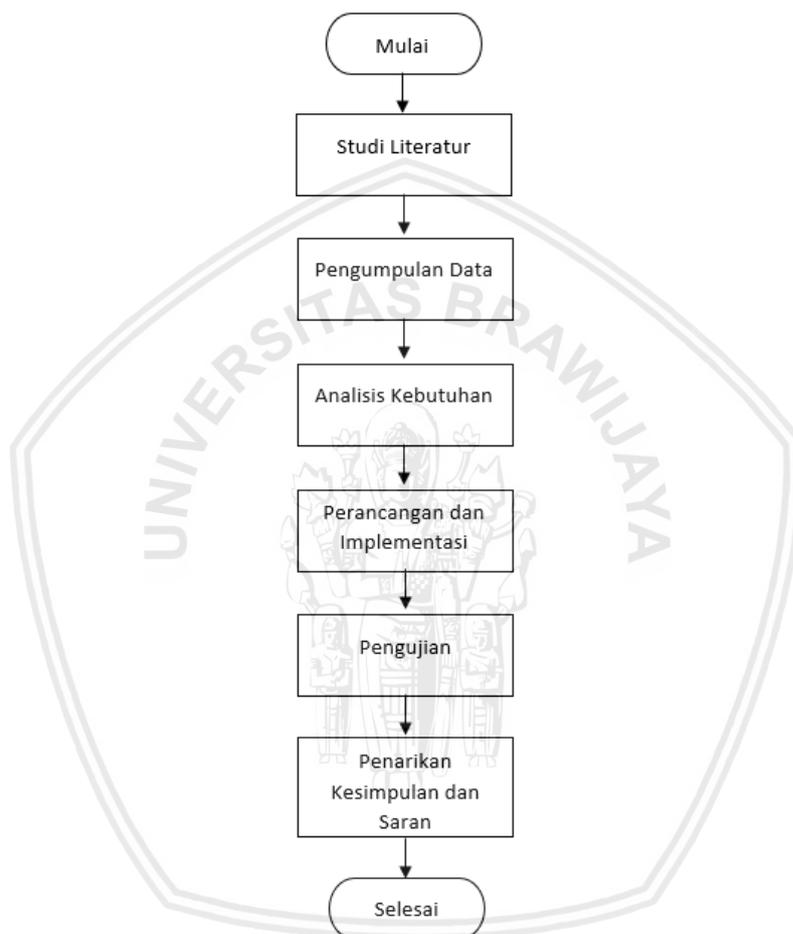
Tabel 2.2 Keterangan Severity Ratings

<i>Severity Ratings</i>	Keterangan
0	Tidak ditemukan kejanggalan dan sistem mudah digunakan.
1	Terdapat sedikit kejanggalan namun tidak akan memengaruhi aktivitas pengguna. Tidak diperlukan perubahan sistem apabila waktu pengerjaan mendesak.
2	Terdapat kejanggalan yang dapat menyulitkan pengguna dalam melakukan aktivitas dalam sistem. Sebaiknya dilakukan perbaikan dengan prioritas rendah.
3	Terdapat kejanggalan yang menyulitkan pengguna melakukan aktivitas dalam sistem. Sebaiknya dilakukan perbaikan dengan prioritas tinggi.
4	Ditemukan kejanggalan yang membuat perlu adanya perubahan sebelum sistem dapat digunakan oleh pengguna.



BAB 3 METODOLOGI

Secara terstruktur terdapat enam tahapan metodologi penelitian dimulai dari melakukan studi literatur, pengumpulan data, analisis kebutuhan, perancangan dan implementasi, pengujian, dan terakhir mengambil kesimpulan juga saran atas hasil akhir penelitian. Diagram alir metodologi penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Metodologi Penelitian

3.1 Studi Literatur

Pada tahap ini dilakukan kajian terhadap teori-teori yang mendukung dalam proses penyelesaian masalah penelitian. Studi literatur pada penelitian ini dilakukan dengan mengumpulkan data dari berbagai sumber seperti jurnal, buku, atau dokumen yang berkaitan terhadap masalah yang sedang dikaji. Teori-teori tersebut diantaranya adalah rekayasa perangkat lunak, model *waterfall*, pendekatan berorientasi objek, kampanye, media sosial, konsep *model view controller*, Laravel, Twitter API, *web scraping*, *lexicon based sentiment analysis*, *weighted product*, pengujian perangkat lunak, pengujian *white box*, pengujian *black box*, dan pengujian *usability*.

3.2 Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data yang digunakan untuk mendukung jalannya penelitian. Pengumpulan data sendiri dilakukan dengan dua cara yaitu mengakses Twitter API dan melakukan teknik *scraping* pada *web* Twitter dan *web* penyedia tren topik Indonesia. Data yang diperoleh melalui Twitter API berupa profil Twitter pengguna seperti nama, foto, *bio* Twitter, tanggal akun dibuat, lokasi pengguna, jumlah *followers*, jumlah *following*, jumlah *tweet*, dan *tweet-tweet* pengguna beserta jumlah *retweet* dan *likes tweet* tersebut. Data *replies* tidak disediakan oleh Twitter API sehingga untuk mendapatkan data tersebut dilakukan teknik *scraping* dari *web* Twitter. Selain itu teknik *scraping* juga digunakan untuk mendapatkan data tren pada *web* penyedia tren topik indonesia untuk selanjutnya dilakukan teknik *scraping* kembali untuk mengambil *sample tweet* yang membahas tentang tren tersebut melalui *web* Twitter. Data *replies* yang telah berhasil diambil langsung diproses menggunakan metode *lexicon based sentiment analysis* untuk mendapatkan data polarisasi sentimen masyarakat terhadap pengguna. Selain itu data jumlah *retweet*, *likes*, dan *replies* juga langsung diproses menggunakan metode *weighted product* untuk mendapatkan data *tweet* terbaik pengguna dan contoh *tweet* yang membahas mengenai tren topik tertentu. Seluruh data hasil pengumpulan disimpan pada *database* agar mempermudah dalam proses pengaksesan data.

3.3 Analisis Kebutuhan

Pada tahap ini dilakukan analisis kebutuhan untuk memperoleh kebutuhan yang akan diterapkan dalam sistem untuk memonitor aktivitas media sosial. Tahapan pertama yang dilakukan dalam proses analisis kebutuhan adalah elisitasi kebutuhan. Elisitasi kebutuhan penelitian ini menggunakan metode *brainstorming* yang dilakukan pada PT Astra Graphia Information Technology. Hasil dari *brainstorming* tersebut menjadi acuan fitur untuk dikembangkan dalam sistem. Selanjutnya kebutuhan fungsional maupun non-fungsional akan dijelaskan secara lebih detail pada tahap spesifikasi kebutuhan. Setelah itu dilakukan tahap pemodelan sistem dengan pendekatan berorientasi objek. Pemodelan ini dilakukan untuk mengilustrasikan siapa saja aktor yang ada pada sistem dan bagaimana aktor tersebut berinteraksi dengan sistem. Hasil dari tahap pemodelan tersebut antara lain *use case diagram* dan *use case scenario*.

3.4 Perancangan dan Implementasi

Pada tahap ini dilakukan perancangan sistem untuk memonitor aktivitas media sosial sesuai dengan hasil analisis kebutuhan yang telah dilakukan. Hasil dari tahap perancangan ini akan digunakan sebagai acuan dalam melakukan implementasi sistem. Tahap perancangan terdiri dari perancangan arsitektur sistem, perancangan komponen, perancangan data dan perancangan antarmuka. Pada tahap perancangan arsitektur sistem akan dibuat perancangan berupa *sequence diagram* dan *class diagram* sistem. Selanjutnya perancangan komponen akan melakukan perancangan terhadap algoritme sistem. Perancangan data berisikan

perancangan *Entity Relationship Diagram* (ERD) dan *Physical Data Model* (PDM). Sedangkan perancangan antarmuka akan menggambarkan tampilan dari sistem dengan menggunakan perancangan *layout*.

Setelah tahap perancangan selesai dilakukan maka dilanjutkan dengan tahap implementasi. Pada tahap ini seluruh hasil perancangan diimplementasikan menjadi sistem. Implementasi sendiri terdiri dari spesifikasi sistem, implementasi kode program, basis data, dan antarmuka. Tahap spesifikasi sistem akan menjelaskan mengenai perangkat keras dan perangkat lunak yang digunakan dalam melakukan implementasi sistem. Selanjutnya tahap implementasi program akan mengubah perancangan algoritme menjadi kode program. Tahap implementasi basis data akan menjelaskan mengenai *Data Definition Language* (DDL) yang digunakan dalam mengimplementasikan perancangan basis data. Terakhir tahap implementasi antarmuka merupakan implementasi dari perancangan *layout* yang telah dilakukan sebelumnya.

3.5 Pengujian

Pada tahap ini dilakukan pengujian yang bertujuan untuk menguji hasil implementasi sistem selain itu memastikan kembali bahwa hasil akhir implementasi sistem telah sesuai dengan kebutuhan yang telah didefinisikan sebelumnya. Terdapat tiga tahap pengujian yang dilakukan pada penelitian ini yaitu pengujian *white box*, *black box*, dan *usability*. Pengujian *white box* bertujuan untuk menguji jalur dasar dari algoritme. Pengujian *black box* akan melakukan pengujian terhadap daftar kebutuhan dan memastikannya telah berjalan dengan benar. Sedangkan pengujian *usability* akan melakukan pengujian untuk memastikan sistem telah mudah untuk digunakan oleh pengguna.

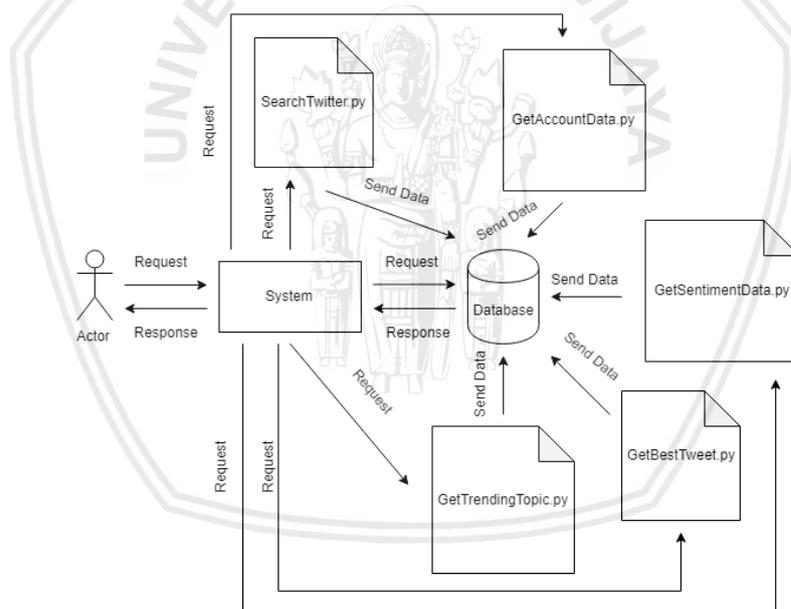
3.6 Penarikan Kesimpulan dan Saran

Pada tahap ini dilakukan pemberian kesimpulan dan saran dari penulis yang bertujuan untuk memberikan kesimpulan dari hasil penelitian dengan judul “Pengembangan Sistem untuk Memonitor Aktivitas Media Sosial Berbasis Web (Studi Kasus Media Sosial Twitter Calon Legislatif pada Pemilihan Legislatif 2019)”. Hasil kesimpulan yang diberikan harus sesuai dengan rumusan masalah yang telah didefinisikan pada tahap awal penelitian. Selain itu terdapat saran dari penulis terkait pengembangan sistem untuk memonitor aktivitas media sosial agar pihak pengembang selanjutnya dapat mengetahui dan memperbaiki kekurangan dari sistem ini.

BAB 4 ANALISIS KEBUTUHAN

4.1 Deskripsi Umum Sistem

Sistem untuk memonitor aktivitas media sosial merupakan sistem yang dapat membantu dalam memantau akun media sosial Twitter sehingga harapannya dapat meningkatkan efektivitas kampanye melalui media sosial. Fitur-fitur yang terdapat pada sistem ini yaitu dapat memonitor perkembangan profil akun Twitter pengguna seperti melihat jumlah *followers*, *tweet*, *likes*, *replies*, sentimen masyarakat terhadap pengguna, dan *tweet* terbaik pengguna. Selain itu sistem ini menyediakan fitur komparasi aktivitas pengguna terhadap kompetitor, hasil komparasi tersebut digunakan sebagai acuan dalam memberikan rekomendasi aktivitas yang perlu dilakukan oleh pengguna. Sistem ini memberikan 10 daftar tren topik Indonesia yang diurutkan berdasarkan topik yang bertahan paling lama sebagai tren selama 24 jam terakhir juga terdapat contoh-contoh *tweet* yang membahas mengenai tren tersebut. Gambaran umum sistem terkait interaksi antar aktor, sistem, *database*, dan Python dapat dilihat pada Gambar 4.1. Sistem ini untuk selanjutnya akan disebut sebagai Momment.



Gambar 4.1 Gambaran Umum Sistem

4.2 Elisitasi Kebutuhan

Elisitasi kebutuhan pada penelitian ini dilakukan dengan cara *brainstorming* pada divisi *Analytics* PT Astra Graphia Information Technology dengan peserta tertera pada Tabel 4.1. Dalam prosesnya teknik *brainstorming* akan mencari ide sebanyak-banyaknya untuk menyelesaikan suatu permasalahan. Hasil dari *brainstorming* terlampir dan dapat dilihat pada Lampiran B. Setelah terkumpul banyak ide dilakukan analisis kebutuhan untuk dapat mendefinisikan kebutuhan fungsional sistem. Dalam penelitian ini analisis kebutuhan dilakukan dua tahap.

Tahap pertama menganalisis hasil ide yang telah terkumpul dengan mengurutkannya berdasarkan ide yang dianggap paling menyelesaikan inti permasalahan. Lalu tahap kedua menganalisis hasil ide yang telah terkumpul dengan mengurutkannya berdasarkan ketersediaan data untuk mengimplementasikan ide tersebut. Hasil akhir dari tahapan ini menghasilkan 23 kebutuhan fungsional yang akan dijelaskan lebih lanjut pada Tabel 4.3. Karena calon pengguna tidak terlibat pada saat melakukan proses *brainstorming* maka hasil dari kebutuhan fungsional tersebut diverifikasi dan divalidasi kembali oleh *head of product development and delivery* kepada pengguna sistem yaitu calon legislatif untuk memastikan bahwa kebutuhan fungsional telah sesuai dengan sistem yang diharapkan dengan bukti keterangan terlampir pada Lampiran A.

Tabel 4.1 Peserta *Brainstorming*

No	Nama	Jabatan
1	Wahyu Adhi Arifiyanto	<i>Head of product development and delivery</i>
2	Wahyu Nugroho	<i>Mysights business development</i>
3	Michael Seno Adhy	<i>Mysights product manager</i>
4	Donny Fernando	<i>Data Analyst</i>
5	Pradina Duta	<i>Software Engineer</i>
6	Vasha Farisi Sarwan Halim	<i>Freelancer</i>
7	Muhammad Ainun Najib	<i>Freelancer</i>
8	Mochamad Dearifaldi	<i>Freelancer</i>
9	Jesy Thesalonicha	<i>Freelancer</i>

4.3 Identifikasi Aktor

Identifikasi aktor merupakan tahap yang dilakukan untuk mengidentifikasi aktor yang terlibat dan berinteraksi secara langsung dengan sistem. Pada Tabel 4.2 terdapat kolom aktor yang akan menjelaskan siapa saja yang akan berperan dalam sistem dan pada kolom deskripsi aktor akan disebutkan peran aktor tersebut dalam sistem.

Tabel 4.2 Identifikasi aktor

No	Aktor	Deskripsi Aktor
1	Tamu	Tamu merupakan aktor yang belum melakukan proses <i>login</i> .
2	Pengguna	Pengguna merupakan aktor yang telah melakukan proses <i>login</i> dan dapat melakukan seluruh aktivitas di dalam sistem.

4.4 Kebutuhan Fungsional

Kebutuhan fungsional merupakan daftar kebutuhan yang harus disediakan oleh sistem. Pada Tabel 4.3 terdapat kolom kode kebutuhan yang digunakan untuk mengidentifikasi setiap kebutuhan dengan susunan kode MM-X-Y-Z. MM merupakan singkatan dari nama aplikasi, X sebagai kode dalam membedakan antara kebutuhan fungsional dan non-fungsional, Y sebagai kode urutan kebutuhan, dan Z sebagai kode urutan spesifikasi kebutuhan. Selain itu terdapat kolom nama *use case* sebagai penamaan dari kebutuhan tersebut serta kolom deskripsi untuk mendeskripsikan kebutuhan itu sendiri.

Tabel 4.3 Kebutuhan fungsional

No	Kode Kebutuhan	Nama <i>Use Case</i>	Deskripsi
1	MM-F-001	<i>Login</i>	<p>Sistem harus mampu menyediakan fungsi untuk mengizinkan tamu masuk kedalam sistem sebagai pengguna.</p> <p>Spesifikasi Kebutuhan:</p> <ol style="list-style-type: none"> 1. Sistem menyediakan <i>field e-mail</i> yang digunakan untuk menerima masukan <i>e-mail</i> dari tamu (MM-F-001-01). 2. Sistem menyediakan <i>field password</i> yang digunakan untuk menerima masukan password dari tamu (MM-F-001-02). 3. Sistem menyediakan tombol <i>login</i> untuk memulai proses otorisasi tamu (MM-F-001-03). 4. Sistem dapat memberikan pesan "Silahkan isi field ini terlebih dahulu" apabila tamu telah menekan tombol <i>login</i> namun <i>field e-mail</i> dan/atau <i>password</i> belum menerima masukan (MM-F-001-04). 5. Sistem dapat memberikan pesan "<i>e-mail/password</i> tidak cocok, silahkan coba lagi" apabila tamu gagal melakukan proses otorisasi (MM-F-001-05).

Tabel 4.3 Kebutuhan Fungsional (Lanjutan)

No	Kode Kebutuhan	Nama <i>Use Case</i>	Deskripsi
2	MM-F-002	Daftar Akun	<p>Sistem harus mampu menyediakan fungsi bagi tamu untuk mendaftar sebagai pengguna.</p> <p>Spesifikasi Kebutuhan:</p> <ol style="list-style-type: none"> 1. Sistem menyediakan <i>field</i> nama lengkap yang digunakan untuk menerima masukan nama lengkap dari tamu (MM-F-002-01). 2. Sistem menyediakan <i>field e-mail</i> yang digunakan untuk menerima masukan <i>e-mail</i> dari tamu (MM-F-002-02). 3. Sistem menyediakan <i>field password</i> yang digunakan untuk menerima masukan <i>password</i> dari tamu (MM-F-002-03). 4. Sistem menyediakan <i>field konfirmasi password</i> yang digunakan untuk menerima masukan <i>password</i> untuk mengkonfirmasi <i>password</i> yang dimasukan telah sama (MM-F-002-04). 5. Sistem menyediakan tombol daftar yang digunakan tamu untuk memulai proses pendaftaran akun (MM-F-002-05). 6. Sistem dapat memberikan pesan "Silahkan isi <i>field</i> ini terlebih dahulu" apabila terdapat <i>field</i> yang belum mendapatkan masukan (MM-F-002-06). 7. Sistem dapat memberikan pesan "E-mail telah terdaftar" (MM-F-002-07)

Tabel 4.3 Kebutuhan Fungsional (Lanjutan)

No	Kode Kebutuhan	Nama Use Case	Deskripsi
3	MM-F-003	Menetapkan akun Twitter	Sistem harus mampu menyediakan fungsi bagi pengguna untuk menetapkan akun Twitter yang akan dimonitor.
			<p>Spesifikasi Kebutuhan:</p> <ol style="list-style-type: none"> 1. Sistem menyediakan tombol “tetapkan akun” yang digunakan dalam menetapkan akun Twitter hasil pencarian untuk dimonitor aktivitasnya (MM-F-003-01). 2. Sistem mampu menampilkan pesan “Akun telah terdaftar” apabila setelah dilakukan pencarian akun dengan <i>username</i> sesuai masukan pengguna telah terdaftar sebelumnya (MM-F-003-02).
4	MM-F-004	Lihat profil akun Twitter	Sistem harus mampu menyediakan fungsi agar pengguna dapat melihat profil media sosialnya.
			<p>Spesifikasi Kebutuhan:</p> <ol style="list-style-type: none"> 1. Sistem menyediakan <i>card</i> yang digunakan untuk menampilkan nama, <i>username</i>, foto profil, foto <i>banner</i>, <i>bio</i>, jumlah tweet, jumlah <i>followers</i>, jumlah <i>following</i>, lokasi, dan tanggal akun Twitter pengguna dibuat (MM-F-004-01).
5	MM-F-005	Lihat penilaian aktivitas selama seminggu terakhir	Sistem harus mampu menyediakan fungsi agar pengguna dapat melihat penilaian aktivitas selama seminggu terakhir.

Tabel 4.3 Kebutuhan Fungsional (Lanjutan)

No	Kode Kebutuhan	Nama Use Case	Deskripsi
			<p>Spesifikasi Kebutuhan:</p> <ol style="list-style-type: none"> 1. Sistem menampilkan pesan “Aktivitas anda seminggu ini baik” serta <i>icon smile</i> apabila rata-rata aktivitas pengguna lebih baik dibanding kompetitor (MM-F-005-01). 2. Sistem menampilkan pesan “Aktivitas anda seminggu ini perlu ditingkatkan” serta <i>icon frown</i> apabila rata-rata aktivitas pengguna tidak lebih baik dibanding kompetitor (MM-F-005-02).
6	MM-F-006	Lihat rekomendasi dalam pengelolaan akun Twitter	<p>Sistem harus mampu menyediakan fungsi agar pengguna dapat melihat rekomendasi aksi yang dapat digunakan sebagai pertimbangan dalam mengelola akun Twitter.</p> <p>Spesifikasi Kebutuhan:</p> <ol style="list-style-type: none"> 1. Sistem menampilkan pesan rekomendasi “<i>posting tweet</i> hari ini” apabila pengguna terdeteksi belum melakukan <i>posting</i> pada hari tersebut (MM-F-006-01). 2. Sistem menampilkan pesan rekomendasi jumlah <i>tweet</i> yang perlu <i>diposting</i> agar aktivitas <i>posting</i> pengguna lebih baik dibanding kompetitornya (MM-F-006-02). 3. Sistem menampilkan pesan rekomendasi “menanggapi <i>mention</i> masuk kepada anda” apabila pengguna terdeteksi belum menanggapi <i>mention</i> pada hari tersebut (MM-F-006-03).

Tabel 4.3 Kebutuhan Fungsional (Lanjutan)

No	Kode Kebutuhan	Nama Use Case	Deskripsi
			<p>4. Sistem menampilkan pesan rekomendasi jam <i>posting</i> (MM-F-006-04).</p> <p>5. Sistem menampilkan pesan rekomendasi untuk melihat dan menanggapi tren topik pada hari tersebut (MM-F-006-05).</p>
7	MM-F-007	Lihat rata-rata peningkatan <i>followers</i> pengguna dan kompetitor	<p>Sistem harus mampu menyediakan fungsi agar pengguna dapat melihat jumlah rata-rata peningkatan <i>followers</i> pengguna dan kompetitor.</p> <p>Spesifikasi Kebutuhan:</p> <p>1. Sistem menyediakan <i>card</i> untuk menampilkan rata-rata peningkatan jumlah <i>followers</i> pengguna selama 7 hari terakhir, apabila <i>card</i> ditekan maka akan menampilkan rata-rata peningkatan jumlah <i>followers</i> kompetitor selama 7 hari terakhir (MM-F-007-01).</p>
8	MM-F-008	Lihat rata-rata <i>posting tweet</i> pengguna dan kompetitor	<p>Sistem harus mampu menyediakan fungsi agar pengguna dapat melihat jumlah rata-rata <i>posting tweet</i> pengguna dan kompetitor.</p> <p>Spesifikasi Kebutuhan :</p> <p>1. Sistem menyediakan <i>card</i> untuk menampilkan rata-rata <i>posting tweet</i> pengguna selama 7 hari terakhir, apabila <i>card</i> ditekan maka akan menampilkan rata-rata <i>posting tweet</i> kompetitor selama 7 hari terakhir (MM-F-008-01).</p>

Tabel 4.3 Kebutuhan Fungsional (Lanjutan)

No	Kode Kebutuhan	Nama Use Case	Deskripsi
9	MM-F-009	Lihat rata-rata <i>retweet</i> pengguna dan kompetitor	Sistem harus mampu menyediakan fungsi agar pengguna dapat melihat jumlah rata-rata <i>retweet</i> yang didapatkan pengguna dan kompetitor.
			<p>Spesifikasi Kebutuhan:</p> <ol style="list-style-type: none"> 1. Sistem menyediakan <i>card</i> untuk menampilkan rata-rata <i>retweet</i> yang didapatkan pengguna selama 7 hari terakhir, apabila <i>card</i> ditekan maka akan menampilkan rata-rata <i>retweet</i> yang didapatkan kompetitor selama 7 hari terakhir (MM-F-009-01).
10	MM-F-0010	Lihat rata-rata <i>likes</i> pengguna dan kompetitor	Sistem harus mampu menyediakan fungsi agar pengguna dapat melihat jumlah rata-rata <i>likes</i> yang didapatkan pengguna dan kompetitor.
			<p>Spesifikasi Kebutuhan:</p> <ol style="list-style-type: none"> 1. Sistem menyediakan <i>card</i> untuk menampilkan rata-rata <i>likes</i> yang didapatkan pengguna selama 7 hari terakhir, apabila <i>card</i> ditekan maka akan menampilkan rata-rata <i>likes</i> yang didapatkan kompetitor selama 7 hari terakhir (MM-F-010-01).
11	MM-F-011	Lihat rata-rata <i>replies</i> pengguna dan kompetitor	Sistem harus mampu menyediakan fungsi agar pengguna dapat melihat jumlah rata-rata <i>replies</i> yang didapatkan pengguna dan kompetitor.

Tabel 4.3 Kebutuhan Fungsional (Lanjutan)

No	Kode Kebutuhan	Nama Use Case	Deskripsi
			<p>Spesifikasi Kebutuhan:</p> <ol style="list-style-type: none"> 1. Sistem menyediakan <i>card</i> untuk menampilkan rata-rata <i>replies</i> yang didapatkan pengguna selama 7 hari terakhir, apabila <i>card</i> ditekan maka akan menampilkan rata-rata <i>replies</i> yang didapatkan kompetitor selama 7 hari terakhir (MM-F-011-01).
12	MM-F-012	Lihat hasil analisis sentiment	<p>Sistem harus mampu menyediakan fungsi agar pengguna dapat melihat analisis sentimen masyarakat terhadap pengguna.</p> <p>Spesifikasi Kebutuhan:</p> <ol style="list-style-type: none"> 1. Sistem menyediakan <i>pie chart</i> untuk menampilkan sentimen positif, negatif, dan netral dalam satuan persen (MM-F-012-01). 2. Sistem menyediakan <i>card</i> untuk menampilkan 5 sentimen positif dan negatif (MM-F-012-02).
13	MM-F-013	Lihat tweet terbaik	<p>Sistem harus mampu menyediakan fungsi agar pengguna dapat melihat 5 tweet terbaik berdasarkan pembobotan jumlah <i>retweet</i>, <i>likes</i>, <i>replies</i> dan jumlah sentimen positif-negatif dari <i>replies</i> tweet tersebut.</p> <p>Spesifikasi Kebutuhan:</p> <ol style="list-style-type: none"> 1. Sistem menyediakan <i>card</i> untuk menampilkan 5 tweet terbaik pengguna disertai keterangan jumlah <i>retweet</i>, <i>likes</i>, <i>replies</i>, dan jumlah sentimen positif-negatif dari <i>replies</i> tweet tersebut (MM-F-013-01).

Tabel 4.3 Kebutuhan Fungsional (Lanjutan)

No	Kode Kebutuhan	Nama Use Case	Deskripsi
14	MM-F-014	Mengelola daftar kompetitor	Sistem harus mampu menyediakan fungsi agar pengguna dapat melihat daftar kompetitor yang sebelumnya telah ditambahkan.
			<p>Spesifikasi Kebutuhan:</p> <ol style="list-style-type: none"> 1. Sistem menyediakan <i>card</i> untuk menampilkan daftar nama dan <i>username</i> Twitter kompetitor (MM-F-014-01).
15	MM-F-015	Tambah kompetitor	Sistem harus mampu menyediakan fungsi agar dapat menambahkan kompetitor sesuai masukan pengguna.
			<p>Spesifikasi Kebutuhan:</p> <ol style="list-style-type: none"> 1. Sistem menyediakan tombol “+” yang digunakan untuk menampilkan halaman pencarian akun Twitter yang akan ditambahkan sebagai kompetitor (MM-F-015-01).
16	MM-F-016	Hapus kompetitor	Sistem harus mampu menyediakan fungsi agar dapat menghapus kompetitor sesuai pilihan pengguna.
			<p>Spesifikasi Kebutuhan:</p> <ol style="list-style-type: none"> 1. Sistem menyediakan tombol “hapus” yang digunakan untuk menghapus kompetitor sesuai pilihan pengguna (MM-F-016-01).
17	MM-F-017	Lihat detail perbandingan aktivitas pengguna dan kompetitor	Sistem harus mampu menyediakan fungsi agar pengguna dapat melihat detail perbandingan aktivitas media sosial pengguna dan kompetitor.

Tabel 4.3 Kebutuhan Fungsional (Lanjutan)

No	Kode Kebutuhan	Nama Use Case	Deskripsi
			<p>Spesifikasi Kebutuhan:</p> <ol style="list-style-type: none"> 1. Sistem menyediakan tombol “detail” yang digunakan untuk menampilkan halaman detail perbandingan aktivitas media sosial pengguna dan kompetitor (MM-F-017-01). 2. Sistem menyediakan <i>card</i> yang digunakan untuk menampilkan nama, <i>username</i>, foto profil, foto <i>banner</i>, <i>bio</i>, jumlah tweet, jumlah <i>followers</i>, jumlah <i>following</i>, lokasi, dan tanggal akun Twitter kompetitor pilihan pengguna dibuat (MM-F-017-02). 3. Sistem menyediakan tabel yang digunakan untuk menampilkan perbandingan jumlah peningkatan <i>followers</i>, <i>posting tweet</i>, <i>retweet</i>, <i>likes</i>, <i>replies</i> pengguna dan kompetitor (MM-F-017-03). 4. Sistem menyediakan grafik garis untuk menampilkan perbandingan peningkatan <i>followers</i>, <i>posting tweet</i>, <i>retweet</i>, <i>likes</i>, dan <i>replies</i> pengguna dan kompetitor (MM-F-017-04).
18	MM-F-018	<i>Posting tweet</i>	<p>Sistem harus mampu menyediakan fungsi agar pengguna dapat melakukan <i>posting tweet</i> melalui sistem.</p> <p>Spesifikasi Kebutuhan:</p> <ol style="list-style-type: none"> 1. Sistem menyediakan tombol tweet agar pengguna dapat melakukan <i>posting tweet</i> melalui sistem. (MM-F-018-01)

Tabel 4.3 Kebutuhan Fungsional (Lanjutan)

No	Kode Kebutuhan	Nama Use Case	Deskripsi
19	MM-F-019	Lihat tren topik Indonesia	Sistem harus mampu menyediakan fungsi agar pengguna dapat melihat 10 daftar tren topik Indonesia
			<p>Spesifikasi Kebutuhan:</p> <ol style="list-style-type: none"> 1. Sistem menyediakan 10 daftar tren topik Indonesia (MM-F-019-01). 2. Sistem menyediakan tombol detail yang dapat ditekan untuk menampilkan contoh tweet dari tren topik tersebut (MM-F-019-02)
20	MM-F-020	Lihat contoh <i>tweet</i> berdasarkan tren	Sistem harus mampu menyediakan fungsi agar pengguna dapat melihat contoh <i>tweet</i> berdasarkan tren.
			<p>Spesifikasi Kebutuhan:</p> <ol style="list-style-type: none"> 1. Sistem menyediakan <i>modal pop-up</i> untuk menampilkan 10 contoh <i>tweet</i> berdasarkan tren (MM-F-020-01).
21	MM-F-021	Cari akun Twitter	Sistem harus mampu menyediakan fungsi agar pengguna dapat mencari akun Twitter melalui sistem.
			<p>Spesifikasi Kebutuhan:</p> <ol style="list-style-type: none"> 1. Sistem menyediakan <i>field username</i> yang digunakan untuk menerima masukan <i>username</i> akun Twitter pengguna (MM-F-021-01).
22	MM-F-022	<i>Update data</i>	Sistem harus mampu menyediakan fungsi agar pengguna dapat melakukan <i>update</i> data melalui sistem.

Tabel 4.3 Kebutuhan Fungsional (Lanjutan)

No	Kode Kebutuhan	Nama Use Case	Deskripsi
			<p>Spesifikasi Kebutuhan:</p> <p>1. Sistem menyediakan tombol <i>update</i> data yang dapat ditekan oleh pengguna untuk memulai proses <i>update</i> data (MM-F-022-01).</p>
23	MM-F-023	<i>Logout</i>	<p>Sistem harus mampu menyediakan fungsi agar pengguna dapat keluar dari sistem.</p> <p>Spesifikasi Kebutuhan:</p> <p>1. Sistem menyediakan tombol <i>logout</i> yang digunakan untuk mengeluarkan pengguna dari sistem dan kembali ke halaman <i>login</i> (MM-F-023-01).</p>

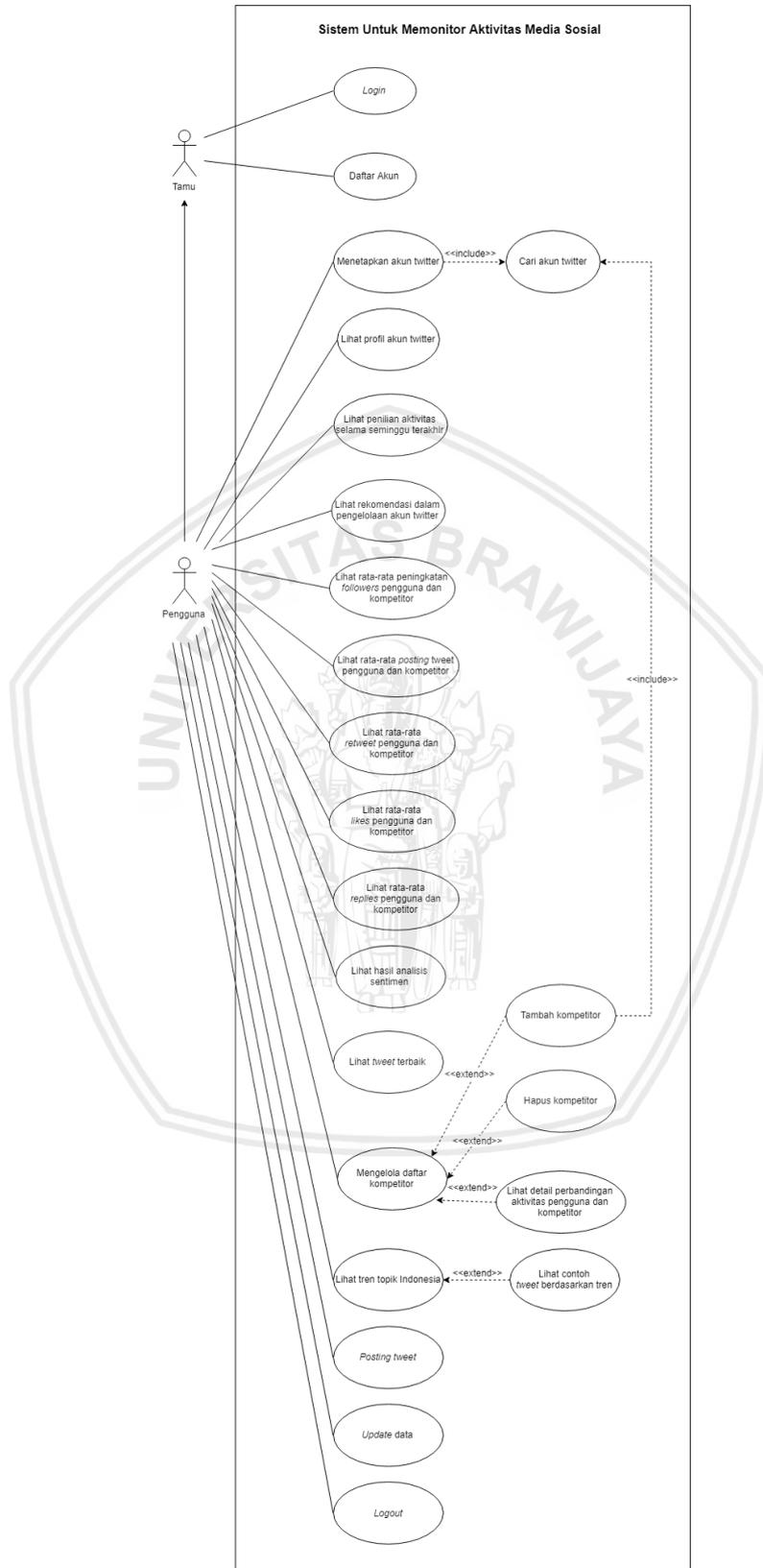
4.5 Kebutuhan Non-Fungsional

Tabel kebutuhan non-fungsional dapat dilihat pada Tabel 4.4.

Tabel 4.4 Kebutuhan Non-Fungsional

No	Kode Fungsi	Nama Fungsi	Deskripsi
1	MM-NF-01	<i>Usability</i>	Kemudahan dalam menggunakan aplikasi yang digunakan oleh pengguna

4.6 Use Case Diagram



Gambar 4.2 Use Case Diagram

4.7 Use Case Scenario

Use case scenario menjelaskan tentang alur dari masing-masing *use case diagram* yang telah didefinisikan pada Gambar 4.2. *Use case scenario* pada penelitian ini menggunakan format lengkap karena menjelaskan bagian-bagian pendukung seperti *objective*, *actor*, *pre-condition*, *main flow*, *alternative flow*, dan *post-condition*.

4.7.1 Use Case Scenario Login

Use case scenario login akan menjelaskan alur dalam melakukan proses *login*. Tujuan dari *use case scenario login* adalah tamu dapat masuk kedalam sistem sebagai pengguna. *Use case scenario login* akan dijelaskan pada Tabel 4.5.

Tabel 4.5 Use Case Scenario Login

<i>Login</i>	
Kode Kebutuhan	MM-F-001
<i>Objective</i>	Tamu masuk kedalam sistem sebagai pengguna.
<i>Actor</i>	Tamu
<i>Pre-condition</i>	Halaman <i>login</i> ditampilkan.
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Tamu memasukan <i>e-mail</i> dan <i>password</i>. 2. Tamu menekan tombol <i>login</i>. 3. Sistem melakukan proses otorisasi. 4. Sistem menampilkan halaman <i>dashboard</i>.
<i>Alternative flow</i>	<ol style="list-style-type: none"> 1. Kegagalan otorisasi tamu: <ol style="list-style-type: none"> 1.1 Jika <i>field e-mail</i> dan/atau <i>password</i> belum menerima masukan maka sistem akan menampilkan pesan "Silahkan isi <i>field</i> ini terlebih dahulu" apabila. 1.2 Jika proses otorisasi gagal maka sistem akan menampilkan pesan "<i>e-mail/password</i> tidak cocok, silahkan coba lagi!".
<i>Post-condition</i>	Tamu berhasil diotorisasi oleh sistem dan masuk ke halaman <i>dashboard</i> sebagai pengguna.

4.7.2 Use Case Scenario Daftar Akun

Use case scenario daftar akun akan menjelaskan alur dalam melakukan proses daftar akun. Tujuan dari *use case scenario daftar akun* adalah tamu dapat terdaftar dalam sistem sebagai pengguna. *Use case scenario* daftar akun akan dijelaskan pada Tabel 4.6.

Tabel 4.6 Use Case Scenario Daftar Akun

Daftar Akun	
Kode Kebutuhan	MM-F-002
Objective	Tamu terdaftar pada sistem sebagai pengguna.
Actor	Tamu
Pre-condition	Halaman <i>register</i> ditampilkan.
Main flow	<ol style="list-style-type: none"> 1. Tamu memasukan nama lengkap, <i>e-mail</i>, <i>password</i>, dan konfirmasi <i>password</i>. 2. Tamu menekan tombol daftar. 3. Sistem menyimpan data tamu. 4. Sistem menampilkan halaman menetapkan akun.
Alternative flow	<ol style="list-style-type: none"> 1. Kegagalan menyimpan data tamu: <ol style="list-style-type: none"> 1.1 Jika terdapat <i>field</i> yang belum menerima masukan maka sistem akan menampilkan pesan “Silahkan isi <i>field</i> ini terlebih dahulu”. 1.2 Jika masukan <i>field password</i> dan konfirmasi <i>password</i> tidak sama maka sistem akan menampilkan pesan “<i>Password</i> tidak sesuai”. 1.3 Jika <i>e-mail</i> masukan tamu sebelumnya telah terdaftar maka sistem akan menampilkan pesan “<i>e-mail</i> telah terdaftar”.
Post-condition	Data tamu disimpan pada <i>database</i> sebagai pengguna.

4.7.3 Use Case Scenario Menetapkan Akun Twitter

Use case scenario menetapkan akun Twitter akan menjelaskan alur dalam proses menetapkan akun Twitter pengguna yang baru mendaftar pada sistem. Tujuan dari *use case scenario* menetapkan akun Twitter adalah pengguna dapat menetapkan akun Twitternya sebagai acuan dalam proses monitor. *Use case scenario* menetapkan akun akan dijelaskan pada Tabel 4.7.

Tabel 4.7 Use Case Scenario Menetapkan Akun Twitter

Menetapkan akun Twitter	
Kode Kebutuhan	MM-F-003
Objective	Pengguna menetapkan akun Twitter.
Actor	Pengguna
Pre-condition	Pengguna <i>login</i> dan akun Twitter belum ditetapkan.
Main flow	<ol style="list-style-type: none"> 1. Sistem menampilkan halaman <i>engage</i>.

Tabel 4.7 Use Case Scenario Menetapkan Akun Twitter (Lanjutan)

Menetapkan akun Twitter (Lanjutan)	
	<ol style="list-style-type: none"> 2. Pengguna memasukkan <i>username</i> Twitter lalu menekan tombol pencarian. 3. Sistem menampilkan hasil pencarian. 4. Pengguna menekan tombol “Tetapkan Akun”. 5. Sistem menampilkan <i>pop-up</i> konfirmasi dalam menetapkan akun media sosial. 6. Pengguna menekan tombol “Ya”. 7. Sistem menampilkan halaman <i>dashboard</i>.
<i>Alternative flow</i>	<ol style="list-style-type: none"> 1. Kegagalan dalam menetapkan akun: <ol style="list-style-type: none"> 1.1 Jika akun hasil pencarian telah ditetapkan sebelumnya, maka tombol “Tetapkan Akun” akan berubah menjadi keterangan “Akun telah terdaftar”.
<i>Post-condition</i>	Akun Twitter pengguna ditetapkan.

4.7.4 Use Case Scenario Lihat Profil Akun Twitter

Use case scenario lihat profil akun Twitter akan menjelaskan alur dalam menampilkan profil akun Twitter. Tujuan dari *use case scenario* lihat profil akun Twitter adalah pengguna dapat melihat profil akun Twitter melalui sistem. *Use case scenario* lihat profil akun Twitter akan dijelaskan pada Tabel 4.8.

Tabel 4.8 Use Case Scenario Lihat Profil Akun Twitter

Lihat profil akun Twitter	
Kode Kebutuhan	MM-F-004
<i>Objective</i>	Pengguna dapat melihat profil Twitter.
<i>Actor</i>	Pengguna
<i>Pre-condition</i>	Halaman <i>dashboard</i> ditampilkan.
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan nama, <i>username</i>, foto profil, foto <i>banner</i>, <i>bio</i>, jumlah tweet, jumlah <i>followers</i>, jumlah <i>following</i>, lokasi, dan tanggal akun Twitter pengguna dibuat.
<i>Alternative flow</i>	-
<i>Post-condition</i>	Profil akun Twitter pengguna ditampilkan.

4.7.5 Use Case Scenario Lihat Penilaian Aktivitas Selama Seminggu Terakhir

Use case scenario lihat penilaian aktivitas selama seminggu terakhir akan menjelaskan alur dalam melihat penilaian aktivitas selama seminggu terakhir. Tujuan dari *use case scenario* lihat penilaian aktivitas selama seminggu terakhir adalah pengguna dapat melihat penilaian aktivitas media sosialnya selama seminggu terakhir. *Use case scenario* lihat penilaian aktivitas selama seminggu terakhir akan dijelaskan pada Tabel 4.9.

Tabel 4.9 Use Case Scenario Lihat Penilaian Aktivitas Selama Seminggu Terakhir

Lihat penilaian aktivitas selama seminggu terakhir	
Kode Kebutuhan	MM-F-005
<i>Objective</i>	Pengguna dapat melihat penilaian aktivitas selama seminggu terakhir.
<i>Actor</i>	Pengguna
<i>Pre-condition</i>	Halaman <i>dashboard</i> ditampilkan.
<i>Main flow</i>	1. Sistem menampilkan pesan “Aktivitas anda seminggu ini baik!” dan <i>icon smile</i> .
<i>Alternative flow</i>	1. Rata-rata aktivitas kompetitor lebih baik dari pengguna 1.1. Jika rata-rata aktivitas kompetitor lebih dari pengguna, maka sistem menampilkan pesan “Aktivitas anda seminggu ini perlu ditingkatkan!” dan <i>icon frown</i> .
<i>Post-condition</i>	Penilaian aktivitas pengguna selama seminggu terakhir ditampilkan.

4.7.6 Use Case Scenario Lihat Rekomendasi Dalam Pengelolaan Akun Twitter

Use case scenario lihat rekomendasi dalam pengelolaan akun Twitter akan menjelaskan alur dalam melihat rekomendasi dalam pengelolaan akun Twitter. Tujuan dari *use case scenario* lihat rekomendasi dalam pengelolaan akun Twitter adalah pengguna dapat melihat rekomendasi dalam pengelolaan akun Twitter. *use case scenario* lihat rekomendasi dalam pengelolaan akun Twitter akan dijelaskan pada Tabel 4.10.

Tabel 4.10 Use Case Scenario Lihat Rekomendasi Dalam Pengelolaan Akun Twitter

Lihat rekomendasi dalam pengelolaan akun Twitter	
Kode Kebutuhan	MM-F-006
<i>Objective</i>	Pengguna dapat melihat rekomendasi dalam pengelolaan akun Twitter.
<i>Actor</i>	Pengguna
<i>Pre-condition</i>	Halaman <i>dashboard</i> ditampilkan.
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan pesan “Sebaiknya anda <i>posting</i> tweet pada pukul sesuai dengan jam yang frekuensinya sering muncul dalam <i>tweet</i> terbaik” 2. Sistem menampilkan pesan “menanggapi tren topik hari ini”.
<i>Alternative flow</i>	<ol style="list-style-type: none"> 1. Alternatif rekomendasi: <ol style="list-style-type: none"> 1.2. Jika pengguna telah melakukan <i>posting tweet</i> pada hari tersebut maka pesan “<i>posting tweet</i> hari ini” tidak ditampilkan. 1.3. Jika jumlah rata-rata <i>posting tweet</i> pengguna lebih banyak dari kompetitor maka pesan “sebaiknya anda <i>posting</i> sejumlah <i>tweet</i> lagi” tidak ditampilkan. 1.4. Jika pengguna telah menanggapi <i>mention</i> pada hari tersebut maka pesan “menanggapi <i>mention</i> masuk kepada anda” tidak ditampilkan.
<i>Post-condition</i>	Rekomendasi dalam pengelolaan akun Twitter ditampilkan.

4.7.7 Use Case Scenario Lihat Rata-Rata Peningkatan *Followers* Pengguna dan Kompetitor

Use case scenario lihat rata-rata peningkatan *followers* pengguna dan kompetitor akan menjelaskan alur dalam melihat rata-rata peningkatan *followers* pengguna dan kompetitor. Tujuan dari *use case scenario* lihat rata-rata peningkatan *followers* pengguna dan kompetitor adalah pengguna dapat melihat rata-rata peningkatan *followers* pengguna dan kompetitor. *Use case scenario* lihat rata-rata peningkatan *followers* pengguna dan kompetitor akan dijelaskan pada Tabel 4.11.

Tabel 4.11 Use Case Scenario Lihat Rata-Rata Peningkatan Followers Pengguna dan Kompetitor

Lihat rata-rata peningkatan <i>followers</i> pengguna dan kompetitor	
Kode Kebutuhan	MM-F-007
<i>Objective</i>	Pengguna dapat melihat rata-rata peningkatan <i>followers</i> pengguna dan kompetitor.
<i>Actor</i>	Pengguna
<i>Pre-condition</i>	Halaman <i>dashboard</i> ditampilkan.
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan rata-rata peningkatan <i>followers</i> pengguna. 2. Pengguna menekan <i>card</i>. 3. Sistem menampilkan rata-rata peningkatan <i>followers</i> kompetitor.
<i>Alternative flow</i>	<ol style="list-style-type: none"> 1. Kegagalan dalam menampilkan rata-rata peningkatan <i>followers</i> kompetitor: <ol style="list-style-type: none"> 1.1 Jika pengguna belum menambahkan kompetitor maka sistem akan menampilkan pesan “kompetitor tidak ditemukan”.
<i>Post-condition</i>	Rata-rata peningkatan <i>followers</i> pengguna dan kompetitor ditampilkan.

4.7.8 Use Case Scenario Lihat Rata-Rata Posting Tweet Pengguna dan Kompetitor

Use case scenario lihat rata-rata *posting tweet* pengguna dan kompetitor akan menjelaskan alur dalam melihat rata-rata *posting tweet* pengguna dan kompetitor. Tujuan dari *use case scenario* lihat rata-rata *posting tweet* pengguna dan kompetitor adalah pengguna dapat melihat rata-rata *posting tweet* pengguna dan kompetitor. *Use case scenario* lihat rata-rata *posting tweet* pengguna dan kompetitor akan dijelaskan pada Tabel 4.12.

Tabel 4.12 Use Case Scenario Lihat Rata-Rata Posting Tweet Pengguna dan Kompetitor

Lihat rata-rata <i>posting tweet</i> pengguna dan kompetitor	
Kode Kebutuhan	MM-F-008
<i>Objective</i>	Pengguna dapat melihat rata-rata <i>posting tweet</i> pengguna dan kompetitor.
<i>Actor</i>	Pengguna
<i>Pre-condition</i>	Halaman <i>dashboard</i> ditampilkan.

Tabel 4.12 Use Case Scenario Lihat Rata-Rata *Positing Tweet* Pengguna dan Kompetitor (Lanjutan)

Lihat rata-rata <i>posting tweet</i> pengguna dan kompetitor (Lanjutan)	
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan rata-rata <i>posting</i> pengguna. 2. Pengguna menekan <i>card</i>. Sistem menampilkan rata-rata <i>posting</i> kompetitor.
<i>Alternative flow</i>	<ol style="list-style-type: none"> 1. Kegagalan dalam menampilkan rata-rata <i>posting</i> kompetitor: <ol style="list-style-type: none"> 1.1. Jika pengguna belum menambahkan kompetitor maka sistem akan menampilkan pesan “kompetitor tidak ditemukan”.
<i>Post-condition</i>	Rata-rata <i>posting</i> pengguna dan kompetitor ditampilkan.

4.7.9 Use Case Scenario Lihat Rata-Rata *Retweet* Pengguna dan Kompetitor

Use case scenario lihat rata-rata *retweet* pengguna dan kompetitor akan menjelaskan alur dalam melihat rata-rata *retweet* pengguna dan kompetitor. Tujuan dari *use case scenario* lihat rata-rata *retweet* pengguna dan kompetitor adalah pengguna dapat melihat rata-rata *retweet* pengguna dan kompetitor. *Use case scenario* lihat rata-rata *retweet* pengguna dan kompetitor akan dijelaskan pada Tabel 4.13.

Tabel 4.13 Use Case Scenario Lihat Rata-Rata *Retweet* Pengguna dan Kompetitor

Lihat rata-rata <i>retweet</i> pengguna dan kompetitor	
Kode Kebutuhan	MM-F-009
<i>Objective</i>	Pengguna dapat melihat rata-rata <i>retweet</i> pengguna dan kompetitor.
<i>Actor</i>	Pengguna
<i>Pre-condition</i>	Halaman <i>dashboard</i> ditampilkan.
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan rata-rata <i>retweet</i> pengguna. 2. Pengguna menekan <i>card</i>. 3. Sistem menampilkan rata-rata <i>retweet</i> kompetitor.
<i>Alternative flow</i>	<ol style="list-style-type: none"> 1. Kegagalan dalam menampilkan rata-rata <i>retweet</i> kompetitor: <ol style="list-style-type: none"> 1.1 Jika pengguna belum menambahkan kompetitor maka sistem akan menampilkan pesan “kompetitor tidak ditemukan”.
<i>Post-condition</i>	Rata-rata <i>retweet</i> pengguna dan kompetitor ditampilkan.

4.7.10 Use Case Scenario Lihat Rata-Rata Likes Pengguna dan Kompetitor

Use case scenario lihat rata-rata *likes* pengguna dan kompetitor akan menjelaskan alur dalam melihat rata-rata *likes* pengguna dan kompetitor. Tujuan dari *use case scenario* lihat rata-rata *likes* pengguna dan kompetitor adalah pengguna dapat melihat rata-rata *likes* pengguna dan kompetitor. *Use case scenario* lihat rata-rata *likes* pengguna dan kompetitor akan dijelaskan pada Tabel 4.14.

Tabel 4.14 Use Case Scenario Lihat Rata-Rata Likes Pengguna dan Kompetitor

Lihat rata-rata <i>likes</i> pengguna dan kompetitor	
<i>Kode Kebutuhan</i>	MM-F-010
<i>Objective</i>	Pengguna dapat melihat rata-rata <i>likes</i> pengguna dan kompetitor.
<i>Actor</i>	Pengguna
<i>Pre-condition</i>	Halaman <i>dashboard</i> ditampilkan.
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan rata-rata <i>likes</i> pengguna. 2. Pengguna menekan <i>card</i>. 3. Sistem menampilkan rata-rata <i>likes</i> kompetitor.
<i>Alternative flow</i>	<ol style="list-style-type: none"> 1. Kegagalan dalam menampilkan rata-rata <i>likes</i> kompetitor: <ol style="list-style-type: none"> 1.1 Jika pengguna belum menambahkan kompetitor maka sistem akan menampilkan pesan “kompetitor tidak ditemukan”.
<i>Post-condition</i>	Rata-rata <i>likes</i> pengguna dan kompetitor ditampilkan.

4.7.11 Use Case Scenario Lihat Rata-Rata Replies Pengguna dan Kompetitor

Use case scenario lihat rata-rata *replies* pengguna dan kompetitor akan menjelaskan alur dalam melihat rata-rata *replies* pengguna dan kompetitor. Tujuan dari *use case scenario* lihat rata-rata *replies* pengguna dan kompetitor adalah pengguna dapat melihat rata-rata *replies* pengguna dan kompetitor. *Use case scenario* lihat rata-rata *replies* pengguna dan kompetitor akan dijelaskan pada Tabel 4.15.

Tabel 4.15 Use Case Scenario Lihat Rata-Rata Replies Pengguna dan Kompetitor

Lihat rata-rata <i>replies</i> pengguna dan kompetitor	
Kode Kebutuhan	MM-F-011
<i>Objective</i>	Pengguna dapat melihat rata-rata <i>replies</i> pengguna dan kompetitor.
<i>Actor</i>	Pengguna
<i>Pre-condition</i>	Halaman <i>dashboard</i> ditampilkan.
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan rata-rata <i>replies</i> pengguna. 2. Pengguna menekan <i>card</i>. 3. Sistem menampilkan rata-rata <i>replies</i> kompetitor.
<i>Alternative flow</i>	<ol style="list-style-type: none"> 1. Kegagalan dalam menampilkan rata-rata <i>replies</i> kompetitor: <ol style="list-style-type: none"> 1.1 Jika pengguna belum menambahkan kompetitor maka sistem akan menampilkan pesan “kompetitor tidak ditemukan”.
<i>Post-condition</i>	Rata-rata <i>replies</i> pengguna dan kompetitor ditampilkan.

4.7.12 Use Case Scenario Lihat Hasil Sentimen Analisis

Use case scenario lihat hasil sentimen analisis akan menjelaskan alur dalam melihat hasil sentimen analisis. Tujuan dari *use case scenario* lihat hasil sentimen adalah pengguna dapat melihat hasil sentimen analisis. *Use case scenario* lihat hasil sentimen analisis akan dijelaskan pada Tabel 4.16.

Tabel 4.16 Use Case Scenario Lihat Hasil Sentimen Analisis

Lihat hasil sentimen analisis	
Kode Kebutuhan	MM-F-012
<i>Objective</i>	Pengguna dapat melihat hasil sentimen analisis.
<i>Actor</i>	Pengguna
<i>Pre-condition</i>	Halaman <i>dashboard</i> ditampilkan.
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan <i>pie chart</i> hasil analisis sentimen <i>replies</i> terhadap pengguna dan 5 sentimen positif negatif.
<i>Alternative flow</i>	<ol style="list-style-type: none"> 1. Kegagalan dalam menampilkan sentimen analisis: <ol style="list-style-type: none"> 1.1 Jika data sentimen belum diambil maka sistem akan menampilkan pesan “Mohon maaf data sentimen belum diambil, silahkan coba sesaat lagi!”.
<i>Post-condition</i>	Hasil analisis sentimen ditampilkan.

4.7.13 Use Case Scenario Lihat Tweet Terbaik

Use case scenario lihat tweet terbaik akan menjelaskan alur dalam melihat tweet terbaik. Tujuan dari *use case scenario* lihat tweet terbaik adalah Pengguna dapat melihat tweet terbaik selama seminggu terakhir. *Use case scenario* lihat tweet terbaik akan dijelaskan pada Tabel 4.17.

Tabel 4.17 Use Case Scenario Lihat Tweet Terbaik

Lihat tweet terbaik	
Kode Kebutuhan	MM-F-013
<i>Objective</i>	Pengguna dapat melihat tweet terbaik.
<i>Actor</i>	Pengguna
<i>Pre-condition</i>	Halaman <i>dashboard</i> ditampilkan.
<i>Main flow</i>	1. Sistem menampilkan 5 tweet terbaik pengguna.
<i>Alternative flow</i>	1. Kegagalan dalam menampilkan tweet terbaik: 1.1 Jika data <i>tweet</i> pengguna belum diambil maka sistem akan menampilkan pesan “Mohon maaf data tweet terbaik belum diambil, silahkan coba sesaat lagi!”.
<i>Post-condition</i>	Tweet terbaik pengguna ditampilkan.

4.7.14 Use Case Scenario Mengelola Daftar Kompetitor

Use case scenario mengelola daftar kompetitor akan menjelaskan alur dalam mengelola daftar kompetitor. Tujuan dari *use case scenario* mengelola daftar kompetitor adalah pengguna dapat mengelola daftar kompetitor. *Use case scenario* mengelola daftar kompetitor akan dijelaskan pada Tabel 4.18.

Tabel 4.18 Use Case Scenario Mengelola Daftar Kompetitor

Mengelola daftar kompetitor	
Kode Kebutuhan	MM-F-014
<i>Objective</i>	Pengguna dapat mengelola daftar kompetitor.
<i>Actor</i>	Pengguna
<i>Pre-condition</i>	Halaman <i>competitor</i> ditampilkan.
<i>Main flow</i>	1. Sistem menampilkan daftar kompetitor yang telah ditambahkan pengguna sebelumnya.

Tabel 4.18 Use Case Scenario Mengelola Daftar Kompetitor (Lanjutan)

Mengelola daftar kompetitor (Lanjutan)	
<i>Alternative flow</i>	1. Kegagalan dalam menampilkan daftar kompetitor: 1.1 Jika pengguna belum menambahkan kompetitor maka sistem akan menampilkan pesan “kompetitor tidak ditemukan”.
<i>Post-condition</i>	Daftar kompetitor ditampilkan.

4.7.15 Use Case Scenario Tambah Kompetitor

Use case scenario tambah kompetitor akan menjelaskan alur dalam menambah kompetitor. Tujuan dari *use case scenario* tambah kompetitor adalah pengguna dapat menambah daftar kompetitor. *Use case scenario* tambah kompetitor akan dijelaskan pada Tabel 4.19.

Tabel 4.19 Use Case Scenario Tambah Kompetitor

Tambah kompetitor	
Kode Kebutuhan	MM-F-01
<i>Objective</i>	Pengguna dapat menambah daftar kompetitor.
<i>Actor</i>	Pengguna
<i>Pre-condition</i>	Halaman <i>competitor</i> ditampilkan.
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pengguna menekan tombol “+” 2. Halaman pencarian akun Twitter ditampilkan 3. Pengguna memasukkan <i>username</i> Twitter lalu menekan tombol pencarian. 4. Sistem menampilkan hasil pencarian. 5. Pengguna menekan tombol “Tambahkan kompetitor”. 6. Sistem menampilkan <i>pop-up</i> konfirmasi dalam menambahkan kompetitor. 7. Pengguna menekan tombol “Ya”. 8. Sistem menampilkan daftar kompetitor.
<i>Alternative flow</i>	<ol style="list-style-type: none"> 1. Kegagalan dalam menambahkan kompetitor: 1.1 Jika akun hasil pencarian sebelumnya telah ditambahkan sebagai kompetitor, maka tombol “Tambahkan kompetitor” akan berubah menjadi keterangan “X Kompetitor”.
<i>Post-condition</i>	Daftar kompetitor ditambahkan kedalam <i>database</i> sistem.

4.7.16 Use Case Scenario Hapus Kompetitor

Use case scenario hapus kompetitor akan menjelaskan alur dalam menghapus kompetitor. Tujuan dari *use case scenario* hapus kompetitor adalah pengguna dapat menghapus kompetitor. *Use case scenario* hapus kompetitor akan dijelaskan pada Tabel 4.20.

Tabel 4.20 Use Case Scenario Hapus Kompetitor

Hapus kompetitor	
Kode Kebutuhan	MM-F-016
<i>Objective</i>	Pengguna dapat menghapus kompetitor.
<i>Actor</i>	Pengguna
<i>Pre-condition</i>	Halaman <i>competitor</i> ditampilkan.
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pengguna menekan tombol "Hapus". 2. Sistem menampilkan <i>pop-up</i> konfirmasi menghapus kompetitor. 3. Pengguna menekan tombol "ya" untuk mengkonfirmasi menghapus kompetitor. 4. Sistem menampilkan kembali daftar kompetitor.
<i>Alternative flow</i>	-
<i>Post-condition</i>	Data kompetitor dihapus dari <i>database</i> .

4.7.17 Use Case Scenario Lihat Detail Perbandingan Aktivitas Pengguna dan Kompetitor

Use case scenario lihat detail perbandingan aktivitas pengguna dan kompetitor akan menjelaskan alur dalam melihat detail perbandingan aktivitas pengguna dan kompetitor. Tujuan dari *use case scenario* lihat detail perbandingan aktivitas pengguna dan kompetitor adalah pengguna dapat melihat detail perbandingan aktivitas pengguna dan kompetitor. *Use case scenario* lihat detail perbandingan aktivitas pengguna dan kompetitor akan dijelaskan pada Tabel 4.21.

Tabel 4.21 Use Case Scenario Lihat Detail Perbandingan Aktivitas Pengguna dan Kompetitor

Lihat detail perbandingan aktivitas pengguna dan kompetitor	
Kode Kebutuhan	MM-F-017
<i>Objective</i>	Pengguna melihat detail perbandingan aktivitas pengguna dan kompetitor
<i>Actor</i>	Pengguna
<i>Pre-condition</i>	Halaman <i>competitor</i> ditampilkan.

Tabel 4.21 Use Case Scenario Lihat Detail Perbandingan Aktivitas Pengguna dan Kompetitor (Lanjutan)

Lihat detail perbandingan aktivitas pengguna dan kompetitor (Lanjutan)	
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pengguna menekan tombol “detail”. 2. Sistem menampilkan halaman detail perbandingan aktivitas pengguna dan kompetitor. 3. Sistem menampilkan profil Twitter kompetitor seperti nama, <i>username</i>, foto profil, foto <i>banner</i>, <i>bio</i>, jumlah tweet, jumlah <i>followers</i>, jumlah <i>following</i>, lokasi, dan tanggal akun Twitter kompetitor dibuat. 4. Sistem menampilkan tabel komparasi performa dengan menampilkan perbandingan peningkatan <i>followers</i>, <i>posting tweet</i>, total <i>retweet</i>, total <i>likes</i>, dan total <i>replies</i>. 5. Sistem menampilkan detail grafik komparasi dengan menampilkan grafik perbandingan peningkatan <i>followers</i>, jumlah tweet, jumlah retweet, jumlah likes, dan jumlah replies.
<i>Alternative flow</i>	-
<i>Post-condition</i>	Halaman detail perbandingan aktivitas pengguna dan kompetitor ditampilkan.

4.7.18 Use Case Scenario Posting Tweet

Use case scenario posting tweet akan menjelaskan alur dalam melakukan *posting tweet*. Tujuan dari *use case scenario posting tweet* adalah pengguna dapat melakukan *posting tweet* melalui sistem. *Use case scenario posting tweet* akan dijelaskan pada Tabel 4.22.

Tabel 4.22 Use Case Scenario Posting Tweet

<i>Posting Tweet</i>	
Kode Kebutuhan	MM-F-018
<i>Objective</i>	Pengguna dapat <i>posting tweet</i> .
<i>Actor</i>	Pengguna
<i>Pre-condition</i>	Pengguna telah login.
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>icon</i> Twitter pada <i>navbar</i>. 2. Sistem menampilkan pop-up <i>posting tweet</i>. 3. Pengguna menulis <i>tweet</i>. 4. Pengguna menekan tombol <i>tweet</i>. 5. Sistem menutup <i>pop-up posting tweet</i>.

Tabel 4.22 Use Case Scenario Posting Tweet (Lanjutan)

Posting Tweet (Lanjutan)	
<i>Alternative flow</i>	-
<i>Post-condition</i>	<i>Tweet pengguna diposting.</i>

4.7.19 Use Case Scenario Lihat Tren Topik Indonesia

Use case scenario lihat tren topik Indonesia akan menjelaskan alur dalam melihat tren topik Indonesia. Tujuan dari *use case scenario* lihat tren topik Indonesia adalah pengguna dapat melihat daftar tren topik Indonesia. *Use case scenario* lihat tren topik Indonesia akan dijelaskan pada Tabel 4.23.

Tabel 4.23 Use Case Scenario Lihat Tren Topik Indonesia

Lihat Tren Topik Indonesia	
Kode Kebutuhan	MM-F-019
<i>Objective</i>	Pengguna dapat melihat daftar tren topik Indonesia.
<i>Actor</i>	Pengguna
<i>Pre-condition</i>	Halaman <i>trend</i> ditampilkan.
<i>Main flow</i>	1. Sistem menampilkan 10 daftar tren topik Indonesia.
<i>Alternative flow</i>	-
<i>Post-condition</i>	Tren topik Indonesia ditampilkan

4.7.20 Use Case Scenario Lihat Contoh Tweet Berdasarkan Tren

Use case scenario lihat contoh *tweet* berdasarkan tren akan menjelaskan alur dalam melihat contoh *tweet* berdasarkan tren. Tujuan dari *use case scenario* lihat contoh *tweet* berdasarkan tren adalah pengguna dapat melihat contoh *tweet* berdasarkan tren yang dipilih. *Use case scenario* lihat contoh *tweet* berdasarkan tren akan dijelaskan pada Tabel 4.24.

Tabel 4.24 Use Case Scenario Lihat Contoh Tweet Berdasarkan Tren

Lihat Contoh <i>Tweet</i> Berdasarkan Tren	
Kode Kebutuhan	MM-F-02
<i>Objective</i>	Pengguna dapat melihat contoh <i>tweet</i> berdasarkan tren yang dipilih.
<i>Actor</i>	Pengguna
<i>Pre-condition</i>	Halaman <i>trend</i> ditampilkan.
<i>Main flow</i>	1. Pengguna menekan tombol detail pada salah satu tren.

Tabel 4.24 Use Case Scenario Lihat Contoh Tweet Berdasarkan Tren (Lanjutan)

Lihat Contoh Tweet Berdasarkan Tren (Lanjutan)	
	2. Sistem mengeluarkan <i>modal pop-up</i> untuk menampilkan contoh <i>tweet</i> berdasarkan tren pilihan pengguna.
<i>Alternative flow</i>	-
<i>Post-condition</i>	Contoh <i>tweet</i> berdasarkan tren ditampilkan.

4.7.21 Use Case Scenario Cari Akun Twitter

Use case scenario cari akun Twitter akan menjelaskan alur dalam melakukan pencarian akun Twitter. Tujuan dari *use case scenario* cari akun Twitter adalah pengguna dapat mencari akun Twitter. *use case scenario* cari akun Twitter akan dijelaskan pada Tabel 4.25.

Tabel 4.25 Use Case Scenario Cari Akun Twitter

Cari akun Twitter	
Kode Kebutuhan	MM-F-021
<i>Objective</i>	Pengguna dapat mencari akun Twitter.
<i>Actor</i>	Pengguna
<i>Pre-condition</i>	Halaman pencarian akun Twitter ditampilkan.
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pengguna memasukan <i>username</i> Twitter. 2. Pengguna menekan tombol pencarian. 3. Sistem menampilkan hasil pencarian
<i>Alternative flow</i>	<ol style="list-style-type: none"> 1. Kegagalan dalam mencari akun Twitter: <ol style="list-style-type: none"> 1.1 Jika akun hasil pencarian tidak ditemukan, maka sistem menampilkan pesan “akun tidak ditemukan”
<i>Post-condition</i>	Hasil pencarian akun Twitter ditampilkan.

4.7.22 Use Case Scenario Update Data

Use case scenario update data akan menjelaskan alur dalam melakukan proses *update data*. Tujuan dari *use case scenario update data* adalah pengguna dapat melakukan proses *update data* melalui sistem. *Use case scenario update data* akan dijelaskan pada Tabel 4.26.

Tabel 4.26 Use Case Scenario Update Data

Update Data	
Kode Kebutuhan	MM-F-022
<i>Objective</i>	Pengguna dapat melakukan proses <i>update data</i> melalui sistem.

Tabel 4.26 Use Case Scenario Update Data (Lanjutan)

Update Data (Lanjutan)	
<i>Actor</i>	Pengguna
<i>Pre-condition</i>	Pengguna telah <i>login</i> .
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>update data</i>. 2. Sistem melakukan <i>update data</i> akun Twitter pengguna dengan memanggil <i>script</i> Python <i>GetAccountData.py</i>. 3. Sistem melakukan <i>update data</i> sentimen pengguna dengan memanggil <i>script</i> Python <i>GetSentimentData.py</i>. 4. Sistem melakukan <i>update data</i> <i>tweet</i> terbaik pengguna dengan memanggil <i>script</i> Python <i>GetBestTweet.py</i>. 5. Sistem melakukan <i>update data</i> tren topik Indonesia dengan memanggil <i>script</i> Python <i>GetTrendingTopic.py</i>.
<i>Alternative flow</i>	<ol style="list-style-type: none"> 1. Sistem mendeteksi akun belum melakukan <i>update data</i>: <ol style="list-style-type: none"> 1.1 Jika sistem mendeteksi akun belum melakukan <i>update data</i> pada hari tersebut, maka sistem akan secara otomatis melakukan proses <i>update data</i> tanpa pengguna perlu menekan tombol <i>update data</i>.
<i>Post-condition</i>	Data aktivitas akun Twitter pengguna berhasil di <i>update</i> .

4.7.23 Use Case Scenario Logout

Use case scenario logout akan menjelaskan alur dalam melakukan proses *logout*. Tujuan dari *use case scenario logout* adalah pengguna dapat keluar dari sistem. *Use case scenario logout* akan dijelaskan pada Tabel 4.27.

Tabel 4.27 Use Case Scenario Logout

Logout	
Kode Kebutuhan	MM-F-023
<i>Objective</i>	Pengguna dapat keluar dari sistem.
<i>Actor</i>	Pengguna
<i>Pre-condition</i>	Pengguna telah <i>login</i> .
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>logout</i>. 2. Sistem menampilkan halaman <i>login</i>.
<i>Alternative flow</i>	-
<i>Post-condition</i>	Pengguna keluar dari sistem.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan Sistem

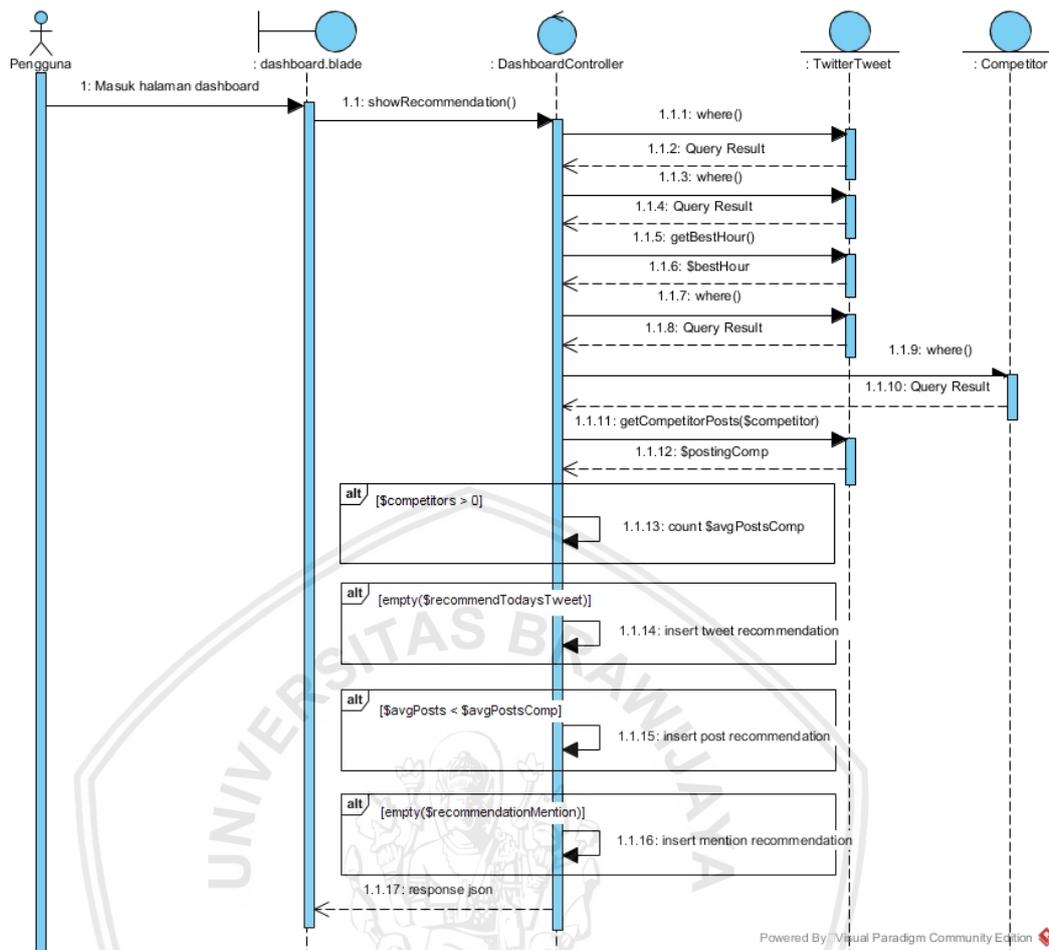
Setelah menyelesaikan tahap analisis kebutuhan maka dilanjutkan dengan tahapan berikutnya yaitu perancangan. Terdapat beberapa perancangan yang dilakukan pada tahapan ini yaitu perancangan arsitektur sistem, perancangan data, perancangan komponen, dan perancangan antarmuka. Perancangan arsitektur sistem akan menjelaskan mengenai *sequence diagram* dan *class diagram* sistem. Perancangan komponen menjelaskan perancangan algoritme yang merupakan gambaran proses logika yang dikerjakan sistem. Perancangan data menjelaskan tentang perancangan *database* yang digambarkan melalui *Entity Relationship Diagram* (ERD) dan *Physical Data Model* (PDM). Perancangan antarmuka menggambarkan antarmuka yang akan digunakan dan diimplementasikan pada sistem.

5.1.1 Perancangan Arsitektur Sistem

Pada bagian perancangan arsitektur sistem akan dijelaskan mengenai *sequence diagram* untuk menggambarkan interaksi antar objek yang bekerja pada sistem. Terdapat tiga *sequence diagram* yang akan dijelaskan pada bagian ini yaitu proses lihat rekomendasi, lihat hasil sentimen analisis, dan lihat tren topik Indonesia. Selain itu terdapat penjelasan mengenai *class diagram* sistem untuk menunjukkan *class* pembentuk sistem dan hubungannya satu sama lain.

5.1.1.1 *Sequence Diagram* Lihat Rekomendasi Dalam Pengelolaan Akun Twitter

Pada Gambar 5.1 *sequence diagram* lihat rekomendasi dalam pengelolaan akun Twitter, terdapat 4 objek yang saling berinteraksi yaitu *boundary dashboard.blade*, *controller DashboardController*, dan dua *entity* yaitu *TwitterTweet* dan *Competitor*. Dalam melihat rekomendasi, pengguna masuk pada halaman *dashboard* lalu sistem akan memanggil fungsi *showRecommendation()* yang terletak pada *class DashboardController*. Pada fungsi *showRecommendation()* sistem akan mengambil data *posting tweet* pengguna pada hari tersebut melalui *query* pada *class TwitterTweet* lalu dilanjutkan dengan mengambil data *mention* pada hari tersebut melalui *query* pada *class TwitterTweet* dilanjutkan dengan mengambil data jam terbaik *posting* pengguna melalui *method getBestHour()* dari *class TwitterTweet* dilanjutkan dengan mengambil data *tweet* pengguna selama seminggu terakhir melalui *query* pada *class TwitterTweet* dilanjutkan dengan mengambil data *competitor* melalui *query* pada *class Competitor* dan terakhir mengambil *posting competitor* melalui *method getCompetitorPosts()* dari *class TwitterTweet* setelah itu *method showRecommendation()* melakukan pemrosesan data dan mengembalikan hasil pemrosesan data dalam bentuk JSON pada *dashboard.blade*.

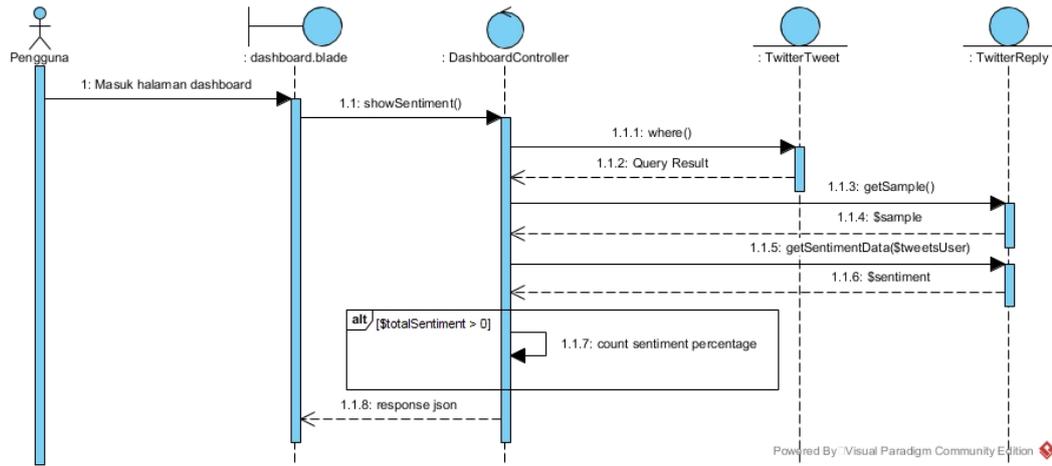


Gambar 5.1 Sequence Diagram Lihat Rekomendasi Dalam Pengelolaan Akun Twitter

5.1.1.2 Sequence Diagram Lihat Hasil Sentimen Analisis

Pada Gambar 5.2 *sequence diagram* lihat hasil sentimen analisis, terdapat 4 objek yang saling berinteraksi yaitu *boundary dashboard.blade*, *controller DashboardController* dan dua *entity* yaitu *TwitterTweet* dan *TwitterReply*. Dalam melihat hasil sentimen analisis pengguna masuk pada halaman *dashboard.blade* lalu sistem memanggil *method showSentiment* pada *class DashboardController*. Selanjutnya mengambil data *tweet* pengguna melalui *query* pada *class TwitterTweet* dilanjutkan dengan mengambil data *sample* sentimen melalui *method getSample* pada *class TwitterReply* lalu dilanjutkan mengambil data jumlah sentimen melalui *method getSentimentData(\$tweetsUser)* setelah itu pada *method showSentiment()* dilakukan pemrosesan data dan mengembalikan hasil pemrosesan data dalam bentuk JSON pada *dashboard.blade*.

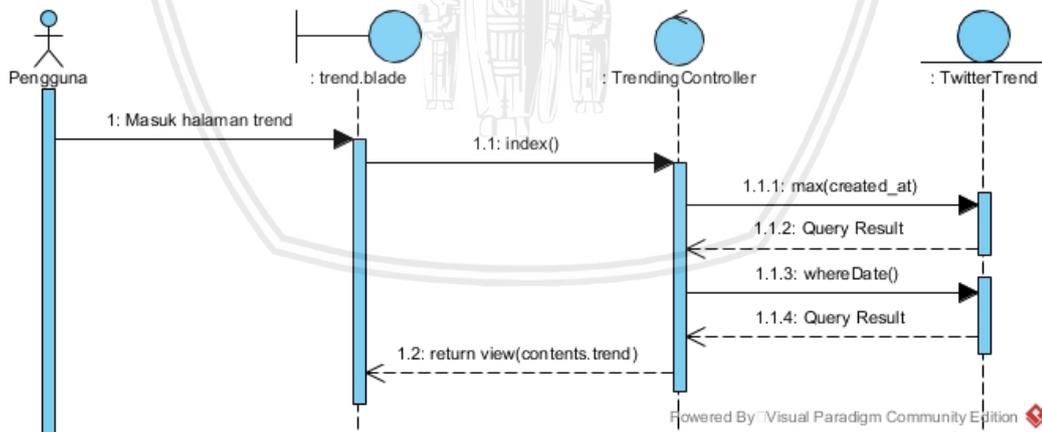




Gambar 5.2 Sequence Diagram Lihat Hasil Sentimen Analisis

5.1.1.3 Sequence Diagram Lihat Tren Topik Indonesia

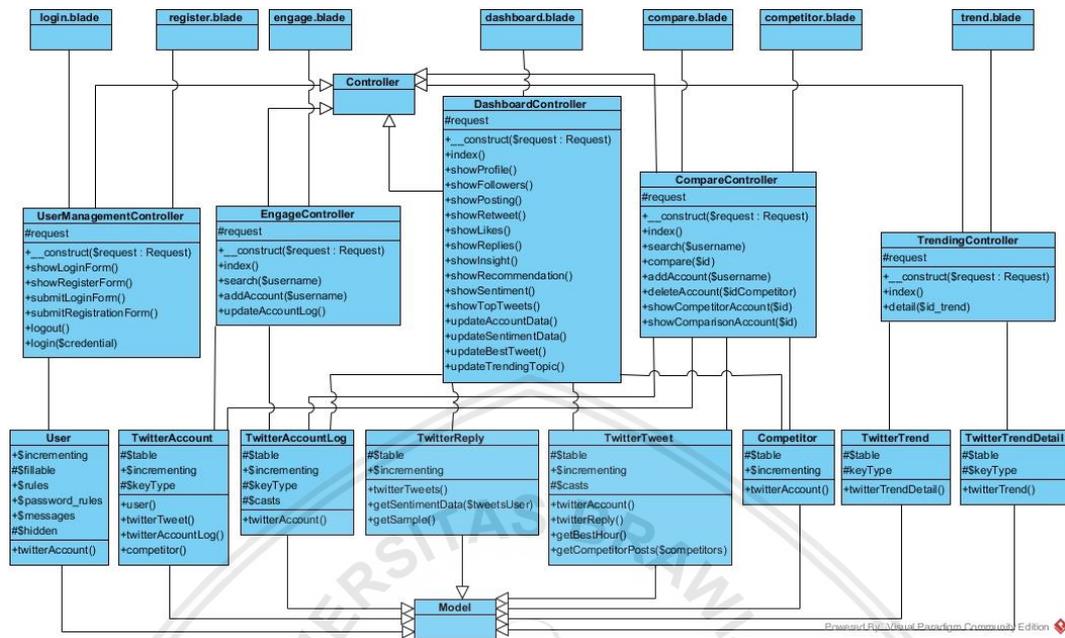
Pada Gambar 5.3 *sequence diagram* lihat tren topik Indonesia, terdapat 3 objek yang saling berinteraksi yaitu *boundary trend.blade*, *controller TrendingController*, dan *entity yaitu TwitterTrend*. Dalam melihat daftar tren topik Indonesia, pengguna masuk pada halaman *trend* lalu sistem akan memanggil fungsi *index()* yang terletak pada *class TrendingController*. Pada fungsi *index()* sistem akan mengambil daftar tren topik yang terakhir ditambahkan pada *database* melalui *entity TwitterTrend*. Terakhir sistem akan menampilkan daftar tren tersebut pada halaman *trend*.



Gambar 5.3 Sequence Diagram Lihat Tren Topik Indonesia



5.1.1.4 Class Diagram

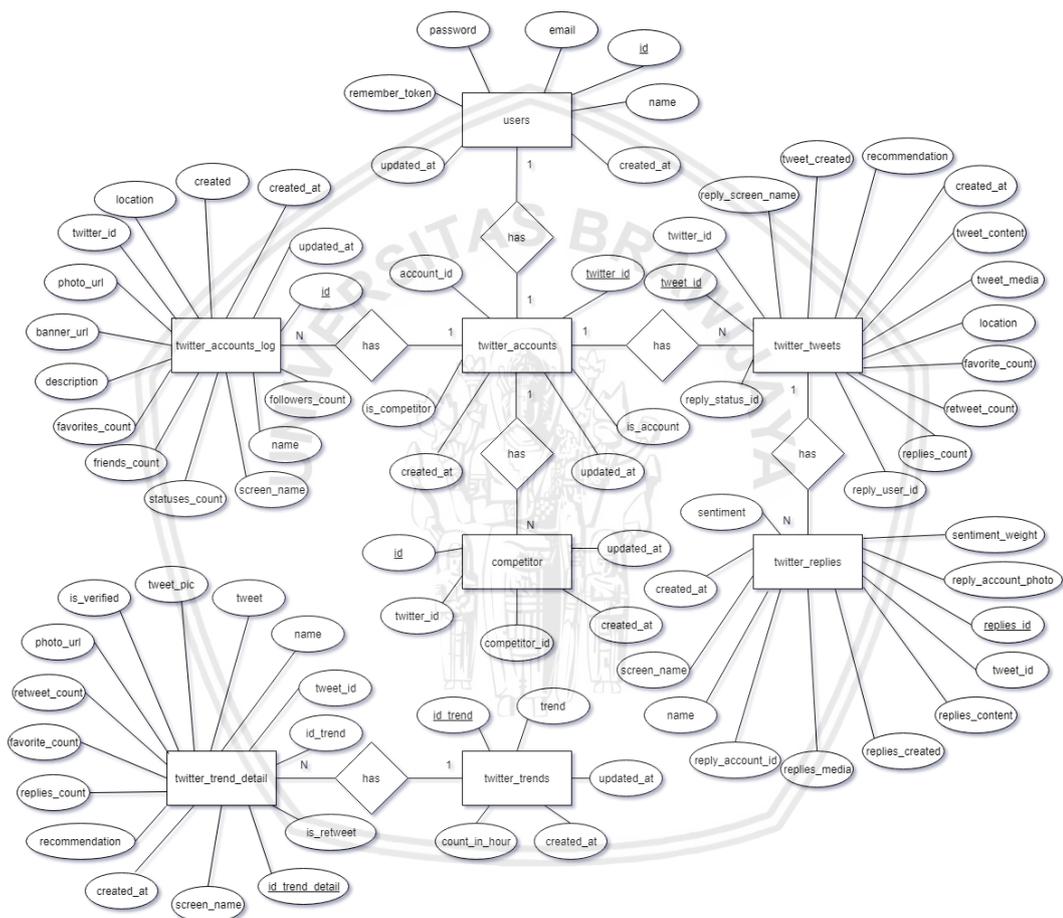


Gambar 5.4 Class Diagram Sistem

Pada Gambar 5.4 menjelaskan tentang *class diagram* sistem yang didalamnya terdapat 5 *class controller* yaitu *UserManagementController* yang berfungsi untuk mengatur aktivitas otorisasi tamu, *EngageController* yang berfungsi untuk mengatur aktivitas dalam menetapkan akun Twitter pengguna, *DashboardController* yang berfungsi untuk mengatur dalam menampilkan rekomendasi serta aktivitas media sosial pengguna, *CompareController* yang berfungsi untuk mengatur dalam proses komparasi aktivitas pengguna dengan kompetitor, dan *TrendingController* yang berfungsi untuk mengatur dalam menampilkan tren topik Indonesia. Selain itu juga terdapat 8 *class model* yaitu *class User* yang berfungsi untuk mengatur alur data akun sistem moment, *class TwitterAccount* yang berfungsi mengatur alur data akun media sosial Twitter, *class TwitterAccountLog* yang berfungsi mengatur alur data *log* akun Twitter pengguna, *class TwitterTweet* yang berfungsi mengatur alur data *tweet* pengguna dan kompetitor, *class TwitterReply* yang berfungsi mengatur alur data *reply* terhadap pengguna, *class TwitterTrend* yang berfungsi untuk mengatur data tren topik Indonesia, dan *class TwitterTrendDetail* yang berfungsi untuk mengatur data detail setiap tren topik Indonesia. Terakhir, terdapat 7 *class view* yaitu *class login.blade*, *register.blade*, *engage.blade*, *dashboard.blade*, *compare.blade*, *competitor.blade*, dan *trend.blade*.

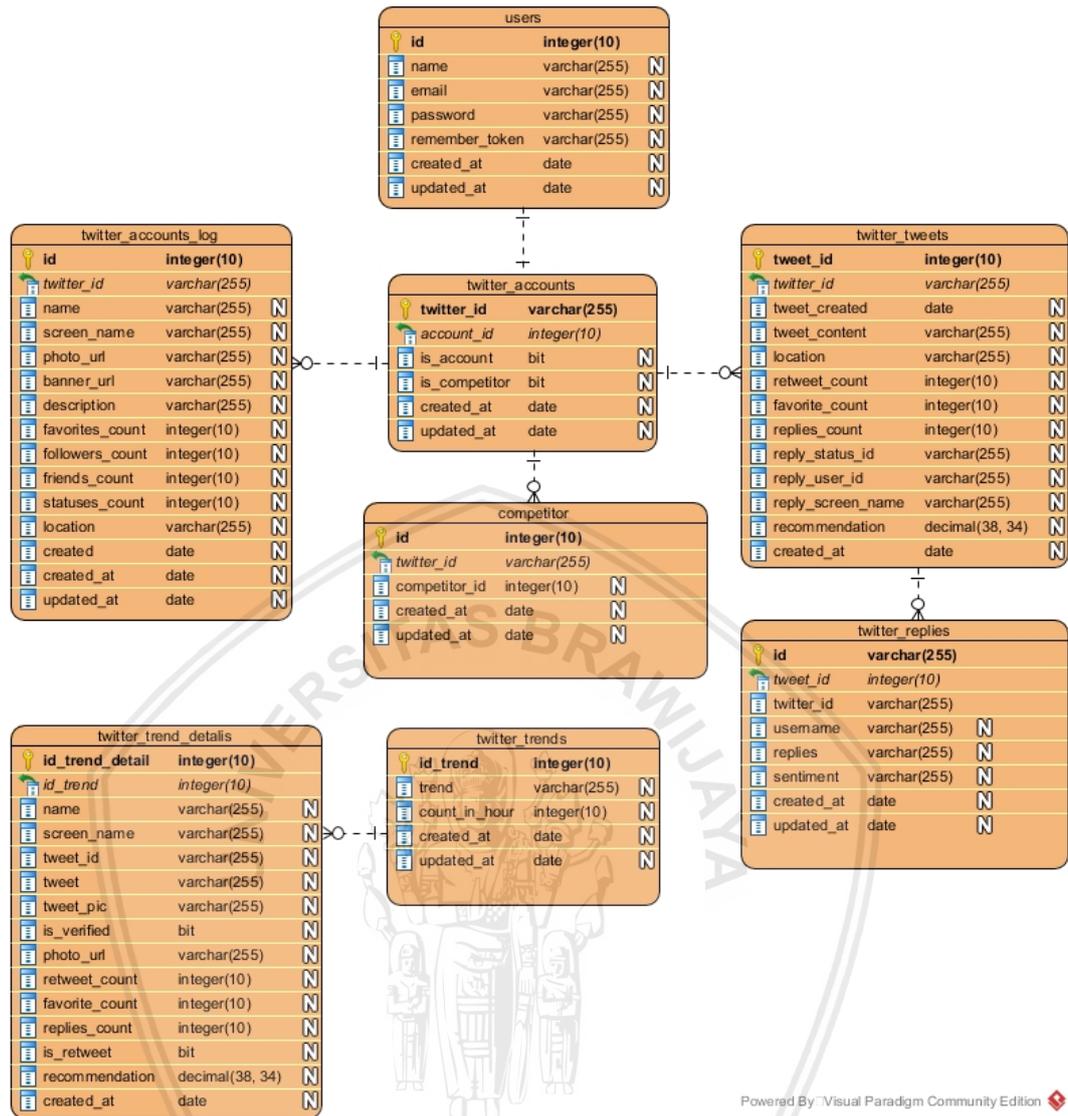
5.1.2 Perancangan Data

Pada bagian ini akan menjelaskan perancangan data yang akan digambarkan melalui *Entity Relationship Diagram* (ERD). Gambar 5.5 menunjukkan ERD sistem untuk memonitor aktivitas media sosial, terdapat 8 entitas yaitu *users*, *twitter_accounts*, *twitter_accounts_log*, *twitter_tweets*, *twitter_replies*, *competitor*, *twitter_trends*, *twitter_trend_detail*. Diantara entitas tersebut terdapat hubungan satu dengan lainnya untuk mempermudah proses pengambilan serta pertukaran data.



Gambar 5.5 Entity Relationship Diagram

Perancangan *Physical Data Model* (PDM) dibuat sebagai perancangan dalam membuat *database*. Jumlah tabel pada PDM harus sesuai dengan jumlah entitas pada perancangan ERD. Perancangan PDM ditunjukkan pada Gambar 5.6.



Gambar 5.6 Physical Data Model

5.1.3 Perancangan Komponen

Pada perancangan komponen akan dijelaskan mengenai komponen algoritme penyusun sistem, perancangan tersebut akan menggambarkan logika dasar dalam menyelesaikan suatu proses dalam sistem. Terdapat tiga perancangan algoritme yang akan dijelaskan pada bagian ini yaitu perancangan algoritme dalam melihat rekomendasi, lihat hasil sentimen analisis, dan melihat daftar tren topik Indonesia.

5.1.3.1 Perancangan Algoritme Lihat Rekomendasi Dalam Pengelolaan Akun Media Sosial

Pada Tabel 5.1 menunjukkan perancangan Algoritme dalam melihat rekomendasi pengelolaan akun media sosial. Dimulai dengan sistem mengambil mengambil data *posting tweet*, *mention*, *jam posting* terbaik pengguna pada hari tersebut. Lalu data *tweet* pengguna selama seminggu terakhir, menghitung jumlah *tweet* pengguna selama seminggu terakhir. Dilanjutkan dengan mengambil data



kompetitor beserta data *posting tweet* dari kompetitor tersebut. Jika pengguna belum melakukan *posting tweet* pada hari tersebut maka sistem akan menampilkan pesan rekomendasi untuk melakukan *posting*. Selanjutnya sistem membandingkan jumlah rata-rata *posting tweet* pengguna dan kompetitor, jika jumlah rata-rata *posting tweet* kompetitor lebih banyak maka sistem akan menampilkan pesan rekomendasi untuk melakukan *posting tweet* sebanyak selesih jumlah rata-rata *posting tweet* pengguna dan kompetitor ditambah satu. Terakhir jika pengguna belum menanggapi *mention* maka sistem akan menampilkan pesan rekomendasi untuk menanggapi *mention* pada hari tersebut.

Tabel 5.1 Perancangan Algoritme Lihat Rekomendasi Dalam Pengelolaan Akun Media Sosial

Perancangan algoritme lihat rekomendasi dalam pengelolaan akun media sosial	
1	Declaration variables avgPostsComp,
2	tweetRecommendation, postingRecommendation,
3	mentionRecommendation
4	Get user's today tweet data
5	Get user's today mention data
6	Get user's best hour to post tweet
7	Get user's tweet data in a week
8	Count user's tweet data
9	Get user's competitor
10	Get competitor's post
11	count user's average posting per week
12	if count competitor > 0
13	count competitor's average posting per week
14	endif
15	if user's today tweet data is empty
16	initiate tweetRecommendation = "posting tweet hari ini"
17	endif
18	if user's average posting < competitor's average posting
19	initiate postRecommendation = competitor's
20	average posting - user's average posting
21	initiate postingRecommendation = "Sebaiknya
22	hari ini anda post + postRecommendation + tweet lagi"
23	endif
24	if user's today mention data is empty
25	initiate mentionRecommendation = "Menanggapi mention
26	masuk kepada anda"
27	endif
28	return response json

5.1.3.2 Perancangan Algoritme Lihat Hasil Sentimen Analisis

Pada Tabel 5.2 menunjukan perancangan algoritme dalam melihat hasil sentimen analisis. Dimulai dengan mengambil data *tweet* pengguna, contoh sentimen, dan data sentimen setelah itu menghitung jumlah total sentimen. Jika total sentimen lebih dari 0 maka sistem menghitung nilai persen setiap sentimen. Terakhir sistem mengembalikan respon dalam bentuk JSON.



Tabel 5.2 Perancangan Algoritme Lihat Hasil Sentimen Analisis

Perancangan algoritme lihat hasil sentimen analisis	
1	Get user's tweet
2	Get sentiment sample
3	Get sentiment data
4	Count total sentiment
5	If total sentiment > 0
6	Count sentiment percentage
7	End if
8	Return response json

5.1.3.3 Perancangan Algoritme Lihat Tren Topik Indonesia

Pada Tabel 5.3 menunjukkan perancangan algoritme dalam melihat tren topik Indonesia. Dimulai dengan mengambil data tanggal tren terakhir *diupdate* lalu mengambil data tren berdasarkan tanggal tersebut. Terakhir data tren yang telah diambil akan ditampilkan pada halaman *trend*.

Tabel 5.3 Perancangan Algoritme Lihat Tren Topik Indonesia

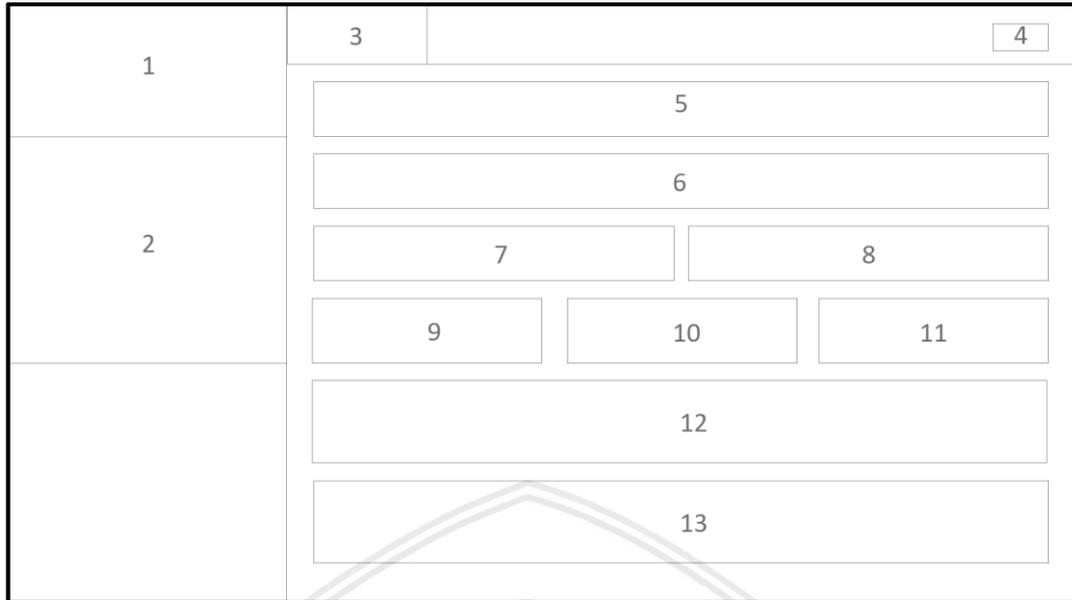
Perancangan algoritme lihat tren topik Indonesia	
1	Get trend latest created date
2	Get twitter trend based on latest created date
3	Return view trend

5.1.4 Perancangan Antarmuka

Pada bagian ini akan menjelaskan mengenai perancangan antarmuka sistem yang digambarkan melalui perancangan *layout*, perancangan ini akan menjadi acuan implementasi antarmuka sistem yang akan digunakan pengguna untuk dapat berinteraksi dengan sistem. Terdapat tiga perancangan antarmuka yang akan dijelaskan pada bagian ini yaitu perancangan antarmuka halaman *dashboard*, kompetitor, dan tren.

5.1.4.1 Perancangan Antarmuka Halaman Dashboard

Perancangan antarmuka halaman *dashboard* dapat dilihat pada Gambar 5.7. Terdapat 13 komponen pada perancangan ini dengan penjelasannya dapat dilihat pada Tabel 5.4.



Gambar 5.7 Perancangan Antarmuka Halaman *Dashboard*

Tabel 5.4 Keterangan Perancangan Antarmuka Halaman *Dashboard*

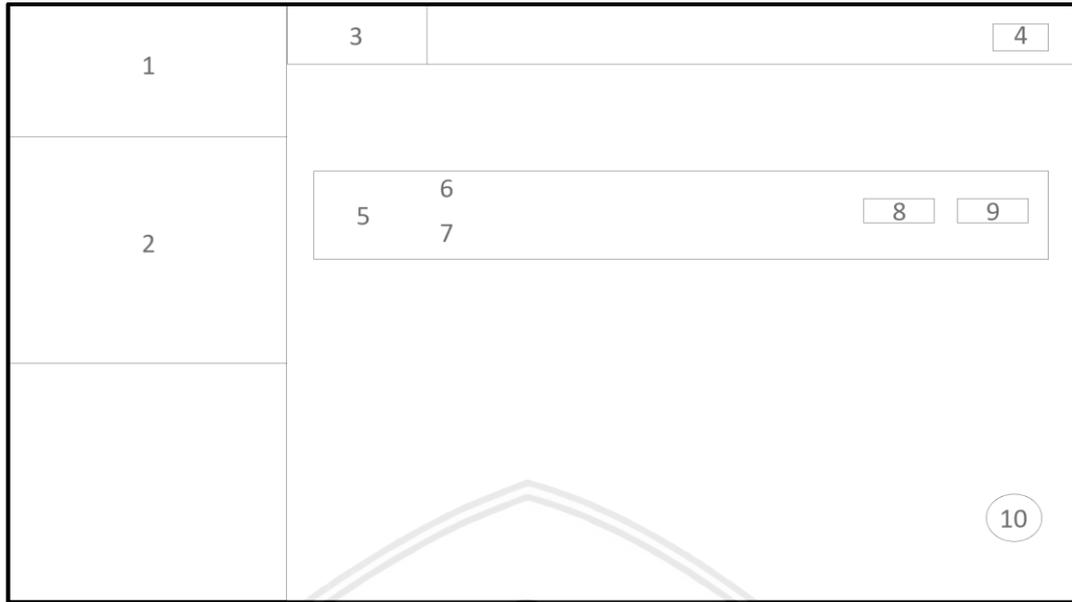
No.	Nama Objek	Tipe	Keterangan
1	Nama sistem	Teks	Nama sistem momment.
2	Navigasi	Tombol	Tombol navigasi untuk pindah halaman.
3	Nama halaman	Teks	Informasi nama halaman yang sedang dikunjungi pengguna.
4	Tombol <i>tweet</i>	Tombol	Tombol untuk membuka <i>pop-up posting tweet</i> .
5	Profil akun Twitter pengguna	Teks	Informasi profil akun Twitter pengguna seperti foto profil, <i>banner</i> , total <i>tweet</i> , <i>followers</i> , <i>following</i> , lokasi, dan tanggal daftar akun.
6	Laporan aktivitas	Teks	Informasi penilaian aktivitas pengguna dan rekomendasi.
7	Rata-rata mendapatkan <i>followers</i>	<i>Card</i>	<i>Card</i> yang digunakan untuk menampilkan informasi jumlah rata-rata peningkatan <i>followers</i>

Tabel 5.4 Keterangan Perancangan Antarmuka Halaman *Dashboard* (Lanjutan)

No.	Nama Objek	Tipe	Keterangan
			pengguna dan kompetitor.
8	Rata-rata <i>posting tweet</i>	<i>Card</i>	<i>Card</i> yang digunakan untuk menampilkan informasi jumlah rata-rata <i>posting tweet</i> pengguna dan kompetitor.
9	Rata-rata <i>retweet</i>	<i>Card</i>	<i>Card</i> yang digunakan untuk menampilkan informasi jumlah rata-rata <i>retweet</i> pengguna dan kompetitor.
10	Rata-rata <i>likes</i>	<i>Card</i>	<i>Card</i> yang digunakan untuk menampilkan informasi jumlah rata-rata <i>likes</i> pengguna dan kompetitor.
11	Rata-rata <i>replies</i>	<i>Card</i>	<i>Card</i> yang digunakan untuk menampilkan informasi jumlah rata-rata <i>replies</i> pengguna dan kompetitor.
12	Analisis Sentimen	<i>Card</i>	Grafik menampilkan analisis sentimen terhadap <i>posting tweet</i> pengguna dan teks terkait analisis sentimen positif dan negatif.
13	<i>Tweet</i> terbaik pengguna	Teks	Informasi mengenai 5 <i>tweet</i> terbaik pengguna.

5.1.4.2 Perancangan Antarmuka Halaman *Competitor*

Perancangan antarmuka halaman *competitor* dapat dilihat pada Gambar 5.8. Terdapat 10 komponen pada perancangan ini dengan penjelasannya dapat dilihat pada Tabel 5.5.



Gambar 5.8 Perancangan Antarmuka Halaman *Competitor*

Tabel 5.5 Keterangan Perancangan Antarmuka Halaman *Competitor*

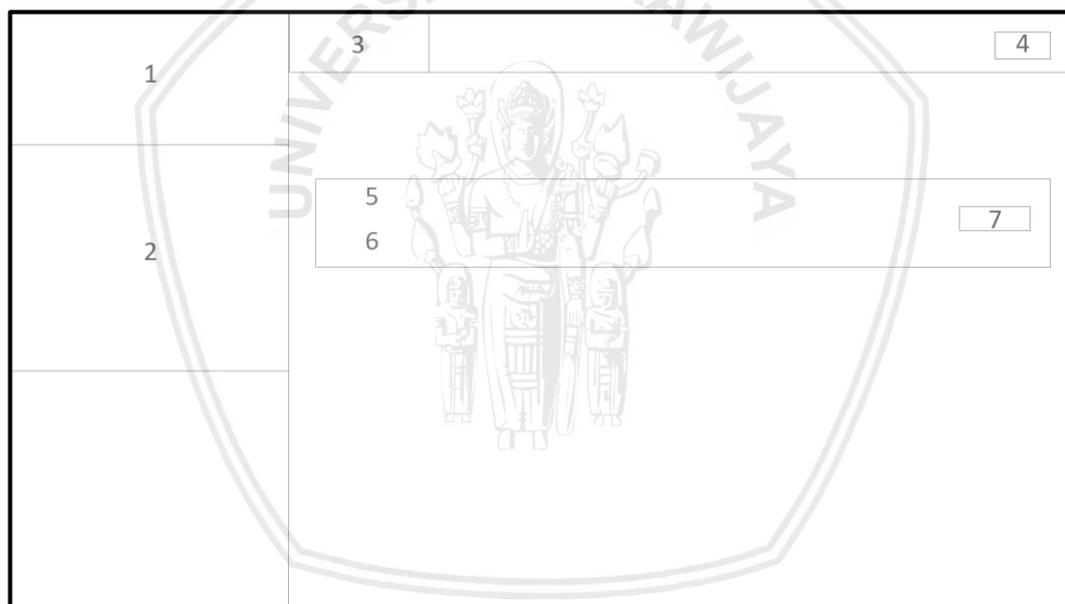
No.	Nama Objek	Tipe	Keterangan
1	Nama sistem	Teks	Nama sistem moment.
2	Navigasi	Tombol	Tombol navigasi untuk pindah halaman.
3	Nama halaman	Teks	Informasi nama halaman yang sedang dikunjungi pengguna.
4	Tombol <i>tweet</i>	Tombol	Tombol untuk membuka <i>pop-up posting tweet</i> .
5	Foto profil kompetitor	Gambar	Informasi foto profil kompetitor.
6	Nama kompetitor	Teks	Informasi nama kompetitor.
7	<i>Username</i> kompetitor	Teks	Informasi <i>username</i> kompetitor.
8	Tombol hapus	Tombol	Tombol yang digunakan untuk menghapus kompetitor.

Tabel 5.5 Keterangan Perancangan Antarmuka Halaman Competitor (Lanjutan)

No.	Nama Objek	Tipe	Keterangan
9	Tombol detail	Tombol	Tombol yang digunakan untuk melihat detail perbandingan aktivitas media sosial pengguna dan kompetitor.
10	Tombol tambah	Tombol	Tombol yang digunakan untuk menambah kompetitor.

5.1.4.3 Perancangan Antarmuka Halaman *Trend*

Perancangan antarmuka halaman *trend* dapat dilihat pada Gambar 5.9. Terdapat 7 komponen pada perancangan ini dengan penjelasannya dapat dilihat pada Tabel 5.6.



Gambar 5.9 Perancangan Antarmuka Halaman *Trend*

Tabel 5.6 Keterangan Perancangan Antarmuka Halaman *Trend*

No.	Nama Objek	Tipe	Keterangan
1	Nama sistem	Teks	Nama sistem moment.
2	Navigasi	Tombol	Tombol navigasi untuk pindah halaman.



Tabel 5.6 Keterangan Perancangan Antarmuka Halaman *Trend* (Lanjutan)

No.	Nama Objek	Tipe	Keterangan
3	Nama halaman	Teks	Informasi nama halaman yang sedang dikunjungi pengguna.
4	Tombol <i>tweet</i>	Tombol	Tombol untuk membuka <i>pop-up posting tweet</i> .
5	Nama tren	Teks	Informasi mengenai tren topik Indonesia.
6	Durasi tren	Teks	Informasi durasi suatu topik menjadi tren.
7	Tombol detail	Tombol	Tombol yang digunakan untuk menampilkan contoh <i>tweet</i> berdasarkan tren pilihan pengguna.

5.2 Implementasi Sistem

Setelah proses perancangan selesai dilakukan, tahap berikutnya adalah melakukan implementasi sistem. Implementasi sistem sendiri terdiri dari menjabarkan spesifikasi sistem yang digunakan dalam melakukan implementasi, implementasi kode program, implementasi basis data, dan implementasi antarmuka. Implementasi kode program akan mengubah perancangan algoritme menjadi sebuah kode program, implementasi basis data merupakan hasil implementasi yang dilakukan berdasarkan perancangan data, dan implementasi antarmuka merupakan hasil implementasi yang dilakukan berdasarkan perancangan *layout* yang telah didefinisikan sebelumnya.

5.2.1 Spesifikasi Sistem

Pada bagian spesifikasi sistem akan dijelaskan mengenai spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam melakukan penelitian. Spesifikasi perangkat keras terdiri dari penjabaran komponen *processor*, *main memory*, dan *graphics card* sedangkan pada spesifikasi perangkat lunak menjabarkan komponen editor dokumentasi, editor perancangan, editor pemrograman, bahasa pemrograman, *database management system*, dan *browser*. Spesifikasi perangkat keras dapat dilihat pada Tabel 5.7 dan spesifikasi perangkat lunak dapat dilihat pada Tabel 5.8.

Tabel 5.7 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
<i>Processor</i>	Intel Core i5 8250 U
<i>Main Memory</i>	DDR4 8 GB
<i>Graphics Card</i>	Nvidia Geforce MX150

Tabel 5.8 Spesifikasi Perangkat Lunak

Jenis Komponen	Spesifikasi
<i>Documentation Program</i>	Microsoft Office Word 2016
<i>Design Program</i>	Visual Paradigm
<i>Programming Editor</i>	Atom
<i>Programming Languages</i>	PHP, HTML, Python
<i>Database Management System</i>	Microsoft SQL
<i>Browser</i>	Google Chrome

5.2.2 Implementasi Kode Program

Implementasi kode program dilakukan berdasarkan perancangan komponen yang telah dilakukan sebelumnya. Pada penelitian ini hasil perancangan komponen akan diimplementasikan dengan bahasa pemrograman PHP. Terdapat tiga implementasi kode program yang akan dijelaskan pada bagian ini yaitu implementasi kode program lihat rekomendasi, lihat hasil sentimen analisis, dan lihat tren topik Indonesia.

5.2.2.1 Implementasi Kode Program Lihat Rekomendasi Dalam Pengelolaan Akun Media Sosial

Nama Operasi: *showRecommendation()*

Kelas: *DashboardController.php*

Tabel 5.9 Kode Program Lihat Rekomendasi Dalam Pengelolaan Akun Media Sosial

Kode program lihat rekomendasi dalam pengelolaan akun media sosial	
1	<code>public function showRecommendation()</code>
2	<code>{</code>
3	<code> \$tweetRecommendation = '';</code>
4	<code> \$postingRecommendation = '';</code>
5	<code> \$mentionRecommendation = '';</code>
6	<code> \$avgPostsComp = 0;</code>
7	<code> \$recommendTodaysTweet = TwitterTweet::where('twitter_id','=',</code>
8	<code>Auth::user()->twitterAccount->twitter_id)-</code>
9	<code>>whereDate('tweet_created','=',Carbon::today())-</code>
10	<code>>whereNull('reply_screen_name')->first();</code>
11	<code>}</code>

Tabel 5.9 Kode Program Lihat Rekomendasi Dalam Pengelolaan Akun Media Sosial (Lanjutan)

Kode program lihat rekomendasi dalam pengelolaan akun media sosial (Lanjutan)	
12	<code>\$recommendMention = TwitterTweet::where('twitter_id','=',</code>
13	<code>Auth::user()->twitterAccount->twitter_id)-</code>
14	<code>>whereDate('tweet_created','=',Carbon::today())-</code>
15	<code>>whereNotNull('reply_screen_name')->first();</code>
16	<code>\$bestHour = TwitterTweet::getBestHour();</code>
17	<code>\$tweetsUser</code>
18	<code>TwitterTweet::where('twitter_id','=',Auth::user()->twitterAccount-</code>
19	<code>>twitter_id)->where('created_at','>=', Carbon::today()-</code>
20	<code>>subWeek())->get();</code>
21	<code>\$posting = \$tweetsUser->count();</code>
22	<code>\$competitors</code>
23	<code>Competitor::where('twitter_id','=',Auth::user()->twitterAccount-</code>
24	<code>>twitter_id)->get();</code>
	<code>\$postingComp = TwitterTweet::getCompetitorPosts(\$competitors);</code>
25	
	<code>\$avgPosts = \$posting/8;</code>
26	
27	<code>if(count(\$competitors)){</code>
28	<code>\$avgPostsComp = \$postingComp/8/count(\$competitors);</code>
29	<code>}</code>
30	<code>if (empty(\$recommendTodaysTweet)) {</code>
31	<code>\$tweetRecommendation = '<i class="fas fa-</code>
32	<code>lightbulb"></i> Posting tweet hari ini
';</code>
33	<code>}</code>
34	<code>if (\$avgPosts < \$avgPostsComp) {</code>
35	<code>\$postRecommendation = round(\$avgPostsComp) - round(\$avgPosts)</code>
	<code>+ 1;</code>
36	<code>\$postingRecommendation = '<i class="fas fa-</code>
37	<code>lightbulb"></i> Sebaiknya hari ini anda post ' . \$postRecommendation</code>
38	<code>. ' tweet lagi
';</code>
39	<code>}</code>
40	<code>if (empty(\$recommendMention)) {</code>
41	<code>\$mentionRecommendation = '<i class="fas fa-</code>
42	<code>lightbulb"></i> Menanggapi mention masuk kepada anda
';</code>
43	<code>}</code>
44	<code>return response()->json(</code>
45	<code>[</code>
46	<code>'status' => 200,</code>
47	<code>'message' => 'success',</code>
48	<code>'response' => [</code>
49	<code>'bestHour' => \$bestHour,</code>
50	<code>'tweetRecommendation' => \$tweetRecommendation,</code>
51	<code>'postingRecommendation' => \$postingRecommendation,</code>
52	<code>'mentionRecommendation' => \$mentionRecommendation</code>
53	<code>]</code>
54	<code>]</code>
55	<code>);</code>
	<code>}</code>

Pada Tabel 5.9 menjelaskan tentang implementasi kode program lihat rekomendasi dalam pengelolaan akun media sosial, dimulai dengan mengambil data *posting tweet* pada hari tersebut dengan memanggil model *TwitterTweet* dilanjutkan dengan mengambil data *mention* pada hari tersebut dengan memanggil model *TwitterTweet* dilanjutkan dengan mengambil data jam *posting tweet* terbaik pengguna melalui model *TwitterTweet* dan selanjutnya mengambil data jumlah *posting tweet* pengguna dan kompetitor selama seminggu terakhir

dengan memanggil model *TwitterTweet* yang selanjutnya menghitung rata-rata jumlah *posting tweet* pengguna dan kompetitor setiap harinya berdasarkan data tersebut. Selanjutnya sistem melakukan pengecekan apabila data *posting tweet* pada hari tersebut tidak ditemukan maka sistem akan menampilkan pesan rekomendasi untuk melakukan *posting tweet* pada hari tersebut. Selanjutnya sistem melakukan pengecekan apabila jumlah rata-rata *posting tweet* pengguna lebih sedikit dibanding kompetitor maka sistem akan menampilkan pesan rekomendasi untuk melakukan *posting tweet* dengan jumlah selisih rata-rata *posting tweet* pengguna dan kompetitor ditambah satu. Selanjutnya sistem melakukan pengecekan apabila data *mention* tidak ditemukan pada hari tersebut maka sistem akan menampilkan pesan rekomendasi untuk menanggapi mention pada hari tersebut. Terakhir sistem mengembalikan data JSON untuk ditampilkan pada halaman *dashboard*.

5.2.2.2 Implementasi Kode Program Lihat Hasil Sentimen Analisis

Nama Operasi: *showSentiment()*

Kelas: *DashboardController.php*

Tabel 5.10 Kode Program Lihat Hasil Sentimen Analisis

Kode program lihat hasil sentimen analisis	
1	public function showSentiment()
2	{
3	\$tweetsUser =
4	TwitterTweet::where('twitter_id', '=', Auth::user()->twitterAccount-
5	->twitter_id)->where('created_at', '>=', Carbon::today()-
6	->subWeek()->get();
7	\$sample = TwitterReply::getSample();
8	\$sentiment = TwitterReply::getSentimentData(\$tweetsUser);
9	
10	\$totalSentiment = array_sum(\$sentiment);
11	if (\$totalSentiment > 0)
12	{
13	\$sentiment[0] = \$sentiment[0] / \$totalSentiment * 100;
14	\$sentiment[1] = \$sentiment[1] / \$totalSentiment * 100;
15	\$sentiment[2] = \$sentiment[2] / \$totalSentiment * 100;
16	}
17	return response()->json(
18	[
19	'status' => 200,
20	'message' => 'success',
21	'response' => [
22	'positiveSentiment' => \$sentiment[0],
23	'neutralSentiment' => \$sentiment[1],
24	'negativeSentiment' => \$sentiment[2],
25	'totalSentiment' => \$totalSentiment,
26	'samplingPositiveContent' =>
27	mb_convert_encoding(\$sample[0], 'UTF-8', 'UTF-8'),
28	'samplingPositiveUser' =>
29	mb_convert_encoding(\$sample[1], 'UTF-8', 'UTF-8'),
30	'samplingNegativeContent' =>
31	mb_convert_encoding(\$sample[2], 'UTF-8', 'UTF-8'),
32	'samplingNegativeUser' =>
33	mb_convert_encoding(\$sample[3], 'UTF-8', 'UTF-8'),
34]
35]
36);

Pada Tabel 5.10 menjelaskan tentang implementasi kode program lihat hasil sentimen analisis, dimulai dengan mengambil data *tweet* pengguna melalui model *TwitterTweet* dilanjutkan dengan mengambil contoh sentimen dan data sentimen melalui model *TwitterReply*. Setelah itu menghitung total jumlah sentimen dan melakukan pengecekan apabila jumlah sentimen lebih dari 0 maka sistem melakukan proses perhitungan nilai persen setiap sentimen. Terakhir sistem mengembalikan respon dalam bentuk JSON.

5.2.2.3 Implementasi Kode Program Lihat Tren Topik Indonesia

Nama Operasi: *Index()*

Kelas: *TrendingController.php*

Tabel 5.11 Kode Program Lihat Tren Topik Indonesia

Kode program lihat tren topik Indonesia	
1	<code>public function index()</code>
2	<code>{</code>
3	<code> \$trendDate = TwitterTrend::max('created_at');</code>
4	<code> \$trendDate = explode(" ", \$trendDate)[0];</code>
5	<code> \$twitterTrend</code>
6	<code>TwitterTrend::whereDate('created_at', '=', \$trendDate)-</code>
7	<code>>orderBy('id_trend', 'asc')->get();</code>
8	<code> return view('contents.trend', [</code>
9	<code> 'twitterTrend' => \$twitterTrend</code>
10	<code>]);</code>
11	<code>}</code>

Pada Tabel 5.11 menjelaskan tentang implementasi kode program dalam melihat tren topik Indonesia, dimulai dengan mengambil tanggal terakhir data tren di *update* melalui model *TwitterTrend* lalu dilanjutkan mengambil data tren berdasarkan tanggal yang telah diambil sebelumnya melalui model *TwitterTrend*. Setelah itu data ditampilkan pada halaman *trend*.

5.2.3 Implementasi Basis Data

Implementasi basis data akan menjelaskan mengenai *Data Definition Language* (DDL) dengan perintah *create table* yang digunakan dalam mengimplementasikan perancangan basis data. Terdapat delapan implementasi tabel basis data yang akan dijelaskan pada bagian ini antara lain implementasi tabel *competitor*, *twitter_accounts*, *twitter_accounts_log*, *twitter_replies*, *twitter_trend_detail*, *twitter_trends*, *twitter_tweets*, dan *users*.

5.2.3.1 Implementasi Tabel *Competitor*

Tabel *competitor* digunakan untuk menyimpan daftar kompetitor yang telah ditambahkan oleh pengguna. Implementasi tabel *competitor* dapat dilihat pada Tabel 5.12.

Tabel 5.12 Implementasi Tabel *Competitor*

Implementasi tabel <i>competitor</i>	
1	CREATE TABLE [dbo].[competitor](
2	[id] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
3	[twitter_id] [varchar](255) NULL,
4	[competitor_id] [varchar](255) NULL,
5	[created_at] [datetime] NULL,
6	[updated_at] [datetime] NULL
7)

5.2.3.2 Implementasi Tabel *Twitter_Accounts*

Tabel *twitter_accounts* digunakan untuk menyimpan data Twitter dengan keterangan apakah jenis akun tersebut sebagai akun Twitter pengguna atau kompetitor. Implementasi tabel *twitter_accounts* dapat dilihat pada Tabel 5.13.

Tabel 5.13 Implementasi Tabel *Twitter_Accounts*

Implementasi tabel <i>twitter accounts</i>	
1	CREATE TABLE [dbo].[twitter_accounts](
2	[twitter_id] [varchar](255) NOT NULL PRIMARY KEY,
3	[account_id] [int] NULL,
4	[is_account] [bit] NULL,
5	[is_competitor] [bit] NULL,
6	[created_at] [datetime] NULL,
7	[updated_at] [datetime] NULL
8)

5.2.3.3 Implementasi Tabel *Twitter_Accounts_Log*

Tabel *twitter_accounts_log* digunakan untuk menyimpan rekam data akun Twitter. Implementasi tabel *twitter_accounts_log* dapat dilihat pada Tabel 5.14.

Tabel 5.14 Implementasi Tabel *Twitter_Accounts_Log*

Implementasi tabel <i>twitter accounts log</i>	
1	CREATE TABLE [dbo].[twitter_accounts_log](
2	[id] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
3	[twitter_id] [varchar](255) NULL,
4	[name] [varchar](255) NULL,
5	[screen_name] [varchar](255) NULL,
6	[photo_url] [nvarchar](255) NULL,
7	[banner_url] [varchar](255) NULL,
8	[description] [text] NULL,
9	[favorites_count] [bigint] NULL,
10	[followers_count] [bigint] NULL,
11	[friends_count] [bigint] NULL,
12	[statuses_count] [bigint] NULL,
13	[location] [varchar](255) NULL,
14	[created] [datetime] NULL,
15	[created_at] [datetime] NULL,
16	[updated_at] [datetime] NULL
17)

5.2.3.4 Implementasi Tabel *Twitter_Replies*

Tabel *twitter_replies* digunakan untuk menyimpan data *reply* dari *tweet* pengguna. Implementasi tabel *twitter_replies* dapat dilihat pada Tabel 5.15.

Tabel 5.15 Implementasi Tabel *Twitter_Replies*

Implementasi tabel <i>twitter_replies</i>	
1	CREATE TABLE [dbo].[twitter_replies](
2	[replies_id] [varchar](255) NOT NULL PRIMARY KEY,
3	[tweet_id] [varchar](255) NULL,
4	[replies_content] [text] NULL,
5	[replies_created] [varchar](255) NULL,
6	[replies_media] [varchar](255) NULL,
7	[reply_account_id] [varchar](255) NULL,
8	[name] [varchar](255) NULL,
9	[screen_name] [varchar](255) NULL,
10	[reply_account_photo] [varchar](255) NULL,
11	[sentiment] [varchar](255) NULL,
12	[sentiment_weight] [decimal](4, 3) NULL,
13	[created_at] [datetime] NULL
14)

5.2.3.5 Implementasi Tabel *Twitter_Trend_Detail*

Tabel *twitter_trend_detail* digunakan untuk menyimpan contoh *tweet* berdasarkan tren yang ada. Implementasi tabel *twitter_trend_detail* dapat dilihat pada Tabel 5.16.

Tabel 5.16 Implementasi Tabel *Twitter_Trend_Detail*

Implementasi tabel <i>twitter trend detail</i>	
1	CREATE TABLE [dbo].[twitter_trend_detail](
2	[id_trend_detail] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
3	[id_trend] [int] NULL,
4	[name] [varchar](255) NULL,
5	[screen_name] [varchar](255) NULL,
6	[tweet_id] [varchar](255) NULL,
7	[tweet] [text] NULL,
8	[tweet_pic] [varchar](255) NULL,
9	[is_verified] [bit] NULL,
10	[photo_url] [varchar](255) NULL,
11	[retweet_count] [int] NULL,
12	[favorite_count] [int] NULL,
13	[replies_count] [int] NULL,
14	[is_retweet] [bit] NULL,
15	[recommendation] [decimal](38, 34) NULL,
16	[created_at] [datetime] NULL
17)

5.2.3.6 Implementasi Tabel *Twitter_Trends*

Tabel *twitter_trends* digunakan untuk menyimpan daftar tren topik Indonesia. Implementasi tabel *twitter_trends* dapat dilihat pada Tabel 5.17.

Tabel 5.17 Implementasi Tabel *Twitter_Trends*

Implementasi tabel <i>twitter trends</i>	
1	CREATE TABLE [dbo].[twitter_trends](
2	[id_trend] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
3	[trend] [text] NULL,
4	[count_in_hour] [smallint] NULL,
5	[created_at] [datetime] NULL,
6	[updated_at] [datetime] NULL
7)



5.2.3.7 Implementasi Tabel *Twitter_Tweets*

Tabel *twitter_tweets* digunakan untuk menyimpan data *tweet* pengguna dan kompetitor. Implementasi tabel *twitter_tweets* dapat dilihat pada Tabel 5.18.

Tabel 5.18 Implementasi Tabel *Twitter_Tweets*

Implementasi tabel <i>twitter_tweets</i>	
1	CREATE TABLE [dbo].[twitter_tweets](
2	[tweet_id] [varchar](255) NOT NULL PRIMARY KEY,
3	[twitter_id] [varchar](255) NULL,
4	[tweet_created] [datetime] NULL,
5	[tweet_content] [text] NULL,
6	[tweet_media] [varchar](255) NULL,
7	[location] [varchar](255) NULL,
8	[retweet_count] [int] NULL,
9	[favorite_count] [int] NULL,
10	[replies_count] [int] NULL,
11	[reply_status_id] [varchar](255) NULL,
12	[reply_user_id] [varchar](255) NULL,
13	[reply_screen_name] [varchar](255) NULL,
14	[recommendation] [decimal](38, 34) NULL,
15	[created_at] [datetime] NULL
16)

5.2.3.8 Implementasi Tabel *Users*

Tabel *Users* digunakan untuk menyimpan data pengguna sistem yang telah terdaftar. Implementasi tabel *users* dapat dilihat pada Tabel 5.19.

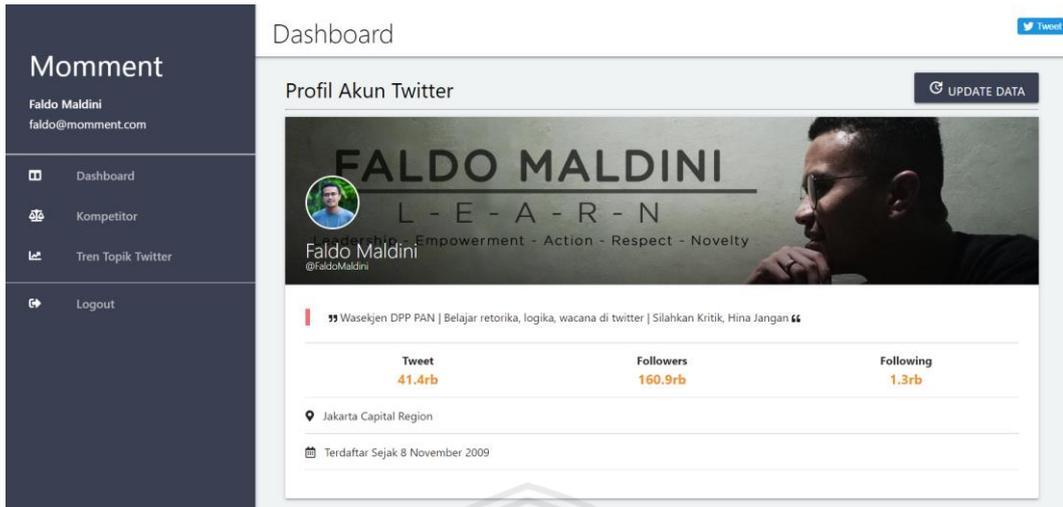
Tabel 5.19 Implementasi Tabel *Users*

Implementasi tabel <i>users</i>	
1	CREATE TABLE [dbo].[users](
2	[id] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
3	[name] [varchar](255) NOT NULL,
4	[email] [varchar](255) NOT NULL,
5	[password] [varchar](255) NOT NULL,
6	[remember_token] [nvarchar](100) NULL,
7	[created_at] [datetime] NULL,
8	[updated_at] [datetime] NULL
9)

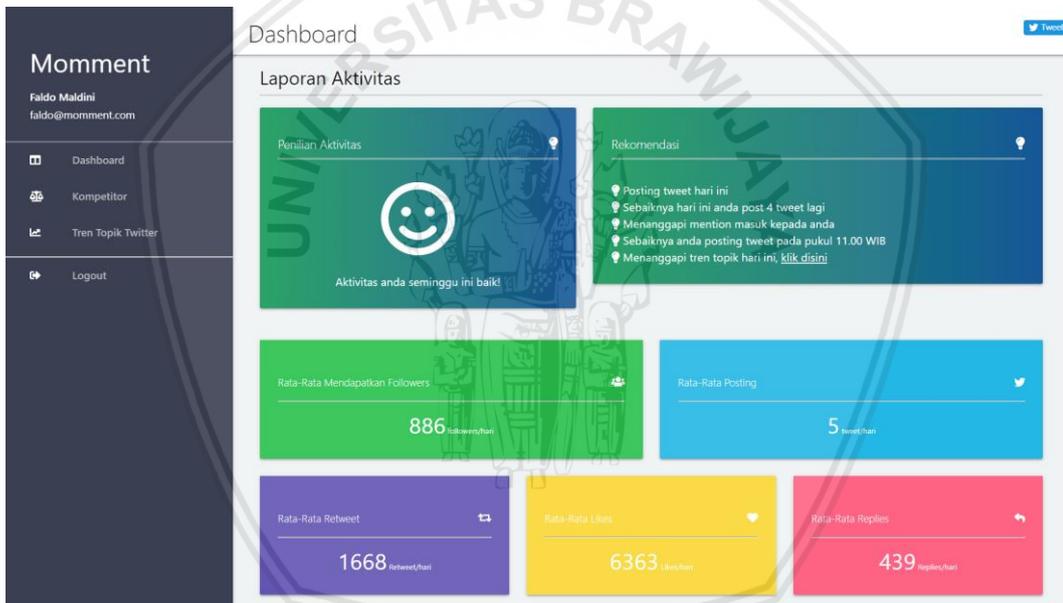
5.2.4 Implementasi Antarmuka

5.2.4.1 Implementasi Antarmuka Halaman *Dashboard*

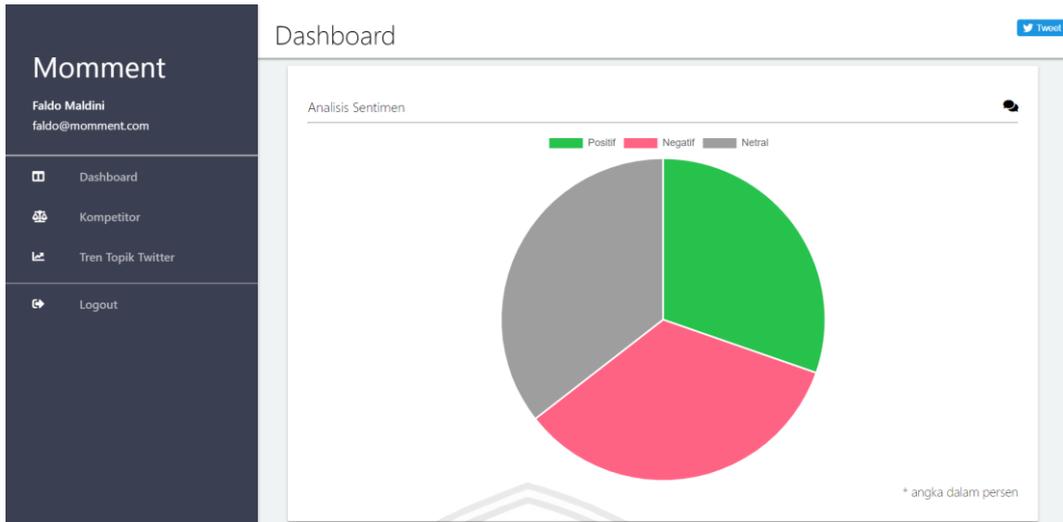
Gambar 5.10, 5.11, 5.12, 5.13, 5.14 merupakan hasil implementasi antarmuka halaman *dashboard*. Pada halaman ini terdapat profil akun Twitter pengguna, laporan aktivitas pengguna, analisis sentiment, dan daftar 5 *tweet* terbaik pengguna.



Gambar 5.10 Implementasi Antarmuka Halaman *Dashboard*



Gambar 5.11 Implementasi Antarmuka Halaman *Dashboard (Lanjutan)*



Gambar 5.12 Implementasi Antarmuka Halaman *Dashboard* (Lanjutan)

Dashboard

Moment
Faldo Maldini
faldo@moment.com

- Dashboard
- Kompetitor
- Tren Topik Twitter
- Logout

Sentimen Positif

Smangt dan sehat terus kang faldo.. insya allah swara saya dan keluarga buat abng 17 april menuju abng ke senayan., amin
@CilokHanet

ya mungkin km dia di bully netizen sih DO, jadi membuat pembenaran2 dgn narasi apa yg tdk terlihat di TV, trio macan di mata najwa itu sungguh menarik, 1 orang tua Said Didu dan 2 Polittisi muda, ente ama jansen itu adlah kolaborasi yang bagus dan terlihat baik utk Indonesia.
@sulemarbun

Bagus nih kalo banyak pemuda kayak gini mau terjun ke dunia politik rela masa mudanya memikirkan pembangunan utk negri, ketimbang anak muda yg foya foya ngabisin duit org tua. Walaupun bersebrangan dlm memilih pilihan capres sih.. Gue apresiasi elo bang.
@Mochhamsyah

Saya suka twittmu ini bang makanya saya LIKE & RT, lebih suka lagi kalau dirimu berkenan polbek akun Kuli Remekan Rengginan ini..suwon bang & selalu semangat, Merdeka!! #PrabowoMenyapaDIY
@RollezB

Bagus lagunya, kreatif banget
@azeemary

Sentimen Negatif

wkwkwk bukan maki2, tapi menghina smbl tertawa kebegooan ente yang ngebenci orang sampe bego, bagaimana tau rasanya dihinna. apalagi ente berhati jahat dan sampah negara.. orang yg sdh berkontribusi aja, anda dan gerombolan anda masih hina2
@AYWarman

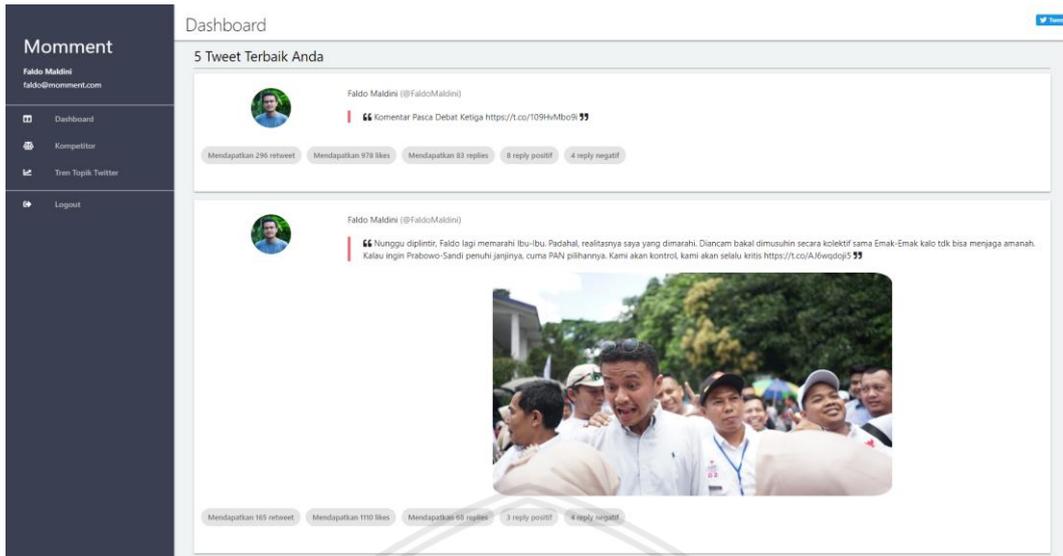
Saling intropeksi aja...lihat keluar sah" saja tp jangan lupa lihat kedalam juga, sehingga persepsi yg dibangun positif, jangan membangun sentimen negatif nanti justru bisa menyiprat wajah sendiri. Karna setiap manusia punya 2 sisi, baik & buruk. #siapapunpresidennyabuatinIndonesia
@StefanusPurwant

jangan samain gerombolan ente dengan kita, kita mah dengan sukarela menghina dan mentertawai ketololan gerombolan ente..gak enak kan di hina ?
@AYWarman

Jangan bicara hukum jika hal begini saja tak bisa dilakukan !
@Cuma_Rofiq

Bro lo satu dapil gak ma @AdianNapitupul1 ?
@BATOKKELAPA2019

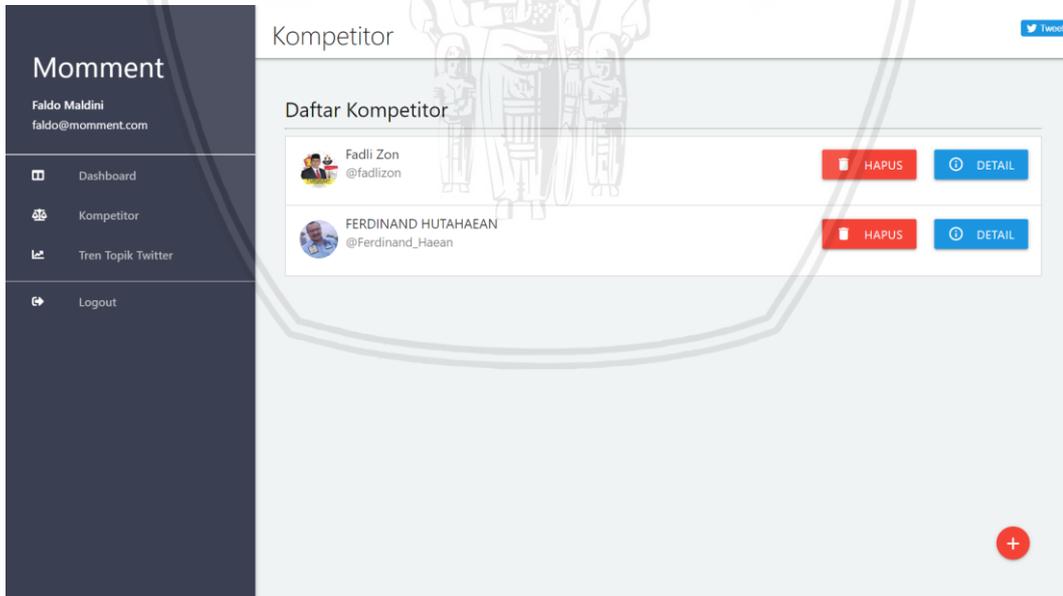
Gambar 5.13 Implementasi Antarmuka Halaman *Dashboard* (Lanjutan)



Gambar 5.14 Implementasi Antarmuka Halaman *Dashboard* (Lanjutan)

5.2.4.2 Implementasi Antarmuka Halaman *Competitor*

Gambar 5.15 merupakan hasil implementasi antarmuka halaman *competitor*. Pada halaman ini terdapat daftar kompetitor pengguna serta tombol “detail” dan “hapus” untuk melihat detail perbandingan aktivitas pengguna dan kompetitor serta menghapus kompetitor tersebut. Selain itu terdapat tombol “+” yang dapat digunakan oleh pengguna untuk menambahkan kompetitor.



Gambar 5.15 Implementasi Antarmuka Halaman *Competitor*

5.2.4.3 Implementasi Antarmuka Halaman *Trend*

Gambar 5.16 merupakan hasil implementasi antarmuka halaman *trend*. Pada halaman ini terdapat 10 daftar tren topik Indonesia dengan keterangan seberapa lama topik tersebut menjadi tren, selain itu terdapat tombol “detail” yang digunakan untuk melihat contoh *tweet* berdasarkan tren tersebut.



Gambar 5.16 Implementasi Antarmuka Halaman *Trend*

BAB 6 PENGUJIAN

6.1 Pengujian Unit

Pengujian unit bertujuan untuk menguji masukan dan keluaran dari unit terkecil sistem apakah sudah sesuai dengan yang diharapkan. Unit yang dimaksud merupakan fungsi dari sebuah kelas. Dalam melakukan pengujian unit dibuat kelas khusus untuk menguji kasus uji yang ada. Terdapat tiga pengujian yang akan dijelaskan yaitu pengujian unit fungsi *showRecommendation()*, *showSentiment()*, dan *index()*.

6.1.1 Pengujian Unit Algoritme Lihat Rekomendasi Dalam Pengelolaan Akun Media Sosial

1. Perancangan algoritme fungsi *showRecommendation* kelas *DashboardController* akan dijelaskan pada Tabel 6.1.

Nama fungsi: *showRecommendation()*

Nama kelas: *DashboardController*

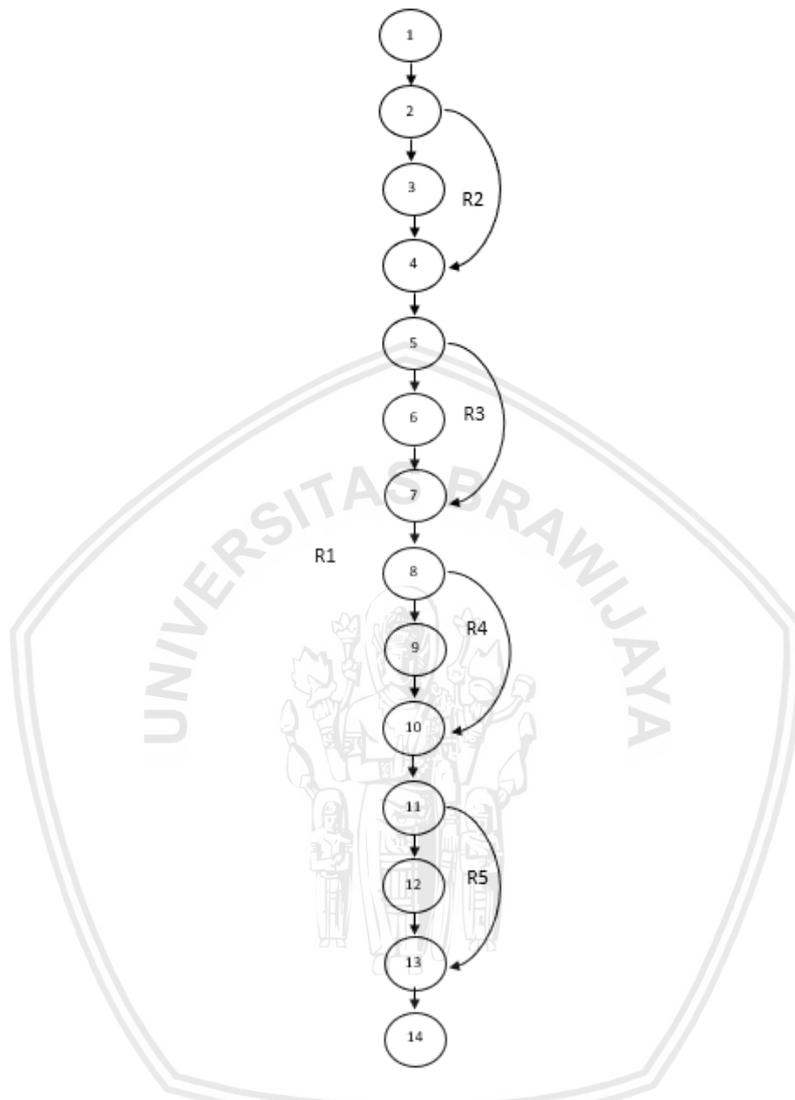
Tabel 6.1 Perancangan Algoritme showRecommendation

Perancangan algoritme lihat rekomendasi dalam pengelolaan akun media sosial		
1	Declaration variables avgPostsComp,	
2	tweetRecommendation, postingRecommendation,	
3	mentionRecommendation	
4	Get user's today tweet data	}
5	Get user's today mention data	
6	Get user's best hour to post tweet	
7	Get user's tweet data in a week	
8	Count user's tweet data	
9	Get user's competitor	
10	Get competitor's post	
11	count user's average posting per week	
12	if count competitor > 0	}
13	count competitor's average posting per week	
14	endif	
15	if user's today tweet data is empty	}
16	initiate tweetRecommendation = "posting tweet hari ini"	
17	endif	
18	if user's average posting < competitor's average posting	}
19	initiate postRecommendation = competitor's	
20	average posting - user's average posting	
21	initiate postingRecommendation = "Sebaiknya	
22	hari ini anda post + postRecommendation + tweet lagi"	
23	endif	
24	if user's today mention data is empty	}
25	initiate mentionRecommendation = "Menanggapi mention	
26	masuk kepada anda"	
27	endif	
28	return response json	



2. Basis Path Testing

a. Flow Graph



Gambar 6.1 Flow Graph Fungsi showRecommendation()

b. Cyclomatic Complexity

$$V(G) = \text{jumlah region} = 5$$

$$V(G) = \text{jumlah edge} - \text{jumlah node} + 2 = 17 - 14 + 2 = 5$$

$$V(G) = \text{jumlah predicate node} + 1 = 4 + 1 = 5$$

c. Independent Path

- Jalur 1: 1-2-4-5-7-8-10-11-13-14

- Jalur 2: 1-2-3-4-5-7-8-10-11-13-14

- Jalur 3: 1-2-3-4-5-6-7-8-10-11-13-14

- Jalur 4: 1-2-3-4-5-6-7-8-9-10-11-13-14
- Jalur 5: 1-2-3-4-5-6-7-8-10-11-12-13-14

Tabel 6.2 Hasil Pengujian Unit Lihat Rekomendasi Dalam Pengelolaan Akun Media Sosial

Jalur	Prosedur Uji	Expected Result	Result	Status
1	Class Driver memanggil fungsi <i>showRecommendation()</i> menggunakan akun dengan <i>id</i> = 125, yang memiliki data <i>tweet</i> pada hari tersebut dengan <i>tweet_id</i> = 323, data <i>mention</i> pada hari tersebut dengan <i>tweet_id</i> = 324, dan data <i>best tweet</i> dengan <i>tweet_id</i> = 325.	Sistem memanggil fungsi <i>showRecommendation()</i> lalu mengembalikan JSON berisi status = 200, <i>message</i> = <i>success</i> , <i>tweetRecommendation</i> , <i>postingRecommendation</i> , dan <i>mentionRecommendation</i> bernilai <i>null</i> .	Sistem memanggil fungsi <i>showRecommendation()</i> lalu mengembalikan JSON berisi status = 200, <i>message</i> = <i>success</i> , <i>tweetRecommendation</i> , <i>postingRecommendation</i> , dan <i>mentionRecommendation</i> bernilai <i>null</i> .	Valid.
2	Class Driver memanggil fungsi <i>showRecommendation()</i> menggunakan akun dengan <i>id</i> = 126 ,	Sistem memanggil fungsi <i>showRecommendation()</i> lalu melakukan pengecekan <i>competitor</i> dan menghitung rata-rata <i>posting competitor</i>	Sistem memanggil fungsi <i>showRecommendation()</i> lalu melakukan pengecekan <i>competitor</i> dan menghitung rata-rata <i>posting</i>	Valid.



Tabel 6.2 Hasil Pengujian Unit Lihat Rekomendasi Dalam Pengelolaan Akun Media Sosial (Lanjutan)

Jalur	Prosedur Uji	Expected Result	Result	Status
	yang memiliki data <i>tweet</i> pada hari tersebut dengan <i>tweet_id</i> = 335, data <i>mention</i> pada hari tersebut dengan <i>tweet_id</i> = 336, data <i>best tweet</i> dengan <i>tweet_id</i> = 337, dan memiliki <i>competitor</i> dengan <i>competitor_id</i> = 1226.	lalu mengembalikan JSON berisi status = 200, <i>message</i> = <i>success</i> , <i>tweetRecommendation</i> , <i>postingRecommendation</i> , dan <i>mentionRecommendation</i> bernilai <i>null</i> .	<i>competitor</i> lalu mengembalikan JSON berisi status = 200, <i>message</i> = <i>success</i> , <i>tweetRecommendation</i> , <i>postingRecommendation</i> , dan <i>mentionRecommendation</i> bernilai <i>null</i> .	Valid.
3	<i>Class Driver</i> memanggil fungsi <i>showRecommendation()</i> menggunakan akun dengan <i>id</i> = 127, yang memiliki data <i>mention</i> pada hari tersebut dengan <i>tweet_id</i> = 338, data <i>best tweet</i> dengan <i>tweet_id</i> = 339, dan memiliki <i>competitor</i> dengan <i>competitor_id</i> = 1228.	Sistem memanggil fungsi <i>showRecommendation()</i> lalu melakukan pengecekan <i>tweet</i> pada hari tersebut dan memberikan rekomendasi untuk melakukan <i>posting tweet</i> lalu mengembalikan JSON berisi status = 200, <i>message</i> = <i>success</i> , <i>postingRecommendation</i> , <i>mentionRecommendation</i> bernilai <i>null</i> , dan <i>tweetRecommendation</i> = ' <i class="fas fa-lightbulb"></i> Posting tweet hari ini
'.	Sistem memanggil fungsi <i>showRecommendation()</i> lalu melakukan pengecekan <i>tweet</i> pada hari tersebut dan memberikan rekomendasi untuk melakukan <i>posting tweet</i> lalu mengembalikan json berisi status = 200, <i>message</i> = <i>success</i> , <i>postingRecommendation</i> , <i>mentionRecommendation</i> bernilai <i>null</i> , dan <i>tweetRecommendation</i> = ' <i class="fas fa-lightbulb"></i> Posting tweet hari ini
'.	Valid.



Tabel 6.2 Hasil Pengujian Unit Lihat Rekomendasi Dalam Pengelolaan Akun Media Sosial (Lanjutan)

Jalur	Prosedur Uji	Expected Result	Result	Status
4	<p><i>Class Driver</i> memanggil fungsi <i>showRecommendation()</i> menggunakan akun dengan <i>id</i> = 128, yang memiliki data <i>mention</i> pada hari tersebut dengan <i>tweet_id</i> = 343, data <i>best tweet</i> dengan <i>tweet_id</i> = 344, memiliki <i>competitor</i> dengan <i>competitor_id</i> = 1230, dan <i>competitor</i> memiliki data <i>tweet</i> dengan <i>tweet_id</i> = 340, 341, 342.</p>	<p>Sistem memanggil fungsi <i>showRecommendation()</i> lalu melakukan pengecekan rata-rata jumlah <i>posting tweet</i> pengguna dan kompetitor dan memberikan rekomedasi jumlah tweet yang sebaiknya dilakukan pengguna lalu mengembalikan JSON berisi status = 200, <i>message</i> = <i>success</i>, <i>mentionRecommendation</i> bernilai <i>null</i>, <i>tweetRecommendation</i> = '<i class="fas fa-lightbulb"></i> Posting tweet hari ini
', dan <i>postingRecommendation</i> = '<i class="fas fa-lightbulb"></i> Sebaiknya hari ini anda post 1 tweet lagi
'.</p>	<p>Sistem memanggil fungsi <i>showRecommendation()</i> lalu melakukan pengecekan rata-rata jumlah <i>posting tweet</i> pengguna dan kompetitor dan memberikan rekomedasi jumlah tweet yang sebaiknya dilakukan pengguna lalu mengembalikan JSON berisi status = 200, <i>message</i> = <i>success</i>, <i>mentionRecommendation</i> bernilai <i>null</i>, <i>tweetRecommendation</i> = '<i class="fas fa-lightbulb"></i> Posting tweet hari ini
', dan <i>postingRecommendation</i> = '<i class="fas fa-lightbulb"></i> Sebaiknya hari ini anda post 1 tweet lagi
'.</p>	Valid.
5	<p><i>Class Driver</i> memanggil fungsi <i>showRecommendation()</i></p>	<p>Sistem memanggil fungsi <i>showRecommendation()</i> lalu melakukan pengecekan <i>mention</i> pada hari tersebut</p>	<p>Sistem memanggil fungsi <i>showRecommendation()</i> lalu melakukan pengecekan <i>mention</i> pada hari tersebut</p>	Valid.



Tabel 6.2 Hasil Pengujian Unit Lihat Rekomendasi Dalam Pengelolaan Akun Media Sosial (Lanjutan)

Jalur	Prosedur Uji	Expected Result	Result	Status
	menggunakan akun dengan <i>id</i> = 129, yang memiliki data <i>best tweet</i> dengan <i>tweet_id</i> = 347, memiliki <i>competitor</i> dengan <i>competitor_id</i> = 1232, dan <i>competitor</i> memiliki data <i>tweet</i> dengan <i>tweet_id</i> = 345, 346.	dan memberikan rekomendasi untuk melakukan <i>posting mention</i> lalu mengembalikan JSON berisi status = 200, <i>message</i> = <i>success</i> , <i>tweetRecommendation</i> = ' <i class="fas fa-lightbulb"></i> Posting tweet hari ini
', dan <i>postingRecommendation</i> = '<i class="fas fa-lightbulb"></i> Sebaiknya hari ini anda post 1 tweet lagi
', <i>mentionRecommendation</i> = '<i class="fas fa-lightbulb"></i> Menanggapi mention masuk kepada anda
'.	dan memberikan rekomendasi untuk melakukan <i>posting mention</i> lalu mengembalikan JSON berisi status = 200, <i>message</i> = <i>success</i> , <i>tweetRecommendation</i> = ' <i class="fas fa-lightbulb"></i> Posting tweet hari ini
', dan <i>postingRecommendation</i> = '<i class="fas fa-lightbulb"></i> Sebaiknya hari ini anda post 1 tweet lagi
', <i>mentionRecommendation</i> = '<i class="fas fa-lightbulb"></i> Menanggapi mention masuk kepada anda
'.	Valid.

Berdasarkan hasil pengujian setiap jalur uji pada pengujian unit algoritme lihat rekomendasi dalam pengelolaan akun media sosial yang telah dijelaskan pada Tabel 6.2 menghasilkan setiap jalur uji telah teruji seluruhnya dan berstatus valid.

6.1.2 Pengujian Unit Algoritme Lihat Hasil Sentimen Analisis

1. Perancangan algoritme fungsi *showSentiment* kelas *DashboardController* akan dijelaskan pada Tabel 6.3.

Nama fungsi: *showSentiment()*

Nama kelas: *DashboardController*

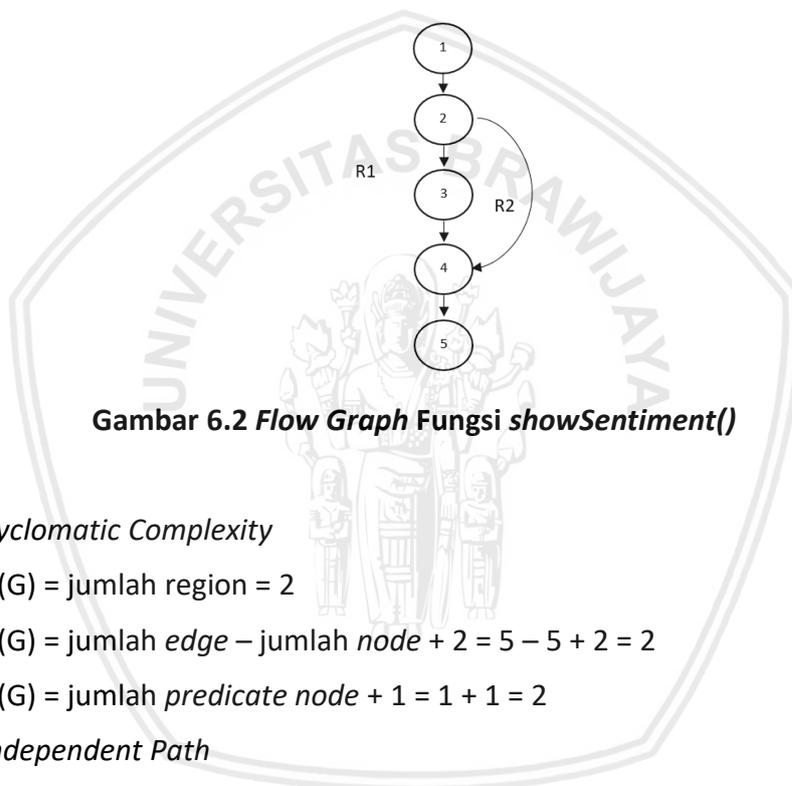


Tabel 6.3 Perancangan Algoritme Lihat Hasil Sentimen Analisis

Perancangan algoritme lihat hasil sentimen analisis		
1	Get user's tweet	}
2	Get sentiment sample	
3	Get sentiment data	
4	Count total sentiment	}
5	If total sentiment > 0	
6	Count sentiment percentage	5
7	End if	4
8	Return response json	3
		2
		1

2. *Basis Path Testing*

a. *Flow Graph*



Gambar 6.2 Flow Graph Fungsi *showSentiment()*

b. *Cyclomatic Complexity*

$V(G) = \text{jumlah region} = 2$

$V(G) = \text{jumlah edge} - \text{jumlah node} + 2 = 5 - 5 + 2 = 2$

$V(G) = \text{jumlah predicate node} + 1 = 1 + 1 = 2$

c. *Independent Path*

- Jalur 1: 1-2-4-5
- Jalur 2: 1-2-3-4-5

Tabel 6.4 Hasil Pengujian Unit Lihat Hasil Sentimen Analisis

Jalur	Prosedur Uji	Expected Result	Result	Status
1	<i>Class Driver</i> memanggil fungsi <i>showSentiment()</i> menggunakan akun dengan <i>id = 122</i> , yang memiliki	Sistem memanggil fungsi <i>showSentiment()</i> lalu mengembalikan JSON berisi status = 200,	Sistem memanggil fungsi <i>showSentiment()</i> lalu mengembalikan JSON berisi status = 200,	Valid.



Tabel 6.4 Hasil Pengujian Unit Lihat Hasil Sentimen Analisis (Lanjutan)

Jalur	Prosedur Uji	Expected Result	Result	Status
	data <i>tweet</i> dengan <i>tweet_id</i> = 322.	<i>message</i> = <i>success</i> , <i>postiveSentiment</i> = 0, <i>neutralSentiment</i> = 0, <i>negativeSentiment</i> = 0, <i>totalSentiment</i> = 0.	<i>message</i> = <i>success</i> , <i>postiveSentiment</i> = 0, <i>neutralSentiment</i> = 0, <i>negativeSentiment</i> = 0, <i>totalSentiment</i> = 0.	Valid.
2	<i>Class Driver</i> memanggil fungsi <i>showSentiment()</i> menggunakan akun dengan <i>id</i> = 234, yang memiliki data <i>tweet</i> dengan <i>tweet_id</i> = 876, data <i>reply</i> dengan <i>replies_id</i> = 1 bersentimen positif, <i>replies_id</i> = 2 bersentimen netral, <i>replies_id</i> = 3 bersentimen negatif.	Sistem memanggil fungsi <i>showSentiment()</i> lalu mengembalikan JSON berisi status = 200, <i>message</i> = <i>success</i> , <i>postiveSentiment</i> = 33.33, <i>neutralSentiment</i> = 33.33, <i>negativeSentiment</i> = 33.33, <i>totalSentiment</i> = 3.	Sistem memanggil fungsi <i>showSentiment()</i> lalu mengembalikan JSON berisi status = 200, <i>message</i> = <i>success</i> , <i>postiveSentiment</i> = 33.33, <i>neutralSentiment</i> = 33.33, <i>negativeSentiment</i> = 33.33, <i>totalSentiment</i> = 3.	Valid.

Berdasarkan hasil pengujian setiap jalur uji pada pengujian unit algoritme lihat hasil sentimen analisis yang telah dijelaskan pada Tabel 6.4 menghasilkan setiap jalur uji telah teruji seluruhnya dan berstatus valid.

6.1.3 Pengujian Unit Algoritme Lihat Tren Topik Indonesia

1. Perancangan algoritme fungsi *index()* kelas *TrendingController* akan dijelaskan pada Tabel 6.5.

Nama fungsi: *index()*

Nama kelas: *TrendingController*

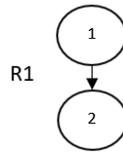
Tabel 6.5 Perancangan Algoritme Lihat Tren Topik Indonesia

Perancangan algoritme lihat tren topik Indonesia		
1	Get trend latest created date	} <u>1</u>
2	Get twitter trend based on latest created date	
3	Return view trend	<u>2</u>



2. Basis Path Testing

a. Flow Graph



Gambar 6.3 Flow Graph Fungsi Index()

b. Cyclomatic Complexity

$V(G) = \text{jumlah region} = 1$

$V(G) = \text{jumlah edge} - \text{jumlah node} + 2 = 1 - 2 + 2 = 1$

$V(G) = \text{jumlah predicate node} + 1 = 0 + 1 = 1$

c. Independent Path

- Jalur 1: 1-2

Tabel 6.6 Hasil Pengujian Unit Lihat Tren Topik Indonesia

Jalur	Prosedur Uji	Expected Result	Result	Status
1	Class <i>Driver</i> memanggil fungsi <i>index()</i> pada class <i>TrendingController</i> .	Sistem memanggil fungsi <i>index()</i> lalu menampilkan tren topik Indonesia.	Sistem memanggil fungsi <i>index()</i> lalu menampilkan tren topik Indonesia.	Valid.

Berdasarkan hasil pengujian setiap jalur uji pada pengujian unit algoritme lihat tren topik Indonesia yang telah dijelaskan pada Tabel 6.6 menghasilkan setiap jalur uji telah teruji seluruhnya dan berstatus valid.

6.2 Pengujian Integrasi

Pengujian integrasi dilakukan untuk menguji interaksi antar komponen sesuai dengan struktur sistem. Pada penelitian ini terdapat fungsi *showSentiment()* dari class *DashboardController* yang memanggil fungsi *getSample()* dan *getSentimentData()* dari class *TwitterReply*. Fungsi *showSentiment()* memanggil fungsi *getSample()* dan *getSentimentData()* untuk mengambil data yang dibutuhkan dalam mengembalikan data yang diperlukan pada *view* dalam bentuk JSON.

1. Perancangan algoritme fungsi *showSentiment* kelas *DashboardController* akan dijelaskan pada Tabel 6.7.

Nama fungsi: *showSentiment()*

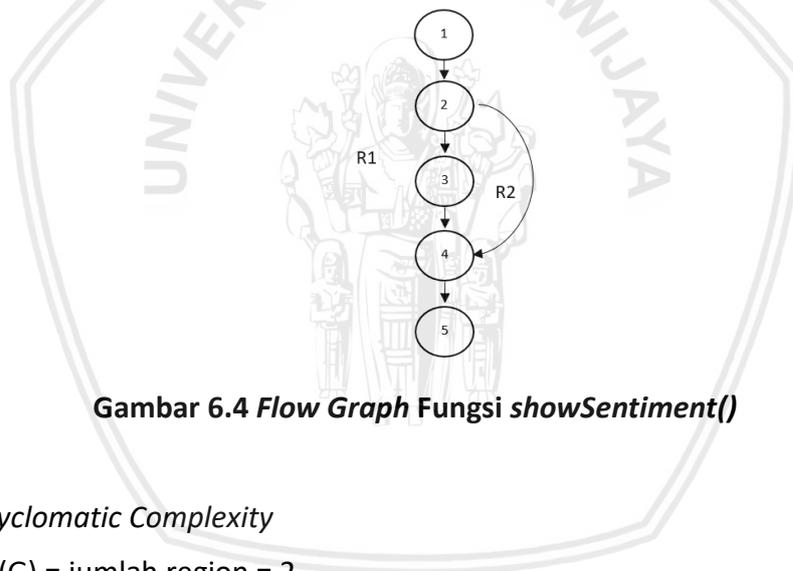
Nama kelas: *DashboardController*

Tabel 6.7 Perancangan Algoritme Lihat Hasil Sentimen Analisis

Perancangan algoritme lihat hasil sentimen analisis		
1	Get user's tweet	}
2	Get sentiment sample	
3	Get sentiment data	
4	Count total sentiment	}
5	If total sentiment > 0	
6	Count sentiment percentage	2
7	End if	3
8	Return response json	4
		5

2. *Basis Path Testing*

a. *Flow Graph*



Gambar 6.4 Flow Graph Fungsi *showSentiment()*

- b. *Cyclomatic Complexity*

$V(G) = \text{jumlah region} = 2$

$V(G) = \text{jumlah edge} - \text{jumlah node} + 2 = 5 - 5 + 2 = 2$

$V(G) = \text{jumlah predicate node} + 1 = 1 + 1 = 2$

- c. *Independent Path*

- Jalur 1: 1-2-4-5
- Jalur 2: 1-2-3-4-5



Tabel 6.8 Hasil Pengujian Integrasi

Jalur	Prosedur Uji	Expected Result	Result	Status
1	Class Driver memanggil fungsi <i>showSentiment()</i> yang memanggil fungsi <i>getSample()</i> dan <i>getSentimentData</i> dengan data pada tabel <i>twitter_replies = null</i> .	Sistem memanggil fungsi <i>showSentiment()</i> yang kemudian memanggil fungsi <i>getSample()</i> dan <i>getSentimentData</i> kemudian keduanya memberikan kembalian <i>null</i> .	Sistem memanggil fungsi <i>showSentiment()</i> yang kemudian memanggil fungsi <i>getSample()</i> dan <i>getSentimentData</i> kemudian keduanya memberikan kembalian <i>null</i> .	Valid.
2	Class Driver memanggil fungsi <i>showSentiment()</i> yang memanggil fungsi <i>getSample()</i> dan <i>getSentimentData</i> dengan data pada tabel <i>twitter_replies</i> masing-masing memiliki sentimen positif, netral, dan negatif berjumlah 1	Sistem memanggil fungsi <i>showSentiment()</i> yang kemudian memanggil fungsi <i>getSample()</i> dan <i>getSentimentData()</i> kemudian fungsi <i>getSample()</i> memberikan kembalian berupa array <i>sampling</i> positif dan negatif sentimen selanjutnya fungsi <i>getSentimentData()</i> memberikan kembalian array <i>[1,1,1]</i> yang merupakan jumlah dari masing-masing sentimen	Sistem memanggil fungsi <i>showSentiment()</i> yang kemudian memanggil fungsi <i>getSample()</i> dan <i>getSentimentData()</i> kemudian fungsi <i>getSample()</i> memberikan kembalian berupa array <i>sampling</i> positif dan negatif sentimen selanjutnya fungsi <i>getSentimentData()</i> memberikan kembalian array <i>[1,1,1]</i> yang merupakan jumlah dari masing-masing sentimen	Valid.

Berdasarkan hasil pengujian setiap jalur uji pada pengujian integrasi antara fungsi *showSentiment()* dengan *getSample()* dan *getSentimentData()* yang telah dijelaskan pada Tabel 6.8 menghasilkan setiap jalur uji telah teruji seluruhnya dan berstatus valid.



6.3 Pengujian Validasi

Pengujian validasi pada penelitian ini menggunakan pendekatan *black box*. Pengujian *black box* bertujuan untuk menguji perangkat lunak berdasarkan spesifikasi fungsional tanpa melihat kode atau desain program. Hasil dari pengujian tersebut memastikan masukan dan keluaran dari setiap fungsi program sesuai dengan spesifikasi yang telah dideskripsikan sebelumnya.

6.3.1 Login

Pengujian validasi *login* dengan kode kebutuhan MM-F-001 akan menguji sistem mampu melakukan otorisasi tamu yang terdaftar sebagai pengguna. Pada pengujian *login* alternatif pertama akan menguji sistem mampu menampilkan pesan untuk mengisi *field* yang belum mendapat masukan dalam melakukan proses *login*. Pada pengujian *login* alternatif kedua akan menguji sistem mampu menampilkan pesan bahwa *e-mail* atau *password* masukan tamu tidak sesuai. Pengujian validasi *login* akan dijelaskan pada Tabel 6.9, pengujian validasi *login* alternatif pertama akan dijelaskan pada Tabel 6.10, dan pengujian validasi *login* alternatif kedua akan dijelaskan pada Tabel 6.11.

Tabel 6.9 Pengujian Validasi Login

Pengujian validasi <i>login</i>	
Nama Kasus Uji	<i>Login</i> .
Objek Uji	Kebutuhan fungsional MM-F-001.
Tujuan Pengujian	Menguji sistem mampu melakukan otorisasi tamu yang terdaftar sebagai pengguna.
Prosedur Uji	<ol style="list-style-type: none"> 1. Membuka halaman <i>login</i>. 2. Memasukan <i>e-mail</i> dan <i>password</i> yang telah terdaftar. 3. Menekan tombol "<i>Login</i>".
Hasil Pengujian yang diharapkan	Sistem menampilkan halaman <i>dashboard</i> .
Hasil pengujian	Sistem menampilkan halaman <i>dashboard</i> .
Status	Valid.

Tabel 6.10 Pengujian Validasi Login – Alternatif Pertama

Pengujian validasi <i>login</i> – alternatif pertama	
Nama Kasus Uji	<i>Login</i> – alternatif pertama.
Objek Uji	Kebutuhan fungsional MM-F-001.
Tujuan Pengujian	Menguji sistem mampu mendeteksi <i>field</i> yang belum mendapatkan masukan.
Prosedur Uji	1. Membuka halaman <i>login</i> . 2. Menekan tombol " <i>Login</i> ".
Hasil Pengujian yang diharapkan	Sistem menampilkan pesan "Silahkan isi <i>field</i> ini terlebih dahulu".
Hasil pengujian	Sistem menampilkan pesan "Silahkan isi <i>field</i> ini terlebih dahulu"
Status	Valid.

Tabel 6.11 Pengujian Validasi *Login* – Alternatif Kedua

Pengujian validasi <i>login</i> – alternatif kedua	
Nama Kasus Uji	<i>Login</i> – alternatif kedua.
Objek Uji	Kebutuhan fungsional MM-F-001.
Tujuan Pengujian	Menguji sistem mampu mendeteksi kegagalan otorisasi tamu yang belum terdaftar sebagai pengguna.
Prosedur Uji	3. Membuka halaman <i>login</i> . 4. Memasukan <i>e-mail</i> dan <i>password</i> yang belum terdaftar. 5. Menekan tombol " <i>Login</i> ".
Hasil Pengujian yang diharapkan	Sistem menampilkan pesan " <i>E-mail/Password</i> tidak cocok, silahkan coba lagi".
Hasil pengujian	Sistem menampilkan pesan " <i>E-mail/Password</i> tidak cocok, silahkan coba lagi".
Status	Valid.

6.3.2 Daftar Akun

Pengujian validasi daftar akun dengan kode kebutuhan MM-F-002 akan menguji sistem mampu menyimpan data pengguna sistem baru. Pengujian validasi daftar akun alternatif pertama akan menguji sistem mampu mendeteksi kondisi ketika terdapat *field* yang belum mendapatkan masukan dalam melakukan proses pendaftaran akun. Pengujian validasi daftar akun alternatif kedua akan menguji sistem mampu mendeteksi perbedaan masukan antara *field password* dan konfirmasi *password*. Pengujian validasi daftar akun alternatif ketiga akan menguji

sistem mampu mendeteksi masukan *e-mail* yang telah terdaftar sebelumnya. Pengujian validasi daftar akun akan dijelaskan pada Tabel 6.12, pengujian validasi daftar akun alternatif pertama akan dijelaskan pada Tabel 6.13, pengujian validasi daftar akun alternatif kedua akan dijelaskan pada Tabel 6.14, dan pengujian validasi daftar akun alternatif ketiga akan dijelaskan pada Tabel 6.15.

Tabel 6.12 Pengujian Validasi Daftar Akun

Pengujian validasi daftar akun	
Nama Kasus Uji	Daftar akun.
Objek Uji	Kebutuhan fungsional MM-F-002.
Tujuan Pengujian	Menguji sistem mampu menyimpan data pengguna sistem baru.
Prosedur Uji	<ol style="list-style-type: none"> 1. Membuka halaman <i>register</i>. 2. Memasukan nama lengkap, <i>e-mail</i> yang belum terdaftar, <i>password</i>, dan konfirmasi <i>password</i>. 3. Menekan tombol "Daftar".
Hasil Pengujian yang diharapkan	Sistem menyimpan data masukan sebagai pengguna baru yang terdaftar lalu menampilkan halaman <i>engage</i> .
Hasil pengujian	Sistem menyimpan data masukan sebagai pengguna baru yang terdaftar lalu menampilkan halaman <i>engage</i> .
Status	Valid.

Tabel 6.13 Pengujian Validasi Daftar Akun – Alternatif Pertama

Pengujian validasi daftar akun – alternatif pertama	
Nama Kasus Uji	Daftar akun – alternatif pertama.
Objek Uji	Kebutuhan fungsional MM-F-002.
Tujuan Pengujian	Menguji sistem mampu mendeteksi <i>field</i> yang belum menerima masukan.
Prosedur Uji	<ol style="list-style-type: none"> 1. Membuka halaman <i>register</i>. 2. Menekan tombol "Daftar".
Hasil Pengujian yang diharapkan	Sistem menampilkan pesan "Silahkan isi <i>field</i> ini terlebih dahulu".
Hasil pengujian	Sistem menampilkan pesan "Silahkan isi <i>field</i> ini terlebih dahulu".
Status	Valid.

Tabel 6.14 Pengujian Validasi Daftar Akun – Alternatif Kedua

Pengujian validasi daftar akun – alternatif kedua	
Nama Kasus Uji	Daftar akun – alternatif kedua.
Objek Uji	Kebutuhan fungsional MM-F-002.
Tujuan Pengujian	Menguji sistem mampu mendeteksi ketidaksesuaian masukan <i>field password</i> dan konfirmasi <i>password</i> .
Prosedur Uji	<ol style="list-style-type: none"> 1. Membuka halaman <i>register</i>. 2. Memasukan nama lengkap, <i>e-mail</i>, <i>password</i>, dan konfirmasi <i>password</i> yang berbeda dengan <i>field password</i>. 3. Menekan tombol “Daftar”.
Hasil Pengujian yang diharapkan	Sistem menampilkan pesan “ <i>Password tidak sesuai</i> ”.
Hasil pengujian	Sistem menampilkan pesan “ <i>Password tidak sesuai</i> ”.
Status	Valid.

Tabel 6.15 Pengujian Validasi Daftar Akun – Alternatif Ketiga

Pengujian validasi daftar akun – alternatif ketiga	
Nama Kasus Uji	Daftar akun – alternatif ketiga.
Objek Uji	Kebutuhan fungsional MM-F-002.
Tujuan Pengujian	Menguji sistem mampu mendeteksi masukan <i>e-mail</i> yang sebelumnya telah terdaftar.
Prosedur Uji	<ol style="list-style-type: none"> 1. Membuka halaman <i>register</i>. 2. Memasukan nama lengkap, <i>e-mail</i> yang telah terdaftar, <i>password</i>, dan konfirmasi <i>password</i>. 3. Menekan tombol “Daftar”.
Hasil Pengujian yang diharapkan	Sistem menampilkan pesan “ <i>E-mail telah terdaftar</i> ”.
Hasil pengujian	Sistem menampilkan pesan “ <i>E-mail telah terdaftar</i> ”.
Status	Valid.

6.3.3 Menetapkan Akun Twitter

Pengujian validasi menetapkan akun Twitter dengan kode kebutuhan MM-F-003 akan menguji sistem mampu menetapkan akun Twitter pengguna. Pengujian validasi menetapkan akun Twitter alternatif pertama akan mendeteksi akun Twitter yang telah terdaftar sebelumnya. Pengujian validasi menetapkan akun Twitter akan dijelaskan pada Tabel 6.16 dan pengujian validasi menetapkan akun Twitter alternatif pertama akan dijelaskan pada Tabel 6.17.

Tabel 6.16 Pengujian Validasi Menetapkan Akun Twitter

Pengujian validasi menetapkan akun Twitter	
Nama Kasus Uji	Menetapkan akun Twitter.
Objek Uji	Kebutuhan fungsional MM-F-003.
Tujuan Pengujian	Menguji sistem mampu menetapkan akun Twitter pengguna.
Prosedur Uji	<ol style="list-style-type: none"> 1. Membuka halaman <i>engage</i>. 2. Memasukan <i>username</i> Twitter lalu menekan tombol pencarian. 3. Menekan tombol "Tetapkan Akun". 4. Menekan tombol "Ya" sebagai konfirmasi menetapkan akun
Hasil Pengujian yang diharapkan	Sistem menetapkan akun Twitter lalu menampilkan halaman <i>dashboard</i> .
Hasil pengujian	Sistem menetapkan akun Twitter lalu menampilkan halaman <i>dashboard</i> .
Status	Valid.

Tabel 6.17 Pengujian Validasi Menetapkan Akun Twitter – Alternatif Pertama

Pengujian validasi menetapkan akun Twitter – alternatif pertama	
Nama Kasus Uji	Menetapkan akun Twitter – alternatif pertama.
Objek Uji	Kebutuhan fungsional MM-F-003.
Tujuan Pengujian	Menguji sistem mampu mendeteksi akun Twitter yang sebelumnya telah ditetapkan oleh pengguna lain.
Prosedur Uji	<ol style="list-style-type: none"> 1. Membuka halaman <i>engage</i>. 2. Memasukan <i>username</i> Twitter lalu menekan tombol pencarian.

Tabel 6.17 Pengujian Validasi Menetapkan Akun Twitter – Alternatif Pertama (Lanjutan)

Pengujian validasi menetapkan akun Twitter – alternatif pertama (Lanjutan)	
Hasil Pengujian yang diharapkan	Sistem mengubah tombol “tetapkan akun” menjadi keterangan “akun telah terdaftar”.
Hasil pengujian	Sistem mengubah tombol “tetapkan akun” menjadi keterangan “akun telah terdaftar”.
Status	Valid.

6.3.4 Lihat Profil Akun Twitter

Pengujian validasi lihat profil akun Twitter dengan kode kebutuhan MM-F-004 akan menguji sistem mampu menampilkan profil akun Twitter pengguna. Pengujian validasi lihat profil akun Twitter akan dijelaskan pada Tabel 6.18.

Tabel 6.18 Pengujian Validasi Lihat Profil Akun Twitter

Pengujian validasi lihat profil akun Twitter	
Nama Kasus Uji	Lihat profil akun Twitter.
Objek Uji	Kebutuhan fungsional MM-F-004.
Tujuan Pengujian	Menguji sistem mampu menampilkan profil akun Twitter pengguna.
Prosedur Uji	1. Membuka halaman <i>dashboard</i> .
Hasil Pengujian yang diharapkan	Sistem menampilkan profil akun Twitter.
Hasil pengujian	Sistem menampilkan profil akun Twitter.
Status	Valid.

6.3.5 Lihat Penilaian Aktivitas Selama Seminggu Terakhir

Pengujian validasi lihat penilaian aktivitas selama seminggu terakhir dengan kode kebutuhan MM-F-005 akan menguji sistem mampu mendeteksi ketika rata-rata aktivitas pengguna lebih baik dari kompetitor. Pengujian validasi lihat penilaian aktivitas selama seminggu terakhir alternatif pertama akan menguji sistem mampu mendeteksi ketika rata-rata aktivitas kompetitor lebih baik dari pengguna. Pengujian validasi lihat penilaian aktivitas selama seminggu terakhir akan dijelaskan pada Tabel 6.19 dan pengujian validasi lihat penilaian aktivitas selama seminggu terakhir alternatif pertama akan dijelaskan pada Tabel 6.20.

Tabel 6.19 Pengujian Validasi Lihat Penilaian Aktivitas Selama Seminggu Terakhir

Pengujian validasi lihat penilaian aktivitas selama seminggu terakhir	
Nama Kasus Uji	Lihat penilaian aktivitas selama seminggu terakhir.
Objek Uji	Kebutuhan fungsional MM-F-005.
Tujuan Pengujian	Menguji sistem mampu mendeteksi ketika rata-rata aktivitas pengguna lebih baik dari kompetitor.
Prosedur Uji	1. Membuka halaman <i>dashboard</i> dengan akun pengguna ketika rata-rata aktivitasnya lebih baik dari kompetitor.
Hasil Pengujian yang diharapkan	Sistem menampilkan pesan “aktivitas anda seminggu ini baik!” dan <i>icon smile</i> .
Hasil pengujian	Sistem menampilkan pesan “aktivitas anda seminggu ini baik!” dan <i>icon smile</i> .
Status	Valid.

Tabel 6.20 Pengujian Validasi Lihat Penilaian Aktivitas Selama Seminggu Terakhir – Alternatif Pertama

Pengujian validasi lihat penilaian aktivitas selama seminggu terakhir – alternatif pertama	
Nama Kasus Uji	Lihat penilaian aktivitas selama seminggu terakhir - alternatif.
Objek Uji	Kebutuhan fungsional MM-F-005.
Tujuan Pengujian	Menguji sistem mampu mendeteksi ketika rata-rata aktivitas kompetitor lebih baik dari pengguna.
Prosedur Uji	1. Membuka halaman <i>dashboard</i> dengan akun pengguna ketika rata-rata aktivitas kompetitornya lebih baik dari pengguna.
Hasil Pengujian yang diharapkan	Sistem menampilkan pesan “aktivitas anda seminggu ini perlu ditingkatkan!” dan <i>icon frown</i> .
Hasil pengujian	Sistem menampilkan pesan “aktivitas anda seminggu ini perlu ditingkatkan!” dan <i>icon frown</i> .
Status	Valid.

6.3.6 Lihat Rekomendasi Dalam Pengelolaan Akun Twitter

Pengujian validasi lihat rekomendasi dalam pengelolaan akun Twitter dengan kode kebutuhan MM-F-006 akan menguji sistem mampu menampilkan rekomendasi untuk menanggapi tren topik Indonesia. Pengujian validasi lihat rekomendasi dalam pengelolaan akun Twitter alternatif pertama akan menguji

sistem mampu mendeteksi kondisi pengguna belum melakukan *posting tweet* pada hari tersebut. Pengujian validasi lihat rekomendasi dalam pengelolaan akun Twitter alternatif kedua akan menguji sistem mampu mendeteksi kondisi rata-rata *posting tweet* kompetitor lebih banyak dari pengguna. Pengujian validasi lihat rekomendasi dalam pengelolaan akun Twitter alternatif ketiga akan menguji sistem mampu mendeteksi kondisi pengguna belum menanggapi *mention* pada hari tersebut. Pengujian validasi lihat rekomendasi dalam pengelolaan akun Twitter akan dijelaskan pada Tabel 6.21, pengujian validasi lihat rekomendasi dalam pengelolaan akun Twitter alternatif pertama akan dijelaskan pada Tabel 6.22, pengujian validasi lihat rekomendasi dalam pengelolaan akun Twitter alternatif kedua akan dijelaskan pada Tabel 6.23, pengujian validasi lihat rekomendasi dalam pengelolaan akun Twitter alternatif ketiga akan dijelaskan pada Tabel 6.24.

Tabel 6.21 Pengujian Validasi Lihat Rekomendasi Dalam Pengelolaan Akun Twitter

Pengujian validasi lihat rekomendasi dalam pengelolaan akun Twitter.	
Nama Kasus Uji	Lihat rekomendasi dalam pengelolaan akun Twitter.
Objek Uji	Kebutuhan fungsional MM-F-006.
Tujuan Pengujian	Menguji sistem mampu menampilkan rekomendasi jam <i>posting</i> dan pesan “menanggapi tren topik hari ini”.
Prosedur Uji	1. Membuka halaman <i>dashboard</i> akun pengguna dengan kondisi pengguna telah melakukan <i>posting tweet</i> dan <i>mention</i> pada hari tersebut juga jumlah rata-rata <i>tweet</i> pengguna lebih baik dari kompetitor.
Hasil Pengujian yang diharapkan	Sistem menampilkan rekomendasi jam <i>posting</i> dan pesan “menanggapi tren topik hari ini”.
Hasil pengujian	Sistem menampilkan rekomendasi jam <i>posting</i> dan pesan “menanggapi tren topik hari ini”.
Status	Valid.

Tabel 6.22 Pengujian Validasi Lihat Rekomendasi Dalam Pengelolaan Akun Twitter – Alternatif Pertama

Pengujian validasi lihat rekomendasi dalam pengelolaan akun Twitter – alternatif pertama	
Nama Kasus Uji	Lihat rekomendasi dalam pengelolaan akun Twitter – alternatif pertama.
Objek Uji	Kebutuhan fungsional MM-F-006.
Tujuan Pengujian	Menguji sistem mampu menampilkan rekomendasi pesan “ <i>posting tweet</i> hari ini”

Tabel 6.22 Pengujian Validasi Lihat Rekomendasi Dalam Pengelolaan Akun Twitter – Alternatif Pertama (Lanjutan)

Pengujian validasi lihat rekomendasi dalam pengelolaan akun Twitter – alternatif pertama (Lanjutan)	
Prosedur Uji	1. Membuka halaman <i>dashboard</i> akun pengguna dnengan kondisi pengguna belum melakukan <i>posting tweet</i> pada hari tersebut.
Hasil Pengujian yang diharapkan	Sistem menampilkan rekomendasi pesan “ <i>posting tweet hari ini</i> ”.
Hasil pengujian	Sistem menampilkan rekomendasi pesan “ <i>posting tweet hari ini</i> ”.
Status	Valid.

Tabel 6.23 Pengujian Validasi Lihat Rekomendasi Dalam Pengelolaan Akun Twitter – Alternatif Kedua

Pengujian validasi lihat rekomendasi dalam pengelolaan akun Twitter - alternatif kedua	
Nama Kasus Uji	Lihat rekomendasi dalam pengelolaan akun Twitter – alternatif kedua.
Objek Uji	Kebutuhan fungsional MM-F-006.
Tujuan Pengujian	Menguji sistem mampu menampilkan rekomendasi pesan “sebaiknya anda <i>posting sejumlah tweet lagi</i> ”
Prosedur Uji	1. Membuka halaman <i>dashboard</i> akun pengguna dengan kondisi rata-rata <i>posting tweet</i> pengguna lebih buruk dibandingkan dengan kompetitor.
Hasil Pengujian yang diharapkan	Sistem menampilkan rekomendasi pesan “sebaiknya anda <i>posting sejumlah tweet lagi</i> ”.
Hasil pengujian	Sistem menampilkan rekomendasi pesan “sebaiknya anda <i>posting sejumlah tweet lagi</i> ”.
Status	Valid.

Tabel 6.24 Pengujian Validasi Lihat Rekomendasi Dalam Pengelolaan Akun Twitter – Alternatif Ketiga

Pengujian validasi lihat rekomendasi dalam pengelolaan akun Twitter – alternatif ketiga	
Nama Kasus Uji	Lihat rekomendasi dalam pengelolaan akun Twitter – alternatif ketiga.
Objek Uji	Kebutuhan fungsional MM-F-006.
Tujuan Pengujian	Menguji sistem mampu menampilkan rekomendasi pesan “menanggapi <i>mention</i> masuk kepada anda”.
Prosedur Uji	1. Membuka halaman <i>dashboard</i> ketika pengguna belum melakukan <i>posting mention</i> pada hari tersebut.
Hasil Pengujian yang diharapkan	Sistem menampilkan rekomendasi pesan “menanggapi <i>mention</i> masuk kepada anda”.
Hasil pengujian	Sistem menampilkan rekomendasi pesan “menanggapi <i>mention</i> masuk kepada anda”.
Status	Valid.

6.3.7 Lihat Rata-Rata Peningkatan *Followers* Pengguna dan Kompetitor

Pengujian validasi lihat rata-rata peningkatan *followers* pengguna dan kompetitor dengan kode kebutuhan MM-F-007 akan menguji sistem mampu menampilkan rata-rata peningkatan *followers* pengguna dan kompetitor. Pengujian validasi lihat rata-rata peningkatan *followers* pengguna dan kompetitor alternatif pertama akan menguji sistem mampu mendeteksi akun pengguna yang tidak memiliki kompetitor. Pengujian validasi lihat rata-rata peningkatan *followers* pengguna dan kompetitor akan dijelaskan pada Tabel 6.25 lalu pengujian validasi lihat rata-rata peningkatan *followers* pengguna dan kompetitor alternatif pertama akan dijelaskan pada Tabel 6.26.

Tabel 6.25 Pengujian Validasi Lihat Rata-Rata Peningkatan *Followers* Pengguna dan Kompetitor

Pengujian validasi lihat rata-rata peningkatan <i>followers</i> pengguna dan kompetitor	
Nama Kasus Uji	Lihat rata-rata peningkatan <i>followers</i> pengguna dan kompetitor
Objek Uji	Kebutuhan fungsional MM-F-007.
Tujuan Pengujian	Menguji sistem mampu menampilkan rata-rata peningkatan <i>followers</i> pengguna dan kompetitor.

Tabel 6.25 Pengujian Validasi Lihat Rata-Rata Peningkatan *Followers* Pengguna dan Kompetitor (Lanjutan)

Pengujian validasi lihat rata-rata peningkatan <i>followers</i> pengguna dan kompetitor (Lanjutan)	
Prosedur Uji	1. Membuka halaman <i>dashboard</i> oleh akun pengguna yang memiliki kompetitor. 2. Menekan <i>card</i> rata-rata peningkatan <i>followers</i> .
Hasil Pengujian yang diharapkan	Sistem menampilkan rata-rata peningkatan <i>followers</i> pengguna dan kompetitor.
Hasil pengujian	Sistem menampilkan rata-rata peningkatan <i>followers</i> pengguna dan kompetitor.
Status	Valid.

Tabel 6.26 Pengujian Validasi Lihat Rata-Rata Peningkatan *Followers* Pengguna dan Kompetitor – Alternatif Pertama

Pengujian validasi lihat rata-rata peningkatan <i>followers</i> pengguna dan kompetitor – alternatif pertama	
Nama Kasus Uji	Lihat rata-rata peningkatan <i>followers</i> pengguna dan kompetitor – alternatif pertama.
Objek Uji	Kebutuhan fungsional MM-F-007.
Tujuan Pengujian	Menguji sistem mampu mendeteksi pengguna yang tidak memiliki kompetitor.
Prosedur Uji	1. Membuka halaman <i>dashboard</i> oleh akun pengguna yang tidak memiliki kompetitor. 2. Menekan <i>card</i> rata-rata peningkatan <i>followers</i> .
Hasil Pengujian yang diharapkan	Sistem menampilkan pesan “kompetitor tidak ditemukan”.
Hasil pengujian	Sistem menampilkan pesan “kompetitor tidak ditemukan”.
Status	Valid.

6.3.8 Lihat Rata-Rata *Posting Tweet* Pengguna dan Kompetitor

Pengujian validasi lihat rata-rata *posting tweet* pengguna dan kompetitor dengan kode kebutuhan MM-F-008 akan menguji sistem mampu menampilkan rata-rata *posting tweet* pengguna dan kompetitor. Pengujian validasi lihat rata-rata *posting tweet* pengguna dan kompetitor alternatif pertama akan menguji

sistem mampu mendeteksi akun pengguna yang tidak memiliki kompetitor. Pengujian validasi lihat rata-rata *posting tweet* pengguna dan kompetitor akan dijelaskan pada Tabel 6.27 dan pengujian validasi lihat rata-rata *posting tweet* pengguna dan kompetitor alternatif pertama akan dijelaskan pada Tabel 6.28.

Tabel 6.27 Pengujian Validasi Lihat Rata-Rata *Posting Tweet* Pengguna dan Kompetitor

Pengujian validasi lihat rata-rata <i>posting tweet</i> pengguna dan kompetitor	
Nama Kasus Uji	Lihat rata-rata <i>posting tweet</i> pengguna dan kompetitor.
Objek Uji	Kebutuhan fungsional MM-F-008.
Tujuan Pengujian	Menguji sistem mampu menampilkan rata-rata <i>posting tweet</i> pengguna dan kompetitor.
Prosedur Uji	1. Membuka halaman <i>dashboard</i> oleh akun pengguna yang memiliki kompetitor. 2. Menekan <i>card</i> rata-rata <i>posting tweet</i> .
Hasil Pengujian yang diharapkan	Sistem menampilkan rata-rata <i>posting tweet</i> pengguna dan kompetitor.
Hasil pengujian	Sistem menampilkan rata-rata <i>posting tweet</i> pengguna dan kompetitor.
Status	Valid.

Tabel 6.28 Pengujian Validasi Lihat Rata-Rata *Posting Tweet* Pengguna dan Kompetitor – Alternatif Pertama

Pengujian validasi lihat rata-rata <i>posting tweet</i> pengguna dan kompetitor – alternatif pertama	
Nama Kasus Uji	Lihat rata-rata <i>posting tweet</i> pengguna dan kompetitor – alternatif pertama.
Objek Uji	Kebutuhan fungsional MM-F-008.
Tujuan Pengujian	Menguji sistem mampu mendeteksi pengguna yang tidak memiliki kompetitor.
Prosedur Uji	1. Membuka halaman <i>dashboard</i> oleh akun pengguna yang tidak memiliki kompetitor. 2. Menekan <i>card</i> rata-rata <i>posting tweet</i> .
Hasil Pengujian yang diharapkan	Sistem menampilkan pesan “kompetitor tidak ditemukan”.
Hasil pengujian	Sistem menampilkan pesan “kompetitor tidak ditemukan”.
Status	Valid.

6.3.9 Lihat Rata-Rata *Retweet* Pengguna dan Kompetitor

Pengujian validasi lihat rata-rata *retweet* pengguna dan kompetitor dengan kode kebutuhan MM-F-009 akan menguji sistem mampu menampilkan rata-rata *retweet* pengguna dan kompetitor. Pengujian validasi lihat rata-rata *retweet* pengguna dan kompetitor alternatif pertama akan menguji sistem dapat mendeteksi akun pengguna yang tidak memiliki kompetitor. Pengujian validasi lihat rata-rata *retweet* pengguna dan kompetitor akan dijelaskan pada Tabel 6.29 dan pengujian validasi lihat rata-rata *retweet* pengguna dan kompetitor alternatif pertama akan dijelaskan pada Tabel 6.30.

Tabel 6.29 Pengujian Validasi Lihat Rata-Rata *Retweet* Pengguna dan Kompetitor

Pengujian validasi lihat rata-rata <i>retweet</i> pengguna dan kompetitor	
Nama Kasus Uji	Lihat rata-rata <i>retweet</i> pengguna dan kompetitor.
Objek Uji	Kebutuhan fungsional MM-F-009.
Tujuan Pengujian	Menguji sistem mampu menampilkan rata-rata <i>retweet</i> pengguna dan kompetitor.
Prosedur Uji	<ol style="list-style-type: none"> 1. Membuka halaman <i>dashboard</i> oleh akun pengguna yang memiliki kompetitor. 2. Menekan <i>card</i> rata-rata <i>retweet</i>.
Hasil Pengujian yang diharapkan	Sistem menampilkan rata-rata <i>retweet</i> pengguna dan kompetitor.
Hasil pengujian	Sistem menampilkan rata-rata <i>retweet</i> pengguna dan kompetitor.
Status	Valid.

Tabel 6.30 Pengujian Validasi Lihat Rata-Rata *Retweet* Pengguna dan Kompetitor – Alternatif Pertama

Pengujian validasi lihat rata-rata <i>retweet</i> pengguna dan kompetitor – alternatif pertama	
Nama Kasus Uji	Lihat rata-rata <i>retweet</i> pengguna dan kompetitor – alternatif pertama.
Objek Uji	Kebutuhan fungsional MM-F-009.
Tujuan Pengujian	Menguji sistem mampu mendeteksi pengguna yang tidak memiliki kompetitor.
Prosedur Uji	<ol style="list-style-type: none"> 1. Membuka halaman <i>dashboard</i> oleh akun pengguna yang tidak memiliki kompetitor. 2. Menekan <i>card</i> rata-rata <i>retweet</i>.

Tabel 6.30 Pengujian Validasi Lihat Rata-Rata *Retweet* Pengguna dan Kompetitor – Alternatif Pertama (Lanjutan)

Pengujian validasi lihat rata-rata <i>retweet</i> pengguna dan kompetitor – alternatif pertama (Lanjutan)	
Hasil Pengujian yang diharapkan	Sistem menampilkan pesan “kompetitor tidak ditemukan”.
Hasil pengujian	Sistem menampilkan pesan “kompetitor tidak ditemukan”.
Status	Valid.

6.3.10 Lihat Rata-Rata *Likes* Pengguna dan Kompetitor

Pengujian validasi lihat *likes* pengguna dan kompetitor dengan kode kebutuhan MM-F-010 akan menguji sistem mampu menampilkan rata-rata *likes* pengguna dan kompetitor. Pengujian validasi lihat *likes* pengguna dan kompetitor alternatif pertama akan menguji sistem mampu mendeteksi akun pengguna yang tidak memiliki kompetitor. Pengujian validasi lihat *likes* pengguna dan kompetitor akan dijelaskan pada Tabel 6.31 dan validasi lihat *likes* pengguna dan kompetitor alternatif pertama akan dijelaskan pada Tabel 6.32.

Tabel 6.31 Pengujian Validasi Lihat *Likes* Pengguna dan Kompetitor

Pengujian validasi lihat <i>likes</i> pengguna dan kompetitor	
Nama Kasus Uji	Lihat rata-rata <i>likes</i> pengguna dan kompetitor.
Objek Uji	Kebutuhan fungsional MM-F-010.
Tujuan Pengujian	Menguji sistem mampu menampilkan rata-rata <i>likes</i> pengguna dan kompetitor.
Prosedur Uji	<ol style="list-style-type: none"> 1. Membuka halaman <i>dashboard</i> oleh akun pengguna yang memiliki kompetitor. 2. Menekan <i>card</i> rata-rata <i>likes</i>.
Hasil Pengujian yang diharapkan	Sistem menampilkan rata-rata <i>likes</i> pengguna dan kompetitor.
Hasil pengujian	Sistem menampilkan rata-rata <i>likes</i> pengguna dan kompetitor.
Status	Valid.

Tabel 6.32 Pengujian Validasi Lihat Rata-Rata *Likes* Pengguna dan Kompetitor – Alternatif Pertama

Pengujian validasi lihat rata-rata <i>likes</i> pengguna dan kompetitor – alternatif pertama	
Nama Kasus Uji	Lihat rata-rata <i>likes</i> pengguna dan kompetitor – alternatif pertama.
Objek Uji	Kebutuhan fungsional MM-F-010.
Tujuan Pengujian	Menguji sistem mampu mendeteksi pengguna yang tidak memiliki kompetitor.
Prosedur Uji	<ol style="list-style-type: none"> 1. Membuka halaman <i>dashboard</i> oleh akun pengguna yang tidak memiliki kompetitor. 2. Menekan <i>card</i> rata-rata <i>likes</i>.
Hasil Pengujian yang diharapkan	Sistem menampilkan pesan “kompetitor tidak ditemukan”.
Hasil pengujian	Sistem menampilkan pesan “kompetitor tidak ditemukan”.
Status	Valid.

6.3.11 Lihat Rata-Rata *Replies* Pengguna dan Kompetitor

Pengujian validasi lihat *replies* pengguna dan kompetitor dengan kode kebutuhan MM-F-011 akan menguji sistem mampu menampilkan rata-rata *replies* pengguna dan kompetitor. Pengujian validasi lihat *replies* pengguna dan kompetitor alternatif pertama akan menguji sistem mampu mendeteksi akun pengguna yang tidak memiliki kompetitor. Pengujian validasi lihat *replies* pengguna dan kompetitor akan dijelaskan pada Tabel 6.33 lalu pengujian validasi lihat *replies* pengguna dan kompetitor alternatif pertama akan dijelaskan pada Tabel 6.34.

Tabel 6.33 Pengujian Validasi Lihat *Replies* Pengguna dan Kompetitor

Pengujian validasi lihat <i>replies</i> pengguna dan kompetitor	
Nama Kasus Uji	Lihat rata-rata <i>replies</i> pengguna dan kompetitor.
Objek Uji	Kebutuhan fungsional MM-F-011.
Tujuan Pengujian	Menguji sistem mampu menampilkan rata-rata <i>replies</i> pengguna dan kompetitor.
Prosedur Uji	<ol style="list-style-type: none"> 1. Membuka halaman <i>dashboard</i> oleh akun pengguna yang memiliki kompetitor. 2. Menekan <i>card</i> rata-rata <i>replies</i>.

Tabel 6.33 Pengujian Validasi Lihat *Replies* Pengguna dan Kompetitor (Lanjutan)

Pengujian validasi lihat <i>replies</i> pengguna dan kompetitor	
Hasil Pengujian yang diharapkan	Sistem menampilkan rata-rata <i>replies</i> pengguna dan kompetitor.
Hasil pengujian	Sistem menampilkan rata-rata <i>replies</i> pengguna dan kompetitor.
Status	Valid.

Tabel 6.34 Pengujian Validasi Lihat Rata-Rata *Replies* Pengguna dan Kompetitor – Alternatif Pertama

Pengujian validasi lihat rata-rata <i>replies</i> pengguna dan kompetitor – alternatif pertama	
Nama Kasus Uji	Lihat rata-rata <i>replies</i> pengguna dan kompetitor – alternatif pertama.
Objek Uji	Kebutuhan fungsional MM-F-011.
Tujuan Pengujian	Menguji sistem mampu mendeteksi pengguna yang tidak memiliki kompetitor.
Prosedur Uji	<ol style="list-style-type: none"> 1. Membuka halaman <i>dashboard</i> oleh akun pengguna yang tidak memiliki kompetitor. 2. Menekan <i>card</i> rata-rata <i>replies</i>.
Hasil Pengujian yang diharapkan	Sistem menampilkan pesan “kompetitor tidak ditemukan”.
Hasil pengujian	Sistem menampilkan pesan “kompetitor tidak ditemukan”.
Status	Valid.

6.3.12 Lihat Hasil Sentimen Analisis

Pengujian validasi lihat hasil sentimen analisis dengan kode kebutuhan MM-F-012 akan menguji sistem mampu menampilkan hasil sentimen analisis. Pengujian validasi lihat hasil sentimen analisis alternatif pertama akan menguji sistem mampu mendeteksi akun pengguna dengan kondisi data sentimen belum diambil. Pengujian validasi lihat hasil sentimen analisis akan dijelaskan pada Tabel 6.35 dan pengujian validasi lihat hasil sentimen analisis alternatif pertama akan dijelaskan pada Tabel 6.36.

Tabel 6.35 Pengujian Validasi Lihat Hasil Sentimen Analisis

Pengujian validasi lihat hasil sentimen analisis	
Nama Kasus Uji	Lihat hasil sentimen analisis
Objek Uji	Kebutuhan fungsional MM-F-012.
Tujuan Pengujian	Menguji sistem mampu menampilkan hasil sentimen analisis.
Prosedur Uji	1. Membuka halaman <i>dashboard</i> oleh akun pengguna dengan data sentimen telah diambil.
Hasil Pengujian yang diharapkan	Sistem menampilkan hasil sentimen analisis.
Hasil pengujian	Sistem menampilkan hasil sentimen analisis.
Status	Valid.

Tabel 6.36 Pengujian Validasi Lihat Hasil Sentimen Analisis – Alternatif Pertama

Pengujian validasi lihat hasil sentimen analisis – alternatif pertama	
Nama Kasus Uji	Lihat hasil sentimen analisis – alternatif pertama.
Objek Uji	Kebutuhan fungsional MM-F-012.
Tujuan Pengujian	Menguji sistem mampu mendeteksi akun pengguna dengan data sentimen belum tersedia.
Prosedur Uji	1. Membuka halaman <i>dashboard</i> oleh akun pengguna dengan data sentimen belum diambil.
Hasil Pengujian yang diharapkan	Sistem menampilkan pesan “Mohon maaf data sentimen belum diambil, silahkan coba sesaat lagi!”.
Hasil pengujian	Sistem menampilkan pesan “Mohon maaf data sentimen belum diambil, silahkan coba sesaat lagi!”.
Status	Valid.

6.3.13 Lihat *Tweet* Terbaik

Pengujian validasi lihat *tweet* terbaik dengan kode kebutuhan MM-F-013 akan menguji sistem mampu menampilkan *tweet* terbaik pengguna. Pengujian validasi lihat *tweet* terbaik alternatif pertama akan menguji sistem mampu mendeteksi akun pengguna dengan kondisi data *tweet* terbaik belum diambil. Pengujian validasi lihat *tweet* terbaik akan dijelaskan pada Tabel 6.37 dan pengujian validasi lihat *tweet* terbaik alternatif pertama akan dijelaskan pada Tabel 6.38.

Tabel 6.37 Pengujian Validasi Lihat Tweet Terbaik

Pengujian validasi lihat tweet terbaik	
Nama Kasus Uji	Lihat tweet terbaik.
Objek Uji	Kebutuhan fungsional MM-F-013.
Tujuan Pengujian	Menguji sistem mampu menampilkan lima tweet terbaik pengguna.
Prosedur Uji	1. Membuka halaman <i>dashboard</i> oleh akun pengguna dengan data tweet telah diambil.
Hasil Pengujian yang diharapkan	Sistem menampilkan lima tweet terbaik pengguna.
Hasil pengujian	Sistem menampilkan lima tweet terbaik pengguna.
Status	Valid.

Tabel 6.38 Pengujian Validasi Lihat Tweet Terbaik – Alternatif Pertama

Pengujian validasi lihat tweet terbaik – alternatif pertama	
Nama Kasus Uji	Lihat tweet terbaik – alternatif pertama.
Objek Uji	Kebutuhan fungsional MM-F-013.
Tujuan Pengujian	Menguji sistem mampu mendeteksi akun pengguna dengan data tweet belum diambil.
Prosedur Uji	1. Membuka halaman <i>dashboard</i> oleh akun pengguna dengan data tweet belum diambil.
Hasil Pengujian yang diharapkan	Sistem menampilkan pesan “Mohon maaf data tweet terbaik belum diambil, silahkan coba sesaat lagi!”.
Hasil pengujian	Sistem menampilkan pesan “Mohon maaf data tweet terbaik belum diambil, silahkan coba sesaat lagi!”.
Status	Valid.

6.3.14 Mengelola Daftar Kompetitor

Pengujian validasi mengelola daftar kompetitor dengan kode kebutuhan MM-F-014 akan menguji sistem mampu menampilkan daftar kompetitor. Pengujian validasi mengelola daftar kompetitor alternatif pertama akan menguji sistem mampu mendeteksi akun pengguna yang tidak memiliki kompetitor. Pengujian validasi mengelola daftar kompetitor akan dijelaskan pada Tabel 6.39 dan pengujian validasi mengelola daftar kompetitor alternatif pertama akan dijelaskan pada Tabel 6.40.

Tabel 6.39 Pengujian Validasi Mengelola Daftar Kompetitor

Pengujian validasi mengelola daftar kompetitor	
Nama Kasus Uji	Mengelola daftar kompetitor.
Objek Uji	Kebutuhan fungsional MM-F-014.
Tujuan Pengujian	Menguji sistem mampu menampilkan daftar kompetitor.
Prosedur Uji	1. Membuka halaman <i>competitor</i> oleh akun pengguna yang memiliki kompetitor.
Hasil Pengujian yang diharapkan	Sistem menampilkan daftar kompetitor.
Hasil pengujian	Sistem menampilkan daftar kompetitor.
Status	Valid.

Tabel 6.40 Pengujian Validasi Mengelola Daftar Kompetitor – Alternatif Pertama

Pengujian validasi mengelola daftar kompetitor – alternatif pertama	
Nama Kasus Uji	Mengelola daftar kompetitor – alternatif pertama.
Objek Uji	Kebutuhan fungsional MM-F-014.
Tujuan Pengujian	Menguji sistem mampu mendeteksi akun pengguna yang tidak memiliki kompetitor.
Prosedur Uji	1. Membuka halaman <i>competitor</i> oleh akun pengguna yang tidak memiliki kompetitor.
Hasil Pengujian yang diharapkan	Sistem menampilkan pesan “kompetitor tidak ditemukan”.
Hasil pengujian	Sistem menampilkan pesan “kompetitor tidak ditemukan”.
Status	Valid.

6.3.15 Tambah Kompetitor

Pengujian validasi tambah kompetitor dengan kode kebutuhan MM-F-015 akan menguji sistem mampu menambahkan kompetitor. Pengujian validasi tambah kompetitor alternatif akan menguji sistem mampu mendeteksi akun pencarian yang telah terdaftar sebagai kompetitor. Pengujian validasi tambah kompetitor akan dijelaskan pada Tabel 6.41 dan pengujian validasi tambah kompetitor alternatif pertama akan dijelaskan pada Tabel 6.42.

Tabel 6.41 Pengujian Validasi Tambah Kompetitor

Pengujian validasi tambah kompetitor	
Nama Kasus Uji	Tambah Kompetitor.
Objek Uji	Kebutuhan fungsional MM-F-015.
Tujuan Pengujian	Menguji sistem mampu menambahkan kompetitor.
Prosedur Uji	<ol style="list-style-type: none"> 1. Membuka halaman <i>competitor</i>. 2. Menekan tombol “+” 3. Memberi masukan “fadlizon” pada <i>field</i> pencarian <i>username</i> Twitter. 4. Menekan tombol “Ya”
Hasil Pengujian yang diharapkan	Sistem menambahkan kompetitor lalu menampilkan daftar kompetitor keseluruhan.
Hasil pengujian	Sistem menambahkan kompetitor lalu menampilkan daftar kompetitor keseluruhan.
Status	Valid.

Tabel 6.42 Pengujian Validasi Tambah Kompetitor – Alternatif Pertama

Pengujian validasi tambah kompetitor – alternatif pertama	
Nama Kasus Uji	Tambah Kompetitor – alternatif pertama.
Objek Uji	Kebutuhan fungsional MM-F-015.
Tujuan Pengujian	Menguji sistem mampu mendeteksi hasil pencarian kompetitor yang sebelumnya telah terdaftar.
Prosedur Uji	<ol style="list-style-type: none"> 1. Membuka halaman <i>competitor</i>. 2. Menekan tombol “+” 3. Memberi masukan “Ferdinand_Haeen” pada <i>field</i> pencarian <i>username</i> Twitter.
Hasil Pengujian yang diharapkan	Sistem mengubah tombol “tambahkan kompetitor” menjadi “X kompetitor”.
Hasil pengujian	Sistem mengubah tombol “tambahkan kompetitor” menjadi “X kompetitor”.
Status	Valid.

6.3.16 Hapus Kompetitor

Pengujian validasi hapus kompetitor dengan kode kebutuhan MM-F-016 akan menguji sistem mampu menghapus kompetitor. Pengujian validasi hapus kompetitor akan dijelaskan pada Tabel 6.43.

Tabel 6.43 Pengujian Validasi Hapus Kompetitor

Pengujian validasi hapus kompetitor	
Nama Kasus Uji	Hapus Kompetitor.
Objek Uji	Kebutuhan fungsional MM-F-016.
Tujuan Pengujian	Menguji sistem mampu menghapus kompetitor.
Prosedur Uji	<ol style="list-style-type: none"> 1. Membuka halaman <i>competitor</i>. 2. Menekan tombol “Hapus” pada kolom kompetitor budimandjatri. 3. Menekan tombol “Ya”
Hasil Pengujian yang diharapkan	Sistem menghapus kompetitor lalu menampilkan daftar kompetitor keseluruhan
Hasil pengujian	Sistem menghapus kompetitor lalu menampilkan daftar kompetitor keseluruhan.
Status	Valid.

6.3.17 Lihat Detail Perbandingan Aktivitas Pengguna dan Kompetitor

Pengujian validasi lihat detail perbandingan aktivitas pengguna dan kompetitor dengan kode kebutuhan MM-F-017 akan menguji sistem mampu menampilkan detail perbandingan aktivitas pengguna dan kompetitor. Pengujian validasi lihat detail perbandingan aktivitas pengguna dan kompetitor akan dijelaskan pada Tabel 6.44.

Tabel 6.44 Pengujian validasi lihat detail perbandingan aktivitas pengguna dan kompetitor

Pengujian validasi lihat detail perbandingan aktivitas pengguna dan kompetitor	
Nama Kasus Uji	Lihat detail perbandingan aktivitas pengguna dan kompetitor.
Objek Uji	Kebutuhan fungsional MM-F-017.
Tujuan Pengujian	Menguji sistem menampilkan detail perbandingan aktivitas pengguna dan kompetitor.
Prosedur Uji	<ol style="list-style-type: none"> 1. Membuka halaman <i>competitor</i>. 2. Menekan tombol “detail” pada kolom kompetitor fadlizon.

Tabel 6.44 Pengujian Validasi Lihat Detail Perbandingan Aktivitas Pengguna dan Kompetitor (Lanjutan)

Pengujian validasi lihat detail perbandingan aktivitas pengguna dan kompetitor (Lanjutan)	
Hasil Pengujian yang diharapkan	Sistem menampilkan detail perbandingan aktivitas pengguna dan kompetitor.
Hasil pengujian	Sistem menampilkan detail perbandingan aktivitas pengguna dan kompetitor.
Status	Valid.

6.3.18 Posting Tweet

Pengujian validasi *posting tweet* dengan kode kebutuhan MM-F-018 akan menguji sistem mampu menampilkan *pop-up posting tweet*. Pengujian validasi *posting tweet* akan dijelaskan pada Tabel 6.45.

Tabel 6.45 Pengujian Validasi Posting Tweet

Pengujian validasi <i>posting tweet</i>	
Nama Kasus Uji	<i>Posting tweet</i> .
Objek Uji	Kebutuhan fungsional MM-F-018.
Tujuan Pengujian	Menguji sistem dapat menampilkan <i>pop-up posting tweet</i> .
Prosedur Uji	1. Menekan tombol <i>icon Twitter</i> pada <i>navbar</i> .
Hasil Pengujian yang diharapkan	Sistem menampilkan <i>pop-up posting tweet</i> .
Hasil pengujian	Sistem menampilkan <i>pop-up posting tweet</i> .
Status	Valid.

6.3.19 Lihat Tren Topik Indonesia

Pengujian validasi lihat tren topik Indonesia dengan kode kebutuhan MM-F-019 akan menguji sistem mampu menampilkan 10 daftar tren topik Indonesia. Pengujian validasi lihat tren topik Indonesia akan dijelaskan pada Tabel 6.46.

Tabel 6.46 Pengujian Validasi Lihat Tren Topik Indonesia

Pengujian validasi lihat tren topik Indonesia	
Nama Kasus Uji	Lihat tren topik Indonesia.
Objek Uji	Kebutuhan fungsional MM-F-019.
Tujuan Pengujian	Menguji sistem dapat menampilkan 10 daftar tren topik Indonesia.
Prosedur Uji	1. Membuka halaman <i>trend</i> .
Hasil Pengujian yang diharapkan	Sistem menampilkan 10 daftar tren topik Indonesia.
Hasil pengujian	Sistem menampilkan 10 daftar tren topik Indonesia.
Status	Valid.

6.3.20 Lihat Contoh *Tweet* Berdasarkan Tren

Pengujian validasi lihat contoh *tweet* berdasarkan tren dengan kode kebutuhan MM-F-020 akan menguji sistem mampu menampilkan *pop-up* berisi contoh *tweet* berdasarkan tren. Pengujian validasi lihat contoh *tweet* berdasarkan tren akan dijelaskan pada Tabel 6.47.

Tabel 6.47 Pengujian Validasi Lihat Contoh *Tweet* Berdasarkan Tren

Pengujian validasi lihat contoh <i>tweet</i> berdasarkan tren	
Nama Kasus Uji	Lihat contoh <i>tweet</i> berdasarkan tren.
Objek Uji	Kebutuhan fungsional MM-F-020.
Tujuan Pengujian	Menguji sistem dapat menampilkan <i>pop-up</i> berisi contoh <i>tweet</i> berdasarkan tren.
Prosedur Uji	1. Membuka halaman <i>trend</i> . 2. Menekan tombol “detail” pada kolom tren #17April2019PeoplePower.
Hasil Pengujian yang diharapkan	Sistem menampilkan <i>pop-up</i> berisi 10 contoh <i>tweet</i> mengenai #17April2019PeoplePower.
Hasil pengujian	Sistem menampilkan <i>pop-up</i> berisi 10 contoh <i>tweet</i> mengenai #17April2019PeoplePower.
Status	Valid.

6.3.21 Cari Akun Twitter

Pengujian validasi cari akun Twitter dengan kode kebutuhan MM-F-021 akan menguji sistem dapat melakukan pencarian akun Twitter. Pengujian validasi cari akun Twitter alternatif pertama akan menguji sistem dapat mendeteksi apabila hasil pencarian tidak ditemukan. Pengujian validasi cari akun Twitter akan dijelaskan pada Tabel 6.48 dan pengujian validasi cari akun Twitter alternatif pertama akan dijelaskan pada Tabel 6.49.

Tabel 6.48 Pengujian Validasi Cari Akun Twitter

Pengujian validasi cari akun Twitter	
Nama Kasus Uji	Cari akun Twitter.
Objek Uji	Kebutuhan fungsional MM-F-021.
Tujuan Pengujian	Menguji sistem dapat melakukan pencarian akun Twitter.
Prosedur Uji	<ol style="list-style-type: none"> 1. Membuka halaman pencarian akun. 2. Memberi masukan fadlizon pada <i>field username</i>. 3. Menekan tombol pencarian.
Hasil Pengujian yang diharapkan	Sistem menampilkan profil akun Twitter fadlizon.
Hasil pengujian	Sistem menampilkan profil akun Twitter fadlizon.
Status	Valid.

Tabel 6.49 Pengujian Validasi Cari Akun Twitter – Alternatif Pertama

Pengujian validasi cari akun Twitter – alternatif pertama	
Nama Kasus Uji	Cari akun Twitter – alternatif pertama.
Objek Uji	Kebutuhan fungsional MM-F-021.
Tujuan Pengujian	Menguji sistem dapat memberikan pesan apabila hasil pencarian tidak ditemukan.
Prosedur Uji	<ol style="list-style-type: none"> 1. Membuka halaman pencarian akun. 2. Memberi masukan testingmomment pada <i>field username</i>. 3. Menekan tombol pencarian.
Hasil Pengujian yang diharapkan	Sistem menampilkan pesan “Mohon maaf akun testingmomment tidak ditemukan!”.
Hasil pengujian	Sistem menampilkan pesan “Mohon maaf akun testingmomment tidak ditemukan!”.
Status	Valid.

6.3.22 Update Data

Pengujian validasi *update* data dengan kode kebutuhan MM-F-022 akan menguji sistem mampu melakukan *update* data melalui sistem. Pengujian validasi *update* data akan dijelaskan pada Tabel 6.50.

Tabel 6.50 Pengujian Validasi Update Data

Pengujian validasi <i>update</i> data	
Nama Kasus Uji	<i>Update</i> data.
Objek Uji	Kebutuhan fungsional MM-F-022.
Tujuan Pengujian	Menguji sistem mampu melakukan <i>update</i> data melalui sistem.
Prosedur Uji	1. Menekan tombol <i>update</i> data.
Hasil Pengujian yang diharapkan	Data aktivitas akun Twitter pengguna berhasil <i>diupdate</i> .
Hasil pengujian	Data aktivitas akun Twitter pengguna berhasil <i>diupdate</i> .
Status	Valid.

6.3.23 Logout

Pengujian validasi *logout* dengan kode kebutuhan MM-F-023 akan menguji sistem mampu mengeluarkan pengguna dari sistem. Pengujian validasi *logout* akan dijelaskan pada Tabel 6.51.

Tabel 6.51 Pengujian Validasi Logout

Pengujian validasi <i>logout</i>	
Nama Kasus Uji	<i>Logout</i> .
Objek Uji	Kebutuhan fungsional MM-F-023.
Tujuan Pengujian	Menguji sistem dapat mengeluarkan pengguna dari sistem menjadi tamu.
Prosedur Uji	1. Menekan tombol <i>logout</i> .
Hasil Pengujian yang diharapkan	Sistem menampilkan halaman <i>login</i> .
Hasil pengujian	Sistem menampilkan halaman <i>login</i> .
Status	Valid.

6.4 Pengujian *Usability*

Pengujian *usability* dilakukan untuk menguji tingkat kemudahan sistem untuk digunakan oleh pengguna. Dalam penelitian ini metode yang digunakan untuk mengukur *usability* sistem adalah *heuristic evaluation* (HE). Pengujian dilakukan dengan melibatkan lima evaluator dari tim *analytics* PT Astra Graphia Information Technology. Proses pengujian dimulai dari evaluator melakukan evaluasi terhadap sistem untuk menemukan kejanggalan pada sistem yang diteliti. Kejanggalan yang ditemukan kemudian dikategorikan sesuai dengan 10 aspek HE sekaligus memberikan nilai *severity ratings* sesuai dengan tingkat keparahan dari kejanggalan yang ditemukan. Nilai *severity ratings* tersebut nantinya akan menjadi acuan dalam pertimbangan untuk melakukan perubahan pada sistem.

6.4.1 Analisis Hasil Pengujian *Usability*

Setelah melalui pengujian yang melibatkan lima evaluator dari tim *analytics* PT Astra Graphia Information Technology ditemukan kejanggalan sistem pada dua aspek HE yaitu *visibillity of system status* dan *user control and freedom* yang akan dijelaskan lebih lanjut pada Tabel 6.52.

Tabel 6.52 Hasil Pengujian *Heuristic Evaluation*

No	Aspek <i>Heuristic Evaluation</i>	<i>Severity Ratings</i>	Keterangan
1	<i>Visibility of System Status</i>	1	1. Informasi yang kurang jelas pada halaman <i>dashboard</i> terkait kompetitor. Apakah rata2 semua kompetitor atau satu kompetitor.
2	<i>User Control and Freedom</i>	1	1. Pada <i>pop-up</i> detail tren topik tidak terdapat <i>button close</i> , sehingga dapat menyulitkan pengguna baru.

Berdasarkan hasil pengujian HE terdapat temuan kejanggalan pada aspek *visibillity of system status* dengan *severity ratings* bernilai 1 yaitu kurangnya informasi pada halaman *dashboard* mengenai keterangan rata-rata kompetitor. Selain itu juga terdapat temuan kejanggalan pada aspek *user control and freedom* dengan *severity ratings* bernilai 1 yaitu tidak ditemukannya tombol *close* pada *pop-up* detail tren topik sehingga dapat menyulitkan pengguna baru. Hasil temuan kejanggalan sistem memiliki tingkat *severity ratings* bernilai 1 yang berarti tidak dibutuhkan perubahan mendesak pada sistem. Sehingga dapat disimpulkan bahwa sistem sudah cukup baik untuk digunakan oleh pengguna.

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan didapatkan kesimpulan sebagai berikut:

1. Analisis kebutuhan yang dilakukan dengan metode *brainstorming* pada PT Astra Graphia Information Technology menghasilkan 23 kebutuhan fungsional dan 1 ketutuhan non fungsional. Berdasarkan hasil analisis kebutuhan tersebut kemudian menghasilkan pemodelan dengan pendekatan *object oriented* (OO) dalam bentuk *use case diagram* dan *use case scenario*.
2. Hasil perancangan pada penelitian ini menghasilkan perancangan arsitektur yang terdiri dari *sequence diagram* dan *class diagram*, perancangan data yang terdiri dari *Entity Relationship Diagram* (ERD) yang kemudian menjadi dasar dalam pembuatan *Physical Data Model* (PDM), perancangan komponen, dan perancangan antarmuka yang menjadi dasar *layout* dalam implementasi sistem. Kemudian pada tahap implementasi menghasilkan penjabaran spesifikasi sistem, implementasi kode program, implementasi basis data, dan implementasi antarmuka sesuai dengan perancangan yang telah didefinisikan sebelumnya.
3. Hasil dari tahap pengujian yang dilakukan dengan cara pengujian unit, pengujian integrasi, dan pengujian validasi menghasilkan 100% valid. Pada pengujian *usability* ditemukan beberapa kejanggalan pada sistem, namun sesuai dengan tingkat *severity ratings*nya tidak dibutuhkan perbaikan pada sistem dan sistem sudah cukup baik digunakan oleh pengguna.

7.2 Saran

Saran yang diberikan untuk pengembangan lanjut sistem untuk memonitor aktivitas media sosial antara lain sebagai berikut:

1. Menambah informasi yang akan dimonitor pada akun media sosial untuk memperluas pengetahuan pengguna terhadap kondisi akun media sosialnya.
2. Melakukan optimasi pada metode analisis sentimen serta menambah metode untuk mendeteksi kategori tren topik tertentu sehingga pengguna dapat memilih kategori tren secara spesifik.
3. Melakukan pengujian pada aspek *compatibility* untuk menguji bahwa sistem yang telah dikembangkan dapat digunakan di berbagai jenis *browser*.

DAFTAR REFERENSI

- Alshamrani, A. dan Abdullah, B., 2015. A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model. *International Journal of Computer Science Issues (IJCSI)*, [daring] 12(1), hal.106–111. Tersedia pada: <https://s3.amazonaws.com/academia.edu.documents/36637147/SDLC.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1548951809&Signature=txRfv2w1RPUOR%2B%2FHKagtJdrxflU%3D&response-content-disposition=inline%3Bfilename%3DA_Comparison_Between_Three_SDLC_Models_>>.
- Ardha, B., 2014. Social Media Sebagai Media Kampanye Partai Politik 2014 Di Indonesia. *Jurnal Visi Komunikasi*, 13(01), hal.105–120.
- Banjarnahor, B.B. dan Hartomo, K.D., 2016. Penerapan Laravel Framework Dalam Perancangan Sistem Informasi Promosi Produk Unggulan UKM Berbasis Web (Studi Kasus Dinas Perindustrian Perdagangan dan UMKM Kota Salatiga) Artikel. *IOSR Journal of Economics and Finance*, [daring] 3(1), hal.56. Tersedia pada: <https://www.bertelsmann-stiftung.de/fileadmin/files/BSt/Publikationen/GrauePublikationen/MT_Globalization_Report_2018.pdf><http://eprints.lse.ac.uk/43447/1/India_globalisation%2C_society_and_inequalities%28Isero%29.pdf><<https://www.quora.com/What-is-the>>.
- Basri, 2017. Metode Weightd Product (Wp) Dalam Sistem Pendukung Keputusan Penerimaan Beasiswa Prestasi. *Jurnal INSYPRO (Information System and Processing)*, 2(1), hal.1–6.
- Dang-xuan, S.S.L., 2014. Social Media and Political Communication - A Social Media Analytics Social media and political communication : a social media analytics framework. (January).
- Fatimah, S., 2018. Kampanye sebagai Komunikasi Politik : Esensi dan Strategi dalam Pemilu. 1(1), hal.5–16.
- Fitri, A., 2019. DINAMIKA DAN TANTANGAN JELANG PEMILU PRESIDEN TAHUN 2019 Adelia Fitri. 3(01), hal.113–131.
- Josi, A., Abdillah, L.A. dan Suryayusra, 2014. Penerapan Teknik Web Scraping Pada Mesin Pencari Artikel Ilmiah. *Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Bina Darma*.
- Kartini, N.A. dan Bachtiar, F.A., 2019. Klasifikasi Rating Berdasarkan Komentar Tempat Wisata Di Media Sosial Dengan Menggunakan Metode Fuzzy K-Nearest Neighbor. 3(5), hal.4164–4169.
- Kurniawan, A., Rokhmawati, R.I. dan Rachmadi, A., 2018. Evaluasi User Experience dengan Metode Heuristic Evaluation dan Persona (Studi pada : Situs Web Dalang Ki Purbo Asmoro). *Jurnal Pengembangan Teknologi Informasi dan Ilmu*

- Komputer*, [daring] 2(8), hal.2918–2926. Tersedia pada: <<http://j-ptiik.ub.ac.id>>.
- Kusumawati, I., 2017. Analisa Sentimen Menggunakan Lexicon Based Kenaikan Harga Rokok Pada Media Sosial Twitter.
- Lesmana, G.N.A., 2017. Tesis: Analisis Pengaruh Media Sosial Twitter Terhadap Pembentukan Brand Attachment (Studi: PT. XL AXIATA), (Program Magister Manajemen, Fakultas Ekonomi, Universitas Indonesia). hal, 10-11 1. hal.10–11.
- Pastima Simanjuntak & Arwin Kasnady, 2016. Analisis Model View Controller (Mvc) Pada Bahasa Php. *Jurnal ISD*, 2(2), hal.56–66.
- Pressman, R.S., 2010. *CS605-Software Engineering Practitioner's Approach*. [daring] Tersedia pada: <https://www.google.com/mt/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=0ahUKEwi75_374OPTAhXia5oKHd4HDmkQFggzMAI&url=http%3A%2F%2Fdewiar.staff.gunadarma.ac.id%2FDownloads%2Ffiles%2F13560%2FEBookPressman__Software_Engineering.pdf&usg=AFQjCNHbI0figoqsZUtlICO>.
- Rosmala, D. dan Syafei, R.R., 2012. Impelementasi Web Crawler Pada Social Media Monitoring. *Jurusan Teknik Informatika Institut Teknologi Nasional Bandung*, 2(2), hal.44–52.
- Soegoto, E.S., 2018. Implementing Laravel framework website as brand image in higher-education institution. *IOP Conference Series: Materials Science and Engineering*, 407(1).
- Sommerville, I., 2011. *Ninth edition of the G.B. Morgagni Awards Program*. *European Journal of Clinical Investigation*, .
- Suendri, 2018. PENERAPAN KONSEP MODEL VIEW CONTROLLER PADA PERANCANGAN SISTEM MANAJEMEN. 3(2), hal.36–45.
- Vargiu, E. dan Urru, M., 2012. Exploiting web scraping in a collaborative filtering-based approach to web advertising. *Artificial Intelligence Research*, [daring] 2(1), hal.44–54. Tersedia pada: <<http://www.sciedu.ca/journal/index.php/air/article/view/1390>>.
- Vonderschmitt, K., 2012. The Growing Use of Social Media in Political Campaigns : How to use Facebook , Twitter and YouTube to Create an Effective Social Media Campaign. *International Journal of Social Sciences and Entrepreneurship*, 2(3), hal.25–29.

LAMPIRAN A KETERANGAN MELAKUKAN PENELITIAN

SURAT KETERANGAN MELAKUKAN PENELITIAN

Dengan Hormat,

Yang bertanda tangan di bawah ini, menerangkan bahwa:

Nama : Vasha Farisi Sarwan Halim

NIM : 155150200111053

Program Studi : S1 Teknik Informatika

Konsentrasi : Rekayasa Perangkat Lunak

Adalah benar telah melakukan penelitian pada PT Astra Graphia Information Technology dengan judul "Pengembangan Sistem untuk Memonitor Aktivitas Media Sosial Berbasis Web (Studi Kasus Media Sosial Twitter Calon Legislatif pada Pemilihan Legislatif 2019)" periode 25 Juni 2018 – 20 Juni 2019. Adapun penelitian yang dilakukan meliputi:

1. Elisitasi kebutuhan yang telah diverifikasi dan divalidasi oleh calon pengguna
2. *Usability testing*

Demikian surat keterangan ini dibuat agar dapat digunakan sebagaimana mestinya.

Manager

Mysights Product



agit

Michael Seno Adhy

LAMPIRAN B HASIL BRAINSTROMING

1. Trend
Yang sedang viral di masyarakat → bisa dari aggregator (google trend, twitter, dll)
sdsmed (word cloud)
2. Background influencer
↳ Parameter untuk menentukan interest
3. Menampilkan profile kompetitor dan fitur komparasi
4. Harapan Masyarakat
↳ Untuk menentukan visi dan misi
5. Faktor-faktor yang membuat orang tsb terpilih
↳ caleg yang sudah terpilih tahun periode sebelumnya (Masa kampanye)
6. Landing page berupa action recommendation →
Sumber: Poin 1 dan 3
7. Pendapat masyarakat tentang orang tsb (caleg)
8. Pendapat masyarakat tentang Partai Politik / tokoh (AHY dll)
9. Rekomendasi harus posting jam berapa
10. Grafik perbandingan sosial media di akun tsb.
11. Prioritas lokasi kampanye dan topiknya.
12. Mengetahui jumlah kompetitor per-dapil berdasarkan Partai Politik
13. Rekomendasi potential caleg

LAMPIRAN C MANUALISASI METODE LEXICON BASED SENTIMENT ANALYSIS

1. Sampling Data

No	replies_content
1	Kasihlah kalah hahaha
2	Setiap doa, adalah wujud rasa percaya, dan dengan setiap kerja keras, itu tanda bahwa kita layak menerima apa yang terucap dalam doa.
	Jalan masih panjang bang @FaldoMaldini ... Tetap tegak, gentarkan lawan dengan semangat membara.
3	Aamiin.. Insya Allah jadi
4	Di akhir kegelapan menuju terang akan di hantarkan pajar yg sejuk lebih indah daripada senja yg menghantarkan pulang siang menuju malam yg di penuhi kegelapan.
5	Semangaaat
6	Seperti karang di terpa ombak..... Alam terkembang jadikan guru..... Insya Allah..... Badai pasti berlalu..... Semangat Bung.....kita bisa
7	Sabar @FaldoMaldini kito tunggu proyek Listrik @jokowi selesai biar faldo ndak kalam-kalam. Hihi
8	Maju terus bro @FaldoMaldini jangan pernah berfikir negatif ttng sebuah keputusan.!!! Apa yg anda hari ini perjuangkan akan anda tuai esok.!! Karna allah tak pernah mengingkari janji2nya untuk membalas perbuatan kita.... Anda orang Baik!
9	Semangat udaa, semoga lelah mu menjadi lillah
10	IKHLASKAN Gelap itu hanyalah keterbatasan kemampuan mata menterjemahkan gelombang cahaya, bukan kondisi sebenarnya. Cahaya pun adalah ilusi fikiran yang mendeskripsikan bentuk Gelap tidak membuat kita takut, terang tidak membuat kita kagum
11	Dari Pemalang untuk PRABOWO-SANDI
12	Dalam gelap, hitam dan putih tetap bisa dibedakan.
	#KPUNgejarQuickCount
	#KPUJanganJiplakQuickCount

2. Preprocessing Data

```
# 2. Preprocessing
def preprocessTweet(tweet):

    tweet = tweet.lower()
    tweet = re.sub('((www\.[^\s]+)|(https?://[^\s]+))', 'URL', tweet)
    tweet = re.sub('@[^\s]+', 'AT_USER', tweet)
    tweet = re.sub(r'#([^\s]+)', r'\1', tweet)
    tweet = re.sub(r'[\^\w\s]', '', tweet)
    tweet = re.sub('[\s]+', ' ', tweet)
    tweet = tweet.strip('\ ')
    return tweet
```

kasihan kalah hahaha
 setiap doa adalah wujud rasa percaya dan dengan setiap kerja keras itu tanda bahwa kita layak menerima apa yang terucap dalam d
 oa jalan masih panjang bang AT_USER tetap tegak gentarkan lawan dengan semangat membara
 aamiin insya allah jadi
 di akhir kegelapan menuju terang akan di hantarkan pajar yg sejuk lebih indah daripada senja yg menghantarkan pulang siang menu
 ju malam yg di penuhi kegelapan
 semangaaat
 seperti karang di terpa ombak alam terkembang jadikan guru insya allah badai pasti berlalu semangat bungkita bisa
 sabar AT_USER kito tunggu proyek listrik AT_USER selesai biar faldo ndak kalamkalam hihi
 maju terus bro AT_USER jangan pernah berfikir negatif ttng sebuah keputusan apa yg anda hari ini perjuangkan akan anda tuai eso
 k karna allah tak pernah mengingkari janji2nya untuk membalas perbuatan kita anda orang baik
 semangat udaa semoga lelah mu menjadi lillah
 ikhlaskan gelap itu hanyalah keterbatasan kemampuan mata menterjemahkan gelombang cahaya bukan kondisi sebenarnya cahaya pun ad
 alah ilusi fikiran yang mendeskripsikan bentuk gelap tidak membuat kita takut terang tidak membuat kita kagum
 dari pemalang untuk prabowosandi
 dalam gelap hitam dan putih tetap bisa dibedakan kpungejarquickcount kpujanganjiplakquickcount

3. Pemecahan kata dan pembobotan

```
# 4. pemecahan kata
def getFeatureVector(tweet):
    featureVector = []
    words = tweet.split()
    stopWords = getStopWordList()
    for w in words:
        val = re.search(r"^[a-zA-Z][a-zA-Z0-9]*$", w)
        if(w in stopWords or val is None):
            continue
        else:
            featureVector.append(w.lower())
    return featureVector
```

```
# 5. Lexicon based sentiment analysis
def sentimentAnalysis(repliesId, featureVectors) :
    print(featureVectors)
    sentimentScore = []
    temp = []
    dictionary = open("data/dictio.txt", "r", encoding="utf8")

    for word in dictionary :
        check = word.split("|")
        for featureVector in featureVectors :
            if (check[3] == featureVector and check[3] not in temp) :
                sentimentScore.append(float(check[4]))
                sentimentScore.append(-float(check[5]))
                temp.append(check[3])

    print(sentimentScore)
    print("total nilai polaritas: " + str(sum(sentimentScore)))
    result = sum(sentimentScore)
    if (result > 0) :
        sentiment = 'positif'
        print("sentiment = positif")
    elif (result < 0) :
        sentiment = 'negatif'
        print("sentiment = negatif")
    else :
        sentiment = 'netral'
        print("sentiment = netral")
```

```

['kasihan', 'kalah', 'hahaha']
[0.0, -0.5, 0.333, -0.667]
total nilai polaritas: -0.8340000000000001
sentiment = negatif

['setiap', 'doa', 'adalah', 'wujud', 'rasa', 'percaya', 'dan', 'dengan', 'setiap', 'kerja', 'keras', 'itu', 'tanda', 'bahwa',
'kita', 'layak', 'menerima', 'apa', 'yang', 'terucap', 'dalam', 'doa', 'jalan', 'masih', 'panjang', 'bang', 'tetap', 'tegak',
'gentarkan', 'lawan', 'dengan', 'semangat', 'membara']
[0.125, -0.0, 0.0, -0.0, 0.5, -0.25, 0.0, -0.0, 0.0, -0.0, 0.125, -0.375, 0.125, -0.0, 0.0, -0.0, 0.125, -0.125, 0.5, -0.0, 0.0,
0, -0.125, 0.0, -0.0, 0.0, -0.0, 0.0, -0.0, 0.0, -0.0, 0.125, -0.125, 0.25, -0.0, 0.0, -0.5, 0.0, -0.0, 0.0, -0.0, 0.0, -0.0,
0.0, -0.0, 0.0, -0.0, 0.0, -0.25, 0.0, -0.0, 0.0, -0.0]
total nilai polaritas: 0.125
sentiment = positif

['aamiin', 'insya', 'allah', 'jadi']
[0.0, -0.0]
total nilai polaritas: 0.0
sentiment = netral

['di', 'akhir', 'kegelapan', 'menuju', 'terang', 'akan', 'di', 'hantarkan', 'pajar', 'yg', 'sejuk', 'lebih', 'indah', 'daripad
a', 'senja', 'yg', 'menghantarkan', 'pulang', 'siang', 'menuju', 'malam', 'yg', 'di', 'penuhi', 'kegelapan']
[0.0, -0.0, 0.0, -0.0, 0.375, -0.125, 0.25, -0.25, 0.0, -0.0, 0.125, -0.25, 0.0, -0.5, 0.0, -0.0, 0.125, -0.0, 0.0, -0.0, 0.0,
-0.0, 0.0, -0.0, 0.0, -0.5, 0.0, -0.25, 0.0, -0.0, 0.0, -0.0, 0.0, -0.0]
total nilai polaritas: -1.0
sentiment = negatif

['semangaaat']
[]
total nilai polaritas: 0
sentiment = netral

['seperti', 'karang', 'di', 'terpa', 'ombak', 'alam', 'terkembang', 'jadikan', 'guru', 'insya', 'allah', 'badai', 'pasti', 'ber
lalu', 'semangat', 'bungkita', 'bisa']
[0.0, -0.0, 0.0, -0.0, 0.0, -0.125, 0.0, -0.375, 0.5, -0.0, 0.25, -0.0, 0.0, -0.0, 0.0, -0.0, 0.0, -0.0, 0.0, -0.0, 0.0, -0.0, 0.0,
5, 0.0, -0.0, 0.0, -0.0, 0.125, -0.0]
total nilai polaritas: 0.25
sentiment = positif

['sabar', 'kito', 'tunggu', 'proyek', 'listrik', 'selesai', 'biar', 'faldo', 'ndak', 'kalamkalam', 'hihi']
[0.0, -0.0, 0.375, -0.125, 0.0, -0.0, 0.0, -0.25, 0.125, -0.5, 0.0, -0.0]
total nilai polaritas: -0.375
sentiment = negatif

['maju', 'terus', 'bro', 'jangan', 'pernah', 'berfikir', 'negatif', 'ttng', 'sebuah', 'keputusan', 'apa', 'yg', 'anda', 'hari',
'ini', 'perjuangkan', 'akan', 'anda', 'tuai', 'esok', 'karna', 'allah', 'tak', 'pernah', 'mengingkari', 'janjinya', 'untuk',
'membalas', 'perbuatan', 'kita', 'anda', 'orang', 'baik']
[0.0, -0.0, 0.75, -0.0, 0.0, -0.0, 0.375, -0.0, 0.0, -0.0, 0.0, -0.625, 0.0, -0.0, 0.0, -0.625, 0.0, -0.0, 0.0, -0.5, 0.0, -0.8
75, 0.0, -0.0, 0.0, -0.0, 0.125, -0.0, 0.0, -0.0, 0.0, -0.0, 0.0, -0.125, 0.0, -0.25, 0.0, -0.0, 0.0, -0.0]
total nilai polaritas: -1.75
sentiment = negatif

```



LAMPIRAN D MANUALISASI METODE WEIGHTED PRODUCT

1. Kriteria Dalam Penentuan *Tweet* Terbaik

Kriteria	
Benefit	<i>Positive Sentiment, Retweet Count, Replies Count, Favorite Count</i>
Cost	<i>Negative Sentiment</i>

Kriteria	Bobot
<i>Positive Sentiment</i>	1
Negative Sentiment	1
Retweet Count	3
Replies Count	2
Favorite Count	1

2. Manualisasi *Weighted Product* Dalam Penentuan *Tweet* Terbaik

```
def weightedProduct(tweet_ids, datas) :  
    Wj = []  
    Si = []  
    temp = []  
    Vi = []  
  
    # Penentuan bobot  
    # positive sentiment, negative sentiment, retweet_count, replies_count, likes_count  
    weights = [1,1,3,2,1]  
  
    # Total bobot [W]  
    totalWeights = sum(weights)  
  
    # Normalisasi bobot[Wj]  
    # [0.125, 0.125, 0.375, 0.25, 0.125]  
    for counter, weight in enumerate(weights) :  
        Wj.append(round(weight/totalWeights,4))  
  
    # Normalisasi Keputusan[Si]  
    for counter, data in enumerate(datas) :  
        temp = []  
        for counter2, element in enumerate(data) :  
            if (counter2 == 1) :  
                temp.append((element+0.1)**(-Wj[counter2]))  
            else :  
                temp.append((element+0.1)**(Wj[counter2]))  
        Si.append(np.prod(temp))  
    print("====Si====")  
    print(Si)  
    print("====Si====")  
  
    # Nilai Preferensi[Vi]  
    for s in Si :  
        Vi.append(s/sum(Si))  
    print(Vi)  
  
    for counter, recommendationResult in enumerate(Vi) :  
        print("hasil: tweet id " + str(tweet_ids[counter][0]) + " nilai rekomendasi " + str(recommendationResult))
```

Sampling Data	Benefit		Cost		Benefit		Benefit		Benefit	
	0.125	0.125	0.125	0.375	0.25	0.125	SI	VI		
tweet_id	positive_sentiment	negative_sentiment	retweet_count	replies_count	favourite_count	SI	VI			
1116345081974247424	5	4	154	39	988	40.24098301	0.026244672			
1116292710120353792	11	5	114	52	1087	41.92222599	0.027341158			
1116155430571397120	6	7	89	81	587	35.18092859	0.022944567			
1116153509278179328	4	6	75	81	504	31.39543681	0.020475716			
1116152483242397696	0	0	20	81	110	16.640913	0.010852998			
1116151543890231296	1	0	23	81	102	23.43753909	0.015285674			
1116150651250077696	1	0	49	81	160	32.89529423	0.021453905			
1116149652028399616	2	5	52	81	203	22.98075221	0.014987763			
1116148810323910656	1	0	258	81	1198	78.82453808	0.051408391			
1117088487545524225	10	6	590	198	3106	119.4590608	0.077909725			
1117086163481681920	3	10	278	47	541	40.96097912	0.026714245			
1117080366395772928	3	3	15	12	156	9.704832871	0.006329372			
1117078708022468613	0	0	410	0	1003	12.73508043	0.008305662			
1117078319311147008	8	7	163	44	758	40.54142676	0.026440618			
1117071666641694720	3	4	1562	725	4998	229.0409497	0.14937768			
1117062667431137281	3	7	164	27	454	29.92711803	0.019518097			
1117058911117250562	2	2	288	44	704	48.90943684	0.031898131			
1116970088429572096	11	4	345	43	1739	65.98673451	0.043035734			
1116964295147786241	3	4	100	64	606	34.23872246	0.022330072			
1117361971970445312	5	3	518	116	1742	92.52803674	0.060345643			
1117361448038965248	3	7	273	44	1244	46.408776	0.030267231			
1117360332207321089	0	0	1	0	20	0.848050597	0.000553088			
1117360253186629632	4	2	4	104	35	9.196881452	0.005998093			
1117358345885630464	2	1	811	104	3144	116.830424	0.076195361			
1118461538606567424	1	0	11	4	70	8.055285158	0.005253558			
1118459439357448192	0	1	8	3	72	3.677817102	0.002398627			
1118449271009763328	6	7	202	81	677	48.68925379	0.03175453			
1118438708473057281	3	4	84	64	485	31.19350274	0.020344017			
1118426742446624770	4	7	5	19	93	6.337333415	0.004133131			
111855682753533328	4	7	35	88	359	22.6634885	0.014780847			
1118555262745333761	3	3	549	197	2758	107.4279278	0.070063169			
1118553670289780736	9	9	214	56	1354	50.42876214	0.038899016			
1118552552759410688	4	4	12	16	175	9.731095868	0.006346501			
1118509424149012480	5	7	71	27	633	24.26143254	0.015823007			

3. Kriteria Dalam Penentuan Contoh Tweet yang Membahas Suatu Tren

Kriteria	
Benefit	<i>Is verified, Retweet Count, Replies Count, Favorite Count</i>

Kriteria	Bobot
Is verified	4
Retweet Count	3
Replies Count	2
Favorite Count	1



4. Manualisasi *Weighted Product* Dalam Penentuan Contoh *Tweet* Yang Membahas Suatu Tren

```

Wj = []
Si = []
temp = []
Vi = []

# Penentuan bobot
# is verified, retweet_count, replies_count, Likes_count
weights = [4,3,2,1]

# Total bobot [W]
totalWeights = sum(weights)

# Normalisasi bobot[Wj]
# [0.4, 0.3, 0.2, 0.1]
for counter, weight in enumerate(weights) :
    Wj.append(round(weight/totalWeights,4))

# Normalisasi Keputusan[Si]
for counter, data in enumerate(datas) :
    temp = []
    for counter2, element in enumerate(data) :
        temp.append((element+0.1)**(Wj[counter2]))
    Si.append(np.prod(temp))
print("====Si====")
print(Si)
print("====Si====")

# Nilai Preferensi[Vi]
for s in Si :
    Vi.append(s/sum(Si))
print(Vi)

for counter, recommendationResult in enumerate(Vi) :
    print("hasil: tweet id " + str(tweet_ids[counter][0]) + " nilai rekomendasi: " + str(recommendationResult))

```

Sampling Data	benefit 0.4	benefit 0.3	benefit 0.2	benefit 0.1	Si	Vi
tweet_id	is_verified	retweet_count	replies_count	favourite_count		
1123529553672970000	1	406	61	1658	30.08266344	0.251563604
1123505901711700000	1	94	31	498	15.0285712	0.125675093
1123412067740980000	1	73	1	74	5.901910849	0.049354206
1123465116631090000	1	41	1	35	4.608067895	0.038534559
1123432603951030000	1	26	29	110	8.679736798	0.072583529
1123394210751630000	1	21	2	29	4.213810185	0.035237614
1123449723019490000	1	30	0	203	3.096692021	0.025895812
1123405092231580000	1	38	8	129	7.649344047	0.063966961
1123382180212680000	1	27	1	71	4.364287562	0.036495968
1123400798883330000	1	17	1	20	3.350071297	0.028014674
1123533415637080000	1	20	4	83	5.272696394	0.044092456
1123435019312270000	1	44	3	45	5.93715851	0.049648961
1123480657987030000	1	21	0	40	2.366771047	0.019791913
1123397055802740000	1	10	3	25	3.598265436	0.030090176
1123421786442360000	1	16	0	14	1.965743047	0.016438352
1123348312508070000	1	13	1	31	3.230673272	0.027016219
1123377899849910000	1	13	0	16	1.872496654	0.015658587
1123372309316680000	1	6	0	8	1.389967118	0.011623477
1121750665359810000	1	18	5	126	5.56475682	0.046534785
1123600350362180000	0	0	0	0	0.1	0.000836241
1123599329879560000	0	0	0	0	0.1	0.000836241
1123599302205460000	0	0	0	1	0.127098162	0.001062847
1123599268990730000	0	0	0	0	0.1	0.000836241
1123599195552640000	0	0	0	0	0.1	0.000836241
1123599156516230000	1	2	0	2	0.88195168	0.007375243

