BAB 2
LANDASAN TEORI

2.1 Twitter

Twitter adalah layanan jejaring sosial dan mikroblog yang memungkinkan penggunanya untuk mengirim dan membaca pesan berbasis teks hingga 140 karakter, yang dikenal dengan sebutan kicauan (tweet). Twitter didirikan pada bulan Maret 2006 oleh Jack Dorsey, dan situs jejaring sosialnya diluncurkan pada bulan Juli. Sejak diluncurkan, Twitter telah menjadi salah satu dari sepuluh situs yang paling sering dikunjungi di Internet, dan dijuluki dengan "pesan singkat dari Internet". Di Twitter, pengguna tak terdaftar hanya bisa membaca kicauan, sedangkan pengguna terdaftar bisa menulis kicauan melalui antarmuka situs web, pesan singkat (SMS), atau melalui berbagai aplikasi untuk perangkat seluler [11].

Twitter mengalami pertumbuhan yang pesat dan dengan cepat meraih popularitas di seluruh dunia. Hingga bulan Januari 2013, terdapat lebih dari 500 juta pengguna terdaftar di Twitter, 200 juta di antaranya adalah pengguna aktif. Lonjakan penggunaan Twitter umumnya berlangsung saat terjadinya peristiwa-peristiwa populer. Pada awal 2013, pengguna Twitter mengirimkan lebih dari 340 juta kicauan per hari, dan Twitter menangani lebih dari 1,6 miliar permintaan pencarian per hari. Hal ini menyebabkan posisi Twitter naik ke peringkat kedua sebagai situs jejaring sosial yang paling sering dikunjungi di dunia, dari yang sebelumnya menempati peringkat dua puluh dua [11].

Tingginya popularitas Twitter menyebabkan layanan ini telah dimanfaatkan untuk berbagai keperluan dalam berbagai aspek, misalnya sebagai sarana protes, kampanye politik, sarana pembelajaran, dan sebagai media komunikasi darurat. Twitter juga dihadapkan pada berbagai masalah dan kontroversi seperti masalah keamanan dan privasi pengguna, gugatan hukum, dan penyensoran [11].

Pesan pada awalnya diatur hanya mempunyai batasan sampai 140 karakter disesuaikan dengan kompatibilitas dengan pesan SMS, memperkenalkan singkatan notasi dan slang yang biasa digunakan dalam pesan SMS. Batas karakter 140 juga meningkatkan penggunaan layanan memperpendek URL seperti bit.ly, goo.gl, dan tr.im, dan jasa hosting konten, seperti Twitpic, Tweephoto, memozu.com dan

NotePub untuk mengakomodasi multimedia isi dan teks yang lebih panjang daripada 140 karakter (Twitter, 2013). Twitter menggunakan bit.ly untuk memperpendek otomatis semua URL yang dikirim-tampil. Fitur yang terdapat dalam Twitter, antara lain:

1. Laman Utama (*Home*)

   Pada halaman utama kita bisa melihat Tweets yang dikirimkan oleh orang-orang yang menjadi teman kita atau yang kita ikuti (following).

2. Profil (*Profile*)

   Pada halaman ini yang akan dilihat oleh seluruh orang mengenai profil atau data diri serta Tweets yang sudah pernah kita buat.

3. *Followers*

   Pengikut adalah pengguna lain yang ingin menjadikan kita sebagai temaBila pengguna lain menjadi pengikut akun seseorang, maka Tweets seseorang yang ia ikuti tersebut akan masuk ke dalam halaman utama.

4. *Following*

   Kebalikan dari pengikut, following adalah akun seseorang yang mengikuti akun pengguna lain agar Tweets yang dikirim oleh orang yang diikuti tersebut masuk ke dalam halaman utama.

5. *Mentions*

   Biasanya konten ini merupakan balasan dari percakapan agar sesama pengguna bisa langsung menandai orang yang akan diajak bicara.

6. *Favorite*

   Tweets ditandai sebagai favorit agar tidak hilang oleh halaman sebelumnya.

7. Pesan Langsung (*Direct Message*)

   Fungsi pesan langsung lebih bisa disebut SMS karena pengiriman pesan langsung di antara pengguna.

8. *Hashtag*

   Hashtag "#" yang ditulis di depan topik tertentu agar pengguna lain bisa mencari topik yang sejenis yang ditulis oleh orang lain juga

2.2 Emosi

Emosi adalah suatu perasaan yang mendorong individu untuk merespon atau bertingkah laku terhadap stimulus, baik yang berasal dari dalam maupun dari luar dirinya. Emosi cenderung terjadi dalam kaitannya dengan perilaku yang mengarah (approach) atau menyingkiri (avoidance) terhadap sesuatu, dan perilaku tersebut pada umumnya disertai adanya ekspresi kejasmanian, sehingga orang lain dapat mengetahui bahwa seseorang sedang mengalami emosi.

Beberapa studi tentang emosi manusia telah dilakukan sehingga ada kesepakatan tentang emosi dasar yaitu, takut digambarkan sebagai ancaman fisik atau sosial bagi diri sendiri, kemarahan sebagai ketidakpuasan atau frustrasi dari peran atau tujuan yang dicapai oleh orang lain, jijik saat yang jarak, kelalaian, atau penolakan dari ide untuk diri sendiri, kesedihan sebagai kegagalan atau penderitaan pada peran dan tujuan, kebahagiaan atau sukacita sebagai keberhasilan atau gerakan untuk prestasi peran berharga [12].

Baru-baru ini, para peneliti telah menyelidiki beberapa aspek emosi manusia untuk sampai pada satu set kategori emosi yang dapat diterima secara universal (Picard, 1997). Beberapa karya pada bidang ini telah dilaporkan dalam literatur (Tomkins, 1962; Izard, 1977; Plutchik, 1980; Ortony et. Al., 1988; dan Ekman, 1992). Pada tabel 2.1 ditampilkan daftar kategori dasar emosi yang diidentifikasi oleh para peneliti yang berbeda [3].

Tabel 2-1 Kategori dasar emosi yang diidentifikasi oleh para peneliti

Tomkins (1962)	**Izard (1977)**	**Plutchik (1980)**	**Ortony et.al. (1988)**	**Ekman (1992)**
joy	enjoyment	joy	joy	happiness
anguish	sadness	sorrow	sadness	sadness
fear	fear	fear	fear	fear
anger	anger	anger	anger	anger
disgust	disgust	disgust	disgust	disgust
surprise	surprise	surprise	surprise	surprise
interest	interest	acceptance		
shame	shame	anticipation		

Tomkins (1962)	Izard (1977)	Plutchik (1980)	Ortony et.al. (1988)	Ekman (1992)
	shyness			
	guilt			

2.3 Text Mining

Text Mining merupakan ilmu yang mengkombinasikan teknik dari *data mining, machine learning, natural learning processing*, dan *information retrieval.* Salah satu contoh implementasi *text mining* termasuk mengklasifikasikan dokumen ke dalam topik yang spesifik atau *supervised learning*, mengelompokan dokumen yang mempunyai kemiripan atau *unsupervised learning*, dan diimplementasikan ke dalam sistem perncari dokumen atau *information retirieval*. *Text data mining* fokus terhadap metode *data mining* yang diaplikasikan pada tekstual data. Beberapa bagian dari *text mining* adalah *term, corpus, lexicon*, dan dokumen. *Term* adalah kata atau kombinasi kata atau prasa. Dokumen adalah sekumpulan kata dan tanda baca, dokumen dapat terdiri dari beberapa segmen teks dan memiliki panjang yang tidak ditentukan. Contoh dari dokumen adalah kalimat, paragraf, bagian sebuah buku atau bab, buku, halaman web, email dan lain-lain. *Corpus* adalah koleksi dokumen. *Lexicon* adalah sekumpulan kata yang bersifat unik yang terdapat pada corpus [13].

2.4 Tata Bahasa

Istilah tata bahasa dapat dipakai secara luas dengan cakupan yang meliputi semua bagian dalam deskripsi suatu bahasa, termasuk *tata bunyi*, *tata kata*, *tata kalimat*, *tata tulis*, *perbendaharaan kata dan tata makna* [14]. Akan tetapi pada penelitian ini istilah ‘tata bahasa’ dipakai secara sempit dengan cakupan yang meliputi *tata kalimat* (atau sintaksis) dan *tata kata* saja.

2.4.1 Kata

Kata dalam Bahasa Indonesia dapat dibagi ke dalam beberapa jenis, bergantung pada kriteria yang digunakan. Berdasarkan tugas pokok dan perilaku

gramatikalnya dikenal dua kelas kata utama*; kata leksikal dan kata tugas* [14]. *Kata leksikal* merupakan pembawa makna leksikal, artinya kata leksikal mengandung konsep yang mengacu kepada barang, peristiwa, tindakan atau keadaan tertentu. Sedangkan *kata tugas* bukan pembawa makna leksikal, artinya tidak mengacu pada barang, peristiwa, tindakan, atau keadaan tertentu, tetapi mempunyai tugas menunjukan hubungan gramatikal di antara unsur-unsur suatu konstruksi [14]. Berikut adalah daftar jenis kata yang akan dipakai pada penelitian dan mengacu pada kedua kelas kata di atas.

Tabel 2-2 Jenis Kata

NO	JENIS KATA	CIRI-CIRI	CONTOH
1	Kata Benda / Nomina (NN)	1. Kata dasar yang tidak dapat diurai ke bentuk lain dan menunjukan identitas suatu benda. 2. Dapat diperluas dengan kata ‘yang’ (kata benda + yang + kata sifat). 3. Dapat ditambahkan pewatas negasi, (bukan + kata benda)	*1. buku* 2. *air* yang panas 3. bukan *kursi*
2	Kata Kerja / verba (VB)	1. Kata yang menyatakan suatu perbuatan. 2. Kata yang dapat diikuti dengan kata ‘dengan’. 3. Kata berimbuhan yang berawalan **me-**, **ber-**, **di-**, **ter-**, **per-**, dan berakhiran **–kan**, **-i**.	*1. Menyanyi* *2. Mandorong*
3	Kata Sifat / Adjektiva (JJ)	1. Kata yang dipakai untuk menunjukan sifat atau keadaan. 2. Kata yang dapat diikuti dengan kata keterangan ’sekali’ serta dapat dibentuk menjadi kata ulang berimbuhan gabung **se-nya**	*1. Rajin* *2. Sepanas-panasnya* *3. Dingin sekali*
4	Kata Keterangan / Adverbia (ADV)	1. kata yang memberikan keterangan pada kata kerja, kata sifat, dan kata bilangan bahkan mampu memberikan keterangan pada seluruh kalimat.	1. Memang, pasti, tentu 2. Barangkali, mungkin, ... 3. Semoga, moga-moga, ... 4. Seringkali, sesekali, ... 5. Sudah, telah, sedang, ... 6. Ingin, mau, hendak, suka, ... 7. Boleh, wajib, harus, ...
5	Kata Ganti / Pronomina (PR)	Kata yang mengacu pada nomina lain.	1. **Kata Ganti Orang Pertama** : Saya, Aku, Kami, Kita

NO	JENIS KATA	CIRI-CIRI	CONTOH
			2. **Kata Ganti Orang Kedua** : Kamu, Engkau, Anda, Kalian 3. **Kata Ganti Orang Ketiga** : Dia, Nya, Beliau, Mereka
6	Kata Bilangan / Numeralia (NUM)	kata yang digunakan untuk menghitung banyaknya sesuatu.	1. Satu, dua, tiga, ... 2. Pertama, kedua, ketiga,
7	Kata Penghubung (CC)	Kata yang digunakan untuk menghubungkan kata dengan kata, klausa dengan klausa, atau kalimat dengan kalimat.	1. Dan, dengan, serta, ... 2. Karena, sebab 3. Ketika, sewaktu, sebelum, sesudah, ... 4. Sampai, hingga, ... 5. Seperti, laksana, ...

2.4.2 Frasa

Frasa adalah satuan gramatikal yang dibentuk dari dua atau beberapa kata yang bersama-sama mendukung satu fungsi gramatikal[14]. Sebagai suatu konstruksi, frasa disusun oleh beberapa unsur pembentuk yang saling berhubungan secara fungsional [15]. Sebagai contoh, frasa *telur asin* terdiri atas nomina yang diikuti adjektiva. Kedua unsur tersebut memiliki hubungan fungsi sebagai berikut : Kata *telur* berfungsi sebagai unsur inti dan kata *asin* sebagai pewatas. Hubungan keduanya menghasilkan makna ‘rasa’ yang berarti *telur yang rasanya asin*. Konstruksi frasa ini termasuk frasa nomina karena intinya berupa nomina sehingga memiliki fungsi yang sama dengan nomina. Frasa dapat dikelompokan menjadi tujuh jenis frasa, yaitu; frasa nomina, frasa verba, frasa adjektiva, frasa numerilia, frasa pronomina, frasa adverbia dan frasa preposisional.

2.4.2.1 Frasa Verba

Frasa verba (FVB) atau frasa kerja adalah frasa yang berintikan kata kerja (VB), frasa verba merupakan hasil perluasan. Perluasan itu terjadi dari suatu inti yang berupa kata kerja yang bergabung dengan kata lain. Susunan kata pada frasa verba dapat dilihat pada tabel berikut.

Tabel 2-3 Pola Frasa Verba

Pola	Contoh	Pewatas	Inti
adverbia + VB	Lagi mandi	Lagi	Mandi
VB + adverbia	Datang lagi	Lagi	Datang
VB + nomina	Uji materi	Materi	Uji
VB + adjektiva	Menulis indah	Indah	menulis

2.4.2.2 Frasa Nomina

Frasa Nomina (FN) atau frasa benda adalah frasa yang berintikan kata nomina (NN). Pewatas yang terletak sebelum nomina inti disebut pewatas depan. Pada frasa nomina, pewatas depan biasanya berupa kata numeralia dan adverbia. Sedangkan pewatas yang berada setelah nomina inti disebut pewatas belakang, biasanya berupa nomina, adjektiva, verba, adverbia, numeralia, dan determinan (ini, itu) [15]. Untuk lebih jelasnya dapat dilihat pada tabel berikut ini.

Tabel 2-4 Jenis Kata Nomina

Pola	Contoh	Pewatas	Inti
Numralia + NN	Enam mahasiswa	Enam	Mahasiswa
Adverbia + NN	Tanpa bunga	Tanpa	Bunga
NN + NN	Orang desa	Orang	Desa
NN + adjektiva	Gadis cantik	Cantik	Gadis
NN + verba	Ruang kerja	Kerja	Ruang
NN + numeralia	Kuliah pertama	Pertama	Kuliah
NN + adverbia	Orang tadi	Tadi	Orang
NN + determinan	Dosen itu	Itu	dosen

2.4.2.3 Frasa adjektiva

Frasa adjektiva (FJJ) adalah satuan sintaksis yang terbentuk dari dua kata atau lebih yang dapat menggantikan kategori adjektiva. Adjektiva (JJ) berfungsi sebagai inti[15].

Tabel 2-5 Jenis Kata Adjektiva

Pola	Contoh	Pewatas	Inti
Adverbia + JJ	Lebih pintar	Lebih	Pintar
JJ + adverbia	Sakit lagi	Lagi	Sakit
JJ + nomina	Gagah perwira	Perwira	Gagah
JJ + JJ	Hijau tua	Tua	Hijau
JJ + verba	Berani tempur	Tempur	Berani

2.4.2.4 Frasa numeralia

Frasa numeralian (FCD) adalah satuan sintaksis yang terbentuk dari dua kata atau lebih, yang dapat menggantikan kategori numeralia. Numeralia (CD) sebagai inti. Umumnya, frasa ini dibentuk dengan menambahkan kata penggolong, adverbia, atau kata gugus setelah numeralia[15].

Tabel 2-6 Jenis Kata Numeralia

Pola	Contoh	Pewatas	Inti
CD + penggolong	Dua ekor	Ekor	Dua
Adverbia + CD	Hanya satu	Hanya	Satu
CD + adverbia	Lima saja	Saja	Lima
CD + gugus	Tiga ratus/ribu/..	Ratus/ribu/...	Tiga
CD + dan + CD	Satu dan dua	-	Satu, dua

2.4.2.5 Frasa Pronomina

Frasa pronomina (FPR) adalah satuan sintaksis yang terbentuk dari dua kata atau lebih yang dapat menggantikan kategori pronomina. Pronomina (PR) berfungsi sebagai inti[15].

Tabel 2-7 Jenis Kata Pronomina

Pola	Contoh	Pewatas	Inti
PR + numeralia	Kita berempat	Berempat	Kita
PR + determinan	Kami ini	Ini	Kami
PR + adverbia	Saya saja	Saja	Saya
Adverbia + PR	Hanya kamu	Hanya	kamu

2.4.2.6 Frasa Adverbia

Frasa adverbia (FRB) adalah satuan sintaksis yang terbentuk dari dua kata atau lebih dengan adverbia (RB) yang berfungsi sebagai inti [15].

Tabel 2-8 Jenis Kata Adverbia

Pola	Contoh	Pewatas	Inti
RB + nomina	Tadi malam	Malam	Tadi
RB + determinan	Sekarang ini	Ini	Sekarang
RB + RB	Nanti saja	Saja	nanti

2.4.2.7 Frasa Preposisional

Frasa preposisional (FIN) merupakan frasa eksosntris, tidak terdiri atas inti dan pewatas, tetapi terdiri dari perangkai dan sumbu. Preposisi (IN) berfungsi sebagai perangkai, sedangkan jenis kata yang berfungsi sebagai sumbu adalah nomina, adjektiva, atau adverbia [15].

2.4.2.8 Tagset dan Rangkuman Pola Frasa

Dari penjelasan kata dan frasa di atas, berikut adalah rangkuman simbol-simbol atau tagset yang akan dipakai untuk identifikasi frasa.

Tabel 2-9 Rangkuman Tagset

Tag	Keterangan	Contoh
NN	Kata nomina	buku
FN	Frasa nomina	buku itu
VB	Kata verba	memasak
FVB	Frasa Verba	sedang memasak
JJ	Kata adjektiva / kata sifat	pintar
FJJ	Frasa Adjektiva	lebih pintar
CD	Kata Numeralia	satu
FCD	Frasa numeralia	hanya satu
PR	Kata Pronomina / kata ganti	saya
FPR	Frasa pronomina	saya saja
RB	Kata adverbia	tadi
FRB	Frasa adverbia	tadi malam
IN	Kata preposisi	di
FIN	Frasa preposisi	di pasar
DT	Determinan	Ini, itu
CC	Konjungsi	Sedangkan, tetapi, dan

Dan berikut ini adalah rangkuman pola frasa yang telah dirincikan pada subbab sebelumnya.

Tabel 2-10 Rangkuman Pola Frasa

NO	Kombinasi	Frasa
1	NN + NN	FN
2	NN + NN + NN	FN

NO	Kombinasi	Frasa
3	NN + CD	FN
4	NN + DT	FN
5	NN + PR	FN
6	NN + VB	FN
7	NN + VB + DT	FN
8	NN + JJ	FN
9	NN + DT	FN
10	NN + JJ + JJ + PR + DT	FN
11	NN + PR + CC + JJ + DT	FN
12	VB + RB	FVB
13	VB + NN	FVB
14	VB + JJ	FVB
15	JJ + JJ	FJJ
16	JJ + RB	FJJ
17	JJ + NN	FJJ
18	JJ + VB	FJJ
19	CD + CD	FCD
20	CD + RB	FCD
21	CD + NN	FCD
22	CD + NN + DT	FCD
23	PR + CD	FPR
24	PR + RB	FPR
25	PR + DT	FPR
26	RB + NN	FN
27	RB + VB	FVB
28	RB + JJ	FJJ
29	RB + CD	FCD
30	RB + DT	FRB
31	RB + PR	FPR

NO	Kombinasi	Frasa
32	RB + PR + CD + RB	FPR
33	RB + RB + VB	FVB
34	RB + RB + DT	FRB
35	MD + RB + VB	FVB
36	MD + JJ	FJJ
37	MD + JJ +VB	FVB
38	MD + VB	FVB
39	IN + NN	FIN
40	IN + NN + NN	FIN
41	IN + NN + JJ	FIN
42	IN + NN + VB	FIN
43	IN + NN + VB + DT	FIN
44	IN + PR	FIN
45	NEG + VB	FVB
46	NEG + VB + VB	FVB

2.4.3 Kalimat

Kalimat adalah satuan bahasa yang disusun konstituen dasar yang umumnya berupa klausa, kata penghubung (jika ada) dan intonasi final. Dalam ragam tulisan intonasi final ini dinyatakan dengan titik (.), tanda tanya (?), atau tanda seru (!) [14]. Menurut strukturnya, sebuah kalimat sederhana dalam bahasa Indonesia memiliki pola sebagai berikut[15][14].

Tabel 2-11 Pola Kalimat Sederhana Berdasarkan Strukturnya

NO	STRUKTUR	CONTOH
1	Subjek + Predikat (S-P)	1. Para pedagang mengeluh. 2. Orang itu sedang tidur.
2	Subjek + Predikat + Objek (S-P-O)	1. Ilmu mengatur pengetahuan. 2. Ayahnya membeli mobil baru.

NO	STRUKTUR	CONTOH
3	Subjek + Predikat + Pelengkap (S-P-Pel)	1. Beliau menjadi ketua koperasi.
4	Subjek + Predikat + Objek + Pelengkap (S-P-O-Pel)	1. Ia menyebutnya si hitam. 2. Dian mengambilkan adiknya air minum.
5	Subjek + Predikat + Objek + Keterangan (S-P-O-K)	1. Pak Raden memasukan uang ke bank.
6	Subjek + Predikat + Keterangan (S-P-K)	1. Kami tinggal di jakarta

Dalam tata bahasa Indonesia, fungsi gramatikal *subjek, predikat,* dan *objek* dapat ditempati oleh berbagai macam jenis kata maupun frasa. Berikut adalah daftar fungsi gramatikal dan jenis kata atau frasa yang dapat menempatinya.

Tabel 2-12 Daftar Fungsi Gramatikal dan Pengisinya

Fungsi Gramatikal	Keterangan	Pengisi
Subjek	Sebuah kata atau frasa yang letaknya di depan predikat.	1. Kata benda / nomina 2. Frasa nomina 3. Kata kerja / verba 4. Kata bilangan / numeralia
Predikat	Bagian dari klausa yang memberikan informasi tentang subjek.	1. Kata kerja / verba 2. Frasa verba 3. Adjektiva 5. Kata benda / nomina 6. Frasa nomina 7. Kata / Frasa numeralia

Fungsi Gramatikal	Keterangan	Pengisi
Objek	Bagian dari klausa yang terkena tindakan yang disebut di dalam predikat.	1. Kata benda / nomina 2. Frasa nomina
Pelengkap	Bagian dari klausa yang mengikuti atau melengkapi predikat.	1. Adverbia 2. Frasa adverbia
Keterangan	Bagian dari klausa yang menjelaskan jenis tindakan yang dinyatak di dalam predikat. Biasanya keterangan berupa kata keterangan. Posisinya dalam kalimat bisa diubah (depan,tengah, belakang)	Frasa nomina

2.5 Text Pre-Processing

Text Pre-Processing adalah serangkaian proses yang dilakukan untuk mempersiapkan data sebelum proses lebih lanjut. Tahapan ini memiliki tujuan untuk mengubah data agar dapat dilanjukan ke proses klasifikasi. Terdapat beberapa proses yang harus dilakukan dalam tahap *text pre-processing* atau persiapan dokumen teks dengan sumber data twitter, yaitu *cleansing, case folding, formalizatin, stemming.*

2.5.1 *Cleansing*

Tahap ini akan menghapus semua karakter selain alfabetis dengan tujuan untuk mengurangi *noise*. Selain karakter khusus, , URL, hashtag (#), username (@username), tanda koma(,), tanda titik(.), tanda seru (!), tanda titik koma (;), tanda titik dua (:), tanda hubung (-), tanda elipsis (...), tanda tanya (?), tanda kurung ((..)), tanda kurung siku ({..}), tanda petik (".."), tanda petik tunggal („..‟), tanda garis miring (/) dan (\), dan tanda penyingkat („) akan dihilangkan. contoh proses *cleansing*.

Teks awal :

@ridwankamil @PemumBdg Kegiatan hari ultah ke 17 dharma wanita persatuan tk Kota Bandung, kamis 15 des 2016 https://t.co/i27yIyDumb

Setelah *cleansing*:

Kegiatan hari ultah ke 17 dharma wanita persatuan tk Kota Bandung, kamis 15 des 2016

2.5.2 *Case Folding*

Case folding adalah proses merubah semua huruf dalam teks menjadi huruf kapital atau non-kapital. Fungsi *case folding* dalam sistem ini adalah mencegah

terjadinya kata yang tidak terdeteksi dalam kamus yang disimpan dalam basis data karena kata-kata yang ada di basis data semua memiliki huruf non-kapital. Berikut adalah contoh proses *case folding*.

Teks awal :

Kegiatan hari ultah ke 17 dharma wanita persatuan tk Kota Bandung, kamis 15 des 2016

Setelah case folding :

kegiatan hari ultah ke 17 dharma wanita persatuan tk kota bandung, kamis 15 des 2016

Kata *Kegiatan, Kota* dan *Bandung* berubah menjadi *kegiatan, kota* dan *bandung*.

2.5.3 *Formalization*

Formalization merupakan proses mengubah kata yang tidak baku kedalam kata baku atau formal. Para pengguna media sosial seperti twitter seringkali menggunakan kata – kata tidak baku dalam tulisannya. Contoh kata yang sering digunakan adalah kata ‘abis’, ‘adlh’, ‘akikah’, ‘akyu’, ‘asek’, ‘asoy’, dan lain –lain. Untuk dapat melakukan proses pengklasifikasian emosi kata tidak baku tersebut harus terlebih dahulu diubah ke dalam kata baku. Daftar kata–kata yang akan dilakukan proses *formalization* ditunjukan pada tabel berikut:

Tabel 2-13 Contoh daftar kata formal

Kata Tidak Baku	Kata Formal
7an	tujuan
@	di
ababil	abg labil
abis	habis

Kata Tidak Baku	Kata Formal
acc	accord
ad	ada
adlah	adalah
adlh	adalah
adoh	aduh
afaik	as far as i know
aha	tertawa
ahaha	haha
aing	saya
aj	saja
aja	saja
ajep	dunia gemerlap
ajj	saja
ak	saya
aka	dikenal juga sebagai
akika	aku
akko	aku
akkoh	aku
akku	aku
apasih	apa sih
ape	apa
apes	sial
aplot	unggah
aps	apa
apva	apa
aq	saya
aqu	aku
aquwh	aku
asap	sesegera mungkin
asbun	asal bunyi
aseek	asyik
aseekk	asyik
asek	asyik
asekk	asyik
aseknya	asyiknya
asem	asam
asoy	asyik

Kata Tidak Baku	Kata Formal
aspal	asli tetapi palsu
astrojim	astagfirullahaladzim
astul	asal tulis
ath	kalau begitu
ato	atau
atuh	kalau begitu
au	ah tidak mau tahu
ava	avatar
awak	saya
aws	awas
ay	sayang
ayang	sayang
ayank	sayang
ayok	ayo
b4	sebelum
bacot	banyak bicara
bakalan	akan
bales	balas
bandes	bantuan desa
bangdes	pembangunan desa
bangedh	banget
bangkotan	tua
banpol	bantuan polisi
banpres	bantuan presiden
banpur	bantuan tempur
bansarkas	bantuan sarana kesehatan
basbang	basi
bazis	badan amal, zakat, infak, dan sedekah
bcanda	bercanda
bcoz	karena
bdg	bandung
beb	sayang
begajulan	nakal
bejibun	banyak
beliin	belikan
belom	belum
bencong	banci

Kata Tidak Baku	Kata Formal
bener	benar
bentar	sebentar
ber2	berdua
ber3	bertiga
berdikari	berdiri di atas kaki sendiri
beresin	membereskan
bet	banget
bete	bosan
beti	beda tipis
beud	banget
beut	banget
bg	abang
bgd	banget
bgmn	bagaimana
bgs	bagus
bgt	banget
bhubu	tidur
bijimane	bagaimana
bimbuluh	bimbingan dan penyuluhan
bintal	bimbingan mental
bisi	kalau-kalau
bkl	akan
bkn	bukan
bknnya	bukannya
bl	beli
blegug	bodoh
blg	bilang
blh	boleh
blm	belum
bln	bulan
bls	balas
blum	belum
bnchi	benci
bnci	benci
bngung	bingung
bnran	yang benar
bnyk	banyak

Kata Tidak Baku	Kata Formal
bodor	lucu
bohay	badan aduhai
bokap	ayah
bokep	video porno
boker	buang air besar
bokin	pacar
bokis	bohong
bole	boleh
boljug	boleh juga
bolot	bodoh
bonek	bocah nekat
bonyok	ayah ibu
boyeh	boleh
bpk	bapak
br	baru
brb	segera kembali
brg	bareng
brngkt	berangkat
bro	saudara laki-laki
brp	berapa
bru	baru
brur	saudara laki-laki
bs	bisa
bsa	bisa
bsen	bosan
bsk	besok
bt	bosan sekali
bt	buat
btw	ngomong-ngomong
bu_bu	tidur
buaya	tidak setia
bubarin	bubarkan
bubbu	tidur
buber	buka bersama
bubu	tidur
bujubune	luar biasa
bumil	ibu hamil

Kata Tidak Baku	Kata Formal
buser	buru sergap
bw	bawa
bwhn	bawahan
bwt	buat
byar	bayar
byk	banyak
byr	bayar
byrin	bayarkan
c8	chat
cabal	sabar
cabut	pergi
cadas	keren
caem	cakep
calo	makelar
cama	cama sama-sama
can	belum
cangcut	celana dalam
capcus	pergi
cape	capek
caper	cari perhatian
caur	jelek
ce	cewek

contoh proses *formalization* adalah sebagai berkut:

Teks awal :

kegiatan hari ultah ke 17 dharma wanita persatuan tk kota bandung, kamis 15 des 2016

Setelah *formalization*:

kegiatan hari ulang tahun ke 17 dharma wanita persatuan tk kota bandung, kamis 15 des 2016

Kata *ultah* berubah menjadi *ulang tahun.*

2.5.4 *Stemming*

Stemming adalah salah satu cara untuk meningkatkan performa dalam *text processing* dengan cara mentransformasi kata-kata dalam sebuah dokumen teks ke kata dasarnya. Algoritma stemming untuk masing-masing bahasa berbeda-beda, sebagai contoh bahasa Inggris memiliki morfologi yang berbeda dengan bahasa Indonesia sehingga algoritma stemming untuk kedua bahasa tersebut juga berbeda [16].

Stemming menggunakan aturan-aturan tertentu sebagai contoh kata bersama, kebersamaan dan menyamai mempunyai root word yang sama yaitu "sama". Pada teks berbahasa Inggris, proses yang perlu dilakukan hanya proses menghilangkan suffiks, sedangkan pada teks berbahasa Indonesia, selain suffiks, prefiks dan konfiks juga dihilangkan. Salah satu algoritma untuk stemming pada teks berbahasa Indonesia adalah algoritma nazief & adriani yang dibuat oleh Bobby Nazief dan Mirna Adriani. Algoritma tersebut memiliki tahap-tahap sebagai berikut:

1. Cari kata yang akan di-*stem* dalam kamus, jika ditemukan maka diasumsikan bahwa kata tersebut adalah *root word.*
2. Penghapusan *Inflexion Suffix*, ("-lah", "-kah", "-ku", "-mu", atau "-nya"). Cek kata dalam kamus, jika kata ditemukan maka diasumsikan bahwa kata tersebut adalah *root word.*
3. Penghapusan *derivation suffiks* ("-i", "-an", "kan"). Jika kata ditemukan di kamus, maka algoritma berhenti, jika tidak maka lanjutkan ke langkah 3a.
    a. Jika "-an" telah dihapus dan huruf terakhir dari kata tersebut adalah "-k" , maka "-k" juga dihapus. Jika kata ditemukan dalam kamus maka algoritma berhenti. Jika tidak maka lanjutkan ke langkah 3b.
    b. Akhiran yang dihapus ("-i", "-an", atau "-kan"). Dikembalikan, lanjut ke langkah 4.
4. Hapus *derivation prefix*. Jika pada langkah 3 ada *sufiks* yang dihapus maka pergi ke langkah 4a, jika tidak maka lanjut ke langkah 4b.

a. Periksa tabel kombinasi awalan-akhiran yang tidak diijinkan. Jika ditemukan maka algoritma berhenti, jika tidak lanjutkan ke langkah 4b.
b. Tentukan tipe awalan kemudian hapus awalan. Jika root word belum juga ditemukan lakukan langkah 5.

5. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai *root word.* Proses selesai. Pada tahap ke empat langkah-langkah yang dilakukan proses *stemming*, terdapat pemeriksaan kombinasi awalan dan akhiran yang tidak dizinkan. Kombinasi awalan akhiran yang tidak diizinkan ditunjukan pada tabel berikut:
6.

Tabel 2-14 Kombinasi awalan akhiran

Awalan	Akhiran
be-	-i
di-	-an
ke-	-l, -kan
me-	-an
se-	-l, -kan
te-	-an

Beberapa contoh kata yang kata berimbuhan yang akan dilakukan proses stemming:

1. Awalan Se-
   a. Sebungkus = se + bungkus
   b. Searah = se + arah
2. Awalan Me-
   a. Mengasuh = me + asuh
   b. Memberi = me + beri
   c. Mencuci = me + cuci
   d. Mendidik = me + didik
   e. Menghukum = me + hukum
3. Awalan Pe-
   a. Perhitungan = Pe + Hitung + An

b. Pergelaran = Pe + Gelar + An

contoh proses *stemming* adalah sebagai berkut:

Teks awal :

kegiatan hari ultah ke 17 dharma wanita persatuan tk kota bandung, kamis 15 des 2016

Setelah *stemming*:

giat hari ulang tahun ke 17 dharma wanita satu tk kota bandung kamis 15 des 2016

Kata *kegiatan* dan *persatuan* berubah menjadi *giat* dan *satu.*

2.6 Kamus Data Emosi

Kamus data emosi yang digunakan pada penelitian ini adalah kamus data emosi dari EmoSenticNet [8]. EmoSenticNet adalah sumber leksikal yang terdiri dari enam label emosi dari WordNet yang mengacu pada konsep SenticNet. Ini dapat juga dianggap sebagai perluasan dari kosakata yang ada pada WordNet Affect [8]. Sumber ini berguna untuk sentimen analisis, *opinion mining*, analisis jaringan sosial, analisis emosi, dll [8][17][18][19]. Contoh kamus data emosi dari EmoSenticNet ditunjukan pada tabel berikut:

Tabel 2-15 Kamus data emosi

Kata	Gembira	Marah	Jijik	Sedih	Takut
immortal	1	0	0	0	0
neglect	0	0	0	1	0
abortion	0	0	0	1	0
abrasion	0	0	0	1	0
absent	0	0	0	1	0
lawyer	1	0	0	0	0

adapt	1	0	0	0	0
fair	1	0	0	0	0
adopt	1	0	0	0	0
ouch	0	1	1	0	1
stir	0	1	0	0	0
diverse	0	0	0	0	0
religion	1	0	0	0	0
expert	1	0	0	0	0
ignominiously	0	0	0	1	0
watery	1	0	0	0	0
magical	0	0	0	0	0
learn	1	0	0	0	0
end	0	0	0	1	0
familiar	1	0	0	0	0
accessory	1	0	0	0	0
action	0	1	0	0	0
active	1	0	0	0	0
activist	1	0	0	0	0
actor	1	0	0	0	0
actress	1	0	0	0	0
concede	0	0	0	0	1
acculturation	1	0	0	0	0
account	1	0	0	0	0
accountant	0	0	1	0	0
accuracy	1	0	0	0	0
natural	1	0	0	0	0
reason	1	0	0	0	0
equipment	1	0	0	0	0
allergy	0	0	0	1	0
switch	1	0	0	0	0
algebra	1	0	0	0	0
lye	1	0	0	0	0
bible	1	0	0	0	0
alcoholism	0	1	1	1	0
god	1	0	0	0	0
alternate	1	0	0	0	0
charity	1	0	0	0	0
secure	1	0	0	0	0
amendment	1	0	0	0	0
observe	1	0	0	0	0
threshold	1	0	0	0	0

grab	1	0	0	0	0
udder	0	0	0	0	0
ambition	1	0	0	0	0
ambulance	0	0	0	0	1
amplitude	1	0	0	0	0
robbery	0	0	0	1	0
ammunition	0	0	0	0	1
child	1	0	0	0	0
analgesic	0	0	0	1	0
analyze	0	0	0	0	0
analysis	1	0	0	0	0
anarchy	0	0	0	0	1
anarchism	1	0	0	0	0
threat	1	0	0	0	0
yours	1	0	0	0	0
reliability	1	0	0	0	0
strange	0	0	0	0	0
supposition	0	0	0	0	0
budget	1	0	0	0	0
member	1	0	0	0	0
nod	1	0	0	0	0
wind	1	0	0	0	0
lift	1	0	0	0	0
molest	0	1	0	0	0
animation	0	0	0	0	0
anomic	0	0	0	1	0
antagonist	0	0	0	1	0
between	1	0	0	0	0
henchman	1	0	0	0	0
antibiotic	0	0	1	0	1
antique	1	0	0	0	0
antioxidant	1	0	0	0	0
antipathy	0	1	0	0	0
anticipative	1	0	0	0	0
anthrax	1	0	0	0	0
queue	1	0	0	0	0
sleepy	0	0	1	0	0
wicker	1	0	0	0	0
what	0	0	0	0	0
apathy	0	0	0	1	0
fire	0	0	0	0	1

application	1	0	0	0	0
argument	0	1	0	0	0
fleet	0	1	0	0	1
scent	1	0	0	0	0
arsenic	0	0	0	0	1
artillery	0	0	0	0	1
artist	1	0	0	0	0
artistic	1	0	0	0	0
feeling	1	0	0	0	0
origin	1	0	0	0	0
acid	0	0	0	0	1
asbestos	1	0	0	0	0

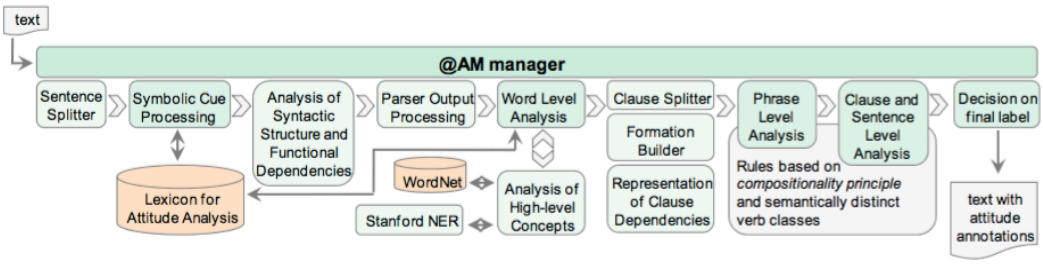
2.7 *ATtitude Analysis Model* (@AM)

@AM (*ATtitude Analysis Model*) adalah metode yang mengklasifikasikan kalimat ke dalam label perilaku atau emosi. @AM (*ATtitude Analysis Model*) bergantung pada *compositional principle*, pendekatan linguistik baru yang didasarkan pada aturan-aturan yang diuraikan untuk kelas kata semantik berbeda, dan metode yang mempertimbangkan konsep hirarki.

2.7.1 Arisitektur @AM (ATtitude Analysis Model) dan Tahapan Utama dari Analisis

Arsitektur dari sistem yang dikembangkan untuk mengenali perilaku emosi pada teks ditunjukan pada gambar 2.1 di bawah ini. Sumber teks akan dimasukan

ke dalam '*Sentence Splitter'* modul yang akan membagi dokumen menjadi paragraf, dan paragraf menjadi kalimat.



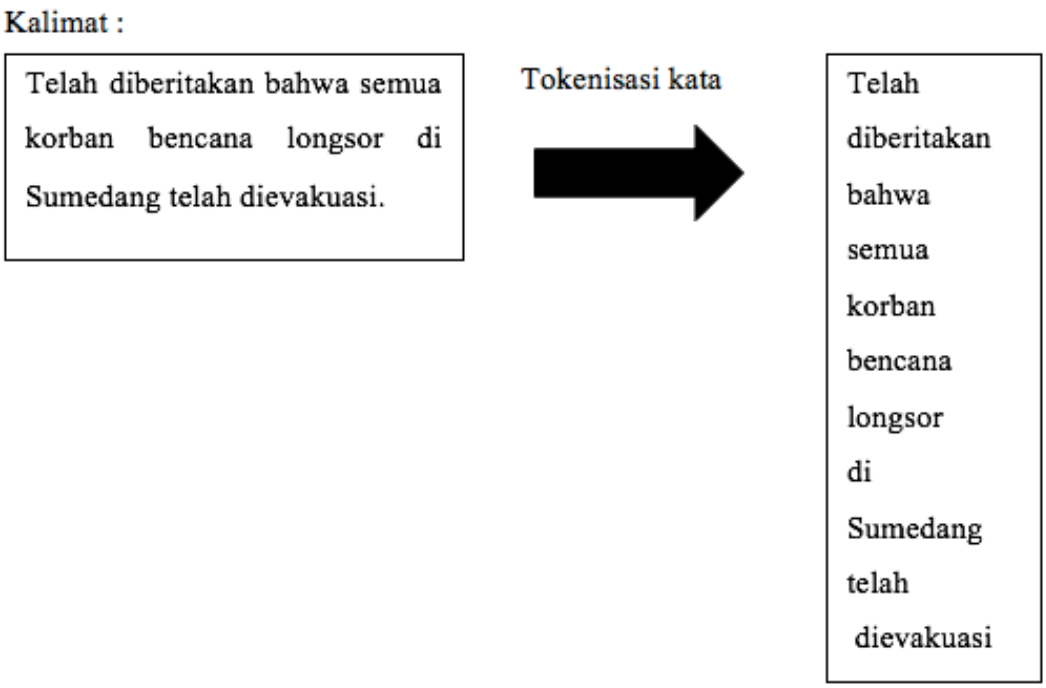
Gambar 2-1 Arsitektur @AM (ATtitude Analysis Model)

2.7.2 Symbolic Cue Processing

Tahap ini merupakan tahap awal yang dilakukan pada proses pengklasifikasian emosi pada teks menggunakan metode @AM Dalam tahap *ini* melibatkan tahapan-tahapan yaitu *tokenization, POS Tagging* dan *named entity recognition.*

2.7.2.1 *Tokenization*

Proses pemahaman bahasa/kalimat diawali dengan tokenisasi (pemotongan) suatu kalimat menjadi unsur-unsurnya yaitu kata. Berdasarkan hal tersebut, tokenisasi dilakukan dengan tujuan untuk mengolah kalimat dimulai dari unsur kalimatnya, yaitu kata. Berikut adalah contoh *tokenization* .



Gambar 2-2 Contoh *Tokenization*

.

2.7.2.2 POS Tagging

Part of Speech Tagging (POS-Tagging) adalah suatu proses yang memberikan label kelas kata secara otomatis pada suatu kata dalam kalimat [20]. Hasil dari POS ini sangat berpengaruh terhadap keluaran dari proses *Parsing*[21].

Sudah ada banyak penelitian yang dilakukan terkait dengan POS *Tagging* untuk teks berbahasa Indonesia. Metode-metode yang dipakai dalam penelitian-penelitian tersebut pun memiliki akurasi yang cukup tinggi, diantaranya Brill *tagger* yang memiliki akurasi 88%[22], *Conditional Random Field (CRF)* dan *Maximum Entropy Method* memiliki akurasi 97,57% [23] dan *Hidden Markov Model* (HMM) yang memiliki akurasi 96,5% [22]. POS tagger yang dipakai pada penelitian ini adalah POS tagger dengan metode HMM[22]. Berikut adalah daftar tagset yang dapat diidentifikasi oleh POS tagger yang bersangkutan[22].

Tabel 2-16 Daftar Tagset POS Tagger

POS	Keterangan	Contoh

OP	Open Parenthesis	({[
CP	Close Parenthesis	]})
GM	Slash	/
;	Semicolon	;
:	Colon	:
“	Quotation	“ ‘
.	Sentence terminator	. ! ?
,	Comma	,
-	Dash	-
...	Elipsis	...
JJ	Adjective	Kaya, manis
RB	Adverb	Nanti, sementara
NN	Common noun	Mobil
NNP	Proper Noun	Bekasi, indonesia
NNG	Genitive noun	Bukunya
VBI	Intrasitive verb	Pergi
VBT	Transitive verb	Membeli
IN	Preposition	Di, ke, dari
MD	Modal	Bisa
CC	Coo-conjunction	Dan, atau, tetapi
SC	Subor-conjucntion	Jika, ketika
DT	Determinan	Para, ini, itu
UH	Interjection	Wah, aduh, oi
CDO	Ordinal numerals	Pertama, kedua
CDC	Colective numerals	Bertiga
CDP	Primary numerals	Satu, dua
CDI	Irregular numerals	Beberapa
PRP	Personal pronouns	Saya, kamu
WP	WH-pronouns	Apa, siapa
PRN	Number pronouns	Kedua-duanya

PRL	Locative pronouns	Sini, situ, sana
NEG	Negation	Bukan, tidak
SYM	Symbols	@#$%^&
RP	Particles	Pin, kah
FW	Foreign words	Foreign, word

Berikut ini adalah contoh dari prosses POS tagging.

Tabel 2-17 Contoh Proses POS Tagging

Kalimat	**Hasil POS Tagging**	
	Kata	**POS Tag**
Ibu pergi ke pasar.	Ibu	NN
	Pergi	VB
	Ke	IN
	pasar	NN

Dari contoh kalimat pada tabel 2.18, kalimat *ibu pergi ke pasar* terdiri dari empat kata di mana setiap katanya memiliki POS tag yang berbeda beda, yaitu *ibu*/NN, *pergi*/VB, *ke*/IN, dan *pasar*/NN.

2.7.2.3 *Named Entity Recognition*

Named Entity Recognition (NER) adalah sebuah proses untuk mengenali dan mengklasifikasi istilah dan frase sebagai entitas nama dari sebuah teks bebas [24]. Ini adalah salah satu komponen utama dari *information extration* yang bertujuan untuk mendeteksi dan mengklasifikasikan named-entity pada suatu teks. NER umumnya digunakan untuk mendeteksi nama orang, nama tempat dan organisasi dari sebuah dokumen, tetapi dapat juga diperluas untuk identifikasi gen, protein dan lainnya sesuai kebutuhan. Penelitian ini menggunakan aplikasi Indonesia NER, dan menggunakan metode HMM [25][26]. Berikut adalah daftar entitas yang dapat diidentifikasi oleh NER yang bersangkutan.

Tabel 2-18 Daftar NER

NER	Contoh
PERSON	Joko Widodo
LOCATION	Bandung, Singapura
ORGANIZATION	Universitas Gadjah Mada
TIME	16.00
QUANTITY	Satu, dua
OTHER	lain-lain

Berikut adalah contoh proses pengenalan entitas bernama.

Tabel 2-19 Contoh Named Entity Recognition

Kalimat	Hasil NER	
	Kata	NER
Joko Widodo Pergi ke Singapura setelah mengunjungi DPR	*Joko Widodo*	PERSON
	Singapura	LOCATION
	DPR	ORGANIZATION

Setelah proses NER dijalankan, akan didapat *named-entity* (NE) atau sering disebut mention beserta tipenya: "Joko Widodo" bertipe PERSON; "Singapura" bertipe LOCATION; dan "DPR" bertipe ORGANIZATION. Dari contoh di atas dapat dilihat bahwa fungsi dari NER adalah mendeteksi kata atau kumpulan kata yang merupakan entitas dan mengkategorikan kata tersebut ke dalam tipe yang sesuai kebutuhan.

2.7.3 Anotasi Level Kata

Pada tahap analisis level kata, sistem akan mengecek ketersediaan token kalimat di dalam database perilaku dan memberikan anotasi sesuai kategorinya. Pada kasus kata yang mewakili emosi atau perilaku, fitur perilaku digambarkan sebagai vektor dari kekuatan perilaku (intensitas):

kata = [POS, NEG, gembira, marah, jijik, sedih, takut].

Sebagai contoh:

kata(‘bersemangat’) = [1, 0, 1, 0, 0, 0, 0];

kata('mengganggu') = [0, 1, 0, 0, 0, 1, 0];

Pada tahap ini pula dilakukan proses *convert emoticon.*

2.7.3.1 *Convert Emoticon*

Emoticon adalah kata gabungan dari “emotion” dan “icon” yang berarti icon yang digunakan untuk mengekspresikan emosi sebuah pernyataan tertulis, dan bisa mengubah serta meningkatkan interpretasi terhadap tulisan tersebut. *Emoticon* (emotion icon) merupakansalah satu cara pengungkapan perasaan secara tekstual. Hal ini tentu akan membantu dalam menentukan sentimen suatu tweet. Emosi yang bisa menunjukan sebagai sentimen positif ataupun negatif. Oleh karena itu setiap *emoticon* harus dikonversikan ke dalam bentuk kata yang mewakili arti dari *emoticon* tersebut agar bisa diproses. Setiap *emoticon* yang dikonversi akan diberi pemisah spasi untuk mengantisipasi *emoticon* yang berdampingan rapat oleh spasi dengan kata sebelum atau sesudah *emoticon*. Sehingga hasil konversi menjadi lebih mudah diproses. Daftar *emoticon* yang akan dikonversi.ditunjukan pada tabel berikut:

.

Tabel 2-20 Daftar *emoticon* [27]

Emoticon	**Deskripsi**
":-)", ":)", ":-]", ":]", ":-3", ":3", ":->", ":>", "8-)", "8)", ":-}", ":}", ":o)", ":c)", ":^)", "=]", "=)", ":-))", ":-D", ":D", "8-D", "8D", "x-D", "xD", "X-D", "XD", "=D", "=3", "B^D", ":'-)", ":')", "O:-)", "O:)", "0:-3", "0:3", "0:-)", "0:)", "0;^)"	**Gembira**
":-(", ":(", ":-c", ":c", ":-<", ":<", ":-[", ":[", ":-||", ">:[", ":{", ":@", ">:(", ">:-)", ">:)", "}:-)", "}:)", "3:-)", "3:)", ">;)"	**Marah**
"D:", "D-':"	**Jijik**
":'-(", ":'(", "D:<", "D8", "D;", "D=", "DX", ":$"	**Sedih**

":-/", ":/", ":-.", ">: >:/", ": =/", "= :L", "=L", ":S"	**Takut**

Setelah anotasi level kata, sistem lalu melanjutkan ke dalam tahap penerapan aturan-aturan kalimat yang sudah ditentukan untuk menentukan final label emosi pada suatu kalimat. Final label emosi ditentukan oleh nilai maksimum dari vektor emosi yang terbentuk.

2.7.4 *Compositionality Principle*

'The full story of how lexical items reflect attitudes is more complex than simply counting the valences of terms." Polanyi and Zaenen (2004)

Kata dalam kalimat saling berkaitan, oleh karena itu setiap kata dapat mempengaruhi keseluruhan makna dan dapat membiaskan pernyataan perilaku. Algoritma @AM untuk mengklasifikasikan perilaku di desain berdasarkan *compositionality principle,* dimana akan menentukan parilaku dalam suatu kalimat dengan menyusun setiap bagian yang sesuai dengan arti kata dalam kamus atau tipe unsur pokok lain dari suatu bahasa sesuai dengan aturan *polarity reversal, aggregation (*gabungan), *propagation, domination, neutralization,* dan *intensification* diberbagai tingkatan tata bahasa.

1. *Polarity reversal*

Polarity reversal berarti dimana suatu frasa atau pernyataan mengandung istilah yang menyampaikan perilaku / frasa dengan polaritas yang cenderung positif menjadi negatif atau sebaliknya. Aturan *polarity reversal* berlaku pada tiga kasus:

1) Kata negatif berelasi dengan suatu pernyataan perilaku:
   'tidak' & POS('sukses') => NEG('tidak sukses').
2) Kata keeterangan berelasi dengan suatu pernyataan perilaku:
   'hampir' & POS('santai') => NEG('hampir santai').
3) Kata 'fungsional' tipe *reversing* berelasi dengan suatu pernyataan perilaku:
   adjective 'kurang' & POS('antusias') => NEG('kurang antusias').

Penggunaan aturan *polarity reversal* dilakukan secara langsung dimana (‘POS’ ⇔ ‘NEG’). Tetapi tidak mudah menemukan lawan dari emosi pada kasus pengklasifikasian emosi sesuai dengan label perilaku, kecuali untuk kategori ‘Gembira’ dan ‘Sedih’, maka diasumsikan sebagai berikut:

1) Lawan emosi untuk emosi positif yaitu ‘Gembira’ adalah ‘Sedih’ contoh (‘POS’ => ‘Sedih’).
2) Lawan emosi dari empat emosi negatif yaitu, 'Marah', ‘Jijik’, 'Takut', dan ‘Sedih’ adalah ‘Gembira’ contoh (‘NEG’ => ‘Gembira’)

2. Aggregation

Aturan *aggregation (*penggabungan) adalah sebagai berikut:

1) Jika polaritas dari istilah yang menyampaikan perilaku dalam verb-adjective, noun-noun, frasa memiliki polaritas yang berlawanan, percampuran polaritas dengan dominan polaritas dari sebuah *premodifier* yang ada dalam frasa dalam hal ini kata adjective, contoh:

   NEG(‘tarung’) & POS(‘cantik’) => neg-POS(‘*tarung cantik’*);
2) Sebaliknya, hasil polaritas berdasarkan pada nilai polaritas yang sama dari suatu istilah, dan kekuatan perilaku diukur dari skor maksimum antara polaritas (max(Skor1, Skor2)).

3. Domination

Aturan *domination* akan digunakan apabila kalimat campuran digabungkan klausanya menggunakan konektor ‘tapi’, emosi utama dalam klausa setelah konektor dianggap dominan, contohnya:

> NEG(sulit untuk mendaki gunung semalaman), tapi POS(pemandangan yang indah menanti pendaki dipagi hari). => POS(seluruh kalimat).

4. Neutralization

Aturan *neutralization* diterapkan ketika:

1) Pernyataan yang menyampaikan perilaku disampaikan oleh suatu kata depan (*preposition)*:

‘meski’ & NEG(‘khawatir’) => NEUTRAL(‘meskipun khawatir’).

2) Pernyataan kondisional:
‘walau’ & NEG(‘mereka tidak sempurna’) => NEUTRAL(‘walau mereka tidak sempurna’).

Daftar kata yang termasuk dalam aturan *netralization* ditunjukan pada tabel 3-4 berikut.

Tabel 2-21 Daftar kata *netralization*

meski
walau
biar
meskipun
walaupun
biarpun

5. Intensification

Aturan *intensification* berarti menguatkan (atau melemahkan) skor perilaku (intensitas), dan berlaku ketika:

1) Kata keterangan berelasi dengan istilah yang menyampaikan perilaku:
Pos_score(‘hampir bahagia’) < Pos_score(‘bahagia’) < Pos_score(‘lebih bahagia’) < Pos_score(‘sangat bahagia’).
2) Kata sifat atau kata keterangan yang menyampaikan perilaku digunakan dalam bentuk komparatif atau superlatif
Pos_score(‘cantik’) < Pos_score(‘lebih cantik’) < Pos_score (‘sangat cantik’).

2.7.5 Evaluasi Dan Pengujian Metode @AM (ATtitude Analysis Model)

Metode @AM (ATtitude Analysis Model) menggunakan survey untuk menguji hasil klasifikasi. Final label emosi yang digunakan pada data survey adalah

kategori emosi yang paling banyak dipilih oleh para responden. Data hasil survey lalu akan diuji keandalannya menggunakan metode *fleiss' kappa.*

Sedangkan untuk besar akurasi yang dihasilkan akan diperoleh dari persentase hasil pembagian antara jumlah data klasifikasi emosi hasil metode @AM (ATtitude Analysis Model) dengan jumlah data klasifikasi emosi hasil survey. Persamaan perhitungan akurasi ditunjukan pada persamaan 2-1 sebagai berikut:

$$akurasi = \frac{total\ tweet\ yang\ akurat}{total\ tweet} \times 100\% \quad \textbf{(2.1)}$$

2.7.5.1 Fleiss' Kappa

Fleiss' kappa (diambil dari nama penemunya Joseph L. Fleiss) adalah ukuran secara statistik untuk menilai keandalan dari persetujuan antara sejumlah tetap penilai ketika menetapkan kategoris peringkat ke sejumlah item atau item klasifikasi. Berbeda dengan kappas lain seperti Cohen kappa, yang hanya bekerja ketika menilai persetujuan antara penilai tidak lebih dari dua penilai. Fleiss' kappa bekerja untuk berapapun jumlah penilai yang memberikan peringkat pada suatu kategori, kepada sejumlah tetap item [9].

Perjanjian dapat dianggap sebagai berikut, jika sejumlah orang yang tetap menetapkan peringkat numerik untuk sejumlah item maka kappa akan memberikan ukuran untuk bagaimana konsistensi suatu peringkat. Kappa $\boldsymbol{\kappa}$ dapat didefinisikan sebagai:

$$\kappa = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e} \quad \textbf{(2.2)}$$

Faktor $1 - \bar{P}_e$ memberikan tingkat kesepakatan yang dicapai di atas kesempatan dan, $\bar{P} - \bar{P}_e$ memberikan tingkat perjanjian yang benar-benar tercapai

di atas kesempatan. Jika penilai dalam perjanjian lengkap kemudian $\kappa = 1$, Jika tidak ada perjanjian antara para penilai (selain dari kebetulan) maka $\kappa \leq 0$.

2.7.5.1.1 Persamaan Fleiss Kappa

Misalkan N menjadi jumlah mata pelajaran, lalu n jumlah peringkat setiap subjek, dan k menjadi jumlah kategori di mana tugas yang dibuat. Subyek diindeks oleh i = 1,... N dan kategori diindeks oleh j = 1,... k. misalkan nij mewakili jumlah penilai yang ditugaskan i-th tergantung pada kategori j-th.

$$p_j = \frac{1}{Nn}\sum_{i=1}^{N} n_{ij}, \qquad 1 = \sum_{j=1}^{k} p_j \tag{2.3}$$

Sekarang menghitung Pj, sejauh mana penilai setuju untuk subjek i-th (yaitu, menghitung berapa banyak para penilai yang setuju, relatif terhadap jumlah semua kemungkinan para penilai):

$$\begin{aligned} P_i &= \frac{1}{n(n-1)}\sum_{j=1}^{k} n_{ij}(n_{ij}-1) \\ &= \frac{1}{n(n-1)}\sum_{j=1}^{k}(n_{ij}^2 - n_{ij}) \\ &= \frac{1}{n(n-1)}[(\sum_{j=1}^{k} n_{ij}^2) - (n)] \end{aligned} \tag{2.4}$$

Sekarang menghitung $\bar{P}$, mean dari *Pi*, dan $\bar{P}_e$ yang akan dimasukan ke dalam rumus untuk k sebagai berikut:

$$\bar{P} = \frac{1}{N}\sum_{i=1}^{N} P_i$$

$$= \frac{1}{Nn(n-1)}\left(\sum_{i=1}^{N}\sum_{j=1}^{k} n_{ij}^2 - Nn\right) \quad \textbf{(2.5)}$$

Persamaan untuk menghitung nilai Pe adalah sebagai berikut:

$$\bar{P}_e = \sum_{j=1}^{k} p_j^2 \quad \textbf{(2.6)}$$

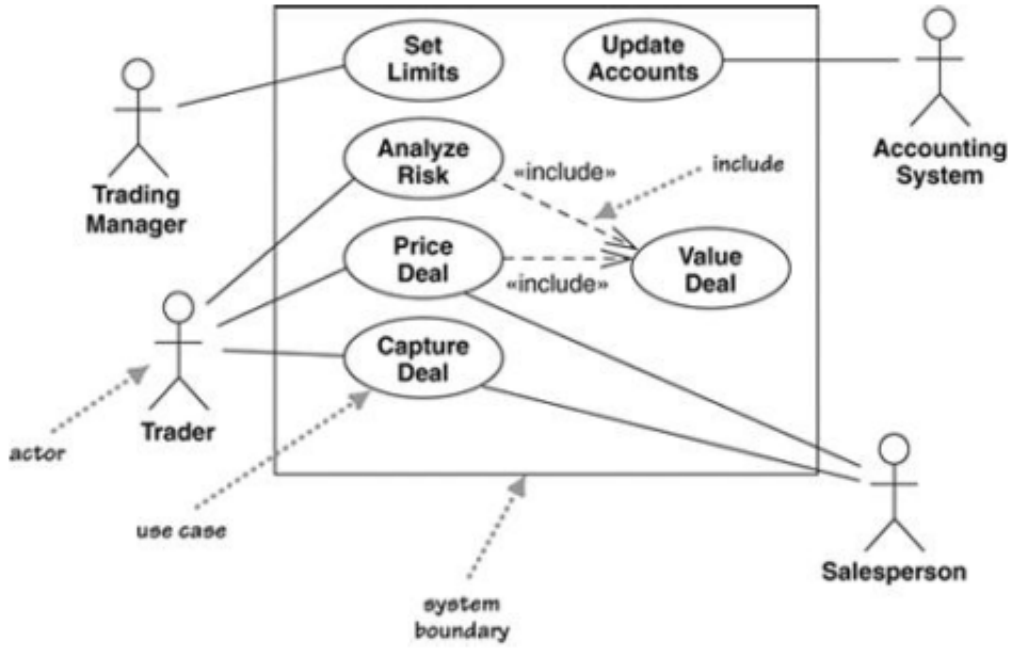
2.8 *Unified Modelling Language* (UML)

Unified Modeling Language (UML) adalah adalah keluarga notasi grafis yang membantu dalam menggambarkan dan merancang sistem perangkat lunak terutama yang dibangun dengan menggunakan orientasi objek. Definisi ini adalah definisi yang telah disederhanakan. Pada kenyataannya, pendapat orang-orang berbeda tentang UML berbeda satu sama lain. Hal ini disebabkan oleh sejarah dan perspektif yang berbeda tentang membuat proses rekayasa perangkat lunak yang efektif [28].

UML yang digunakan dalam penelitian ini adalah UML versi 2.0 [29]. Dan diagram-diagram UML yang digunakan pada sistem ini adalah *use case diagram*, *activity diagram*, *class diagram*, dan *sequence diagram*.

2.8.1 *Use Case Diagram*

Use case adalah teknik untuk menggambarkan kebutuhan fungsional dari sebuah sistem. *Use case* menggambarkan interaksi antara pengguna dengan sistem, menyediakan sebuah cerita bagaimana sebuah sistem digunakan. Dalam bahasan *use case*, para pengguna disebut sebagai aktor. Aktor merupakan sebuah peran yang dimainkan seorang pengguna dalam kaitannya dengan sistem. Aktor tidak harus manusia, jika sebuah sistem melakukan sebuah layanan untuk sistem komputer lain, sistem lain tersebut merupakan aktor [28]. Berikut adalah contoh *use case* diagram.

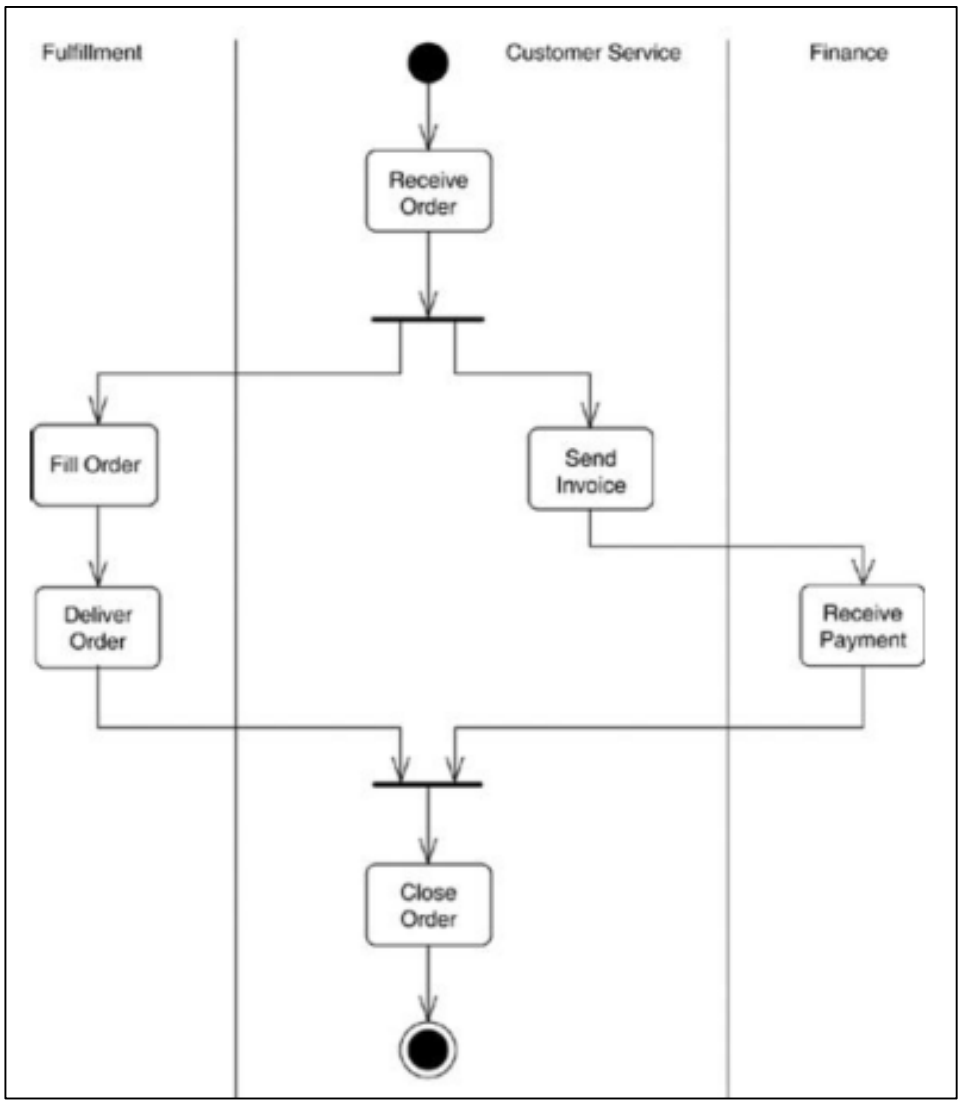


Gambar 2-3 Contoh Use Case [29]

Dari contoh di atas, dapat dilihat bahwa pengguna terdiri dari empat actor, yaitu *Trading manager*, *trader*, *sales person* dan *accounting system*. Tiap *actor* dapat mengakses *use case* yang berbeda-beda sesuai dengan fungsi nya. *Trading manager* dapat mengakses *use case* **Set Limit**, *Trader* dapat mengakses *use case* **Analyze risk, price deal** dan **capture deal**, *Sales person* dapat mengakses *use case* **price deal** dan **captur deal**, sedangkan *Accounting system* dapat mengakses *use case* **update accounts**.

2.8.2 *Activity Diagram*

Activity diagram adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan jalur kerja. Dalam beberapa hal, activity diagram memainkan peran mirip diagram alir, tetapi perbedaan prinsip antara notasi diagram alir adalah *activity* diagram mendukung behavior paralel [28]. Node pada sebuah activity diagram disebut sebagai *action*, sehingga diagram tersebut menampilkan sebuah activity yang tersusun dari action. Berikut adalah contoh activity diagram.



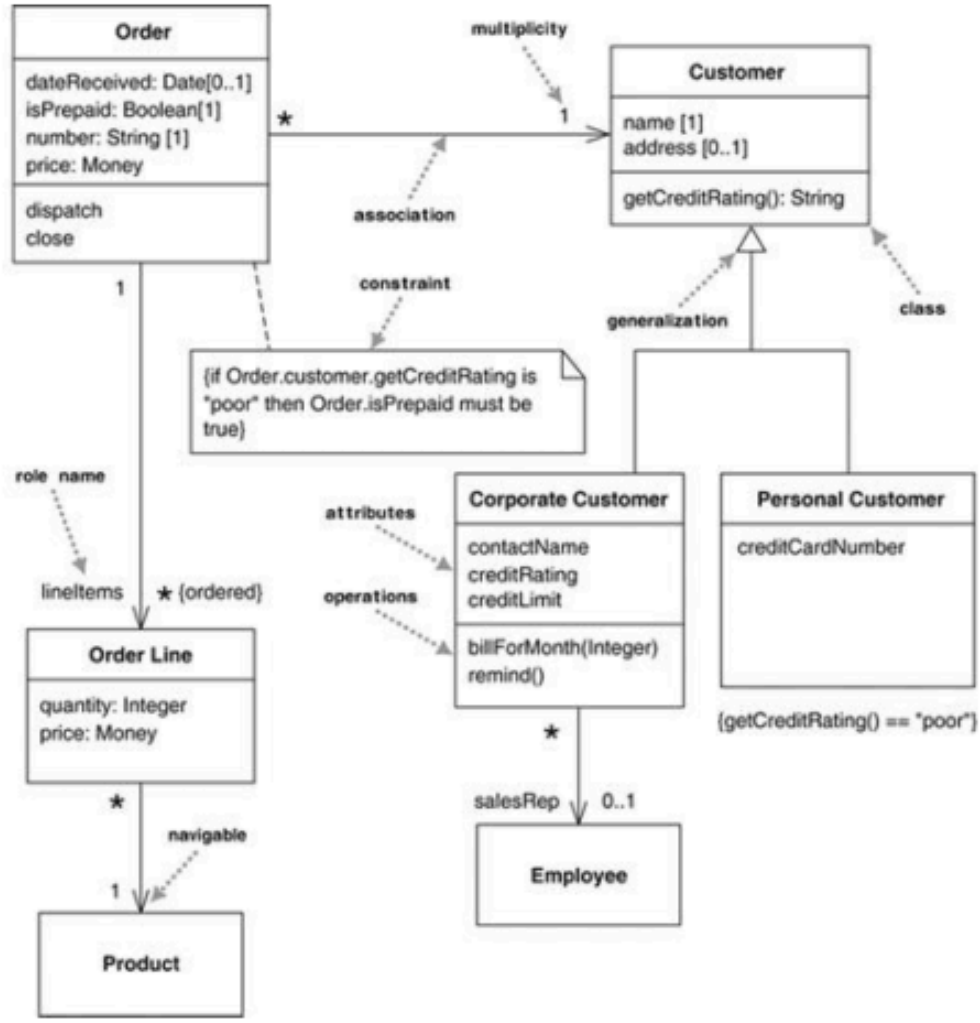
Gambar 2-4 Contoh Activity Diagram [29]

Dari contoh di atas, dapat dilihat aktivitas pemrosesan order. Diawali dengan *customer service* yang menerima *order*, kemudian dilanjutkan dengan pendataan *order* dan pengiriman invoice yang dilakukan secara paralel. Setelah pembayaran diterima oleh finance, dan order telah dikirim maka *customer service* akan menutup order yang bersangkutan.

2.8.3 *Class Diagram*

Class diagram menggambarkan jenis-jenis objek yang terdapat pada sistem dan berbagai macam relasi statis yang ada diantara objek-objek tersebut. *Class*

diagram juga menunjukkan properti-properti dan operasi-operasi pada suatu *class* dan batasan atau *constraint* yang berlaku pada cara objek-objek tersebut saling berhubungan [28]. Berikut adalah contoh class diagram.



Gambar 2-5 Contoh Class Diagram [29]

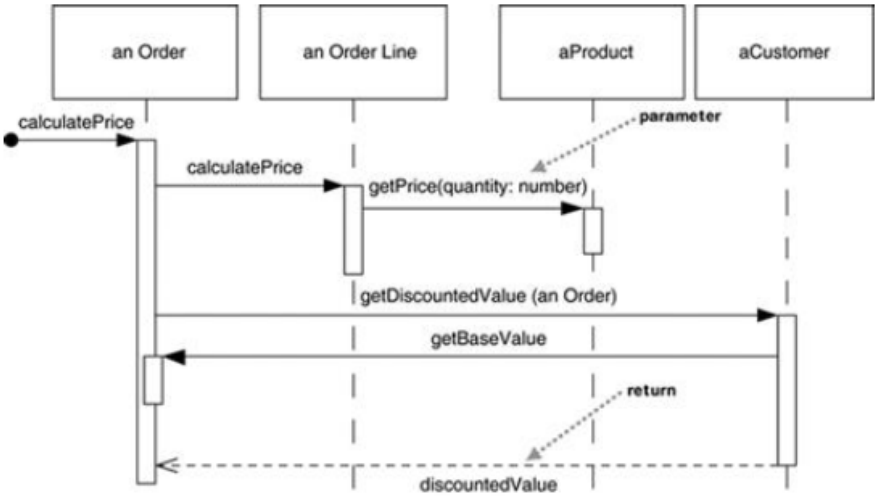
Multiplicity dari suatu properti adalah indikasi seberapa banyak objek yang dapat mengisi properti tersebut. Secara umum, *multiplicity* didefinisikan dengan batas bawah dan batas atas. Batas bawah berupa angka positif dan nol, sedangkan batas atas berupa angka positif dan * (tidak terbatas). Berikut adalah tabel *multiplicity* yang terdapat pada *class diagram*.

Tabel 2-22 Daftar *Multiplicity*

Multiplicity	**Arti**
0..0	Nol
0..1	Nol atau Satu
0..*	Nol atau Banyak
1..1	Tepat satu
1. .*	Satu atau banyak

2.8.4 *Sequence Diagram*

Interaction diagrams menggambarkan bagaimana kumpulan objek dapat berkolaborasi dalam tindakan tertentu. UML mendefinisikan beberapa bentuk *interaction diagram*, salah satunya yang paling umum adalah *sequence diagram*. Secara khusus, *sequence diagram* menangkap perilaku dari suatu skenario. Diagram ini menunjukkan sejumlah objek contoh dan pesan-pesan yang melewati objek-objek tersebut dalam suatu *use case* [28]. Berikut adalah contoh diagram *sequence*.



Gambar 2-6 Contoh Diagram Sequence [29]

2.9 Bahasa Pemrograman

Bahasa pemrograman adalah sebuah instruksi standar untuk memerintah komputer agar menjalankan fungsi tertentu. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Ada banyak macam bahasa pemrograman, diantaranya *Java*, Ruby, PHP, *Phyton*, C, C++, dan lain lain. Dalam sistem yang akan dibangun pada penelitian ini, bahasa pemrograman yang dipakai adalah *Ruby version 2.3.0.*

2.9.1 Ruby

Ruby merupakan bahasa yang seimbang. Pencipta Ruby, Yukihiro "Matz" Matsumoto, menggabungkan bagian-bagian dari bahasa-bahasa favorit beliau (Perl, Smalltalk, Eiffel, Ada dan Lisp) untuk membentuk bahasa baru yang seimbang antara pemrograman fungsional dengan pemrograman imperatif [30]. Matz sering menyebutkan bahwa beliau sedang "mencoba membuat Ruby natural, bukan sederhana," dengan cara membuatnya mirip dengan kehidupan nyata. Berprinsip pada hal ini, Matz menambahkan:

> *"Penampilan Ruby dari luar sederhana, tetapi sangat rumit di bagian dalam, seperti badan manusia kita ini".*

Sejak Ruby pertama kali dirilis ke publik pada tahun 1995, banyak programmer profesional dari seluruh dunia serius ikut mengembangkan Ruby. Pada tahun 2006, Ruby diterima oleh banyak orang. Dengan komunitas pengguna Ruby yang aktif di banyak kota-kota di seluruh dunia dan konferensi-konferensi beserta pertemuan terkait Ruby [30].

Pertama kali, Matz melihat bahasa-bahasa lain untuk mencari sintaks yang ideal. Terkenang pencariannya, Matz berkata, "Saya mau bahasa scripting yang lebih hebat daripada Perl dan lebih berorientasi obyek daripada Python.". Di Ruby, semua adalah obyek. Setiap informasi dan kode bisa diberi property dan action. Pemrograman berorientasi obyek memanggil property dengan nama variabel instan

dan action, yang disebut sebagai metode. Pendekatan murni berorientasi obyek terutama terlihat pada demonstrasi sedikit kode yang diberikan pada number.

```
5.times { print "Kami *cinta* Ruby -- Ruby sungguh aduhai!" }
```

Di banyak bahasa-bahasa lain, number dan tipe primitif bukan obyek. Ruby mengikuti pengaruh bahasa Smalltalk dengan memberikan metode dan variabel instan pada semua tipe. Ini memudahkan menggunakan Ruby, karena peraturan-peraturan mengenai obyek semua berlaku pada Ruby [30].

2.9.1.1 Ruby Fleksibel

Ruby dianggap sebagai bahasa yang fleksibel, karena bagian-bagian dari Ruby bisa diubah-ubah dengan bebas. Bagian-bagian yang esensi di Ruby bisa dihapus maupun didefinisikan ulang. Bagian-bagian yang sudah ada bisa ditambahkan. Ruby mencoba untuk tidak membatasi programmer. Misalnya, penambahan dilakukan dengan operator plus (+). Tetapi, jika Anda ingin menggunakan kata plus yang lebih mudah dibaca, maka Anda dapat menambahkan metode tersebut pada kelas builtin Numeric [30].

```
Class Numeric
  def plus(x)
    self.+(x)
  end
end

y = 5.plus 6
# y sekarang adalah 11
```

2.9.1.2 Blok yang Ekspresif

Blok Ruby juga dianggap sebagai sumber kekuatan Ruby yang sangat fleksibel. Programmer dapat menyertakan closure pada setiap metode, menjelaskan bagaimana metode yang bersangkutan seharusnya berperilaku. Closure disebut blok dan telah menjadi satu diantara banyak fitur-fitur Ruby yang paling populer

pada banyak pendatang baru Ruby dari bahasa-bahasa imperatif lain seperti PHP atau Visual Basic [30]. Blok terinspirasi dari bahasa-bahasa fungsional. Matz berkata, “Saya ingin menghormati kultur Lisp di closure Ruby.”

```
search_engines =
  %w[Google Yahoo MSN].map do |engine|
    "http://www." + engine.downcase + ".com"
  end
```

Pada kode diatas, blok dijelaskan dalam bentuk do ... end. Metode map memberlakukan blok agar menerima array kata-kata (Google, Yahoo dan MSN). Banyak metode-metode lain di Ruby dibiarkan mempunyai hole yang dibuka untuk programmer agar menulis blok mereka sendiri untuk mengisi dengan lebih lengkap apa saja yang seharusnya sebuah metode lakukan [30].

2.9.1.3 Tampilan Ruby secara Visual

Ruby jarang menggunakan tanda baca dan biasanya cenderung menggunakan keyword berbahasa Inggris, biasanya beberapa tanda baca digunakan untuk memperjelas kode Rub [30]. Ruby tidak perlu deklarasi variabel. Ruby menggunakan aturan penamaan yang mudah untuk menyatakan scope suatu variabel [30].

Scope variabel sebagai berikut:

- var adalah variabel lokal.
- @var adalah variabel instan.
- $var adalah variabel global.

2.9.2 SQL

SQL adalah bahasa untuk menghasilkan, memanipulasi, dan mengambil data dari database relasional. Salah satu alasan untuk popularitas database relasional adalah bahwa dia dirancang dengan baik dan dapat menangani sejumlah besar data [31]. SQL merupakan bahasa yang paling banyak diimplentasikan dalam basis data, sehingga menjadi standar bahasa untuk mengelola basis data. Bahasa SQL terbagi

menjadi beberapa bagian, yaitu *schema statements* yang digunakan untuk mendefinisikan struktur data yang tersimpan dalam basis data, *data statements* yang digunakan untuk memanipulasi struktur data yang telah didefinisikan sebelumnya oleh *schema statements*, dan terakhir adalah *transaction statements* yang digunakan untuk memulai, mengakhiri, atau *roll back* transaksi [31].

2.10 Alat Bantu Pengembangan Perangkat Lunak

Pada penelitian ini, tahap pengembangan perangkat lunak akan menggunakan Ruby On Rails dan MySQL sebagai alat bantu bahasa pemograman.

2.10.1 Ruby On Rails

Ruby on Rails diciptakan oleh David Heinemeier Hansson sebagai semacam produk dari Basecamp's development di 37signals pada tahun 2004. Basecamp dibangun menggunakan Ruby karena Hansson merasa PHP dan Java tidak kuat atau cukup fleksibel. Hal ini cukup mengaburkan bahasa, tanpa eco-system besar yang tersedia pada saat itu. Untuk membuat pengembangan lebih mudah, Hansson membuat web framework sendiri, didasarkan pada ide-ide sederhana yang telah terbukti berhasil. Rails didasarkan pada pragmatisme dan mendirikan ide-ide baru yang eksotis. Dan itulah yang membuatnya begitu sukses [32].

Rails didasarkan pada pola Model-View-Controller yang membagi aplikasi menjadi tiga bagian:

- Model, Model mewakili struktur data. Biasanya model berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data dan lain-lain.
- View, View adalah bagian yang mengatur tampilan ke pengguna. Bisa di katakan berupa halaman web.
- Controller, Controller merupakan bagian yang menjembatani model dan view. Controller berisi perintah-perintah yang berfungsi untuk memproses suatu data dan mengirimkannya ke halaman web.

  Rails adalah "opinionated software", artinya ia tidak ingin menjadi segalanya bagi semua orang. Lebih memfokuskan pada salah satu cara dalam

melakukan sesuatu dan merampingkan semua komponen yang ada. Itu bukan berarti tidak ada kemungkinan jika anda perlu untuk melakukan sesuatu yang berbeda. Tetapi anda pasti akan mudah jika anda melakukan hal-hal yang mengikuti aturan pada rails (the Rails Way).

2.10.2 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (database management system) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Berbeda dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius. Perintah-perintah yang ada di SQL antara lain [33] :

a. Menambah data (Insert)

   Perintah SQL yang digunakan untuk menambah data ke database adalah perintah INSERT. Perintah ini mempunyai bentuk umum sebagai berikut: INSERT [INTO] nama_tabel [(daftar field/kolom)] VALUES (daftar_data).

b. Mengakses data (Select)

   Perintah SQL yang digunakan untuk memilih data adalah perintah SELECT. Bentuk umum perintah ini adalah sebagai berikut:

   - SELECT daftar_select FROM daftar_tabel
   - [WHERE kondisi_pencarian]
   - [GROUP BY daftar_group_by] [HAVING kondisi_pencarian]
   - [ORDER BY daftar_order[ASC | DESC]]

c. Mengedit/Mengubah data (Update)

Perintah SQL yang digunakan untuk mengubah data adalah perintah UPDATE SET. Perintah ini mempunyai bentuk umum sebagai berikut:

- UPDATE nama_table SET field1=databaru1 [,data2=databaru2]
- [WHERE kondisi_update].

d. Menghapus data (Delete)

Perintah SQL yang digunakan untuk menghapus data adalah perintah DELETE FROM. Perintah ini mempunyai bentuk umum sebagai berikut: DELETE FROM tabel_hapus WHERE kondisi_hapus.

MySQL memiliki beberapa kelebihan diantaranya [15] :

1. MySQL sebagai Data Management System (DBMS) dan Relation Databese Management System (RDBMS).
2. MySQL adalah software database yang OpenSource, artinya program ini bersifat free atau bebas digunakan oleh siapa saja tanpa harus membeli dan membayar lisensi kepada pembuatnya.
3. MySQL merupakan database server, jadi dengan menggunakan database ini dapat menghubungkan ke media internet sehingga dapat diakses jarak jauh.
4. MySQL merupakan sebuah database client. Selain menjadi server yang melayani permintaan, MySQL juga dapat melakukan query yang mengakses database pada server, jadi MySQL dapat juga berperan sebagai Client.
5. MySQL mampu menerima query yang bertumpuk dalam satu permintaan atau yang disebut Multi-Threading.
6. MySQL merupakan sebuah database yang mampu menyimpan data berkapasitas sangat besar hingga berukuran Gigabyte sekalipun.
7. MySQL didukung oleh driver ODBC, artinya database MySQL dapat diakses menggunakan aplikasi apa saja termasuk berupa visual seperti Delphi maupun Visual Basic.
8. MySQL adalah database menggunakan enkripsi password. Jadi database ini cukup aman karena memiliki password untuk mengaksesnya.

9. MySQL dapat menciptakan lebih dari 16 kunci per tabel dan dalam satu kunci memungkinkan berisi belasan field. Serta MySQL mendukung field yang dijadikan sebagai kunci primer dan kunci unik atau (Unique).
10. MySQL memiliki kecepatan dalam pembuatan tabel maupun peng- update-an tabel.