

## **BAB II**

### **LANDASAN TEORI**

#### **2.1     *Text Mining***

*Text mining* adalah suatu bidang khusus dari data mining. *Text mining* dapat diartikan sebagai suatu proses untuk menggali informasi dimana seorang *user* (pengguna) berinteraksi dengan kumpulan dokumen menggunakan *tool* analisis yang merupakan komponen-komponen dalam data mining. Dalam *text mining* berbeda dengan data mining dimana data mining yang digunakan adalah *structured* (tersusun) data sementara dalam *text mining* umumnya data yang ditemui adalah *semi-structured* (sebagian terstruktur) atau *unstructured* (tidak terstruktur). Sementara keduanya memiliki masalah yang sama yaitu adalah jumlah data yang besar, dimensi yang tinggi, dan data juga struktur yang terus berubah. Struktur teks yang kompleks dan tidak lengkap, arti yang tidak jelas dan tidak standar, dan bahasa yang berbeda ditambah terjemahan yang tidak akurat memberikan tantangan tambahan pada *text mining*. *Text mining* dalam prakteknya mencari pola-pola tertentu, mengasosiasikan satu bagian teks dengan lain berdasar aturan-aturan tertentu, kata-kata yang dapat diwakilkan sehingga dapat dilakukan analisa keterhubungan antar satu dengan lain, dalam kumpulan dokumen yang sangat banyak. Dokumen yang ada bisa bersifat statis, yaitu dokumen yang tidak akan di perbarui lagi ataupun dinamis yaitu dokumen yang akan selalu diperbarui dalam rentang waktu tertentu (Amburika, Chrisnanto, & Uriawan, 2016).

*Text mining* adalah proses untuk menggali sumber-sumber tertulis yang besar untuk menghasilkan informasi yang baru (Mandar, 2014). Data yang diperoleh dapat berupa data yang bersifat kualitatif maupun data yang tidak terstruktur. Secara umum teks mining bertujuan untuk menggali informasi dari sekumpulan dokumen teks dalam ukuran besar dan dalam format yang tidak terstruktur. Secara khusus, tujuan dari *text mining* dapat dibagi menjadi dua (Eirene, 2014):

1. Pengkategorisasian data teks (*text categorization*)

Dalam pengkategorisasian, text mining dipergunakan sebagai alat untuk menemukan kategori yang sesuai dengan kelas yang telah ditentukan (*supervised learning*)

2. Pengelompokan data teks (*text clustering*)

Pada pengelompokan, text mining berfungsi sebagai alat untuk mengelompokkan data teks berdasarkan kesamaan karakteristik, dan clustering dapat digunakan untuk memberikan label pada kelas yang belum diketahui (*unsupervised learning*)

## **2.2 Sentiment Analysis**

*Sentiment analysis* adalah mengekstraksi pendapat, sentimen, evaluasi dan emosi orang tentang suatu topik tertentu yang tertulis menggunakan teknik pemrosesan bahasa alami. Sejumlah karya-karya besar lainnya menyebutkan *sentiment analysis* fokus pada aplikasi spesifik yang mengklasifikasikan mengenai opini positif, negatif dan netral (Sentiaji & Bachtiar, 2015).

*Sentiment analysis* atau yang disebut juga dengan *opini mining* merupakan analisis yang bertujuan untuk melihat opini masyarakat atau kelompok mengenai entitas tertentu (Apriliyanti P, 2015).

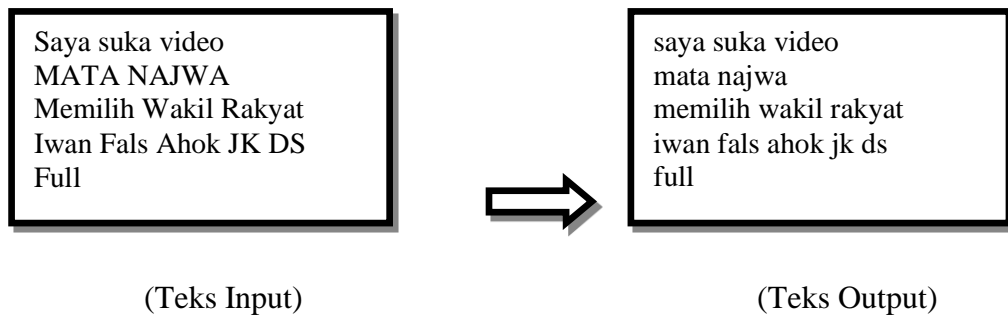
Jadi *sentiment analysis* atau *opinion mining* ini mengacu pada bidang yang luas dari pengolahan bahasa alami, komputasi linguistik dan *text mining*. Secara umum, bertujuan untuk menentukan *attitude* seorang penulis berkenaan dengan topik tertentu. *Attitude* mungkin penilaian atau evaluasi, pernyataan afektif (pernyataan emosional saat menulis) atau komunikasi emosional dimaksud (efek emosional penulis inginkan terhadap pembaca).

### 2.3 Preprocessing

Tahap *preprocessing* diperlukan untuk membersihkan data dari teks yang tidak diperlukan, dimana data teks yang tidak terstruktur akan diubah menjadi data teks yang terstruktur atau semi terstruktur. Tahap dari *preprocessing* untuk mengolah data yaitu *case folding*, *convert emoticon*, *cleansing*, *tokenizing*, *stopword removal* dan *stemming*.

#### 1. Case Folding

*Case folding* merupakan tahapan merubah bentuk kata-kata menjadi sama bentuknya, baik semuanya menjadi *lower case* ataupun menjadi *upper case* (Sentiaji & Bachtiar, 2015). Berikut ini merupakan contoh *case folding* dapat dilihat pada gambar 2.1 :



Gambar 2.1 Contoh *Case Folding*

#### 2. Convert Emoticon

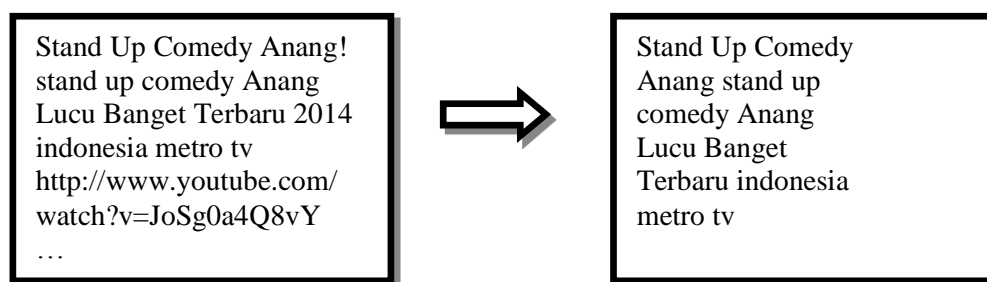
*Convert Emoticon* adalah proses untuk mengubah *emoticon* yang terdapat pada teks menjadi sebuah *string* (Jayanti. dkk, 2016). *Emoticon* merupakan salah satu cara pengungkapan ekspresi perasaan secara tekstual. Hal tersebut akan membantu menentukan sentimen dalam suatu kalimat atau teks (Sentiaji & Bachtiar, 2015). Berikut ini merupakan contoh *convert emoticon* dapat dilihat pada tabel 2.1 :

Tabel 2.1 Daftar *Convert Emoticon*

Emoticons	Deskripsi
:) :-) ;) ;-)^_^ :P o.O :] :-] :3 :c) :> :[] 8) :^ ) :~) O:)	Senang
:v :D :-D B) B  8D x-D xD ==D =D ==3 B^D	Tawa
:-  :@ >:( 3:) :-?	Benci
>:O >:( -_- :- ( :/ :? ( ;( :-c :c :-< :~C :< :-[ :[	Kecewa
<3 :* (y)	Suka

3. *Cleansing*

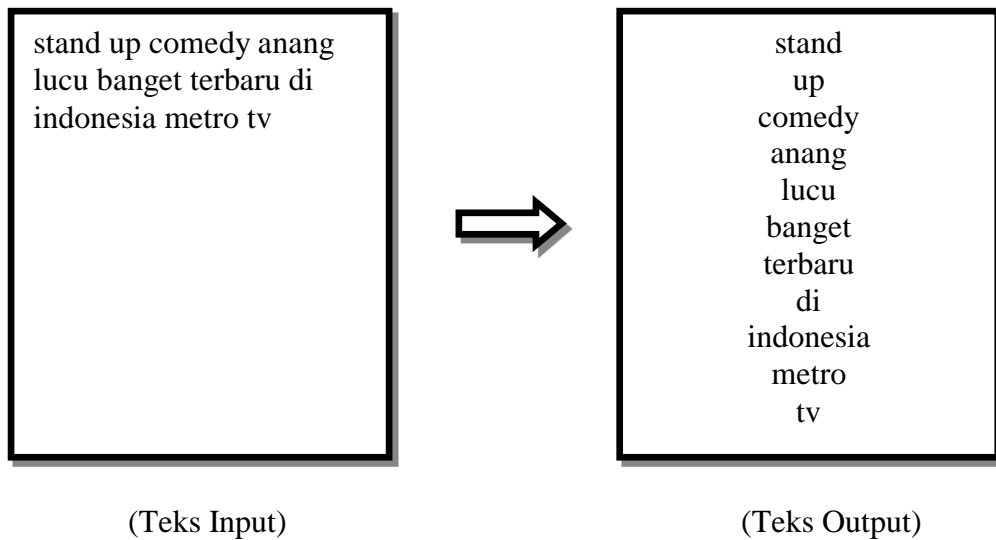
*Cleansing* merupakan proses membersihkan kata-kata yang tidak diperlukan untuk mengurangi noise (Sentiaji & Bachtiar, 2015). Kata yang dihilangkan adalah URL, *hashtag* (#), *username* (@username), email dan tanda baca (!@#\$%^&\*( )\_+ =,.,<>?/|{}[]1234567890). Berikut ini merupakan contoh *cleansing* dapat dilihat pada gambar 2.2 :



Gambar 2.2 Contoh *Cleansing*

4. *Tokenizing*

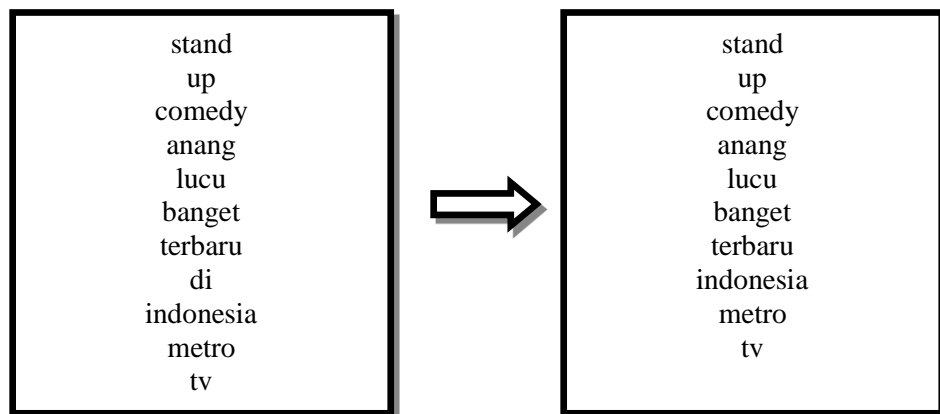
*Tokenizing* berguna untuk memisahkan deretan kata menjadi beberapa urutan yang terpotong oleh spasi atau karakter spesial. (Sentiaji & Bachtiar, 2015). Berikut merupakan contoh *tokenizing* dapat dilihat pada gambar 2.3 :



Gambar 2.3 Contoh *Tokenizing*

5. *Stopword Removal*

*Stopword removal* berperan untuk membuang kata-kata yang sering muncul dan bersifat umum, kurang menunjukkan relevansinya dengan teks (Sentiaji & Bachtiar, 2015). Kata-kata yang akan dibuang tersebut didefinisikan dalam *stopword list*. Contoh beberapa kata yang sering masuk ke dalam *stopword list* adalah “sebuah”, “yang”, “itu”, “di” dan lain-lain. Berikut ini merupakan contoh proses *stopword removal* dapat dilihat pada gambar 2.4 :



(Teks Input)

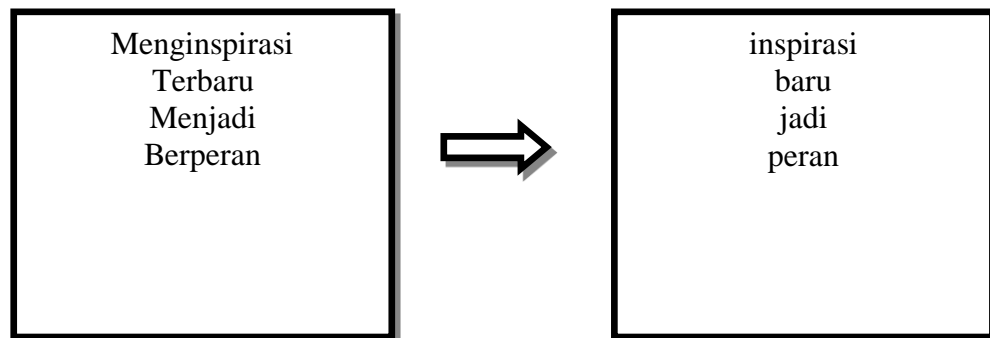
(Teks Output)

Gambar 2.4 Contoh *Stopword Removal*

6. *Stemming*

*Stemming* adalah tahapan untuk membuat kata yang berimbuhan kembali ke bentuk asalnya. Kata-kata yang muncul di dalam dokumen yang sering mempunyai banyak varian morfologik. Karena itu, setiap kata yang bukan *stopwords* direduksi ke *term* yang cocok untuk mendapatkan bentuk akarnya dengan menghilangkan awalan atau akhiran. (Sentiaji & Bachtiar, 2015).

*Stemming* berarti sebuah proses mencari akar kata dari sebuah kata yang diberikan. Proses *stemming* dapat memecahkan setiap imbuhan dari suatu kata seperti awalan (prefiks), sisipan (infiks), akhiran (sufiks), awalan akhiran (konfiks). Hasil dari proses *stemming* ini akan menghasilkan akar kata yaitu bagian dari kata yang tersisa setelah dihilangkan imbuhan. (Febriyani, 2015). Berikut ini merupakan contoh proses *stemming* dapat dilihat pada gambar 2.5 :



Gambar 2.5 Contoh *Stemming*

## 2.4 Facebook

*Facebook* merupakan situs media sosial yang memiliki pengguna aktif terbanyak di dunia. Sejak berdirinya Facebook pada tahun 2004 hingga Juni 2014, Facebook telah memiliki 1,32 Milyar pengguna aktif. Dimana Indonesia menduduki

peringkat ke empat setelah Amerika, Brazil dan India, dengan jumlah 65 juta pengguna Facebook aktif. Sebanyak 33 juta pengguna aktif per harinya, 55 juta pengguna aktif yang memakai perangkat *mobile* dalam pengaksesannya per bulan dan sekitar 28 juta pengguna aktif yang memakai perangkat *mobile* per harinya (kominfo.go.id).

## 2.5 Algoritma Naive Bayes Classifier

Pendekatan Naive Bayes merupakan sebuah metode klasifikasi yang mengacu pada teorema Bayes. Teorema Bayes digunakan untuk menghitung probabilitas ketidakpastian data (Russel & Norvig 2010). Proses pendekatan Naive Bayes Classifier mengasumsikan bahwa ada atau tidaknya suatu fitur pada suatu kelas tidak berhubungan dengan ada atau tidaknya fitur lain di kelas yang sama (Setiawan & Ratnasari 2015). Pada saat klasifikasi, pendekatan Bayes akan menghasilkan tabel kategori yang paling tinggi nilai probabilitasnya yaitu VMAP (Maximum Apriori Probability) dengan atribut inputan  $G_1, G_2, \dots, G_n$  (Aribowo 2010).

Persamaan dari teorema Bayes adalah (Bustami, 2014) :

$$P(H|X) = \frac{P(X|H) \cdot P(H)}{P(X)} \quad (1)$$

Dimana :

X : Data dengan class yang belum diketahui

H : Hipotesis data merupakan suatu class spesifik

$P(H|X)$  : Probabilitas hipotesis H berdasar kondisi X (posteriori probabilitas)

$P(H)$  : Probabilitas hipotesis H (prior probabilitas)

$P(X|H)$  : Probabilitas X berdasarkan kondisi pada hipotesis H

$P(X)$  : Probabilitas X

Untuk menjelaskan metode Naive Bayes, perlu diketahui bahwa proses klasifikasi memerlukan sejumlah petunjuk untuk menentukan kelas apa yang cocok bagi

sampel yang dianalisis tersebut. Karena itu, metode Naive Bayes di atas disesuaikan sebagai berikut:

$$P(C|F_1 \dots F_n) = \frac{P(C)P(F_1 \dots F_n|C)}{P(F_1 \dots F_n)} \quad (2)$$

Di mana Variabel C merepresentasikan kelas, sementara variabel  $F_1 \dots F_n$  merepresentasikan karakteristik petunjuk yang dibutuhkan untuk melakukan klasifikasi. Maka rumus tersebut menjelaskan bahwa peluang masuknya sampel karakteristik tertentu dalam kelas C (Posterior) adalah peluang munculnya kelas C (sebelum masuknya sampel tersebut, seringkali disebut prior), dikali dengan peluang kemunculan karakteristik-karakteristik sampel pada kelas C (disebut juga likelihood), dibagi dengan peluang kemunculan karakteristik-karakteristik sampel secara global (disebut juga evidence). Karena itu, rumus di atas dapat pula ditulis secara sederhana sebagai berikut:

$$\text{Posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}} \quad (3)$$

Nilai Evidence selalu tetap untuk setiap kelas pada satu sampel. Nilai dari posterior tersebut nantinya akan dibandingkan dengan nilai-nilai posterior kelas lainnya untuk menentukan ke kelas apa suatu sampel akan diklasifikasikan. Penjabaran lebih lanjut rumus Bayes tersebut dilakukan dengan menjabarkan  $(C|F_1, \dots, F_n)$  menggunakan aturan perkalian sebagai berikut:

$$\begin{aligned} P(C|F_1, \dots, F_n) &= P(C)P(F_1, \dots, F_n|C) \\ &= P(C)P(F_1|C)P(F_2, \dots, F_n|C, F_1) \\ &= P(C)P(F_1|C)P(F_2|C, F_1)P(F_3, \dots, F_n|C, F_1, F_2) \\ &= P(C)P(F_1|C)P(F_2|C, F_1)P(F_3|C, F_1, F_2)P(F_4, \dots, F_n|C, F_1, F_2, F_3) \\ &= P(C)P(F_1|C)P(F_2|C, F_1)P(F_3|C, F_1, F_2) \dots P(F_n|C, F_1, F_2, F_3, \dots, F_{n-1}) \end{aligned}$$

Dapat dilihat bahwa hasil penjabaran tersebut menyebabkan semakin banyak dan semakin kompleksnya faktor - faktor syarat yang mempengaruhi nilai probabilitas, yang hampir mustahil untuk dianalisa satu persatu. Akibatnya,



perhitungan tersebut menjadi sulit untuk dilakukan. Di sinilah digunakan asumsi independensi yang sangat tinggi (naif), bahwa masing-masing petunjuk ( $F_1, F_2 \dots F_n$ ) saling bebas (independen) satu sama lain. Dengan asumsi tersebut, maka berlaku suatu kesamaan sebagai berikut:

$$P(F_i|F_j) = \frac{P(F_i \cap F_j)}{P(F_j)} = \frac{P(F_i)P(F_j)}{P(F_j)} = P(F_i) \quad (4)$$

Untuk  $i \neq j$ , sehingga

$$P(F_i|C, F_j) = P(F_i|C) \quad (5)$$

Persamaan di atas merupakan model dari teorema Naive Bayes yang selanjutnya akan digunakan dalam proses klasifikasi. Untuk klasifikasi dengan data kontinyu digunakan rumus Densitas Gauss :

$$P(X_i = x_i|Y = y_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}}} e^{-\frac{(x_i - \mu_{ij})^2}{2\sigma_{ij}^2}} \quad (6)$$

Di mana :

P : Peluang

Xi : Atribut ke i

xi : Nilai atribut ke i

Y : Kelas yang dicari

yi : Sub kelas Y yang dicari

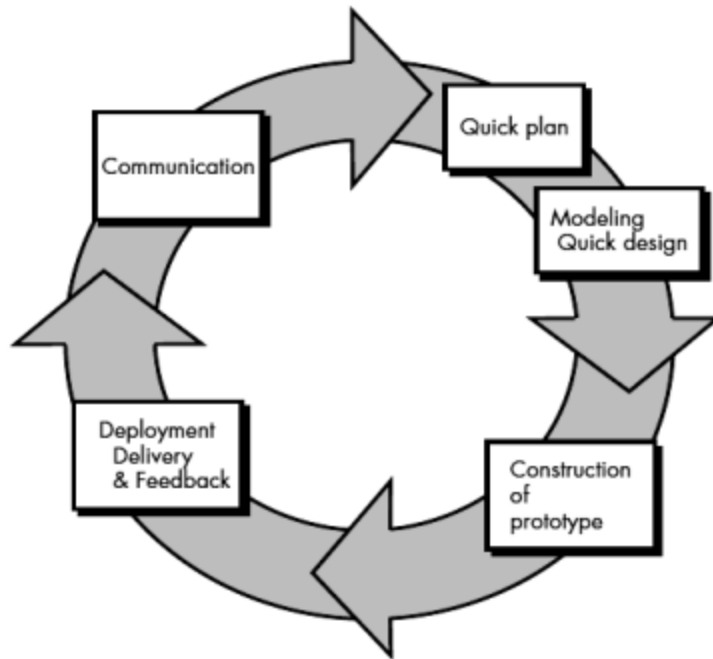
$\mu$  : mean, menyatakan rata – rata dari seluruh atribut

$\sigma$  :Deviasi standar, menyatakan varian dari seluruh atribut.

## **2.5 Metode Pengembangan Perangkat Lunak Menggunakan Metode *Prototype***

Metode penelitian yang digunakan adalah didasarkan pada metoda pengembangan perangkat lunak prototype yang memiliki lima tahap (Pressman, 2002). Alasan mengambil metode pengembangan sistem prototype dalam penelitian ini karena

pengguna dapat dengan mudah beradaptasi dengan sistem yang dikembangkan sesuai dengan kebutuhan pemakai. Selain itu pada saat tahap evaluasi sistem, jika terdapat revisi pada sistem yang dirancang, programmer tidak harus mengulang tahapan dari awal. Gambar 2.6 berikut ini merupakan model Prototype :



Gambar 2.6 Prototyping Model

(Sumber : Pressman, 2002)

Tahapan metoda prototype yang digunakan dalam penelitian ini adalah sebagai berikut :

1. Communication.

Penulis menganalisis sistem dengan mewawancarai pihak-pihak fasilitas dan pengguna sebagai pemakai untuk mendapatkan gagasan dari apa yang diinginkan pemakai terhadap sistem yang dibuat.

2. Quick Plan

Analisis sistem bekerja sama dengan spesialis informasi lain pihak, menggunakan satu atau lebih peralatan prototyping untuk mengembangkan sebuah prototype. (misalnya dengan menambahkan tampilan yang diinginkan pemakai dalam sistem baru seperti menu, interface, fitur, dan database ).

### 3. Modelling Quick Design

Analisis membuat perancangan sistem untuk mengembangkan prototipe dengan menggunakan UML, sebagai dasar perancangan aplikasi. Dalam hal ini programmer menggunakan PHP sebagai bahasa pemrograman dan MySQL sebagai database.

### 4. Construction Of Prototype

Tahapan ini adalah tahapan yang dilakukan setelah kegiatan analisi dan perancangan. Pada bagian ini akan dijelaskan kegiatan-kegiatan yang dilakukan pada tahap pengkodean (coding) sistem operasional, implementasi pembuatan program (programming) dan pengujian (testing).

### 5. Deployment delivery & feedback

Penulis yang menguji sistem baru tersebut dan melakukan uji coba terhadap beberapa calon pengguna sehingga penulis dapat menentukan apakah sistem baru dapat diterima. Pada tahap ini pemakai memberi masukan kepada analis apakah sistem dapat diterima. Jika ya sistem baru yang telah diuji dan terima oleh pengguna aplikasi, jika tidak, langkah 4 dan 5 diulangi.

## ***2.7 Unified Modelling Language (UML)***

UML yang merupakan singkatan dari Unified Modelling Language adalah sekumpulan pemodelan konvensi yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak dalam kaitannya dengan objek. (Rosa dan Shalahuddin, 2013).

UML dapat juga diartikan sebuah bahasa grafik standar yang digunakan untuk memodelkan perangkat lunak berbasis objek. UML pertama kali dikembangkan pada pertengahan tahun 1990an dengan kerjasama antara James Rumbaugh, Grady Booch dan Ivar Jacobson, yang masing-masing telah mengembangkan notasi mereka sendiri di awal tahun 1990an. (Rosa dan Shalahuddin, 2013).

### 2.7.1 Komponen-komponen UML


UML mendefinisikan diagram-diagram berikut ini: (Rosa dan Shalahuddin, 2013).




a. Use case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem.

Tabel 2.1 berikut ini merupakan simbol-simbol pada use case :

Tabel 2.1 Simbol *Use Case*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
3		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.



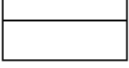
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi


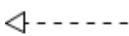
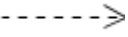

b. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan ( atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Tabel 2.2 berikut merupakan simbol class diagram :

Tabel 2.2. Simbol Class Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.

4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya




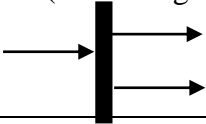
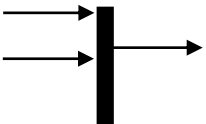
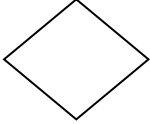
c. Activity Diagram

Activity diagram adalah state diagram khusus, yang sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (internal processing). Oleh karena itu, activity diagram tidak menggambarkan perilaku internal sebuah sistem dan interaksi antar subsistem, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beber apa eksekusi.

Tabel 2.3 berikut ini merupakan simbol pada activity diagram :

Tabel 2.3 Simbol Activity Diagram


No.	Simbol	Keterangan
1	<i>Start State</i> 	<i>Start state</i> adalah sebuah kondisi awal sebuah <i>object</i> sebelum ada perubahan keadaan. Start state digambarkan dengan sebuah lingkaran solid.
2.	<i>End State</i> 	<i>End state</i> adalah menggambarkan ketika objek berhenti memberi respon terhadap sebuah event. <i>End state</i> digambarkan dengan lingkaran solid di dalam sebuah lingkaran kosong.
3.	<i>State/Activities</i> 	<i>State</i> atau <i>activities</i> menggambarkan kondisi sebuah entitas, dan digambarkan dengan segiempat yang pinggirnya.
4.	<i>Fork (Percabangan)</i> 	<i>Fork</i> atau percabangan merupakan pemisalah beberapa aliran konkuren dari suatu aliran tunggal.
5.	<i>Join (Penggabungan)</i> 	<i>Join</i> atau penggabungan merupakan penggabungan beberapa aliran konkuren dalam aliran tunggal.
6.	<i>Decision</i> 	<i>Decision</i> merupakan suatu logika aliran konkuren yang mempunyai dua cabang aliran konkuren.






d. Sequence Diagram

Sequence diagram secara grafis menggambarkan bagaimana objek berinteraksi antara satu sama lain melalui pesan pada sebuah use case atau operasi.

Tabel 2.4 berikut merupakan simbol sequence diagram :

Tabel 2.4 Simbol Sequence Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>State</i>	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.

2		<i>Initial Pseudo State</i>	Bagaimana objek dibentuk atau diawali
3		<i>Final State</i>	Bagaimana objek dibentuk dan dihancurkan
4		<i>Transition</i>	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		<i>Node</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

## 2.8 Basis Data

*Database* adalah kumpulan file-file yang mempunyai kaitan antara satu file dengan file yang lain sehingga membentuk satu bangunan data untuk menginformasikan satu perusahaan, instansi dalam batasan tertentu. (Nugroho, 2011)

Istilah-istilah yang digunakan dalam basis data:

- 1) *File* : merupakan kumpulan dari atribut *record-record* sejenis yang mempunyai panjang elemen yang sama, atribut yang sama namun berbeda-beda dalam data *value*-nya.
- 2) *Record* : merupakan kumpulan dari elemen-elemen yang saling berhubungan atau berkaitan menginformasikan tentang *entry* secara lengkap.
- 3) *Field* : merupakan sekumpulan tanda-tanda yang berbentuk kesatuan tersendiri, merupakan bagian terkecil dari *record* dan bentuknya unik dijadikan *field* kunci yang dapat mewakili *record*-nya.
- 4) *Entity* : merupakan tempat kejadian atau konsep yang informasikan direkam.



## 2.9 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *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. MySQL adalah Relational Database Management System (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat closed source atau komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis (Nugroho,2011 No.Hal).

## 2.10 Penelitian Terdahulu

Tabel 2.5 berikut merupakan Tabel penelitian terdahulu :

Tabel 2.5 Penelitian Terdahulu

No	Peneliti	Judul /Tahun	Uraian
1	Ahmad Fathan Hidayatullah dan Azhari SN	Analisis Sentimen dan Klasifikasi Kategori Terhadap Tokoh Publik Pada Twitter /2014	Penelitian ini menganalisis tweet berbahasa Indonesia yang membicarakan tentang tokoh publik. Metode yang digunakan adalah Naive Bayes Classifier. Klasifikasi sentimen hanya positif dan negatif.  Sedangkan penelitian ini melakukan menganalisis sentimen untuk klasifikasi kategori komentar pada media sosial instagram, facebook, dan youtube dan klasifikasi sentimen ini adalah positif negatif dan netral.
2	Wina Witanti, Herry Rahmanto dan Faiza Renaldi	Pembangunan Sistem Temu Balik Informasi (Information Retrieval)	Penelitian ini menentukan 11 pemain berkualitas dengan menggunakan analisis sentimen dari 120 artikel serta 70 pemain

		Dalam Pemilihan Pemain Sepak Bola Berkualitas di Indonesia berbasis Analisis Sentimen /2016	<p>sepak bola menggunakan metode Naïve Bayes Classifier.</p> <p>Sedangkan penelitian ini Melakukan peneliti untuk menganalisis sentimen klasifikasi kategori komentar pada media sosial yaitu instagram, facebook, dan youtube .</p>
3	Andi Nurul Hidayat	Sentimen Analisis Terhadap Politik Pada Media Masa Online Menggunakan Algoritma Vector Machine dan Naive Bayes (2015)	<p>Penelitian ini menganalisis sentimen terhadap wacana politik media masa Indonesia berbahasa Inggris. Metode yang digunakan adalah Algoritma Vector Machine dan Naive Bayes.</p> <p>Sedangkan penelitian ini melakukan analisis sentimen untuk klasifikasi kategori komentar pada media sosial instagram, facebook, dan youtube.</p>
4	Anindya Apriliyanti P	Sentiment Analysis Dengan Naive Bayes Untuk Melihat Persepsi Masyarakat Terhadap Batik Pada Jejaring Sosial Twitter / 2015	<p>Penelitian ini dilakukan untuk mengklasifikasikan opini atau sentiment masyarakat terhadap batik. Metode yang digunakan adalah metode Naive Bayes, dengan kategori sentiment positif dan negatif</p> <p>Penelitian ini sama dengan penelitian yang dilakukan oleh peneliti. Variabel komentarnya adalah positif negatif, dan netral sedangkan media sosial yang digunakan adalah instagram, facebook, dan youtube.</p>
5	Linda Jayanti, Steven Ray Sentinuwo, Oktavian A. Lantang dan Agustinus Jacobus	Analisa Pola Penyalahgunaan Facebook Sebagai Alat Kejahatan Trafficking Menggunakan Data Mining / 2016	<p>Penelitian ini bertujuan mengidentifikasi data teks (status facebook) yang terindikasi sebagai proses kejahatan trafficking dengan memakai salah satu teknik klasifikasi dengan teorema Naïve Bayes Classifier.</p> <p>Penelitian yang dilakukan oleh peneliti adalah menganalisis sentimen untuk klasifikasi kategori komentar pada media sosial instagram, facebook, dan youtube.</p>