# BAB 2
# TINJAUAN PUSTAKA

## 2.1 Program Acara Televisi

Acara televisi atau program televisi merupakan acara-acara yang ditayangkan oleh stasiun televisi. Secara garis besar, program Televisi dibagi menjadi program berita dan program non-berita. Adapun jenis-jenis acara non-berita adalah sebagai berikut:

*1. Talkshow*

*Talkshow* atau perbincangan adalah program yang menampilakan satu atau beberapa orang untuk membahas topik tertentu yang di pandu oleh seorang pembawa acara (host). Mereka yang di undang adalah orang-orang yang berpengalaman langsung dengan satu peristiwa atau ahli dalam masalah yang tengah di bahas. Contoh acara talkshow di Indonesia seperti acara Ini talkshow pada stasiun .Net, MataNajwa di MetroTV.

*2. Quiz Show/ Game Show*

*Quiz Show* adalah bentuk permainan yang paling sederhana di mana sejumlah peserta saling bersaing untuk menjawab sejumlah pertanyaan. Quiz Show/Game Show adalah jenis program televisi di mana anggota masyarakat, pesohor atau selebriti, kadang-kadang sebagai bagian dari tim, memainkan permainan yang melibatkan menjawab pertanyaan atau memecahkan teka-teki biasanya untuk uang dan/atau hadiah. Pada beberapa menunjukkan kontestan bersaing dengan pemain lain atau tim lain, sementara lainnya menunjukkan melibatkan kontestan bermain sendirian untuk hasil yang baik atau nilai tinggi. Contoh acara Quiz show di Indonesia seperti Deal or No Deal di RCTI, 1 lawan 100 di Indosiar, 24 Hour Quiz show di MetroTV.

*3. Reality show*

*Relaity Show* adalah program yang menyajikan suatu situasi konflik, persaingan atau hubungan berdasarkan realitas sebenarnya. Program ini

menyajikan situasi sebagaimana apa adanya, keadaan yang nyata dengan cara sealamiah mungkin tanpa rekayasa. Tingkat realitas yang disajikan reality show bermacam-macam. Mulai dari yang betul-betul *reality hidden camera* hingga yang terlalu banyak rekayasa namun tetap menggunakan *reality show* dengan tujuan menaikkan penjualan. Contoh acara reality show di Indonesia seperti Super Trap di Trans 7, Master Chef Indonesia di RCTI, Jebakan Bataman di SCTV.

4. *Opera Show*

*Opera* merupakan drama yang menyajikan cerita dari berbagai tokoh secara bersamaan. Masing-masing tokoh memiliki alur cerita sendiri tanpa harus di rankum menjadi satu kesimpulan. Contoh acara Opera show di Indonesia seperti Opera Van Java.




**Gambar 2.1 Kompilasi Stasiun TV dengan Acara TV**

## 2.2 *Twitter*

*Twitter* adalah salah satu layanan *micro-blogging* popular yang memungkinkan penggunanya untuk membuat status (disebut *tweet*). Banyak pengguna twitter yang mengutarakan berbagai pendapat atau opini akan sebuah produk, layanan atau hal lainnya. Adapun keunggulan Twitter dibandingkan media sosial lainnya sebagai berikut:

1. Sederhana

   Tweet yang hanya memuat 140 karakter memudahkan dalam pengiriman melalui sms ataupun email.

2. Fleksibel

   Anda bisa menulis tweet melalui banyak program dan perangkat.

3. *Hot News*

   Tweet langsung diumumkan ke seluruh pemakai twitter.

4. Tanpa Batas

   Anda bisa melihat semua tweet orang tanpa harus menjadi teman terlebih dahulu.

5. Privasi

   Tweet menyediakan layanan komunikasi langsung ke sesame pengguna twitter tanpa terbaca pengguna lainnya. Ini cocok untuk komunikasi bersifat prifat.

6. *Trending Topic*

   *Trending topic* adalah topik-topik yang sedang ramai dibicarakan oleh tweeps, biasanya ditampilkan 10 list di sidebar kiri *timeline*. Trending topik bisa bermanfaat bagi *corporate branding* dalam target pemasaran mereka, bisa pula bermanfaat bagi politikus untuk melihat isu-isu aktual, dan bermanfaat untuk siapa saja dalam mengikuti trend global, contohnya rilis film terbaru. Dan masih banyak manfaat lainnya.

*7. Multi-link*

Tweet bisa dihubungkan dengan aplikasi lain seperti facebook atau wordpress. Hal ini membuat jaringan jauh lebih luas dibandingkan jaringan lain.



**Gambar 2.2 Twitter logo**

## 2.3 *Text Mining*

*Text mining* (penambangan teks) adalah penambangan yang dilakukan oleh komputer untuk mendapatkan sesuatu yang baru, sesuatu yang tidak diketahui sebelumnya atau menemukan kembali informasi yang tersirat secara implisit, yang berasal dari informasi yang diekstrak secara otomatis dari sumber-sumber data teks yang berbeda-beda [5]. Text mining merupakan teknik yang digunakan untuk menangani masalah klasifikasi, *clustering, information extraction dan information retrival* [6].

Pada dasarnya proses kerja dari text mining banyak mengapdopsi dari penelitian *Data Mining* namun yang menjadi perbedaan adalah pola yang digunakan oleh *text mining* diambil dari sekumpulan bahasa alami yang tidak terstruktur sedangkan dalam *data mining* pola yang diambil dari database yang terstruktur [7]. Adapun Kegunaan Text mining adalah salah satu diantaranya adalah analisis sentimen.

## 2.4 Analisis Sentimen

Definisi opini menurut kamus besar bahasa indonesia sangat sederhana, yaitu pendapat, pikiran, atau pendirian. Sedangkan definisi sentimen adalah opini yang didalamnya mengandung perasaan atau emosi. Contoh opini adalah *"Menurut saya, besok akan turun hujan"*, sedangkan contoh sentimen adalah '*Saya menyukai kamera Nikon D3100'.*

Analisis Sentimen atau *opinion mining* mengacu pada bidang yang luas dari pengolahan bahasa alami, komputasi linguistik dan text mining yang bertujuan menganalisa pendapat, sentimen, evaluasi, sikap, penilaian dan emosi seseorang apakah pembicara atau penulis berkenaan dengan suatu topik, produk, layanan, organisasi, individu, ataupun kegiatan tertentu [2].

Tugas dasar dalam analisis sentimen adalah mengelompokkan teks yang ada dalam sebuah kalimat atau dokumen kemudian menentukan pendapat yang dikemukakan dalam kalimat atau dokumen tersebut apakah bersifat positif, negatif atau netral. Analisis Sentimen juga dapat menyatakan perasaan emosional sedih, gembira, atau marah.

Tahap-tahap dari analisis sentimen sebagai berikut:

1. *Preprocessing*
2. Ekstraksi Fitur
3. Klasifikasi

### 2.4.1 *Preprocessing*

Preprocessing merupakan proses menggali, mengolah, mengatur informasi dengan cara menganalisis hubungannya, aturan-aturan yang ada di data tekstual semi terstruktur atau tidak terstruktur. Untuk lebih efektif dalam proses pemrosesan dilakukan langkah transformasi data ke dalam suatu format yang memudahkan untuk kebutuhan pemakai. Proses ini disebut *preprocessing* dokumen. Setelah dalam bentuk yang lebih terstruktur dengan adanya proses

diatas data dapat dijadikan sumber data yang dapat diolah lebih lanjut. Sama halnya *preprocessing* pada *Information Retrieval (IR)*, tahapannya terdiri dari, *case folding*, *cleansing*, *tokenizing*, *stopword removal* dan *stemming*. Namun pada *preprocessing* analisis sentimen, ada tambahan tahapan seperti *convert emoticon* dan *convert negation.*

1. *Convert Emoticon*

*Convert Emoticon* merupakan salah satu cara pengungkapan ekspresi perasaan secara tekstual. Hal tersebut akan membantu dalam menentukan sentimen dalam suatu kalimat atau tweet sekalipun. Setiap *emoticon* akan dikonversikan ke dalam *string* yang bersesuaian. Jenis *emoticon* yang akan diproses oleh adalah *emoticon western style*, dikarenakan jenis tersebut banyak digunakan atau menjadi standar di semua platform seperti *web* ataupun *mobile*. Pada Tabel 2.1 dijelaskan arti dari *emoticon western style* secara umum.

**Tabel 2.1 Daftar Emoticon yang mau dikonversi**

| Emoticon | Deskripsi |
|---|---|
| :-) :) :o) :] :3 :c) :> =] 8) =) :} :^) :⁀) | Senang |
| :-D :D 8-D 8D x-D xD X-D XD =-D =D =-3 =3 B^D | Ketawa |
| :-‖ :@ >:( | Benci |
| :$ >:[ :-( :( :-c :c :-< :⁀C :< :-[ :[ :{ ;( :'-( :'(D:< D: D8 D; D= DX v.v D-': | Kecewa |
| <3 ;-) ;) *-) *) ;-] ;] ;D ;^) :-, | Suka |

Untuk lebih jelasnya pada Tabel 2.2 dijelaskan contoh penerapan Convert Emoticon.

**Tabel 2.2 Contoh Penerapan Convert Emoticon**

| Tweet | |
|---|---|
| Sebelum | Sesudah |
| Pindah dulu ke @Ini_Talkshow cuyyyy bentar lg ada Haruka :3 #MinFre | Pindah dulu ke @Ini_Talkshow cuyyyy bentar lg ada Haruka senang #MinFre |

2. *Cleansing*

*Cleansing* merupakan proses membersihkan kata-kata yang tidak diperlukan untuk mengurangi noise. Kata yang dihilangkan adalah URL, *hashtag (#)*,

*username* (@username), dan email. Selain itu juga tanda baca seperti titik(.), koma(,), dan tanda baca yang lainnya akan dihilangkan. Pada Tabel 2.3 dijelaskan contoh penerapan dari *cleansing*:

**Tabel 2.3 Contoh Penerapan Cleansing**

| Tweet | |
|---|---|
| Sebelum | Sesudah |
| Rasanya ini acara terbaik yang perna tak tonton so far @MataNajwa #5TahunMataNajwa | Rasanya ini acara terbaik yang perna tak tonton so far |

*3. Case Folding*

*Case folding* merupakan tahapan merubah bentuk kata-kata menjadi sama bentuknya, semuanya menjadi *lower case*. Pada Tabel 2.4 dijelaskan contoh dari penerapan proses *case folding*.

**Tabel 2.4 Contoh Penerapan Case Folding**

| Tweet | |
|---|---|
| Sebelum | Sesudah |
| Rasanya ini acara terbaik yang perna tak tonton so far | rasanya ini acara terbaik yang perna tak tonton so far |

*4. Convert Negation*

Seperti halnya ilmu matematika, dalam bahasa terdapat kata yang dapat membalikan arti dari kata tersebut atau bersifat negasi. Kata-kata yang bersifat negasi adalah “kurang”, “tidak”, “enggak”, “ga”, “nggak”, “tak”, dan “gak”. Pada Tabel 2.5 dijelaskan contoh dari penerapan proses *Convert Negation*.

**Tabel 2.5 Contoh Penerapan Convert Negation**

| Tweet | |
|---|---|
| Sebelum | Sesudah |
| gak sabar nunggu acara ini talk show mulainya apalagi klo ada | gak_sabar nunggu acara ini talk show mulainya apalagi klo ada |

*5. Tokenizing*

*Tokenizing* bekerja untuk mengidentifikasi kata-kata dalam teks menjadi beberapa urutan yang terpotong oleh spasi atau karakter spesial. Berikut pada Tabel 2.6 contoh penerapan dari *tokenizing*.

**Tabel 2.6 Contoh Penerapan Tokenizing**

| Tweet | |
|---|---|
| Sebelum | Sesudah |
| gak_sabar nunggu acara ini talk show mulainya apalagi klo ada | String 1 : gak_sabar<br>String 2 : nunggu<br>String 3 : acara<br>String 4 : ini<br>String 5 : talkshow<br>String 6 : mulainya<br>String 7 : apalagi<br>String 8 : klo<br>String 9 : ada |

*6. Stopping*

*Stopping* berperan untuk membuang kata-kata yang sering muncul dan bersifat umum, kurang menunjukan relevansinya dengan teks. Kata-kata yang akan dibuang tersebut didefinisikan dalam *stopword list.* Contoh beberapa kata yang sering masuk ke dalam *stopword list* adalah “sebuah”, “yang”, dan “itu”. Berikut Tabel 2.7 adalah beberapa daftar *stoplist* yang disimpan dalam *database*.

**Tabel 2.7 Stoplist**

| | | | | |
|---|---|---|---|---|
| Masih | Dong | ke | Ada | Yoi |
| Malam | Ya | loe | Pada | Yang |
| Ini | Dan | juga | Kita | Saya |
| untuk | Dari | bagi | Iya | di mana |
| kapan | Bisa | mana | Itu | Sih |
| sudah | Tak | dengan | Anda | Begitu |
| entah | Lalu | yuk | Aku | Adalah |
| gue | Nanti | tunggu | Tau | Kemarin |

Berikut pada Tabel 2.8 contoh penerapan dari Stopping.

**Tabel 2.8 Contoh Penerapan Stopping**

| Tweet | |
|---|---|
| Sebelum | Sesudah |
| rasanya ini acara terbaik yang perna tak tonton so far | rasanya acara terbaik perna tonton so far |

*7. Stemming*

*Stemming* adalah tahapan untuk membuat kata yang berimbuhan kembali ke bentuk asalnya. Contohnya kata "menginspirasi" setelah melewati tahap ini maka akan menjadi "inspirasi". Pada Tabel 2.9 dijelaskan contoh dari penerapan proses *stemming*.

**Tabel 2.9 Contoh Penerapan Stemming**

| Tweet | |
|---|---|
| Sebelum | Sesudah |
| rasanya acara terbaik perna tonton so far | rasanya acara baik perna tonton so far |

Ada banyak algoritma yang digunakan untuk melakukan proses stemming, khususnya stemming bahasa indonesia. Seperti Algoritma Nazief dan Andriani, algoritma Arifin dan Setiono, algoritma Vega, Algoritma Tala, Algoritma Porter, Algoritma Porter[17].

### 2.4.2 Ekstraksi Fitur (*Feature Extraction*)

Berdasarkan tujuan dan tugas pokok, analisis sentimen bisa dibagi menjadi 5 bagian.

1. Ekstraksi dan pengelompokan entitas

Mengekstraksi semua ekspresi entitas dari dokumen dan pengelompokan entitas tersebut. Karena memungkinkan untuk satu entitas ada dua penulisan berbeda.

2. Ekstraksi dan pengelompokan aspek

Aspek kategori merepresentasikan aspek yang unik dari entitas. Sementara ekspresi aspek kata atau frasa yang muncul mengindentifikasikan kategori aspek. Ekspresi aspek bisa berbentuk kata benda, kata kerja, kata sifat, atau keterangan.

3. Ekstraksi dan pengelompokan pemilik opini

Mengekstraksi pemilik opini dari data yang terstruktur kemudian mengkatagorisasikan. Dapat dianalogikan dengan dua tugas diatas.

4. Ekstraksi dan standarisasi waktu

Mengekstraksi waktu ketika opini dikeluarkan dan melakukan standarisasi format waktu yang berbeda.

5. Klasifikasi aspek sentimen

Menentukan apakah sebuah opini apakah termasuk opini negatif ataukah opini positif.

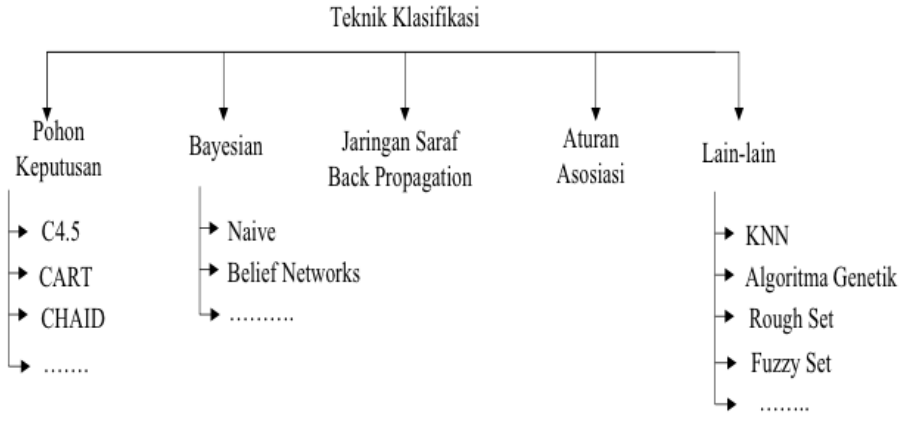
Namun yang akan digunakan dalam penelitian ini adalah nomer 2 yaitu ekstraksi dan pengelompokkan aspek.

### 2.4.3 Klasifikasi

Klasifikasi adalah proses pencarian sekumpulan model atau fungsi yang menggambarkan dan membedakan kelas data. Tujuan dari klasifikasi adalah untuk memprediksi kelas dari suatu obyek yang belum diketahui kelasnya.

Klasifikasi memiliki dua proses yaitu membangun model klasifikasi dari sekumpulan kelas data yang sudah didefinisikan sebelumnya (*training data set*) dan menggunakan model tersebut untuk klasifikasi tes data *(prediction)* serta mengukur akurasi dari model.

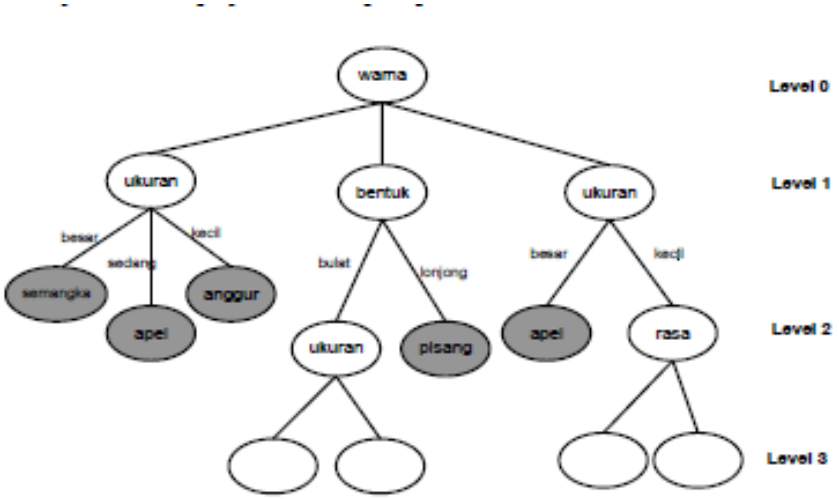
Klasifikasi dapat dimanfaatkan dalam berbagai aplikasi seperti diagnosa medis, *selective marketing*, pengajuan kredit perbankan, *email* dan analisis sentimen. Klasifikasi dapat disajikan dalam berbagai macam model klasifikasi seperti *decision trees, naïve bayes classifier, k-nearest-neighbourhood classifier, neural network* dan lain-lain.




**Gambar 2.3 Teknik Klasifikasi [13]**

### *2.4.3.1 Decision Trees* (Pohon Keputusan)

Dalam decision tree tidak menggunakan vector jarak untuk mengklasifikasikan obyek. Seringkali data observasi mempunyai atribut-atribut yang bernilai nominal. Seperti yang diilustrasikan pada Gambar 2.1, misalkan obyeknya adalah sekumpulan buah-buahan yang bisa dibedakan berdasarkan atribut bentuk, warna, ukuran dan rasa. Bentuk, warna, ukuran dan rasa adalah besaran nominal, yaitu bersifat kategoris dan tiap nilai tidak bisa dijumlahkan atau dikurangkan. Dalam atribut warna ada beberapa nilai yang mungkin yaitu hijau, kuning, merah. Dalam atribut ukuran ada nilai besar, sedang dan kecil. Dengan nilai-nilai atribut ini, kemudian dibuat decision tree untuk menentukan suatu obyek termasuk jenis buah apa jika nilai tiap-tiap atribut diberikan.




**Gambar 2.4 Contoh Pohon Keputusan**

*Decision tree* sesuai digunakan untuk kasus-kasus yang keluarannya bernilai diskrit. Walaupun banyak variasi model decision tree dengan tingkat kemampuan dan syarat yang berbeda, pada umumnya beberapa ciri yang cocok untuk diterapkannya decision tree adalah sebagai berikut :

1. Data dinyatakan dengan pasangan atribut dan nilainya,
2. Label/keluaran data biasanya bernilai diskrit,
3. Data mempunyai missing value (nilai dari suatu atribut tidak diketahui).

Dengan cara ini akan mudah mengelompokkan obyek ke dalam beberapa kelompok. Untuk membuat decision tree perlu memperhatikan hal-hal berikut ini:

1. Atribut mana yang akan dipilih untuk pemisahan obyek,
2. Urutan atribut mana yang akan dipilih terlebih dahulu,
3. Struktur tree,
4. Kriteria pemberhentian ,
5. Pruning [8].

Menurut Larose [9] banyak algotima yang dapat dipakai dalam pembentukan pohon keputusan antara lain CART, ID3, dan C4.5. Tetapi yang dijelaskan adalah algoritma C4.5 saja.

#### 2.4.3.1.1 Algoritma C4.5

Algoritma C4.5 adalah algoritma untuk membuat *decision tree* berdasarkan training data yang telah disediakan. Cabang-cabang pohon keputusan merupakan pertanyaan klasifikasi dan daun-daunnya merupakan kelas-kelasnya.

Algoritma C4.5 merupakan salah satu algoritma machine learning. Dengan algoritma ini, mesin (komputer) akan diberikan sekelompok data untuk dipelajari yang disebut learning dataset. Kemudian hasil dari pembelajaran selanjutnya akan digunakan untuk mengolah data-data yang baru yang disebut test dataset. Karena algoritma C4.5 digunakan untuk melakukan klasifikasi, jadi hasil dari pengolahan test dataset berupa pengelompokkan data ke dalam kelas-kelasnya [13].

Algoritma C4.5 adalah algoritma hasil pengembangan dari algoritma ID3. Perbaikan algoritma C4.5 dari algoritma ID3 dilakukan dalam hal berikut [13]:

1. Bisa mengatasi missing value,
2. Bisa mengatasi data kontinyu,
3. Pruning,
4. Adanya aturan.

Secara umum Algoritma C4.5 membangun pohon keputusan:

a. Pilih atribut sebagai root.
b. Buat cabang untuk masing-masing nilai.
c. Bagi kasus dalam cabang.
d. Ulangi proses untuk masing-masing cabang sampai semua kasus pada cabang memiliki kelas yang sama.

Untuk memilih atribut sebagai akar, didasarkan pada nilai gain tertinggi dari atribut-atribut yang ada.

$$\text{Gain (S,A)} = \text{Entropy (S)} - \sum_{i=1}^{n} \left( \left| \frac{\text{Si}}{S} \right| \right) * \text{Entropy (Si)} \qquad (2.1)$$

Dengan:

S : Himpunan kasus

A : Atribut

n : Jumlah partisi Atribut A

|Si| : Jumlah kasus pada Partisi ke i

|S| : Jumlah kasus dalam S

Semakin kecil nilai Entropy maka semakin baik untuk digunakan dalam mengekstraksi suatu kelas.

$$\text{Entropy (S)} - \sum_{i=1}^{n} - pi * log2\ pi \qquad (2.2)$$

Dengan:

S : Himpunan kasus

n : Jumlah partisi S

p : Proporsi dari Si terhadap S

Pada tahap pembelajaran algoritma C4.5 memiliki dua prinsip kerja yaitu:

1. Pembuatan pohon keputusan.

Tujuan dari algoritma penginduksi pohon keputusan adalah mengkontruksi struktur data pohon yang dapat digunakan untuk memprediksi kelas dari sebuah kasus atau record baru yang belum memiliki kelas. C4.5 melakukan konstruksi pohon keputusan dengan metode divide and conquer. Pada awalnya hanya dibuat node akar dengan menerapkan algoritma divide and conquer. Algoritma ini memilih pemecahan kasus-kasus yang terbaik dengan menghitung dan membandingkan gain ratio, kemudian node-node yang terbentuk di level berikutnya, algoritma divide and conquer akan diterapkan lagi sampai terbentuk daun-daun.

2. Pembuatan aturan-aturan (rule set).

Aturan-aturan yang terbentuk dari pohon keputusan akan membentuk suatu kondisi dalam bentuk if-then. Aturan-aturan ini didapat dengan cara menelusuri pohon keputusan dari akar sampai daun. Setiap node dan syarat percabangan akan membentuk suatu kondisi atau suatu if, sedangkan untuk nilai-nilai yang terdapat pada daun akan membentuk suatu hasil atau suatu then.

## 2.5 *Confussion Matrix*

Confusion Matrix berisi informasi tentang klasifikasi aktual dan yang telah diprediksi yang dilakukan oleh sistem klasifikasi. Kinerja sistem tersebut umumnya dievaluasi dengan menggunakan data dalam matriks. Tabel berikut menunjukkan confusion matrix untuk klasifikasi dua kelas.

**Table 2.1 Confussion Matrix**

| Confussion Matrix | | Predicted Class | |
|---|---|---|---|
| | | Positive | Negative |
| Actual Class | Positive | True Positives | False Positives |
| | Negative | False Negatives | Trus Negatives |

True positives adalah jumlah record positif yang diklasifikasikan sebagai positif, false positives adalah jumlah record positif yang diklasifikasikan sebagai negatif, false negatives adalah jumlah record negatif yang diklasifikasikan sebagai positif, true negatives adalah jumlah record negatif yang diklasifikasikan sebagai negatif. Setiap kolom dari confusion matrix merupakan contoh di kelas yang telah diprediksi, sedangkan setiap baris mewakili contoh di kelas yang sebenarnya. Setelah didapat true positives, false positives, true negatives dan false negatives, selanjutnya hitung nilai precision dan akurasinya. *Precision dan recall* adalah ukuran terhadap suatu kelas yang telah diprediksi. Berikut persamaan dari precision dan akurasi [14].

$$Akurasi = \frac{TP+TN}{TP+FP+TN+FN} \quad (2.1)$$

$$Precision = \frac{TP}{TP+F} \quad (2.2)$$

$$Recall = \frac{TN}{TN+FN} \quad (2.3)$$

Keterangan:

TP = True Positives

TN = True Negatives

FP = False Positives

FN = False Negatives

## 2.6 Pemograman Berorientasi Objek (OOP)

Pemrograman berorientasi objek adalah sebuah pendekatan untuk pembangunan perangkat lunak di mana struktur dari perangkat lunak didasari dari objek yang saling berinteraksi satu sama lain untuk menyelesaikan suatu tugas. Sudut pandang dari pemrograman berorientasi objek menyelesaikan masalah dengan mengunakan objek, persis dengan apa yang ada di dunia nyata. Pemrograman berorientasi objek (PBO) dirancang untuk menutup kelemahan dari pemrograman berorientasi prosedural yang sulit untuk dikelola dan *debug* ketika kode yang dibuat telah mencapai beribu-ribu baris. Maka diperkenalkanlah struktur pemrograman untuk memecah kode ke dalam beberapa segmen atau biasa disebut prosedur atau fungsi. Dengan struktur tersebut kode mengalami perbaikantetapi secara program dapat melakukan fungsionalitas yang lebih kompleks.

Berikut adalah konsep mendasar mengenai bahasa permrograman berorientasi objek.

1. Kelas

Kelas adalah struktur yang akan digunakan oleh objek yang berisikan atribut-atribut yang mendefiniskan data yang akan diolah dan juga berisikan *method* yang dapat berbentuk prosedur ataupun fungsi untuk dapat mengolah data. Dalam konteks PBO biasanya didasari dari pemodelan dari objek yang ada di dunia nyata dalam pemrograman untuk menyelesaikan masalah. Sebagai contoh, dilakukan pemodelan untuk sepeda untuk dijadikan sebuah kelas. Dalam sebuah sepeda misalkan terdiri dari roda, rangka, dan setir. Ketiga unsur tersebut bisa disebut sebagai atribut. Kemampuan dari sepeda itu bisa berjalan dan melompat. Kemampuan tersebut bisa disebut *method* yang ada di dalam kelas.

2. Objek

Objek adalah instansiasi dari sebuah *class*. Kemampuan yang bisa dilakukan oleh suatu objek bergantung apa dari isi yang didefinisikan dari *class* tersebut.

3. Abstraksi

Ketika membuat objek dalam aplikasi PBO, penting untuk menggunakan konsep dari abstraksi. Objek hanya berisi informasi yang relevan dengan konteks dari aplikasi. Contoh untuk pembangunan aplikasi pengiriman. Akan dibuat sebuah objek dengan nama produk dengan atribut ukuran dan berat. Warna akan menjadi informasi yang tidak berhubungan dan bisa diabaikan sebagi atribut dari objek produk. Di kasus lain seperti pembangunan aplikasi pencatatan pemesanan, warna menjadi informasi yang berhubungan dan bisa dijadikan atribut dari suatu objek.

4. Enkapsulasi

Enkapsulasi adalah sebuat perses yang tidak mengizinkan akses langsung kepada data. Jika ingin mengakases dari data tersebut, diharuskan menggunakan objek yang memiliki tanggung jawab untuk terhadap data tersebut. Menggunakan contoh sebelumnya, ketika akan menggunakan *method* berjalan tidak bisa dilakukan secara langsung. Kelas sepeda tersebut harus diinstansiasi terlebih dahulu menjadi objek sepeda sehingga bisa mengakses *method* berjalan.

*5. Polymorphism*

*Polymorphism* adalah kemempuan untuk merespon pesan yang sama dengan cara yang berbeda. Misalkan ada dua objek yaitu sepeda dan motor. Dalam objek tersebut memiliki kemampuan method yang sama yaitu berjalan. Ketika *method* berjalan dari kedua objek tersebut dipanggil. Kedua objek tersebut menampilkan rekasi yang berbeda, sepeda berjalan dengan cara mengayuh pedal, sedangkan motor berjalan dengan cara menarik gas.

*6. Inheritance*

*Inheritance* dalam PBO untuk mengklasifikasikan objek di dalam program berdasarkan karakteristik umum. Dengan *inheritance* pemrograman lebih mudah karena bisa mengkombinasikan karakteristik umum ke dalam kelas induk kemudian bisa diturunkan ke kelas anaknya. Contoh dibuat kelas kendaraan yang

berisikan *method* berjalan. Kemudian dibuat kelas kendaraan darat. Dalam kelas kendaraan darat tidak perlu membuat *method* berjalan di kelas kendaraan darat cukup dengan melakukan *inheritance* atau pewarisan *method* yang ada di kelas kendaraan ke kelas kendaraan darat.

## 2.7 *Unified Modelling Language (UML)*

*Unified modeling language* (UML) adalah bahasa pemodelan standar untuk pembangunan perangkat lunak ataupun sistem yang dibangun dengan pendekatan *object-oriented.* Dalam merancang sistem, model memiliki peranan penting yaitu untuk mengelola kompleksitas. Pemodelan membantu untuk fokus menghubungkan aspek penting dari perancangan sistem. Sebuah model adalah sebuah abstraksi dari benda nyata. Model yang dibangun adalah penyederhanaan dari sistem yang dibangun. Memberikan maksud dan kelangsungan hidup sistem untuk dipahami, dievaluasi, dikritisasi bahkan lebih cepat dibandingkan harus langsung ke sistem sebenarnya.

UML itu sendiri sudah mengalami beberapa generasi, dimulai dari UML 1.0, 1.5, dan yang terakhir 2.0. Versi pertama dari UML memberikan komuniasi rancangan secara jelas, menyampaikan esensi dari suat perancangan, bahkan menangkap peta kebutuhan fungsional dari solusi perangkat lunak. Bagaimanapun, dunia berubah secara fundamental dengan pengenalan pemodelan sistem, dibanding hanya pemodelan perangkat lunak. Dari perubahan itulah yang membuat versi UML itu sendiri berkembang hingga kini 2.0. Secara keseluruhan ada 13 diagram yang dapat dimanfaatkan dari UML 2.0 ini.




**Gambar 2.5 UML Logo**

Namun dalam penelitian ini hanya akan diterapkan empat diagram yaitu *usecase diagram, activity diagram, class diagram* dan *sequence diagram*[15].

Berikut penjelasan masing-masing diagram.

### 2.7.1 Use Case Diagram

Use case biasa digunakan untuk memodelkan suatu fungsi yang disediakan oleh sistem, didalam use case juga memiliki batasan didalam sistem dengan diberi suatu label nama use case tersebut.

Use case juga memiliki karakteristik antara lain :

a. Use case merupakan interaksi antara sistem dengan actor, yaitu tindakan yang akan dilakukan sistem.
b. Use case itu terdapat actor yang dimana juga bisa melibatkan actor lain didalam sistem.
c. Use case mendefinisikan tindakan yang akan dilakukan didalam sistem.
d. Use case class tentunya memiliki objek use case yang dikenal dengan skenario.

Simbol-simbol yang ada di dalam use case diagram, antara lain:

1. Actor

Merupakan seseorang atau pelaku yang akan menggunakan suatu sistem. Sebuah actor memberikan inputan informasi ke sistem, menerima informasi, dan sebaliknya. Actor hanya berinteraksi dengan use case. Actor biasanya disimbolkan dengan Stick Man.

2. Use case

Merupakan fungsionalitas dari suatu sistem, sehingga user mengerti kegunaan dari sistem tersebut di kembangkan atau dibangun nantinya seperi apa. cara menentukan use case dalam suatu sistem adalah :

a. lihat pola perilaku dari perangkat lunak aplikasi
b. lihat gambaran dari tugas sebuah actor
c. lihat apa yang akan dikerjakan oleh suatu perangkat lunak, bukan melihat bagaimana cara mengerjakannya.

3. Relasi

Merupakan hubungan antara actor dengan use case didalam sistem. Ada beberapa relasi yang terdapat didalam use case diagram:

a. *Association,* berfungsi menghubungkan antar elemen.
b. *Generalization,* merupakan inheritance, dimana suatu elemen dapat juga merupakan spesialisasi dari elemen lainnya, contoh : staf akuntan dan staf kreatif digeneralisasi menjadi member staf.
c. *Dependency,* merupakan elemen yang ketergantungan dengan elemen lainnya
d. *Aggregation,* merupakan tipe gabungan yang menunjukkan bahwa elemen berisi atau terdiri dari elemen-elemen lainnya. contoh : mobil terdiri dari ban, setir, dan lain-lain.

Berikut ini akan dijelaskan tipe-tipe relasi yang umum yang ada didalam use case diagram :

a. *Include*, merupakan aktivitas yang harus dipenuhi untuk terjadi sebuah event, dimana kondisi ini sebuah use case adalah bagian dari use case lainnya.
b. *Extends*, merupakan aktivitas yang hanya berjalan dibawah kondisi tertentu.

### 2.7.2 *Activity* Diagram

Diagram ini memperlihatkan aliaran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek.

### 2.7.3 Class Diagram

Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi dan relasi-relasi antar objek.

### 2.7.4 Sequence Diagram

Diagram ini memperlihatkan interaksi yang menekankan pada pengiriman pesan *(message)* dalam suatu waktu tertentu.

## 2.8 *Web Browser*

*Web Browser* adalah suatu program atau software yang digunakan untuk menjelajahi internet atau untuk mencari informasi dari suatu web yang tersimpan didalam komputer. Awalnya, *web browser* berorientasi pada teks dan belum dapat menampilkan gambar. Namun, web browser sekarang tidak hanya menampilkan gambar dan teks saja, tetapi juga memutar *file* multimedia seperti video dan suara. *Web browser* juga dapat mengirim dan menerima email, mengelola HTML, sebagai masukan dan menjadikan halaman web sebagai hasil keluaran yang informatif.

Dengan menggunakan web browser, para pengguna internet dapat mengakses berbagai informasi yang terdapat di internet dengan mudah. Beberapa contoh web browser diantaranya Internet Explorer, Mozilla, Firefox, Safari, Opera, dan sebagainya.

## 2.9 *Web Server*

*Server web* atau pelayan web dapat merujuk baik pada perangkat keras ataupun perangkat lunak yang menyediakan layanan akses kepada pengguna melalui protokol komunikasi HTTP atau HTTPS atas berkas-berkas yang terdapat pada suatu situs web dalam layanan ke pengguna dengan menggunakan aplikasi tertentu seperti peramban web.

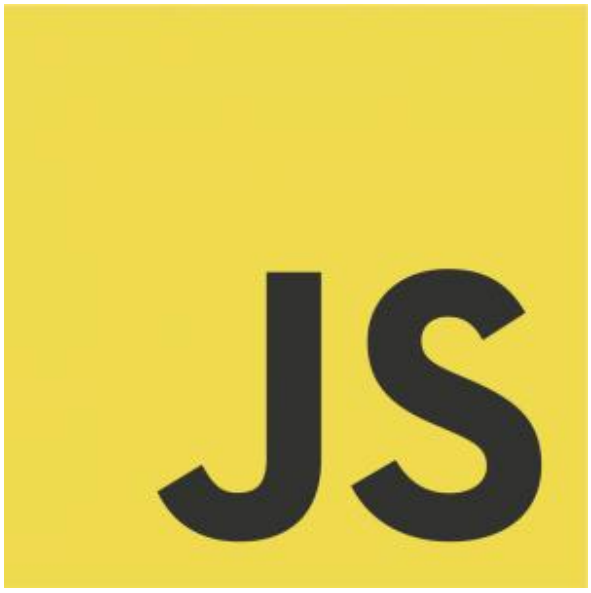
Penggunaan paling umum server web adalah untuk menempatkan situs web, namun pada prakteknya penggunaannya diperluas sebagai tempat peyimpanan data ataupun untuk menjalankan sejumlah aplikasi kelas bisnis.

## 2.10 *JavaScript*

*JavaScript* adalah bahasa pemrograman berbasis *prototype* yang berjalan disisi klien. Jika kita berbicara dalam konteks web, sederhananya, kita dapat memahami JavaScript sebagai bahasa pemrograman yang berjalan khusus untuk *dibrowser* atau halaman web agar halaman web menjadi lebih hidup. Kalau dilihat dari suku katanya terdiri dari dua suku kata, yaitu Java dan Script. Java adalah Bahasa pemrograman berorientasi objek, sedangkan *Script* adalah serangkaian instruksi program[16].

*Javascript* merupakan bahasa *scripting* yang digunakan untuk membuat aplikasi web, sifatnya *client-side* sehingga dapat diolah langsung di *browser* tanpa harus terhubung ke server terlebih dahulu. Walaupun namanya menggunakan kata "Java", Javascript tidak berhubungan dengan bahasa pemprograman java, meskipun keduanya memiliki kemiripan dalam hal syntax yang meniru bahasa C. Nama Asli dari bahasa ini adalah *LiveScript* yang kemudian diganti karena adanya perjanjian kerjasama antara *Netscape* dan *Sun* dengan balasan *Netscape* memperbolehkan untuk membundel *browse* mereka dengan menggunakan *Java* dan *Sun*. "JavaScript" merupakan merk terdaftar milik *Sun Microsystem*, Inc dan dilisensikan oleh *Sun* untuk *Netscape Comunications* dan entitas lainnya seperti *Mozilla Foundation*.

Kegunaan utama *JavaScript* adalah untuk menuliskan fungsi yang disisipkan kedalam HTML baik secara langsung disisipkan maupun diletakan ke file teks dan di link dari dokumen HTML. Secara fungsional, JavaScript digunakan untuk menyediakan akses script pada objek yang dibenamkan (*embedded*). Contoh sederhana dari penggunaan JavaScript adalah membuka halaman pop up, fungsi validasi pada form sebelum data dikirimkan ke server, merubah image kursor ketika melewati objek tertentu, dan lain lain.




**Gambar 2.6 Official Logo JavaScript**

## 2.11 MongoDB

MongoDB adalah sebuah Database berorientasi dokumen. MongoDB merupakan salah satu database noSQL. Penyimpanan data tanpa perlu adanya tabel schema, tidak ada bahasa sql yang terlibat dalam pemakaian database.

MongoDB tidak mengenal yang namanya tabel, kolom dan baris jadi tidak ada schema dalam MongoDB (*schema-less*). Unit paling kecil dari MongoDB adalah *document*, sedangkan kumpulan dari document adalah *collection*. seperti halnya dalam database relasional document itu ibarat *record* dan *collection* sebuah tabel. *Document* dalam MongoDB dapat memiliki atribut yang berbeda-beda dengan document yang lainnya walaupun dalam satu collection.




**Gambar 2.7 Logo MongoDB**

## 2.12 TwitterAPI

API (*Application Programming Interface*) adalah sekumpulan perintah, fungsi, komponen, dan protokol yang disediakan oleh sistem operasi ataupun bahasa pemrograman tertentu yang dapat digunakan oleh programmer saat membangun perangkat lunak.

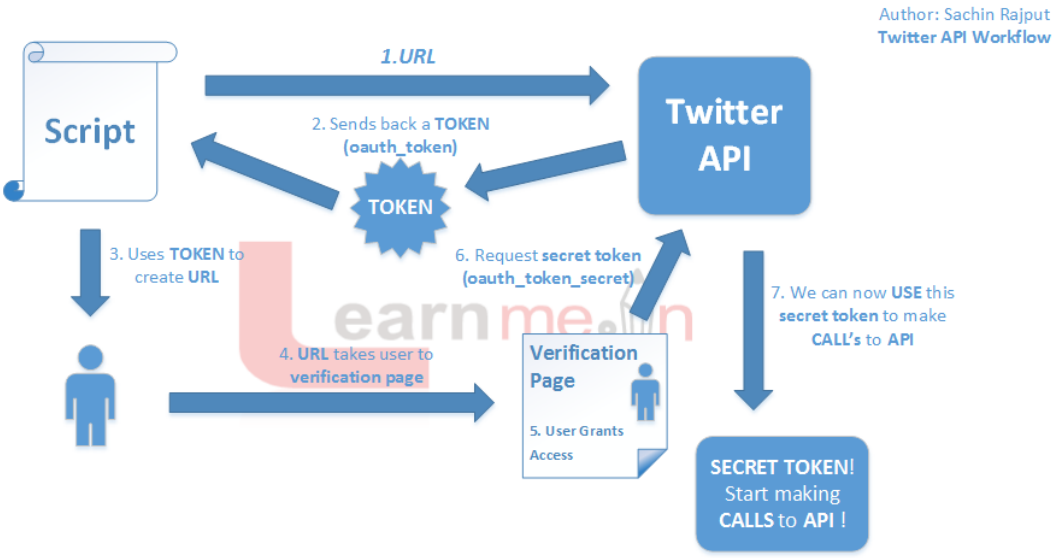
Dalam API terdapat fungsi-fungsi atau perintah-perintah untuk menggantikan bahasa yang digunakan dalam system calls dengan bahasa yang lebih terstruktur dan mudah dimengerti oleh programmer.

OAuth adalah standarisasi dan kebijaksanaan gabungan dari banyak protokol industri mapan. Hal ini mirip dengan protokol lain yang sedang digunakan (Google AuthSub, aol OpenAuth, Yahoo BBAuth, api Mendatang, Flickr api, Amazon Web Services api, dll). Setiap protokol menyediakan metode eksklusif untuk bertukar kredensial pengguna untuk token akses atau ticker.

Berikut ini adalah cara OAuth bekerja:

1. Script mengirimkan URL ke API,
2. API mengirimkan tanda (oauth_token) kembali kepada kita,
3. Kita menggunakan tanda ini untuk membuat URL yang akan membawa pengguna ke halaman verifikasi,
4. Setelah pemberian kita akses kita bisa meminta tanda atau akses rahasia tanda (oauth_token_secret),

5. Tanda rahasia tersebut digunakan untuk membuat panggilan ke API.




**Gambar 2.8 Cara kerja Twitter API**