**2. TINJAUAN PUSTAKA**

* 1. ***Machine learning***

*Machine learning* atau pembelajaran mesin merupakan pendekatan dalam AI yangbanyak digunakan untuk menggantikan atau menirukan perilaku manusia untuk menyelesaikan masalah atau melakukan otomatisasi [10]. *Machine learning* adalah teknik untuk melakukan inferensi terhadap data dengan pendekatan matematis yang merefleksikan pola-pola data [12]. *Machine learning* mencoba menirukan bagaimana manusia belajar dan menggeneralisasi. *Machine learning* membutuhkan data untuk dipelajari yang disebut sebagai data training. metode dalam *machine learning* salah satunya adalah klasifikasi yang digunakan untuk memilih atau mengklasifikasikan objek tertentu seperti manusia membedakan suatu objek dengan objek yang lain.

*Machine learning* merupakan salah satu cabang dari ilmu kecerdasan Buatan, khususnya yang mempelajari tentang bagaimana komputer mampu belajar dari data untuk meningkatkan kecerdasannya. *Machine learning* memiliki fokus pada pengembangan sebuah sistem yang mampu belajar sendiri untuk memutuskan sesuatu, tanpa harus berulang kali diprogram oleh manusia. Dengan metode tersebut, mesin tidak hanya bisa menemukan aturan untuk perilaku optimal dalam pengambilan keputusan, namun juga bisa beradaptasi dengan perubahan yang terjadi. Dalam pembelajaran mesin, Anda menganalisis kumpulan data yang besar untuk menemukan pola[8] .

Menurut [22] dalam situsnya Tipe – tipe *machine learning* dibedakan menjadi 3 yaitu :

1. ***Supervised Learning* *(Predictive)***

Model ini digunakan untuk memprediksi hasil masa depan berdasarkan data historis yang ada. Model prediktif diberikan instruksi yang jelas sejak awal. *Supervised Learning* adalah pembelajaran terarah, artinya ada guru yang mengajarkan dan murid yang diajarkan (mesin). *supervised learning* memerlukan data training agar mampu melakukan prediksi maupun klasifikasi. Algoritma yang termasuk kedalam *supervised learning* adalah:

* *Support Vector Machine*
* *Decision Tree*
* *Naive Bayes Classifier*
* *Artificial Neural Network*
* *Nearest Neighbor Classifier*, dsb

Pendekatan *supervised learning* mempunyai input dan output yang dapat dibuat untuk menjadi sebuah model matematis sehingga mampu melakukan prediksi dan klasifikasi berdasarkan data yang ada sebelumnya. berberapa contoh implementasi dari *supervised learning* adalah teknik klasifikasi untuk spam pada email

1. ***Unsupervised Learning* *(Descriptive)***

Model ini digunakan untuk melatih data yang tidak ditetapkan targetnya. Data tidak diberikan label, tetapi secara otomatis dibagi berdasarkan kemiripan dari data tersebut. Pendekatan *unsupervised learning* tidak menggunakan data latih atau data training untuk melakukan prediksi maupun klasifikasi. Algoritma yang termasuk kedalam *unsupervised learning* adalah:

* *K – Means*
* *Hierarchical Clustering*
* *DBSCAN*
* *Fuzzy C-Means*

Berdasarkan model matematisnya, algoritma ini tidak memiliki target variabel. Salah satu tujuan dari algoritma ini adalah mengelompokkan objek yang hampir sama dalam suatu area tertentu. Salah satu contoh implementasi dari *unsupervised learning* adalah pengelompokan jenis tanah dengan algoritma *clustering k-means*.

1. ***Reinforcement Learning***

Model ini digunakan untuk keadaan dimana mesin dilatih untuk mengambil keputusan yang spesifik berdasarkan kebutuhan bisnis dengan tujuan utama untuk memaksimalkan efisiensi.

Terdapat dua istilah penting dalam pembangunan model *machine learning* yaitu *training* dan *testing*. *Training* adalah proses membangun *model* dan *testing* adalah proses menguji kinerja *model* pembelajaran. *Dataset* adalah kumpulan data sampel dalam statistik. Sample inilah yang digunakan untuk membuat model maupun mengevaluasi model *machine learning*. *Dataset* dibagi 3 yaitu:

1. ***Training set***

Adalah himpunan data yang digunakan untuk melatih data atau membangun sebuah *model*

1. ***Development set***

*Development set* atau *Validation set* adalah himpunan data yang digunakan untuk mengoptimasi saat melatih *model*.

1. ***Testing set***

*Testing set* adalah himpunan data yang digunakan untuk menguji *model* setelah proses *training* selesai.

Pada umumnya, rasio pembagian *dataset* adalah (80%:10%:10%). Development set biasanya tidak digunakan apabila *dataset* berukuran kecil (hanya dibagi menjadi *training* dan *testing* saja). Pembagian *dataset* yang membagi *dataset* menjadi training dan testing saja umumnya memiliki rasio (90% : 10%), (80% : 20%), (70% : 30%) atau (50% : 50%) [12].

* 1. **Analisis Sentimen**

Analisis sentimen atau *opinion mining* merupakan proses memahami, mengekstrak dan mengolah data tekstual secara otomatis untuk mendapatkan informasi sentimen yang terkandung dalam suatu kalimat opini [11]. Analisis sentimen adalah analisis yang dilakukan pada pendapat seseorang tentang suatu organisasi, topik atau perusahaan tertentu. Analisis sentimen atau opinion mining merupakan proses memahami, mengekstrak dan mengolah data tekstual secara otomatis untuk mendapatkan informasi sentiment yang terkandung dalam suatu kalimat opini.

Tujuan pertama ketika seseorang berurusan dengan analisis sentimen biasanya terdiri dalam membedakan antara kalimat subjektif dan objektif. Jika kalimat yang diberikan diklasifikasikan sebagai obyektif, tidak ada tugas mendasar lainnya yang diperlukan, sedangkan jika kalimat tersebut diklasifikasikan sebagai subyektif, polaritasnya perlu diperkirakan [2]. Analisis sentimen dilakukan untuk melihat pendapat atau kecenderungan opini terhadap sebuah masalah atau objek oleh seseorang, apakah cenderung berpandangan negatif atau positif.

Secara umum, *Sentiment analysis* dibagi menjadi 2 kategori umum [6] :

1. *Coarse-grained sentiment analysis*
2. *Fined-grained sentiment analysis*

*Coarse-grained sentiment analysis* – kategori ini melakukan proses analisis pada level dokumen. Singkatnya adalah klasifikasi dilakukan dengan orientasi sebuah dokumen secara keseluruhan.

*Fined - grained sentiment analysis* – kategori ini adalah yang sedang populer sekarang. Obyek yang ingin diklasifikasikan bukan berada pada level dokumen melainkan pada sebuah kalimat di suatu dokumen.

Contoh:

1. Saya tidak suka dengan kpu (negatif)
2. KPU sudah berkerja dengan baik (positif)

Tugas dasar dari analisis sentimen adalah mengelompokan teks yang ada pada sebuah dokumen atau kalimat kemudian menentukan pendapat yang ditemukan dalam kalimat atau dokumen tersebut apakah bersifat positif atau negatif.

Analisis sentimen menjadi hal yang diperhatikan sekarang dikarenakan dengan analisis sentimen kita dapat mencari pendapat tentang produk – produk, merek atau orang – orang dan menentukan apakah mereka dilihat positif atau negatif di web [15]. Hal ini memungkinkan kita untuk mencari informasi tentang:

1. Deteksi Flame (rants buruk).
2. Persepsi produk baru.
3. Persepsi merek.
4. Manajemen reputasi

Dengan kemudahan akses dan banyaknya pengguna twittersebagai sarana penyampaian pendapat, maka akan sangat banyak data yang bisa digunakan sebagai bahan penilaian dan evaluasi melalui data twitter*.* Namun, data yang terkumpul tersebut harus diubah menjadi informasi yang spesifik yang digunakan sebagai bahan penilaian. Oleh karena itu, pengolahan data twitter perlu dilakukan. Pada penelitian ini, pengolahan data twitter akan dilakukan dengan analisis sentimen supaya dapat menganalisis pendapat yang mengungkapkan pandangan positif atau negatif terhadap suatu entitas. Entitas tersebut ialah Komisi Pemilihan Indonesia pada penyelenggaraan pemilu bulan april tahun 2019 lalu.

* 1. ***Text Mining***

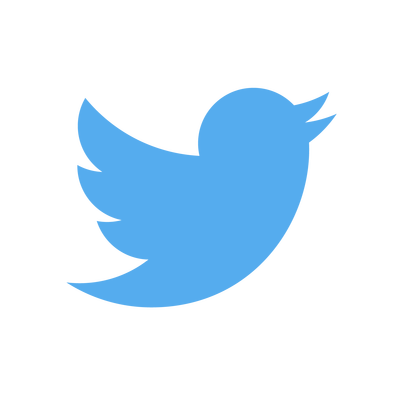
*Text mining*, juga disebut sebagai data *text mining*, kira-kira setara dengan analisis teks, adalah proses untuk memperoleh informasi kualitas tinggi dari teks. [4]. *Text Mining* adalah proses ekstraksi pola (informasi dan pengetahuan yang berguna) dari sejumlah besar sumber data tak terstruktur. Tujuan dari *Text mining* adalah untuk mendapatkan informasi yang berguna dari sekumpulan kata, paragraf, atau dokumen yang tersedia untuk dianalisa dan mencari informasi yang bermanfaat untuk tujuan tertentu. *Text mining* menghasilkan solusi dari berbagai masalah pemrosesan, pengorganisasian dan analisa data tak terstruktur dalam jumlah besar.

Menurut [19] dalam situsnya, *Natural language processing* (*NLP*) dan *text mining* merupakan teknologi *artificial intelligence (AI)* yang memungkinkan penggunanya untuk mengubah konten inti dari sebuah dokumen teks menjadi sebuah data kuantitatif secara cepat. Data kuantitatif tersebut nantinya akan dapat digunakan atau ditindaklanjutkan sesuai keinginan penggunanya. penggunaan *NLP* dan *text mining* dapat membantu untuk meningkatkan efisiensi analisis dokumen teks dengan memberikan kemampuan otomasi pada proses tersebut

*NLP* dan *text mining* biasanya digunakan dalam:

* Menganalisa sebuah survey dimana para peserta menuliskan komentarnya secara bebas
* Mengklasifikasi email sebagai *spam*, *marketing*, atau personal (contoh: Gmail)
* Menganalisa klaim asuransi atau garansi, wawancara diagnostik, dll
* Meneliti kompetitor dengan melakukan *crawling* dalam *website* merek
  1. **Twitter**

Twitter adalah jejaring sosial yang dimiliki dan dikelola oleh Twitter Inc. Twitter menawarkan jaringan sosial mikroblog sehingga memungkinkan penggunanya untuk membaca dan mengirimkan pesan *tweets*. Mikroblog adalah suatu bentuk blog yang memungkinkan penggunanya untuk menulis teks pembaharuan singkat dan mempublikasikannya.



**Gambar 2.1** Logo Twitter

Pengguna Twitter sendiri bisa terdiri dari berbagai macam kalangan yang para penggunanya ini dapat berinteraksi dengan teman, keluarga hingga rekan kerja. Twitter sebagai sebuah situs jejaring sosial memberikan akses kepada penggunanya untuk mengirimkan sebuah pesan singkat yang terdiri dari maksimal 280 karakter (disebut *tweet*). *Tweet* sendiri bisa terdiri dari pesan teks dan foto. Melalui *tweet* inilah pengguna Twitter dapat berinteraksi lebih dekat dengan pengguna Twitter lainnya dengan mengirimkan tentang apa yang sedang mereka pikirkan, apa yang sedang dilakukan, tentang kejadian yang baru saja terjadi, tentang berita terkini serta hal lainnya.

*Tweets* bisa dilihat oleh public, namun pengguna dapat membatasi orang yang melihat *tweets* nya sebatas pada orang – orang yang mengikuti akun pengguna atau bisa disebut followers. Pada twitter, pengguna dapat menulis pesan atau *tweet* berdasarkan topik dengan tanda # (*hastag*). Membalas atau menyebutkan *tweet* dari orang lain dapat menggunakan tanda @(et).

Berberapa fitur yang dimiliki oleh jejaring sosial twitter diantaranya adalah :

1. *Home* (Halaman Awal)

*Home* atau *timeline* adalah halaman yang berisi kumpulan tweet yang ditulis oleh pengguna aktif yang diikuti oleh user tersebut.

1. *Profile* (Profil)

*Profile* adalah halaman yang berisi informasi pribadi dari akun pengguna dan *tweet* apa saja yang sudah dibuat.

1. *Followers* (Pengikut)

*Followers* adalah pengikut yang mengikuti akun yang bersangkutan, *followers* dapat melihat *tweet* orang yang diikuti apabila diset *public*.

1. *Following* (Mengikuti)

*Following* atau mengikuti adalah kebalikan dari *followers*. Fitur ini digunakan apabila ingin melihat setiap *tweet* terbaru dari orang yang ingin diikuti.

1. *Mentions*

*Mention* adalah hal yang dilakukan bila seseorang membuat *tweet* dan ingin memasukan nama user lain di dalam *tweetnya*

1. *Favorite*

Seseorang dapat menekan tombol *favorite* bila menemukan *tweet* yang dirasa penting, dan dapat dibaca kapanpun.

1. *Direct Message*

*Direct Message* adalah fitur pesan langsung yang dapat mengirimkan sebuah pesan dari satu pengguna ke pengguna lain.

1. *Hashtags*

*Hastags* atau tagar adalah sebuah cara untuk mengkategorikan *tweet*, dengan melakukan ini seseorang dapat dengan mudah menemukan *tweet* dengan tema sejenis.

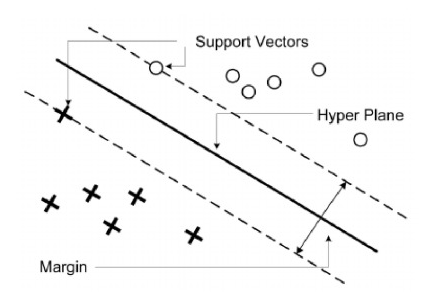
1. *List*

Cara yang digunakan untuk mengelompokan *followers* yang dimiliki pengguna.

Twitter Menyediakan API untuk developer melakukan pengambilan data *tweet*. *API* (*Application Programming interface*) adalah sekumpulan perintah, fungsi, serta protokol yang dapat digunakan oleh programmer saat membangun perangkat lunak tertentu. *API* memungkinkan programmer untuk mengintegrasikan dua bagian dari aplikasi atau dengan aplikasi yang berbeda secara bersamaan. Twitter menyediakan Twitter *API* yang memungkinkan *developer* menggabungkan program dengan layanan dari Twitter. Twitter *API* membuat program komputer "berbicara" satu sama lain agar mereka dapat meminta dan menyajikan informasi. Twitter api menyediakan berberapa kunci yang dapat digunakan untuk developer yang ingin mengambil data dari *Twitter.*

* 1. ***Support Vector Machine***

*Support vector machine (SVM)* adalah suatu teknik yang relatif baru (1995) untuk melakukan prediksi, baik dalam kasus klasifikasi maupun regresi, yang sangat populer belakangan ini. *SVM* berada dalam satu kelas dengan *ANN* dalam hal fungsi dan kondisi permasalahan yang bisa diselesaikan. Keduanya masuk dalam kelas *supervised learning*, dimana dalam implementasinya perlu adanya tahap training dan disusul tahap testing [21]. *Support Vector Machine (SVM)* Adalah salah satu algoritma pada *machine learning* yang paling efektif dari sisi praktis maupun teoritis. *SVM* berusaha menemukan *hyperplane* dengan memaksimalkan jarak antar kelas. Dengan cara ini, *SVM* dapat menjamin kemampuan generalisasi yang tinggi untuk data-data yang akan datang [7]. *SVM* berusaha menemukan *hyperplane* terbaik dengan memaksimalkan *margin* antar kelas.



Gambar 2.2 Ilustrasi Support Vector Machine

Proses pembelajaran *SVM* adalah untuk menentukan *support vector*, hanya cukup mengetahui fungsi kernel yang dipakai dan tidak perlu mengetahui wujud dari fungsi non-linier. *SVM* dapat memisahkan data secara *linear* dan non linear.

1. *Hard Margin SVM*

*Hard Margin* atau *Linearly sparable data* adalah data yang digunakan untuk perhitungan svm yang dapat dipisahkan secara *linear*. Pada kasus ini data yang digunakan adalah data dengan dua kelas yang sudah terpisah secara linear. Data latih dinyatakan oleh *(xi , xj )* dan *xi*adalah atribut set untuk data latih kelas ke – i yakni *xi* = { *x1, x2, x3* ... *xi* }. Untuk y {1,-1} menyatakan label kelas. Pendefinisian persamaan suatu *hyperplane* pemisah dituliskan dengan :

*w \* xi +b* = 0 (2.1)

Dimana,

w = bobot

b= bias

x= *variable* input

Dengan ketentuan jika *w \* xi* + b > 0 untuk yi = +1 adalah *hyperplane*-pendukung (*supporting* *hyperplane*) dari kelas +1, dan *w \* xi* + b < 0 untuk yi = -1 adalah *hyperplane* pendukung dari kelas -1, margin antara dua kelas dapat dihitung dengan mencari jarak pembatas antara kedua kelas dan titik terdekatnya dengan mencari titik minimal dengan persamaan (2.2):

*min* *t*(*w*) = ||*w*||2  (2.2)

sedangkan subjek *constrain* / kendala persamaannya adalah sebagai berikut:

*y(xi* . *w + b)*-1 ≥ 0, i (2.3)

Permasalahan ini dapat dipecahkan dengan berbagai teknik komputasi. Lebih mudah dengan mengubah persamaan diatas ke dalam fungsi lagrangian dan menyederhanakannya menjadi persamaan berikut :

L (*w,b,a*) = ||*w*||2 - i *yi* (*wTxi+b*)+ i  (2.4)

Dimana *a*i adalah lagrange multiplier yang bernilai nol atau positif (*a*i ≥ 0). Nilai optimal dari (2.4) dapat dihitung dengan meminimalkan L terhadap *a*i sekaligus terhadap *w* dan *b* dengan persamaan berikut:

i *-*  , *a*i *a*jyi xi (2.5)

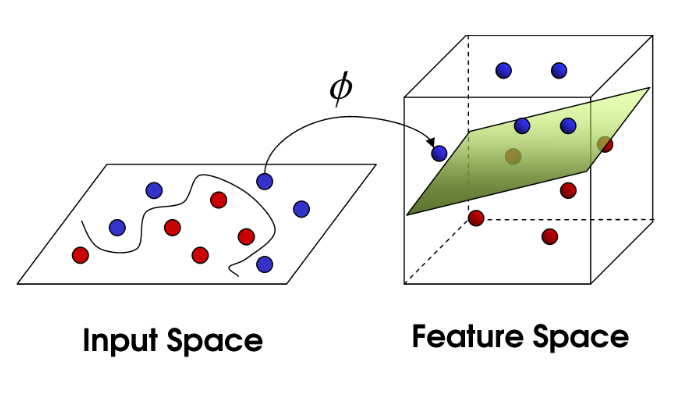
Dengan memperhatikan persamaan berikut :

i yi = 0 , *a*i ≥ 0 (*i,j =* 1,..,n) (2.6)

Diperoleh nilai *a*i yang nantinya akan digunakan untuk menemukan nilai bobot (*w*). Terdapat nilai *a*i untuk setiap data latih. Nilai *a*i yang > 0 adalah merupakan support vector dan dapat digunakan untuk mencari hyperplane(*b*).

1. *Soft Margin SVM*

Apabila data tidak dapat dipisahkan secara linear, dapat dengan cara mentransformasikan data ke dalam dimensi ruang fitur (*feature space*) sehingga dapat dipisahkan secara linear pada ruang fitur. Transformasi dilakukan dengan fungsi pemetaan *xk 🡪* (*xk*) dari input space ke dalam feature space sesuai dengan kelasnya. Permisalan dapat dilihat pada gambar 2.3.



**Gambar 2.3** Transformasi Data

Proses pembelajaran pada metode klasifikasi *SVM* adalah dengan menemukan *support vector* yang hanya bergantung pada dot product dari data pada ruang fitur (*feature space*). Fungsi proses transformasi sangat sulit dipahami, oleh karenanya perhitungan dot product dapat digantikan dengan fungsi kernel yang mendefinisikan fungsi transformasi secara implisit. Fungsi *kernel* inilah yang disebut dengan kernel trick. Dengan *kernel* *trick* ini maka, hanya perlu diketahui fungsi *kernel* yang dipakai untuk menentukan *support vector*. Terdapat 4 fungsi *kernel* yang sering digunakan, diantaranya:

1. *Kernel* linier

(2.7)

*K(x , xk ) = xtk x*

1. *Kernel* *Polynomial*

(2.8)

*K(x , xk ) = ( xtk x + 1 )d*

1. *Kernel* Gaussian (*RBF*, *Radial Basis Function*)

(2.9)

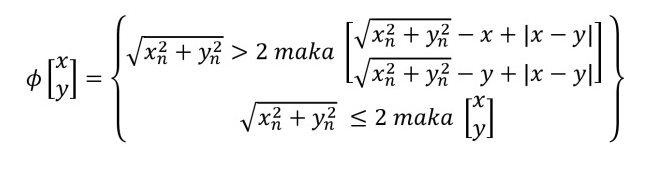
*K(x , xk ) =* exp {-||*x- xk* ||2 / σ2}

1. *Kerne*l Sigmoid

(2.10)

*K(x , xk ) =* tanh[k *xtk x + θ*]

Untuk mendapatkan nilai *a*i, adalah mengubah setiap dokumen menjadi vektor transformasi = . Kemudian nilai vektor transformasi dari setiap abstrak dimasukan ke persamaan (2.11) berikut.



(2.11)

Nilai *x* didapatkan dari persamaan berikut:

xi , (*i, j =* 1 ,..,*n*) (2.12)

Nilai *y* didapatkan dari persamaan berikut:

yi , (*i, j =* 1 ,..,*n*) (2.13)

Nilai *x* dan *y* diberi nilai bias 1, untuk kemudian dicari parameter *a*i dengan persamaan berikut:

, *a*i Dj  = *yi* (2.14)

Setelah parameter *a*i didapatkan, dengan menggunakan persamaan hasilnya digunakan untuk mencari bobot dan *hyperplane* :

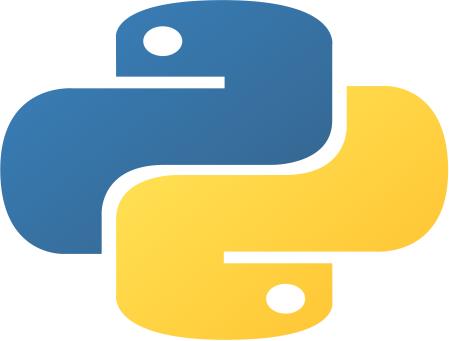
*W =*  i Di (2.15)

Hasil dari persamaan diatas selanjutnya digunakan untuk mencari bobot dan *hyperplane* untuk mengklasifikasikan kedua kelas.

*y = wx+b* (2.16)

* 1. **Bahasa Pemrograman *Python***

Bahasa *Python* adalah salah satu bahasa yang memiliki banyak kegunaan, terutama pada bidang *machine learning*, *Artificial* *intelligence* dan *data* *mining*. *Python* adalah bahasa tingkat tinggi yang melakukan instruksi secara langsung (interpretatif) dengan metode orientasi objek.



Gambar 2.4 Logo Python

*Python* adalah *general*-*purpose*, *high*-*level* *programming* *language*. Filosofi desain *python* menitikberatkan pada *code* *readability*, dan syntax yang memungkinkan programmer untuk mengekspresikan konsepnya dengan *lines* *of* *code* yang lebih sedikit dari bahasa lainnya [4].

*Python* digunakan di berbagai bidang pengembangan, menurut [18] dalam situsnya, aplikasi penggunaan python digunakan di berbagai bidang pengembangan. Berikut berberapa aplikasi penggunaan python yang paling populer :

1. Website dan Internet

Bahasa pemrograman python dapat digunakan sebagai *server* *side* yang diintegrasikan dengan berbagai internet protokol misalnya HTML, JSON, *Email* *Processing*, FTP, dan IMAP. Selain itu, *python* juga mempunyai *library* untuk pengembangan internet.

1. Penelitian Ilmiah dan Numerik

*Python* dapat digunakan untuk melakukan riset ilmiah untuk mempermudah perhitungan numerik. Misalnya penerapan algoritma *KNN*, *Naive* *Bayes*, *Decision* *Tree*, dan lain-lain.

1. *Data* *Science* dan *Big* *Data*

*Python* memungkinkan untuk melakukan analisis data dari *database* *big* *data.*

1. Media Pembelajaran Pemrograman

*Python* dapat digunakan sebagai media pembelajaran di universitas. *Python* sangat mudah dan hemat untuk dipelajari sebagai *Object* *Oriented* Programming dibandingkan bahasa lainnya seperti MATLAB, C++, dan C#.

1. *Graphical* *User* *Interface* (*GUI*)

*Python* dapat digunakan untuk membangun *interface* sebuah aplikasi. Tersedia *library* untuk membuat *GUI* menggunakan *python*, misalnya Qt, win32extension, dan GTK+.

1. Pengembangan *Software*

*Python* menyediakan dukungan struktur kode untuk mempermudah pengembangan *software*.

1. Aplikasi Bisnis

*Python* juga dapat digunakan untuk membuat sistem informasi baik untuk bisnis dan instansi.

Penelitian ini menggunakan berberapa *library* untuk mempermudah pemrosesan data. *Library* atau paket adalah sebuah pustaka berisi *script* yang dapat dipanggil dari dalam program untuk mempermudah proses koding. *Python* adalah salah satu bahasa dengan komunitas yang besar dan memiliki berbagai *library* gratis yang dapat digunakan. *Library* yang digunakan di dalam penelitian ini adalah:

1. Tweepy

Tweepy adalah *library* yang digunakan untuk menghubungkan *python* dengan Twitter *API*. Installasinya yaitu cukup mengetikan pip *install* tweepy pada command prompt apabila pada komputer pengguna sudah ter-*install* *python*.

1. Textblob

Textblob adalah *library* yang digunakan untuk menentukan sentimen dari suatu kalimat yang diteliti. Cara instalasi nya pun sama, yaitu dengan mengetikan perintah pip *install* textblob pada cmd.

1. Pandas

Pandas merupakan *toolkit* yang *powerfull* sebagai alat analisis data dan struktur untuk bahasa pemrograman *Python*. salah satu fiturnya adalah *Dataframe*. Dengan adanya fitur dataframe penulis dapat membaca sebuah file dan menjadikannya *table*.

1. Sastrawi

Sastrawi adalah *library* sederhana yang digunakan untuk proses stemming dalam Bahasa Indonesia ke bentuk dasarnya. Terlepas dari kesederhanaannya, library ini dirancang agar berkualitas tinggi dan didokumentasikan dengan baik. alasan digunakan *stemmer* satrawi pada penelitian ini adalah karena sastrawi adalah *stemmer* *open source* dan sudah mengalami perbaikan dan merupakan stemmer terbaru saat ini.

1. reGex

reGex adalah *library* *python* yang digunakan untuk membantu proses pemotongan dan penggabungan kata dan karakter di dalam sebuah *text*.

1. scikit-learn

scikit-learn adalah *library* *python* yang digunakan untuk membantu proses *machine learning*. Memiliki berbagai macam feature untuk *machine learning*.

* 1. **Bahasa Pemrograman PHP**

PHP dibuat pertama kali oleh seorang perekayasa perangkat lunak (software engineering) yang bernama Rasmus Lerdoff. Rasmus Lerdoff membuat halaman *web* PHP pertamanya pada tahun 1994. PHP4 dengan versi-versi menuju PHP5 sudah mendukung pemrograman berorientasi objek PHP merupakan bahasa pemrograman yang digunakan untuk pemrograman *web* [5].



**Gambar 2.5** Logo Php

Pada awalnya, PHP merupakan kependekan dari *Personal* *Home* *Page* (situs personal). Selanjutnya, Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI pada sekitar tahun 1995, dan diperkenalkan kepada berberapa programmer pemula dengan alasan bahasa yang digunakan oleh PHP cukup sederhana dan mudah dipahami [1].

* 1. **Framework CodeIgniter**

CodeIgniter pertama kali dibuat oleh *Rick Ellis*, CEO *Ellislab*, *Inc*. *(http://ellislab.com)*, sebuah perusahaan yang memproduksi CMS (*Content* *Management* *System*) yang cukup handal, yaitu *Expression* *Engine* *(http://www.expressionengine.com)*. Saat ini, CodeIgniter dikembangkan dan dimaintain oleh *Expression* *Engine* *Development* *Team*.



**Gambar 2.5** Logo CodeIgniter

CodeIgniter merupakan kerangka kerja PHP (*Framework* PHP) sehingga pembuatan web dengan PHP menjadi lebih mudah. CodeIgniter adalah aplikasi open source yang berupa framework dengan model MVC (*Model, View, Controller*) untuk membangun *website* dinamis dengan menggunakan PHP [9].

MVC (Model View Controller) merupakan suatu konsep yang cukup popular dalam pengembangan aplikasi *web*. MVC memisahkan pengembangan aplikasi manipulasi data, user interface, dan bagian yang menjadi control aplikasi. Terdapat 3 jenis komponen yang membangun suatu MVC pattern dalam suatu aplikasi, sebagai berikut:

1. *View*

Merupakan bagian yang menangani *presentation logic*, pada suatu aplikasi *web,* bagian ini biasanya berupa *file template HTML* yang diatur oleh *controller*. *View* berfungsi untuk menerima dan mempresentasikan data kepada pemakai (*user*). Bagian ini tidak memiliki akses langsung terhadap bagian model.

2. *Model*

Biasanya berhubungan langsung dengan *database* untuk memanipulasi data (*insert, update, delete, search*),menangani validasi dari bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian *view*.

3. *Controller*

Merupakan bagian yang mengatur hubungan antara bagian *model* dan bagian *view*. *Controller* berfungsi untuk menerima permintaan (*request*) dan data dari pemakai (*user*) kemudian menentukan apa yang akan diproses oleh aplikasi

* 1. **Flowchart**

Flowchart adalah suatu bagan dengan simbol-simbiol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan suatu proses (Instruksi) dengan proses lainya dalam suatu *program* [17]. Tujuan membuat flowchart

1. Menggambarkan suatu tahapan penyelesaian masalah
2. Secara sederhana, terurai, dan jelas
3. Menggunakan simbol-simbol standar

Simbol-simbol dari flowchart memiliki fungsi yang berbeda antara satu simbol dengan simbol lainya. Fungsi dari simbol-simbol flowchart adalah sebagai berikut

Simbol- simbol yang dipakai dalam flowchart dibagi menjadi 3 kelompok

1. *Flow* *Direction* *Symbols*

* Digunakan untuk menghubungkan simbol satu dengan yang lain
* Disebut juga *connecting line*

1. *Processing* *Symbols*

* Menunjukan jenis operasi pengolahan dalam suatu proses atau prosedur

1. *Input* *Symbols*

* Memnujukan jenis peralatan yang digunakan sebagai media *input* atau *output*

**Tabel 2.1** Flowchart Direction Symbols

|  |  |
| --- | --- |
| Lambang | Penjelasan |
|  | Simbol arus / *flow*, yaitu menyatakan jalannya arus suatu proses |
|  | Simbol *connector*, menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda |
|  | Simbol *connector*, menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda |

**Tabel 2.2** Flowchart *Processing* *Symbols*

|  |  |
| --- | --- |
| Lambang | Penjelasan |
|  | Simbol proses, yaitu menggambarkan proses yang dilakukan oleh komputer |
|  | Simbol manual, yaitu menggambarkan suatu proses yang tidak dilakukan oleh komputer |
|  | Simbol *decision*, yaitu menunjukan suatu kondisi yang akan menghasilkan kemungkinan ya atau tidak |
|  | Simbol *terminal*, yaitu menyatakan permulaan atau akhir program |
|  | Simbol *offline* *storage*, menggambarkan bahwa data akan disimpan ke suatu media tertentu |

**Tabel 2.3** *Input* / *Output* *Symbols*

|  |  |
| --- | --- |
| Lambang | Penjelasan |
|  | Simbol *input/output*, menyatakan *input* atau *output* tanpa tergantung jenis peralatannya |
|  | Simbol *punched* *card*, menyatakan input berasal dari kartu atau output ditulis ke kartu |
|  | Simbol *magnetic* *tape*, menyatakan *input* berasal dari pita magnetis atau *output* disimpan ke pita magnetis |
|  | Simbol *disk* *storage*, menyatakan input berasal dari *disk* atau *output* disimpan ke disk |

* 1. ***Business Process Modeling Notation***

*Business Process Modeling Notation* (*BPMN*) adalah notasi grafis yang menggambarkan logika dari langkah-langkah dalam proses bisnis. Notasi ini telah didesain secara khusus untuk mengkoordinasikan urutan proses dan pesan yang mengalir antara pelaku dalam kegiatan yang berbeda.

Tujuan utama dari *BPMN* adalah menyediakan suatu notasi yang mudah dipahami oleh semua masyarakat terutama pegiat *software*. Dari analis bisnis yang ada kemudian menciptakan *draft* permulaan dari proses-proses sampai dengan pengembangan teknis melipui alur dan pekerjaan dalam bentuk model atau notasi. BPMN juga menciptakan suatu jembatan terstandardisasi untuk gap antara desain proses bisnis dan implementasi proses [24]. Kategori dasar dari elemen *BPNM* adalah:

1. *Flow Objects*

*Flow Objects* dibagi menjadi 3 yaitu, *event, activity,* dan *gateaway*.

1. *Event*

*Event* digambarkan dengan sebuah lingkaran dan merupakan sesuatu yang “terjadi” selama berlangsungnya proses bisnis

**Tabel 2.4** *BPMN Flow Object*

|  |  |
| --- | --- |
| Lambang | Penjelasan |
| new.PNG | *Start*, digunakan untuk memulai aktifitas model |
| inter.PNG | *Intermediate*, *Intermediate* terletak diantara Start *Event* dan *End* *Event*. |
| end.PNG | *End*, digunakan untuk mengakhiri aktifitas *model* |

1. *Activity*

*Activity* merepresentasikan pekerjaan (task) yang harus diselesaikan dan merupakan bentuk umum untuk pekerjaan yang dilakukan. berberapa lambang dari *activity* adalah sebagai berikut

**Tabel 2.5** *BPMN Activity*

|  |  |
| --- | --- |
| Lambang | Penjelasan |
| task.PNG | *Task*, lambang yang digunakan sebagai penggambaran sebuah tugas |
| subproses.PNG | *Sub Proses*, lambang yang digunakan sebagai penggambaran sebuah *sub* *proses* |

1. *Gateaway*

*Gateway* digambarkan dengan bentuk seperti belah ketupat dan digunakan untuk mengontrol percabangan dan penggabungan Sequence Flow. Jadi, gateway menentukan keputusan tradisional, penggabungan, dan penggabungan aliran. *Internal Markers* akan menentukan perilaku dari kontrol.

**Tabel 2.6** *BPMN Gateaway*

|  |  |
| --- | --- |
| Lambang | Penjelasan |
| gateaway.PNG | *Gateaway*, digunakan untuk mengontrol percabangan dan penggabungan *Sequence Flow* |

1. *Data*

*Data* adalah informasi yang diperlukan atau diproduksi ketika menjalankan proses bisnis. Data direpresentasikan dengan empat elemen :

1. *Data Objects*
2. *Data Inputs*
3. *Data Outputs*
4. *Data Stores*

**Tabel 2.7** *BPMN Data*

|  |  |
| --- | --- |
| Lambang | Penjelasan |
| data object.PNG | *Data Object*, Mekanisme untuk menunjukkan bagaimana data dibutuhkan atau diproduksi oleh aktivitas |
| data io.PNG | *Data Input*, Gambar yang menandakan ada data masuk atau keluar. |
| datastore.PNG | *Data Store*, Tempat dimana data disimpan |

1. *Connecting Objects*

*Connecting Object* adalah elemen yang menghubungkan flow object.

**Tabel 2.8** *BPMN Connecting Objects*

|  |  |
| --- | --- |
| Lambang | Penjelasan |
| sequence flow.PNG | *Sequence Flow* , digunakan untuk menunjukkan urutan yang kegiatan akan yang dilakukan dalam sebuah proses |
| message flow.PNG | *Message Flow*, digunakan untuk menunjukkan aliran pesan antara dua entitas yang siap untuk mengirim dan menerima. |
| association.PNG | *Association*, digunakan untuk asosiasi data, informasi dan artefak dengan aliran  benda |

1. *Swimlanes*

*Swimlanes* digambarkan dengan bentuk garis yang memisahkan dan mengelompokkan aktor (pelaku yang berinteraksi dengan system). Banyak metodologi pemodelan menggunakan konsep swimlanes sebagai mekanisme untuk membagi kategori visual yang menggambarkan kemampuan fungsional atau tanggung jawab yang berbeda.

**Tabel 2.9** *BPMN Swimlanes*

|  |  |
| --- | --- |
| Lambang | Penjelasan |
| pool.PNG | *Pool* , swimlanes yang mewakili partisipan dalam sebuah proses dan lane yaitu sub-bagian dalam sebuah pool |
| lane.PNG | *Lanes*, digunakan untuk mengatur dan mengkategorikan aktivitas |

1. *Artifacts*

*Artifacts* adalah elemen yang digunakan untuk memberikan informasi tambahan dari sebuah proses.

**Tabel 2.10** *BPMN Artifacts*

|  |  |
| --- | --- |
| Lambang | Penjelasan |
| group.PNG | *Group*, *Group* dapat digunakan untuk tujuan dokumentasi atau analisis, tetapi tidak mempengaruhi *Sequence* *Flow*. |

* 1. ***Confusion Matrix***

*Confusion Matrix* adalah tabel yang digunakan untuk memahami kinerja model klasifikasi [3]. *Confusion* *matrix* digambarkan dengan tabel yang menyatakan jumlah data uji yang benar diklasifikasikan dan jumlah data uji yang salah diklasifikasikan. *Confusion* *Matrix* seringkali digunakan untuk mengukur Akurasi, Presisi, dan *Recall* pada sebuah model klasifikasi. Akurasi menggambarkan seberapa akurat sistem melakukan klasifikasi, presisi adalah kecocokan antara bagian data yang diambil dengan informasi yang dibutuhkan, recall menggambarkan berapa % data berkategori positif yang terklasifikasikan dengan benar oleh sistem sedangkan, F1 score adalah keseimbangan antara presisi dan recall.

Akurasi, Presisi, dan Recall dapat dicari dengan menggunakan rumus sebagai berikut:

Akurasi = \* 100%

Presisi = \* 100%

Recall = \* 100%

F1 Score = 2 \*

(2.17)

(2.18)

(2.19)

(2.20)

Dimana,

TP = *True* Positif

TN = *True* Negatif

FP = *False* Positif

FN = *False* Negatif

Untuk menentukan keempat nilai diatas. *Confusion* *matrix* perlu dibuat. Tabel untuk *Confusion* *matrix* dapat dilihat pada tabel 2.4

**Tabel 2.11** *Confusion* *Matrix*

|  |  |  |
| --- | --- | --- |
| **Prediction** | **True Values** | |
| **True** | **False** |
| **True** | TP  *Correct Result* | FP  *Unexpected Result* |
| **False** | FN  *Missing Result* | TN  *Correct Absence of Result* |

Berdasarkan nilai *True* *Negative* (TN), *False* *Positive* (FP), *False* *Negative* (FN), dan *True* *Positive* (TP) dapat diperoleh nilai akurasi, presisi dan recall. Nilai akurasi menggambarkan seberapa akurat sistem dapat mengklasifikasikan data secara benar. Dengan kata lain, nilai akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data. Nilai akurasi dapat diperoleh dengan Persamaan 2.17. Nilai presisi menggambarkan jumlah data kategori positif yang diklasifikasikan secara benar dibagi dengan total data yang diklasifikasi positif. Presisi dapat diperoleh dengan Persamaan 2.18. Sementara itu, recall menunjukkan berapa persen data kategori positif yang terklasifikasikan dengan benar oleh sistem. Nilai recall diperoleh dengan Persamaan 2.19

.

* 1. **Komisi Pemilihan Umum Indonesia**

Komisi pemilihan umum atau KPU adalah lembaga yang menyelenggarakan pemilu di Indonesia. Menurut UU No.22 Tahun 2007 Pasal 1 ayat 6, Komisi Pemilihan Umum, selanjutnya disebut KPU, adalah lembaga Penyelenggara Pemilu yang bersifat nasional, tetap, dan mandiri.



**Gambar 2.7** Logo Komisi Pemilihan Umum Indonesia

Visi dan Misi KPU adalah sebagaimana dibawah ini:

Visi :

Menjadi Penyelenggara Pemilihan Umum yang Mandiri, Professional, dan Berintegritas untuk Terwujudnya Pemilu yang LUBER dan JURDIL

Misi :

1. meningkatkan kualitas penyelenggaraan Pemilu yang efektif dan efisien, transparan, akuntabel, serta aksesibel;
2. meningkatkan integritas, kemandirian, kompetensi dan profesionalisme penyelenggara Pemilu dengan mengukuhkan *code* *of* *conduct* penyelenggara Pemilu;
3. menyusun regulasi di bidang Pemilu yang memberikan kepastian hukum, progesif, dan partisipatif;
4. meningkatkan kualitas pelayanan Pemilu untuk seluruh pemangku kepentingan;
5. meningkatkan partisipasi dan kualitas pemilih dalam Pemilu, Pemilih berdaulat Negara kuat; dan
6. mengoptimalkan pemanfaatan kemajuan teknologi informasi dalam penyelenggaraan Pemilu.
   1. **Pembobotan TF-IDF**

Dalam pencarian informasi, TF-IDF (*Term Frequency - Inverse Document Frequency*) adalah statistik numerik yang dimaksudkan untuk mencerminkan betapa pentingnya sebuah kata dalam koleksi atau corpus [4]. TF-IDF adalah metode yang digunakan untuk menghitung bobotan setiap kata yang telah diekstrak.

Model pembobotan TF-IDF merupakan metode yang mengintegrasikan model *term* *frequency* (tf) dan *inverse* *document* *frequency* (idf).*Term* *frequency* (tf) merupakan proses untuk menghitung jumlah kemunculan *term* dalam satu dokumen dan inverse *document* *frequency* (idf) digunakan untuk menghitung *term* yang muncul di berbagai dokumen (komentar) yang dianggap sebagai term umum, yang dinilai tidak penting [17].

Interface menu TF-IDF berisi informasi tentang kata-kata yang telah dipecah dari tweet dan telah memiliki nilai TF-IDF sendiri. Rumus yang digunakan adalah :

(2.21)

TFIDF = TF x IDF = TF x loge

Dimana,

TF = *Term* *Frequency*, banyaknya sebuah kata/istilah muncul

DF = *Document* *Frecuency*, jumlah dokumen di mana kata/istilah tersebut muncul. Minimal 1 dokumen.

| D | = Total seluruh dokumen

* 1. ***Lexicon Based Features***

*Lexcion Based Features* merupakan pendekatan yang menggunakan suatu kamus sentimen berisi kata positif dan kata negatif yang dibandingkan dan dicocokkan dengan kata pada kalimat untuk diketahui tingkat polaritasnya.

*Lexcion Based Features* merupakan suatu kesepakatan dalam pendekatan yang meliputi frase, bentuk ekspresi, atau konten yang berupa teks yang umumnya terdapat pada obrolan, dialog, *post*, *review*,dan lainnya [17]. Dalam proses klasifikasi menggunakan *lexicon* *based*, dapat digunakan rumus sebagai berikut :

(2.22)

Spositif =

Snegatif =

Dimana, (*Spositif*) adalah sentimen positif hasil dari jumlah bobot positif yang memiliki nilai positif lebih besar, dan *(Snegatif)* adalah sentimen negatif yang didapatkan dari jumlah bobot negatif yang lebih besar. Jika total jumlah nilai positif lebih besar dari jumlah nilai negatif maka kalimat akan berorientasi positif. Namun jika total jumlah nilai positif kurang dari jumlah nilai negatif maka kalimat akan berorientasi negatif [14]. Dari persamaan nilai sentimen dalam suatu kalimat maka, diperoleh persamaan (2.23) untuk menentukan orientasi sentimen.

Sentencesentiment  (2.23)

* 1. **Penelitian Terkait**

Berikut adalah penelitian terdahulu yang terkait dalam penelitian ini yang ditunjukan pada tabel 2.1

**Tabel 2.12** Penelitian Terkait

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Author** | **Judul Penelitian** | **Metode** | **Hasil/Output** |
| 1 | Wanda A.L, dkk, (2018) | Analisis Sentimen *Cyberbullying* pada Komentar Instagram dengan Metode *Klasifikasi* *Support* *Vector* *Machine* | klasifikasi *Support* *Vector* *Machine* (SVM) dengan menggunakan fitur *Term* *Frequency*-*Inverse* *Document* *Frequency* (TF-IDF) dan *lexicon based feature* | Dalam penelitian ini, peneliti mengambil data komentar instagram secara luring(*offline*), kemudian melakukan pengujian dengan menggunakan dan tidak menggunakan *lexicon based feature.*  tingkat akurasi terbaik sebesar 90% tanpa mengimplementasikan algoritma *Lexicon Based Features* |
| 2 | Dimas J. H, dkk, (2018) | Analisis Sentimen Review Barang Berbahasa Indonesia Dengan Metode *Support* *Vector* *Machine* Dan *Query* *Expansion* | *Support Vector Machine* kerne Polynomial berderajat dua dengan *Query Expansion.* | Pada penelitian ini, peneliti menggunakan 400 data komentar yang terbagi menjadi dua yaitu positif dan negatif. Peneliti membandingkan penggunaan metode SVM dengan dan tanpa *Query Expansion* Akurasi yang diperoleh dengan menggunakan metode *Support Vector Machine* *dan Query Expansion* sebesar 96.25%, sedangkan akurasi yang diperoleh dengan menggunakan metode *Support Vector Machine* tanpa *Query Expansion* sebesar 94.75%. |
| 3 | Faisal R, dkk, (2016) | Implementasi Twitter *Sentiment* *Analysis* untuk Review film menggunakan Algoritma *Support* *Vector* *Machine* | *Support Vector Machine* | Dalam penelitian ini peneliti menggunakan *dataset* berjumlah 1.027 tweet yang didapatkan dari tweet untuk film populer tahun 2016. Hasil klasifikasi opini terbagi menjadi 3, yaitu opini positif, negatif, dan netral. Hasil akurasi klasifikasi algoritma *Support Vector Machine* menggunakan 60, 70, 80 ,90 persen data training dengan rata – rata yang berbeda – beda. |
| 4 | Elly Indrayuni (2016) | Analisa Sentimen Review Hotel Menggunakan Algoritma *Support* *Vector* *Machine* Berbasis *Particle* *Swarm* *Optimization* | *Support Vector Machine,* *Particle* *Swarm* *Optimization* | Pada penelitian ini terbukti algoritma Support Vector Machine merupakan salah satu algoritma yang paling akurat dengan menghasilkan akurasi sebesar 91.33% dengan nilai AUC sebesar 0.988. Pada penelitian ini terbukti bahwa penggunaan seleksi fitur Particle Swarm Optimizationpada algoritma Support Vector Machinedapat meningkatkan akurasi. |
| 5 | Listari, dkk (2018) | Analisis Sentimen Twitter terhadap Bom Bunuh Diri di Surabaya 13 Mei 2018 menggunakan Pendekatan *Support* *Vector* *Machine* | *Support Vector Machine* | Dalam penelitian ini, peneliti menggunakan 1921 *tweets* sentimen negatif dan 121  *tweets* sentimen positif. Hasil klasifikasi yang diperoleh dengan menggunakan metode SVM memiliki tingkat akurasi sebesar 100% yang diuji dengan data testing sebanyak 1708 data dan dilatih dengan data training sebanyak 334 data. |
| 6 | Noviah Dwi Putranti, Edi Winarko (2014) | Analisis Sentimen Twitter untuk Teks Berbahasa Indonesia dengan *Maximum Entropy* dan *Support Vector Machine* | *Maximum Entropy dan Support Vector Machine* | Pada penelitian ini, peneliti menggunakan data kotor yang berjumlah 81.885 tweet. Setelah melalui proses preprocessing dan POS tagger terdapat 44.006 data bersih yang terdiri dari 12.939 tweet positif, 12.654 tweet negatif dan 18.413 tweet netral yang digunakan sebagai data pelatihan untuk membangun model klasifikasi |