

BAB 2

LANDASAN TEORI

2.1 Perundungan (*Bullying*)

Perundungan atau yang lebih dikenal “*Bullying*” merupakan suatu tindakan yang bersifat melukai seseorang secara fisik ataupun mental. Dalam Kamus Besar Bahasa Indonesia arti perundungan sendiri adalah **rundung**/run·dung/ *v*, **merundung**/me·run·dung/ *v* mengganggu; mengusik terus-menerus; menyusahkan. Beberapa tindakan negatif yang termasuk *bullying*[5] adalah :

1. Mengatakan hal yang tidak menyenangkan ataupun memanggil seseorang dengan julukan buruk;
2. Mengabaikan atau mengucilkan seseorang dari suatu kelompok karena sebuah tujuan;
3. Mengatakan kebohongan atau rumor yang tidak benar mengenai seseorang atau membuat orang lain tidak menyukai seseorang dan hal-hal semacamnya.

Dalam media sosial, perundungan lebih dikenal sebagai “*Cyber Bullying*”.

2.2 Media sosial

Media sosial merupakan sebuah sarana komunikasi digital yang terhubung secara online melalui jaringan Internet. Media sosial memudahkan manusia dalam berkomunikasi dengan jarak dekat maupun jauh dan penyebaran informasi yang cepat. Pengguna media sosial disebut juga sebagai user (pengguna). Kegiatan yang dilakukan di media sosial tidak hanya komunikasi antar user melainkan melakukan promosi, mendapatkan informasi ataupun berita terkini, menanggapi suatu posting (kiriman), melakukan vote (pemilihan), melakukan amal, bertransaksi ataupun belanja online, melakukan pemesanan. Dalam penelitian ini media sosial yang digunakan adalah Facebook.

2.3 Facebook Automation

Facebook Automation merupakan salah satu *tool* untuk memaksimalkan kegiatan marketing secara online. Fitur-fitur yang terdapat pada *tool* yaitu dapat menambah, menghapus, mengubah ataupun mengambil (*scraping*) pesan, status, komentar, dan informasi yang terkandung pada profile individu, grup komunitas, event, halaman yang disukai di Facebook. Dalam penelitian ini Facebook Automation dijadikan sebagai *tool* untuk mengambil data set yang dibutuhkan. Data yang diambil berasal dari komentar suatu kiriman dalam *Fanpage* Facebook.

2.4 Preprocessing

Preprocessing adalah proses pengolahan data untuk mempersiapkan data yang tidak terstruktur menjadi data yang terstruktur dan siap untuk proses selanjutnya[6]. Tahapan – tahapan yang dilakukan dalam penelitian ini yaitu *Case Folding*, *Cleaning*, *Convert Negation*, *Tokenizing*, dan *Stopword Removal*.

1. *Case Folding*

Case Folding merupakan tahapan merubah semua masukan huruf menjadi huruf kecil (*lower case*), dan menghilangkan karakter selain a-z.

2. *Cleaning*

Cleaning merupakan proses membersihkan dokumen dari komponen-komponen yang tidak penting seperti tanda baca.

3. *Convert Negation*

Convert Negation merupakan proses untuk menggabungkan kata negasi dengan kata setelahnya.

4. *Tokenizing*

Tokenizing merupakan tahap pemotongan kalimat berdasarkan tiap kata yang menyusunnya. Pada tahap ini dilakukan penguraian deskripsi yang semula berupa kalimat-kalimat menjadi kata tunggal (*term*).

5. *Stopword Removal*

Stopword Removal merupakan tahap menghilangkan kata-kata yang tidak deskriptif (tidak penting) seperti kata ganti orang, kata penghubung, penunjuk, dan sebagainya dapat dibuang.

2.5 Contextual Features

Contextual feature merupakan pendekatan kontekstual (*contextual*) yang mencakup jumlah istilah, arti kata, dan kedekatan istilah (frasa atau kolokasi)[4]. Namun untuk mendekati tingkat klasifikasi manusia, model pembelajaran harus dapat mengabstraksi informasi pelatihan dan menggabungkan dengan pengetahuan eksternal yang tidak tersedia dalam data pelatihan itu sendiri. Pendekatan sederhana untuk merepresentasikan hubungan kontekstual dari istilah adalah dengan menghitung berapa kali suatu istilah muncul dengan istilah lain dalam kerangka kontekstual yang diberikan. Definisi konteks ini sering disebut sebagai bigrams (konteks dua istilah) atau N-gram (konteks istilah N)[3].

Dalam penelitian ini, Ngram (Bigram) dijadikan metode yang mewakili Contextual Feature.

2.5.1 Ngram

Ngram merupakan teknik pemisahan teks dengan panjang ‘n’ dalam suatu teks[7]. Pada teknik ini nilai ‘n’ menjadi penentu pemisahan teks tersebut. Pembagian Ngram dapat bervariasi tergantung dari pendekatan dalam membagi teks menjadi bentuk Ngram. Ngram dapat berupa unigram (n=1), bigram (n=2), trigram (n=3), dan seterusnya.

2.6 Pembobotan TF-IDF

TF-IDF atau *Term Frequency – Inverse Document Frequency* digunakan untuk menentukan nilai frekuensi sebuah kata atau *term*. Pembobotan diperoleh dari statistik kemunculan kata dan tingkat kepentingan dokumen yang mengandungnya[3]. Metode TF-IDF dapat dirumuskan sebagai berikut :

$$IDF = \log(n/df) \quad (2.1)$$

Dimana :

n : Banyaknya data

df : *Document frequency*

$$w(t, d) = tf(t, d) \times IDF \quad (2.2)$$

Dimana :

tf : *term frequency*

IDF : *Inverse Document Frequency*

d : dokumen ke- d
 t : kata ke- t dari kata kunci
 $w(t, d)$: bobot dokumen ke- d terhadap kata ke- t

$$TF \times IDF = TF(w_i, d) \times IDF(w_i) \quad (2.3)$$

Dimana :

w_i : kata ke i
 d : dokumen
 $TF(w_i, d)$: Jumlah kemunculan kata w_i pada dokumen d
 IDF : *Invers Document Frequency* dari kata w_i .

2.7 Neural Network

Neural network merupakan salah satu pembelajaran dari kecerdasan buatan dimana sistem kerjanya meniru jaringan syaraf manusia. Pembelajaran merupakan proses menambah pengetahuan yang berkesinambung sehingga menjadi pola tertentu. Pada Neural Network terdapat beberapa lapisan (*layer*) yaitu lapisan masukan (*nput layer*), lapisan tersembunyi (*hidden layer*) dna lapisan keluaran (*output layer*). Setiap lapisan memiliki jumlah *neron* yang berbeda dan menerapkan fungsi aktivasi. Terdapat 4 fungsi aktivasi[8] diantaranya : aktivasi linear, aktivasi undak, sigmoid biner, dan sigmoid bipolar.

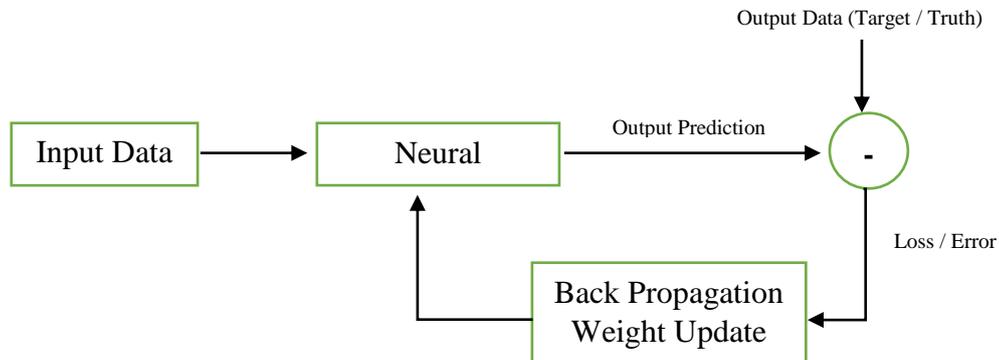
Dalam penelitian ini fungsi aktivasi yang digunakan merupakan sigmoid biner. Fungsi aktivasi ini sebenarnya sama dengan sigmoid biner, hanya batas nilai keluaran yang diberikan adalah -1 smapai 1.

$$y = \text{sign}(v) = \frac{1}{1 + e^{-av}} \quad (2.4)$$

2.7.1 Metode Backpropagation Network

Metode Backpropagation Network ditemukan pada tahun 1970-1n sebagai metode pengoptimalan umum untuk melakukan otomatis fungsi nested yang kompleks. Metode Backpropagation Network merupakan salah satu algoritma pembelajaran *supervised learning*. Backpropagation atau *backward pass* atau

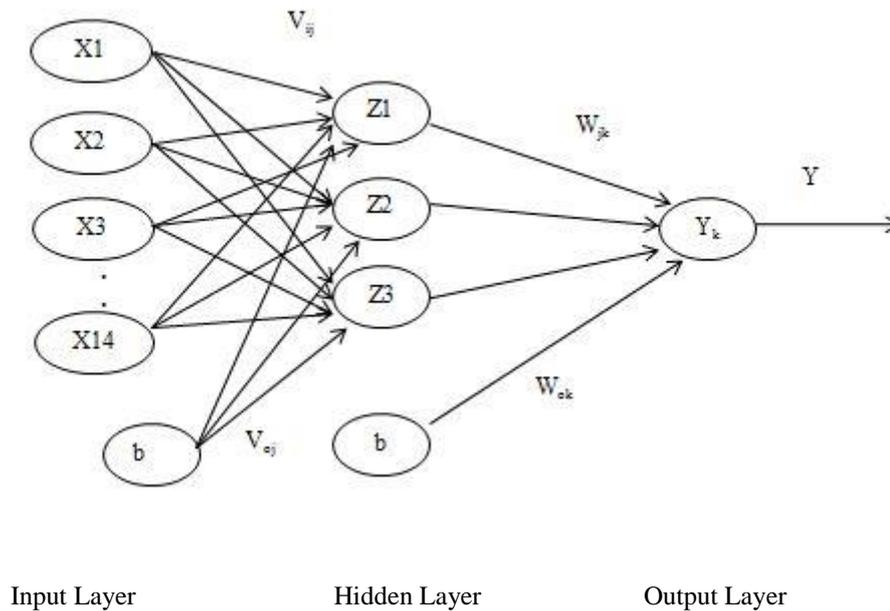
backward propagation of errors, adalah proses untuk menyesuaikan kembali weight dan bias berdasarkan error yang didapat pada tahap *foward pass*[9].



Gambar 2.1 Proses *learning* pada Neural Network

Backpropagation terdiri dari lapisan-lapisan diantaranya lapisan masukan (*input layer*), lapisan yang terdiri dari neuron input baik dari *neuron* pertama sampai *neuron network*. Lapisan tersembunyi (*hidden layer*), terdiri dari minimal satu lapis layer mulai dari neuron tersembunyi pertama sampai neuron ke- j [9].

Pelatihan metode Backpropagation terdiri dari 3 fase yaitu propagasi maju (*foward pass*) yang merupakan sinyal masukan untuk dipropogasikan ke *hidden layer* menggunakan fungsi aktivasi yang telah ditentukan. Propagasi mundur (*backward pass*) merupakan propagasi mulai dari garis yang berhubungan langsung dengan *neuron* di *layer* keluaran. Perubahan bobot adalah bobot semua garis yang dimodifikasi secara bersamaan. Ketiga fase tersebut terus diulang hingga kondisi penghentian terpenuhi.



Gambar 2.2 Contoh Arsitektur *Backpropogation* Maju

Keterangan :

X : Nilai Masukan (*Input layer*)

V_{ij} : Nilai bobot sambungan pada lapisan *input* X ke Unit Z

V_{oj} : Bobot bias pada lapisan tersembunyi Z ke unit b

Z : Bobot pada lapisan tersembunyi (*hidden layer*)

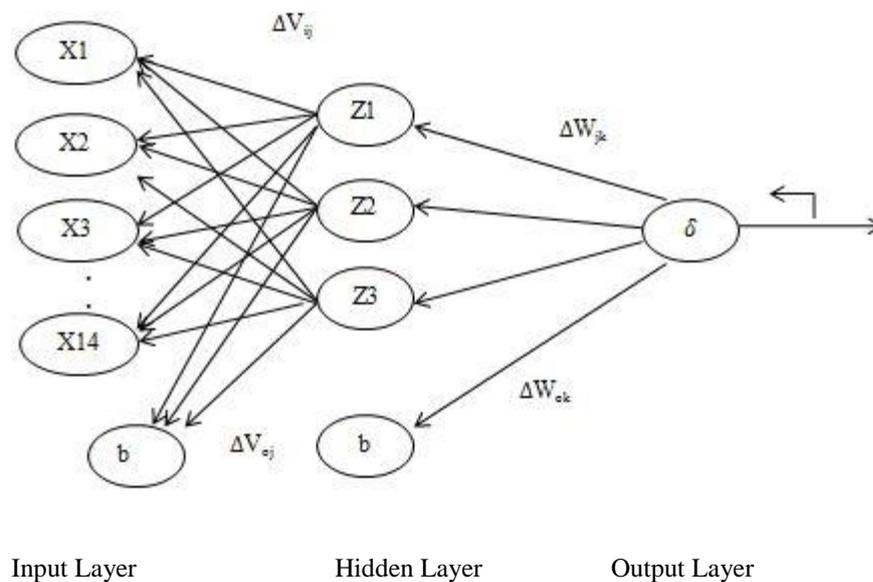
W_{jk} : Nilai perbaikan sambungan pada lapisan tersembunyi Z ke unit Y

W_{ok} : Bobot bias pada lapisan tersembunyi Y ke unit b

Y_k : Keluaran (*output*)

b : Bias

Pada gambar 2.2 merupakan contoh arsitektur *backpropagation* maju. Setelah melakukan *backpropagation* maju terdapat proses berikutnya yaitu perbaikan nilai bobot pada propagation mundur. Perhitungan galat dimulai dari nilai Y dimana dihitung nilai galat Y (*output*) ke Z (*hidden layer*) dan X (*input layer*) untuk menemukan nilai bobot baru yang digunakan kembali pada *backpropagation* maju.



Gambar 2.3 Contoh Arsitektur *Backpropagation* Mundur

Keterangan :

X : Nilai Masukan (*Input layer*)

ΔV_{ij} : Nilai bobot sambungan pada lapisan *input* X ke Unit Z

ΔV_{oj} : Bias sambungan pada lapisan *input* X ke Unit Z

Z_i : Bobot pada lapisan tersembunyi (*hidden layer*)

ΔW_{jk} : Nilai bobot sambungan dari Z ke unit Y

ΔW_{ok} : Bias pada sambungan dari Z ke Y

b : Bias pada lapisan tersembunyi dan lapisan keluaran

δ : Faktor pengaturan nilai penimbang sambungan pada lapisan keluaran.

Proses perbaikan pada lapisan tersembunyi diproses kembali kepada proses pelatihan *backpropagation* maju dengan nilai bobot baru pada bobot penghubung kepada *hidden layer* dan *output layer* sampai kondisi yang di tentukan terpenuhi.

2.7.1.1 Pelatihan *Backpropagation*

Secara umum langkah-langkah dalam pelatihan *Neural Network* menggunakan *Backpropagation* adalah sebagai berikut.

Langkah 0:

Berikan inialisasi penimbang (diberikan nilai kecil secara acak)

Langkah 1:

Jika kondisi berhenti tidak tercapai, maka ulangi langkah 2 sampai 9 hingga kondisi iterasi terpenuhi (kondisi berhenti jika perluangan tercapai maksimal atau *mean square error* (MSE) lebih kecil dari rasio pembelajaran).

Langkah 2:

Untuk masing – masing pasangan data latih (*training data*) lakukan langkah 3 sampai 8.

Propagasi Maju

Langkah 3:

Masukan masing – masing unit menerima sinyal masukan (X_i) dan sinyal disebarkan ke unit – unit bagian berikutnya (hidden layer atau lapisan tersembunyi).

Langkah 4:

Masing – masing dari unit dalam lapisan tersembunyi dikalikan dengan factor penimbang dan dijumlahkan serta ditambahkan (inisialisasi pada langkah 0) dengan bias:

$$Z_{inj} = V_{oj} + \sum_{i=1}^n X_i W_{ij} \quad (2.5)$$

V_{oj} bias pada fungsi tersembunyi dan kemudian hitung dengan sesuai fungsi aktivasi yang digunakan.

$$Z_{inj} = f(Z_{inj}) \quad (2.6)$$

Fungsi sigmoid yang digunakan adalah fungsi sigmoid biner maka gunakan persamaan (2.4)

Langkah 5:

Masing – masing unit keluaran Y_k , ($k =$ karena banyak output yang digunakan satu, maka outputnya keluarnya adalah satu). Unik (Z_p) dikalikan dengan faktor penimbang dan dijumlahkan.

$$Y_{ink} = W_{ok} + \sum_{j=1}^p Z_j W_{kj} \quad (2.7)$$

Wok bias pada unit keluaran dan hitung kembali dengan fungsi aktivasi *backpropagation* dan galatnya.

$$Y_k = f(Y_{in_k}) \quad (2.8)$$

Fungsi sigmoid yang digunakan adalah fungsi sigmoid biner maka gunakan persamaan (2.4)

Propagasi Mundur

Langkah 6:

Masing – masing unit dari keluaran Y_k menerima pola target yang sesuai dengan pola masukan saat pelatihan dan dihitung galatnya.

$$\delta_k = (T_k - Y_k) f'(1 - Y_k) \quad (2.9)$$

Dan hitung perbaikan faktor penimbang kemudian memperbaiki W_{jk}

$$\Delta W_{jk} = a \cdot \delta_k \cdot z_j \quad (2.10)$$

Menghitung perbaikan koreksi

$$\Delta W_{ok} = a \cdot \delta_k \quad (2.11)$$

Dan menggunakan nilai dari δ_k pada semua unit lapisan sebelumnya.

Langkah 7:

Masing – masing penimbang yang menghubungkan unit – unit dari lapisan keluaran dengan unit – unit pada lapisan tersembunyi (*hidden layer*) ke unit – unit lapisan berikutnya.

$$\delta_{inj} = \sum_{k=1}^n \delta_k W_{jk} \quad (2.12)$$

Dan selanjutnya dikalikan dengan turunan fungsi aktivitasnya untuk menghitung galat.

$$\delta_z = \delta_{inj} f'(y_{inj}) \quad (2.13)$$

Menghitung perbaikan penimbang yang digunakan untuk memperbaiki V_{ij}

$$\Delta V_{ij} = a \cdot \delta_z X_i \quad (2.14)$$

Memperbaiki penimbang dan bias

$$\Delta V_{oj} = a \cdot \delta_z \quad (2.15)$$

Langkah 8:

Masing – masing dari keluaran unit Y_k diperbaiki bias dan penimpangnya

$$W_{jk}(\text{baru}) = W_{jk}(\text{lama}) + \Delta W_{jk} \quad (2.16)$$

Masing masing dari unit tersembunyi Z_{jk} , diperbaiki bias dan penimbangannya

$$V_{ij}(\text{baru})=V_{ij}(\text{lama})\Delta V_{ij} \quad (2.17)$$

Langkah 9:

Periksa kondisi akhir iterasi (pemberhentian) pengujian iterasi dilakukan dengan menghitung nilai dari *error*, apakah batas error minimum telah tercapai dan apakah batas iterasi maksimum telah tercapai. Rumus pencarian nilai *error* sebagai berikut.

$$E = \frac{1}{x} \sum_{i=1}^x (T_k - Y_k)^2 \quad (2.18)$$

Daftar keterangan dari notasi *backpropagation* diatas dapat dilihat pada table 2.7 sebagai berikut.

Tabel 2.1 Keterangan Notasi *Backpropagation Networks*

Notasi	Keterangan
T_k	Pola keluaran target dari pelatihan.
X	Lapisan masukan
Z	Lapisan tersembunyi
Y	Lapisan keluaran
Z_{inj}	Keluaran untuk unit Z_j
Y_{ink}	Net masukan untuk unit Y_k
Z_j	Nilai aktifasi dari unit Z_j
W_{ok}	Nilai penimbang sambungan pada bias untuk unit Y_k
W_{jk}	Nilai penimbang sambungan dari Z ke unit Y_k
V_{ok}	Nilai penimbang sambungan pada bias untuk unit Z
V_{ij}	Nilai penimbang sambungan pada X untuk unit Z
∇W_{ik}	Nilai bobot perbaikan sambungan Y_k ke Z
ΔV_{ij}	Nilai bobot perbaikan sambungan Z ke X
∇W_{ok}	Nilai penimbang sambungan pada bias untuk unit Z
ΔV_{oj}	Nilai penimbang sambungan pada bias untuk unit X
Notasi	Keterangan
δ_k	Faktor pengaturan nilai penimbang sambungan pada lapisan keluaran
δ_z	Faktor pengaturan nilai penimbang sambungan pada lapisan tersembunyi

α	<i>learning rate</i>
E	Total galat

2.7.1.2 Pengujian *Backpropagation*

Setelah pelatihan, jaringan saraf *backpropagation* diaplikasikan dengan hanya menggunakan tahap perambatan maju dari algoritma pelatihan. Prosedur aplikasinya adalah sebagai berikut.

Langkah 0: Berikan inisialisasi penimbang (bobot akhir dari pelatihan).

Langkah 1: Untuk tiap vektor masukan, lakukan langkah 2 sampai 4.

Langkah 2: Masukan nilai aktivasi unit masukan X .

Langkah 3: Hitung fungsi lapisan tersembunyi Z beserta biasnya

$$Z_{inj} = V_{oj} + \sum_{i=1}^n X_i V_{ij} \quad (2.5)$$

$$Z_{inj} = f(Z_{inj}) \quad (2.6)$$

Langkah 4: Hitung fungsi lapisan output Y beserta biasnya

$$Y_{ink} = W_{ok} + \sum_{j=1}^p Z_j W_{kj} \quad (2.7)$$

$$Y_k = f(Y_{ink}) \quad (2.8)$$

Langkah 5: Jika hasil perhitungan output $Y_k \geq 0,5$ maka $Y_k = 1$, dan jika $Y_k < 0,5$ maka $Y_k = 0$.

2.7.1.2 Pengujian

Pengujian yang dilakukan dalam penelitian ini menggunakan standar pengukuran akurasi yang biasa digunakan dalam penelitian peringkasan teks diantaranya *recall*, *precision* dan *f-measure*.

1. *Recall*

Recall adalah tingkat keberhasilan dalam sebuah klasifikasi. Adapun rumus untuk menghitung *recall* dapat dilihat pada persamaan berikut:

$$r = \frac{\text{correct}}{\text{correct} + \text{missed}} \quad (2.19)$$

2. *Precision*

Precision badalah tingkat ketepatan dalam sebuah ringkasan. Adapun, rumus untuk menghitung *precision* dapat dilihat pada persamaan berikut.

$$p = \frac{\text{correct}}{\text{correct} + \text{wrong}} \quad (2.20)$$

3. *F-Measure*

F-Measure adalah perhitungan yang dihasilkan dari evaluasi *recall* dan *precision*. Nilai *recall* dan *precision* pada suatu keadaan dapat memiliki bobot yang berbeda. Adapun perhitungan *f-measure* dapat dilihat pada persamaan berikut.

$$f - \text{measure} = \frac{2 * \text{precision} + \text{recall}}{\text{precision} + \text{recall}} \quad (2.21)$$

2.8 Pemrograman Berorientasi Objek

Pemrograman berorientasi objek (*Object-Oriented Programming*) atau disebut juga sebagai OOP merupakan suatu model pemrograman yang berorientasi kepada objek, dimana semua data dan operasi didalamnya disatukan pada *class* atau objek. Konsep dasar OOP sebagai berikut:[10]

1. Objek

Objek didefinisikan sebagai konsep, abstraksi atau benda dengan batasan dan arti untuk suatu masalah[9].

2. Kelas (*class*)

Kelas (*class*) mendefinisikan sebuah tipe dari objek. Dalam kelas dapat mendeklarasikan variabel dan menciptakan objek. Kelas mempunyai anggota yang terdiri dari atribut dan metode (*method*) dimana atribut merupakan semua *field* identitas yang kita berikan pada suatu kelas, dan metode (*method*) dapat diartikan sebagai semua fungsi ataupun prosedur yang merupakan perilaku (*behaviour*) dari suatu kelas.

3. Metode (*Method*)

Metode (*method*) merupakan operasi atau metode yang dapat dipanggil oleh kelas. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi.\

4. Enkapsulasi (*Encapsulation*)

Enkapsulasi adalah metode untuk pembungkusan atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek.

5. Pewarisan (*Inheritance*)

Pewarisan merupakan pewarisan atribut dan metode dari sebuah kelas ke kelas lainnya. Konsep penurunan ini suatu kelas bisa diturunkan menjadi kelas baru dengan mewarisi sifat-sifat kelas orang tuanya. Kelas yang mewarisinya disebut *superclass* dan kelas yang diwarisinya disebut *subclass*.

2.9 UML

UML atau *Unified Modelling Language* adalah salah satu alat bantu dalam pengembangan sistem berorientasi objek. UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, *Object Modeling Technique* (OMT) dan *Object Oriented Software Engineering* (OOSE). UML terdiri dari serangkaian diagram memungkinkan bagi sistem analis untuk membuat cetak biru sistem yang komprehensif kepada klien[11].

2.8.1 Flowchart

Flowchart adalah bagan-bagan yang mempunyai alur yang menggambarkan langkah-langkah penyelesaian suatu masalah. Tujuan membuat flowchart yaitu untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, teratur, rapi dan jelas dengan menggunakan simbol-simbol standar.

2.8.2 Diagram Use Case

Diagram Use Case merupakan suatu gambaran apa saja aktifitas yang ada pada suatu sistem dari sudut pandang pengamatan luar. Diagram use case dekat kaitannya dengan kejadian-kejadian (Scenario) apa saja yang terjadi ketika seseorang berinteraksi dengan sistem. Diagram Use Case berguna dalam 3 hal [10]:

1. Menjelaskan fasilitas yang ada
2. Komunikasi dengan klien
3. Membuat test dari kasus-kasus umum

2.8.4 Diagram Activity

Diagram activity menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem, proses bisnis atau menu yang ada pada perangkat lunak. Diagram activity berfokus pada aktifitas-aktifitas yang terjadi terkait dalam suatu proses tunggal[11]. Diagram activity dapat dibagi menjadi beberapa jalur kelompok yang bertujuan untuk menunjukkan objek mana yang bertanggung jawab untuk suatu aktivitas.

2.8.5 Diagram Sequence

Diagram Sequence menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirim dan diterima antar objek[11]. Diagram sequence merupakan salah satu diagram yang menjelaskan bagaimana suatu operasi dilakukan. Diagram diatur berdasarkan waktu.

2.9 Bahasa Pemrograman

Bahasa pemrograman adalah instruksi standar untuk memerintah komputer. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Fungsi bahasa pemrograman yaitu memerintah \komputer untuk mengolah data sesuai dengan alur berpikir yang kita inginkan. Keluaran dari bahasa pemrograman tersebut berupa program/aplikasi. Berikut adalah beberapa beberapa contoh bahasa pemrograman yaitu C, C++, C#, Pascal, Java, JavaScript, PHP, SQL dan Python. Bahasa Pemrograman yang digunakan pada penelitian ini adalah python.

2.9.1 Python

Python merupakan salah satu bahasa pemrograman tinggi yang dapat melakukan eksekusi sejumlah instruksi secara langsung (interpretatif) dengan metode orientasi objek (OOP). Python sering digunakan di bidang pengembangan web, protokol internet dan infrastruktur server. Python versi 3.7 digunakan dalam penelitian ini.

2.9.1 Spyder



Gambar 2.4 Logo Spyder (<https://www.spyder-ide.org>)

Spyder (*The Scientific PYTHON Development EnviRonment*) merupakan salah satu IDE untuk bahasa pemrograman python dalam melakukan pemrograman saintifik. Beberapa fitur yang ada yaitu *Interactive console*, *Document viewer*, *Variable explorer*, *Find in filfes*, *File explorer*, dan *History log*. Spyder digunakan sebagai *tool* dalam membangun aplikasi dalam penelitian ini.

