# BAB 2
# TINJAUAN PUSTAKA

Berdasarkan latar belakang dan rumusan masalah berikut adalah landasan teori yang dibutuhkan saat penelitian :

## 2.1 *Sentimen Analysis*

*Sentiment* Analysis adalah salah satu teknik dalam mengekstrak informasi berupa pandangan (sentimen) seseorang terhadap suatu isu atau kejadian. Analisis Sentimen dapat digunakan untuk mengungkap opini publik terhadap suatu isu, kepuasan pelayanan, kebijakan, cyber bulliying, memklasifikasi harga saham, dan analisis pesaing berdasarkan data tekstual. Namun, fenomena pertumbuhan data yang terjadi secara eksponensial menjadi tantangan baru dalam proses analisis sentimen. Pendekatakan kovensional bukan lagi jawaban tepat untuk mengungkap dan menentukan jenis sentimen dalam data tekstual. Memperkerjakan manusia untuk mengklasifikasikan jenis sentimen dari suatu kumpulan data tekstual yang sangat besar dan beragam tentu akan membutuhkan waktu dan biaya yang tidak sedikit. Bayangkan jika kita memiliki 100.000 ribu tweet per hari yang harus di tentukan satu per satu jenis sentimen nya, pasti bukan hanya membutuhkan waktu yang lama, tapi juga membutuhkan sumber daya yang sangat besar. Oleh karena itu, dibutuhkan sebuah teknik baru dalam analisis sentimen yang dapat secara otomatis mengekstrak informasi dari data secara cepat dan bertanggungjawab[8].

*Machine Learning* digunakan sebagai tools untuk membuat “robot” yang mampu mengklasifikasikan jenis sentimen dalam data tekstual. Umumnya, terdapat 5 langkah utama dalam melakukan analisis sentimen dengan menggunakan Machine Leaning, yaitu:

1. Data
2. Preprocessing Data

3. *Feature Extraction*
4. Modelling
5. *evaluation* Data

Langkah awal yang dibutuhkan adalah data. Tanpa adanya data, maka tidak akan mungkin bisa melakukan apapun dengan *Machine Learning*. Data yang dimaksud pada tulisan ini adalah data tekstual (tweet, dokumen, buku, kalimat) yang sudah dilabeli oleh annotator. Annotator adalah orang yang bertanggungjawab memberikan label (positif, negatif, netral) pada masing-masing data. Melakukan pelabelan terhadap data adalah usaha pertama yang besar untuk melakukan analisis sentimen.

*Sentiment* Analysis adalah penambangan kontekstual teks yang mengidentifikasi dan mengekstrak informasi subjektif dalam sumber, dan membantu para pembisnis untuk memahami sentimen sosial dari merek, produk atau layanan mereka saat memantau percakapan online. Secara umum, sentimen analisis terbagi menjadi 2 kategori besar yaitu :

1. Coarse-grained *sentiment* analisis

   Coarse-grained *sentiment* analisis melakukan proses analisis pada level dokumen. Pengklasifikasian berorientasi pada sebuah dokumen secara keseluruhan, yaitu positif, netral dan negative.
2. Fined-grained *sentiment* analisis

   Sedangkan fined-grained *sentiment* analisis melakukan proses analisis sebuah kalimat. Contoh sebuah fined-grained *sentiment* analisis :

   1. Saya tidak suka ikan, klasifikasi negatif
   2. Hotel itu sangat indah sekali, klasifikasi positif

### *2.1.1 Metode Sentimen Analysis*

*Machine Learning* Approach memerlukan *dataset* untuk digunakan sebagai data *training*. Oleh karena itu, dibutuhkan effort untuk mengumpulkan dan melakukan *class* tag pada sampel *dataset* tersebut, selain itu proses *training* juga membutuhkan waktu [9]. Akurasi dari pendekatan klasifikasi *Machine Learning* sangat baik, akan

tetapi performa klasifikasinya domain dependent terhadap *dataset* yang digunakan pada saat *training* [10].

Metode-metode yang masuk ke dalam kategori ini adalah sebagai berikut:

1. Naïve Bayes
2. Maximum Entropy
3. SVM
4. Neural Network

*2.1.2 Knowledge-Based Method Approach*

Knowledge-Based adalah pendekatan *sentiment* Analysis pada word level, dimana entitas yang diproses adalah kata. Metode-metode yang masuk di dalam pendekatan ini adalah sebagai berikut:

1. Lexicon-Based PMI (Pointwise Mutual Information) Knowledge Based bergantung pada dictionary atau kamus lexicon yang digunakan untuk melakukan penilaian terhadap fitur yang didapat.
2. Hybrid Approach Pendekatan ini menggabungkan knowledge-based approach dan *Machine Learning* approach. Beberapa penelitian sukses mengaplikasikan keduanya secara bersamaan.

## 2.2 Preprocessing Data

Biasanya data yang kita dapatkan dari flat files atau database berupa data mentah. Algoritma *Machine Learning classification* bekerja dengan data yang akan diformat dengan cara tertentu sebelum mereka memulai proses *training*. Untuk menyiapkan data untuk konsumsi oleh algoritma *Machine Learning*, harus dilakukan processing terlebih dahulu dan mengonversinya menjadi format yang tepat. dalam penerapannya salah satu fungsinya yaitu scikit-learn untuk preprocessing data, dalam library scikit-learn ada banyak fungsi-fungsi yang tersedia[11].

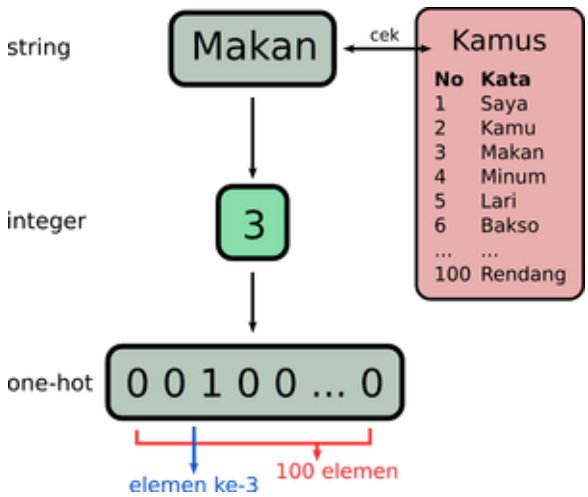
1. Binarization, proses binarization adalah ketika kita ingin mengubah variabel numerik kedalam nilai boolean (0 dan 1)

2. Mean Removal, Mean removal adalah cara umum dalam teknik preprocessing yang digunakan dalam *Machine Learning*, menghilangkan rata-rata biasanya sangat berguna dari variabel, jadi variabel berada ditengah tengah pada angka 0. kita melakukannya untuk menghilangkan bias dari variable.
3. Scaling, biasanya dalam *dataset* yang masih mentah, beberapa variabel memiliki nilai yang sangat bervariasi dan random, jadi sangat penting untuk di scale feature feature tersebut, in my prespective feature-feature tersebut nilainya sangat besar atau kecil hanya karena the nature of measurements.
4. Normalization, biasanya dalam preprocessing, proses normalisasi untuk memodifikasi nilai dalam variabel sehingga dapat mengukurnya dalam skala umum. Dalam *Machine Learning*, kita menggunakan berbagai bentuk normalisasi. Beberapa bentuk normalisasi yang paling umum bertujuan untuk mengubah nilai-nilai sehingga jumlahnya menjadi 1. Normalisasi L1 (di library scikit-learn), yang mengacu pada Penyimpangan Absolut Terkecil, bekerja dengan memastikan bahwa jumlah nilai absolut adalah 1 dalam setiap baris. Normalisasi L2, yang mengacu pada kuadrat terkecil, bekerja dengan memastikan bahwa jumlah kuadrat adalah 1. Secara umum, teknik normalisasi L1 dianggap lebih kuat daripada teknik normalisasi L2. Teknik normalisasi L1 kuat karena tahan terhadap outlier dalam data. Seringkali, data cenderung mengandung outlier dan kita tidak bisa berbuat apa-apa. Kami ingin menggunakan teknik yang dapat dengan aman dan efektif mengabaikannya selama perhitungan. Jika kita memecahkan masalah di mana outlier itu penting, maka mungkin normalisasi L2 menjadi pilihan yang lebih baik
5. Label encoding, Ketika melakukan klasifikasi, biasanya berurusan dengan banyak label. Label-label ini bisa dalam bentuk kata-kata, angka, atau sesuatu yang lain. Fungsi pembelajaran mesin dalam sklearn mengharapkan mereka menjadi angka. Jadi jika mereka sudah menjadi nomor, maka kita dapat menggunakannya secara langsung untuk memulai pelatihan. Tetapi ini tidak biasanya terjadi. Di dunia nyata, label dibuat dalam bentuk kata-kata, karena

kata-kata dapat dibaca manusia. kita melabeli data *training* dengan kata-kata sehingga pemetaan dapat dilacak. Untuk mengonversi label kata menjadi angka, kita perlu menggunakan pembuat label encoding. label encoding mengacu pada proses transformasi label kata menjadi bentuk numerik. dalam hal regresi jika memuat varibel kategori dan nilainya tidak bisa di faktorisasi dalam bentuk tingkatan, dilakukan proses dummy, setiap nilai didalam variabel itu menjadi variabel lain.

## 2.3 Processing Data Menggunakan *Word Embedding*

*Word Embedding* mudahnya adalah istilah yang digunakan untuk teknik mengubah sebuah kata menjadi sebuah vektor atau array yang terdiri dari kumpulan angka. Ketika kita akan membuat model *Machine Learning* yang menerima *input* sebuah teks, tentu *Machine Learning* tidak bisa langsung menerima mentah-mentah teks yang kita miliki. Cara "tradisional" untuk membaca teks tersebut adalah sebagai berikut: Buat kamus kata dengan cara mendaftar semua kata yang ada di *dataset* Setiap menerima sebuah string kata, string tersebut diubah menjadi sebuah integer dengan memberinya nomor. Penomoran ini bisa ditentukan berdasarkan urutan di kamus kata yang kita miliki. Misalnya pada ilustrasi di bawah, string "Makan" menjadi angka 3, string "Lari" menjadi angka 5, dst. Angka-angka tersebut kita ubah lagi menjadi sebuah vektor (array 1 dimensi) yang memiliki panjang sepanjang banyak kata yang kita miliki di kamus. Array tersebut hanya akan bernilai 1 atau 0 (disebut *one hot encoding*). Nilai 1 diposisikan pada indeks yang merupakan nomor kata tersebut sedangkan elemen lainnya bernilai 0. Contohnya untuk kata "makan", dengan banyak kosakata yang kita miliki adalah 100 kata, maka dari kata tersebut kita akan memperoleh sebuah vektor dengan panjang 100 yang berisi 0 semua kecuali pada posisi ke 3 yang bernilai 1[12].




**Gambar 2. 1Ilustrasi Cara Kerja Word Embedding**

Secara sederhana, Word Embedding adalah proses konversi sebuat teks menjadi angka. Sebagian besar algoritma Machine Learning dan arsitektur deep learning tidak mampu melakukan proses analisis pada input data berupa strings atau teks, sehingga membutuhkan angka sebagai input. Contoh sederhana merubah kata menjadi vektor angka. Misal diberikan kalimat berikut: "Word Embedding are Word Converted into numbers." Sebuah kamus akan berisi list seluruh kata yang unique. Sehingga kamus yang terbentuk adalah: ["Word","Embeddings","are","Converted","into","numbers"]. Menggunakan metode one-hot encoding akan menghasilkan vektor dimana 1 merepresentasikan posisi kata tersebut berada, dan 0 untuk kata lainnya. Vektor representasi dari kata "numbers" mengacu pada format kamus diatas adalah [0,0,0,0,0,1] dan kata "Converted" adalah [0,0,0,1,0,0].

## 2.4 *Convolutional Neural Network*

Jaringan saraf konvolusi adalah salah satu struktur jaringan yang representatif dalam pembelajaran mendalam,dan telah menjadi hotspot di bidang analisis dan pengenalan gambar [13].

Berbagi berat struktur jaringan membuatnya lebih mirip dengan jaringan saraf biologis, yang mengurangi kompleksitas model jaringan dan mengurangi jumlah bobot.

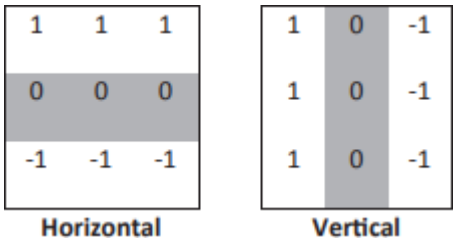
Keuntungan dari metode ini adalah bahwa *input* jaringan lebih jelas ketika *input* multidimensi, sehingga gambar bias masukan langsung ke jaringan, yang menghindari ekstraksi fitur yang kompleks dan rekonstruksi data di algoritma pengakuan tradisional[13]. Jaringan produk adalah perceptron multilayer yang dirancang khusus untuk mengenali bentuk dua dimensi yang sangat berbeda dengan translasi, skala, kemiringan, atau bentuk lain dari deformasi[13].

Kerangka ide penting dari jaringan saraf konvolusi adalah perceptron area lokal; berbagi berat; pengambilan sampel spasial. Tiga karakteristik CNN adalah bahwa distorsi data *input* di ruang sangat kuat. CNN umumnya menggunakan lapisan berbelit-belit dan lapisan sampel secara berurutan mengatur, yaitu, lapisan lapisan berbelit-belit yang terhubung ke lapisan pengambilan sampel, lapisan pengambilan sampel diikuti oleh konvolusi.

Lapisan konvolusi ini untuk mengekstraksi fitur, dan kemudian dikombinasikan untuk membentuk fitur yang lebih abstrak, akhirnya, membentuk deskripsi karakteristik objek gambar. CNN juga dapat diikuti oleh lapisan yang sepenuhnya terhubung[13]. Ketika mulai mendesain CNN, langkah pertama adalah menentukan jumlah lapisan dan jumlah filter dalam setiap lapisan. Biasanya, CNN memiliki lebih dari sekedar lapisan konvolusi (singkatnya), tetapi kita akan mulai hanya menggunakan lapisan ini. CNN dapat menguji diri sendiri untuk menyelesaikannya masalah seperti itu. Pertama, lapisan konvolusi menyelidiki blok bangunan dari struktur bentuk yang kita cari. Jadi, pertanyaan pertama untuk bertanya pada diri sendiri adalah apa yang istimewa dari segi empat dibandingkan dengan segitiga dan lingkaran [13].

Bentuk persegi panjang memiliki empat tepi, dua vertikal dan dua horisontal. Kami dapat mengambil manfaat dari informasi tersebut, tetapi juga perhatikan bahwa properti yang ada dalam persegi panjang seharusnya tidak ada dalam bentuk lain [6]. Bentuk lain sudah memiliki sifat yang berbeda. Tidak satu pun dari dua bentuk lainnya memiliki dua tepi horisontal dan dua tepi vertikal. Ini bagus. Pertanyaan selanjutnya

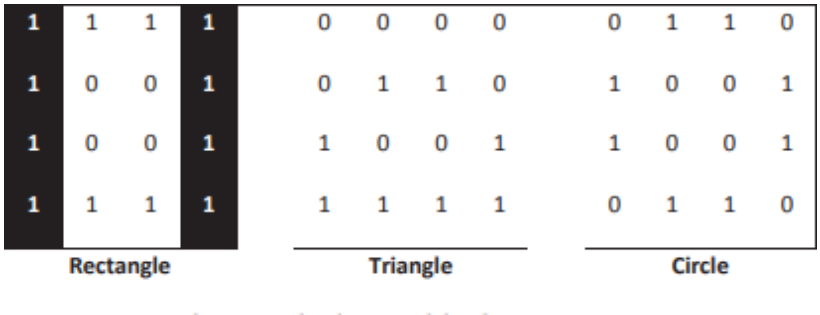
adalah bagaimana membuat lapisan konvolusi mengenali keberadaan tepi [13]. Ingatlah bahwa CNN memulai dengan mengenali elemen-elemen individual dari bentuk dan kemudian menghubungkan elemen-elemen ini bersama-sama. Jadi, kami tidak mencari untuk menemukan empat tepi atau mencari untuk menemukan dua tepi paralel paralel dan dua tepi horisontal paralel, tetapi untuk mengenali setiap tepi vertikal atau horisontal. Jadi, pertanyaannya menjadi lebih spesifik. Bagaimana kita bisa mengenali tepi vertikal atau horisontal? Ini bisa dilakukan dengan menggunakan gradient[13]. Lapisan pertama akan memiliki filter yang mencari tepi horisontal dan filter lain untuk tepi vertikal. Filter ini ditunjukkan pada Gambar 2.2 sebagai matriks 3×3. Jadi, kita tahu berapa banyak filter untuk digunakan di lapisan conv pertamadan juga apa filter ini. Ukuran 3 × 3 dipilih untuk filter karena ini adalah ukuran yang baik di mana struktur tepi horisontal dan vertikal jelas.



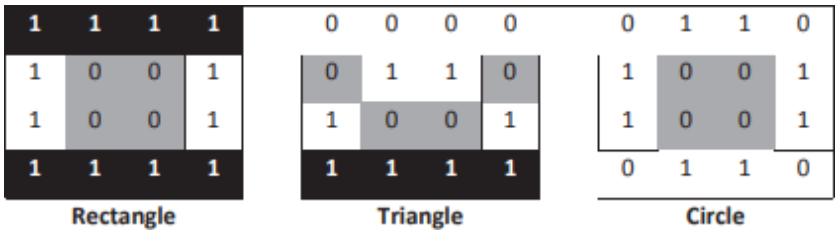

**Gambar 2. 2Filter untuk mengenali tepi horisontal dan vertikal ukuran 3 × 3**

Setelah menerapkan filter ini di atas matriks pada Gambar 2.1, lapisan konvolusi akan dapat mengenali tepi i pada Gambar 2.2 dan tepi horisontal pada Gambar 2.3. [6]. Lapisan ini mampu mengenali tepi horisontal dan vertikal dalam persegi panjang. Itu juga mengenali tepi horisontal di dasar segitiga. Tetapi tidak ada tepi dalam lingkaran. Pada saat ini, CNN memiliki dua kandidat untuk menjadi persegi panjang yang merupakan bentuk yang memiliki setidaknya satu sisi. Meskipun yakin bahwa bentuk ketiga tidak bisa persegi panjang CNN harus menyebarkannya ke lapisan lain sampai membuat

keputusan di lapisan terakhir, karena menggunakan dua filter dalam lapisan konv pertama, ini menghasilkan dua *output*, satu untuk setiap filter .

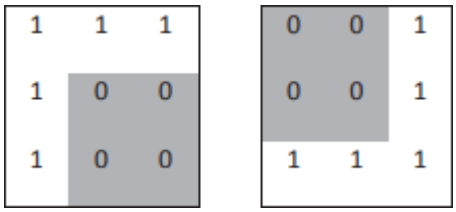



**Gambar 2. 3Tepi vertikal yang diakui berwarna hitam**



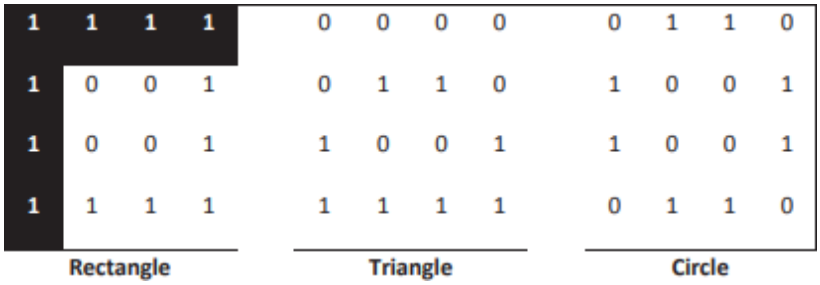

**Gambar 2. 4Tepi horisontal yang dikenali berwarna hitam**

Lapisan konvolusi berikutnya akan menerima hasil dari lapisan konvolusi pertama dan melanjutkan berdasarkan itu. Mari kita ulangi pertanyaan yang sama yang ditanyakan di lapisan pertama. Berapa jumlah filter yang digunakan dan apa strukturnya? Berdasarkan struktur persegi panjang, kami menemukan bahwa setiap tepi horisontal terhubung ke tepi vertikal. Karena ada dua tepi horisontal, ini membutuhkan penggunaan dua filter pada Gambar 2.4 ukuran 3 × 3[13].

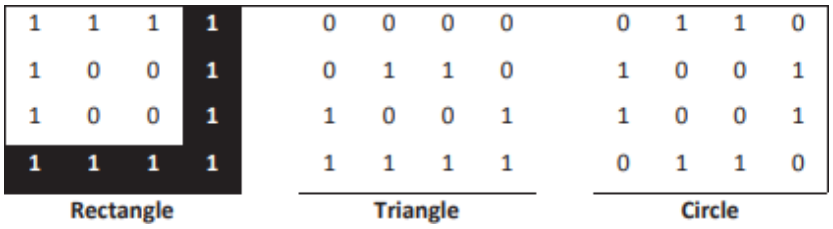



**Gambar 2. 5Filter untuk mengenali terhubungnya horisontal dan vertical pada tepi ukuran 3 × 3**

Setelah menerapkan filter tersebut ke hasil lapisan konvolusi 1, hasil filter yang digunakan pada lapisan konvolusi kedua ditunjukkan masing-masing pada Gambar 2.5 dan Gambar 2.6. Mengenai persegi panjang, filter dapat menemukan dua tepi yang dibutuhkan dan menghubungkannya bersama. Dalam segitiga, hanya ada satu tepi horisontal tanpa tepi vertikal terhubung dengannya. Akibatnya, tidak ada *output* positif untuk segitiga [13].




**Gambar 2. 6Hasil filter pertama di lapisan kedua berwarna hitam**




**Gambar 2. 7Hasil filter kedua di lapisan kedua berwarna hitam**

## 2.5 Tensorflow

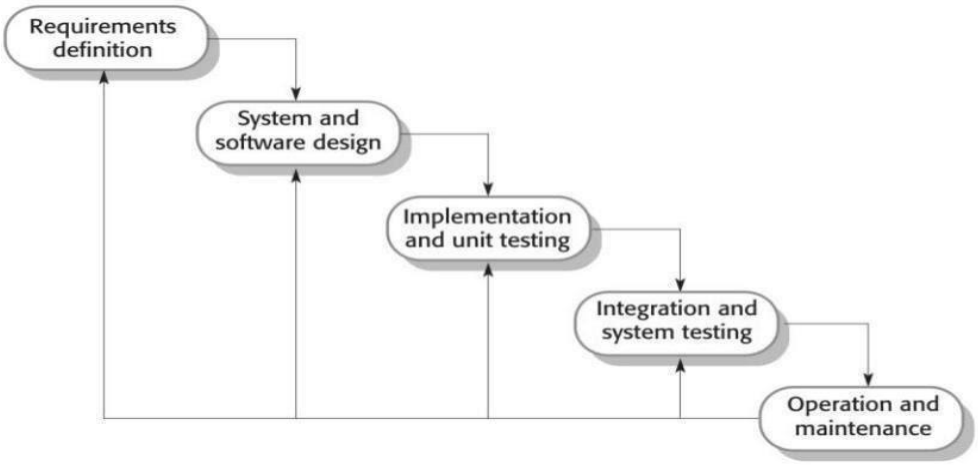
Nama Tensorflow terdiri dari dua kata. Yang pertama adalah tensor yang merupakan unit data yang digunakan TF dalam komputasinya. Kata kedua adalah flow yang mencerminkan bahwa ia menggunakan paradigma aliran data. Akibatnya, TF membangun grafik komputasi yang terdiri dari data yang direpresentasikan sebagai tensor dan operasi yang diterapkan padanya. Untuk membuat hal-hal lebih mudah dipahami, ingatlah bahwa daripada menggunakan variabel dan metode, TF menggunakan tensor dan operasi [6] . Berikut ini beberapa keuntungan menggunakan dataflow dengan TF:

1. Paralelisme: Lebih mudah untuk mengidentifikasi operasi yang dapat dieksekusi secara paralel.
2. Eksekusi Terdistribusi: Program TF dapat dipartisi di beberapa perangkat (CPU, GPU, dan Unit Pemrosesan TF [TPU]). TF sendiri menangani pekerjaan yang diperlukan untuk komunikasi dan kerja sama antar perangkat.

3. Portabilitas: Grafik aliran data adalah representasi kode model yang tidak tergantung bahasa. Grafik aliran data bias dibuat menggunakan Python, disimpan, dan kemudian dipulihkan dalam program C ++ [6].

## 2.6 Modelling Software Waterfall

Metode ini juga disebut “siklus hidup klasik” atau yang sekarang disebut model air terjun. Metode ini adalah metode yang pertama kali diangkat pada tahun 1970 sehingga sering dianggap terlalu kuno, tetapi metode ini sering digunakan oleh para teknisi di Rekayasa Perangkat Lunak (SE)[9]. Metode ini mengambil pendekatan yang sistematis dan tersusun rapi seperti air terjun mulai dari tingkat kebutuhan sistem kemudian berlanjut ke tahapan analisis, desain, coding, pengujian / verifikasi, dan pemeliharaan. Disebut air terjun karena seperti air terjun yang jatuh satu demi satu sehingga penyelesaian tahap sebelumnya kemudian dapat dilanjutkan ke tahap berikutnya dan berjalan-urut. langkah-langkah dalam model air terjun dapat dilihat pada gambar 2.7 . :




**Gambar 2. 8Metode Waterfall**

Menurut Tahapan-tahapan model air terjun adalah sebagai berikut:

1. *Requirement Analysis And Definition*
2. *Sistem And Software Design*
3. *Implementation And Unit testing*
4. *Integration And Sistem testing*
5. *Operation And Maintenance*

Kelebihan menggunakan metode air terjun (*waterfall*) adalah metode ini memungkinkan untuk departementalisasi dan kontrol. proses pengembangan model fase one by one, sehingga meminimalis kesalahan yang mungkin akan terjadi. Pengembangan bergerak dari konsep, yaitu melalui desain, implementasi, pengujian, instalasi, penyelesaian masalah, dan berakhir di operasi dan pemeliharaan.

Kekurangan menggunakan metode *waterfall* adalah metode ini tidak memungkinkan untuk banyak revisi jika terjadi kesalahan dalam prosesnya. Karena setelah aplikasi ini dalam tahap pengujian, sulit untuk kembali lagi dan mengubah sesuatu yang tidak terdokumentasi dengan baik dalam tahap konsep sebelumnya.

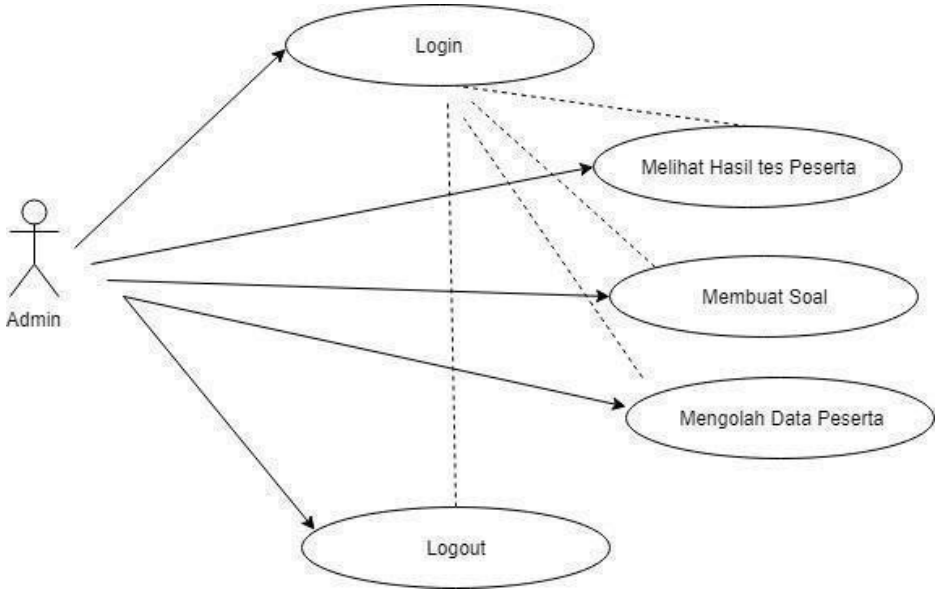
## 2.7. Pengertian *Unified Modelling Languange*

UML (*Unified Modeling Language*) adalah salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek[16]

UML sendiri terdiri atas pengelompokkan diagram-diagram sistem menurut aspek atau sudut pandang tertentu. Diagram adalah yang menggambarkan permasalahan maupun solusi dari permasalahan suatu model. UML mempunyai 5 diagram, yang akan dibahas dalam penulisan skripsi kali ini yaitu.

1. *use case* Diagram

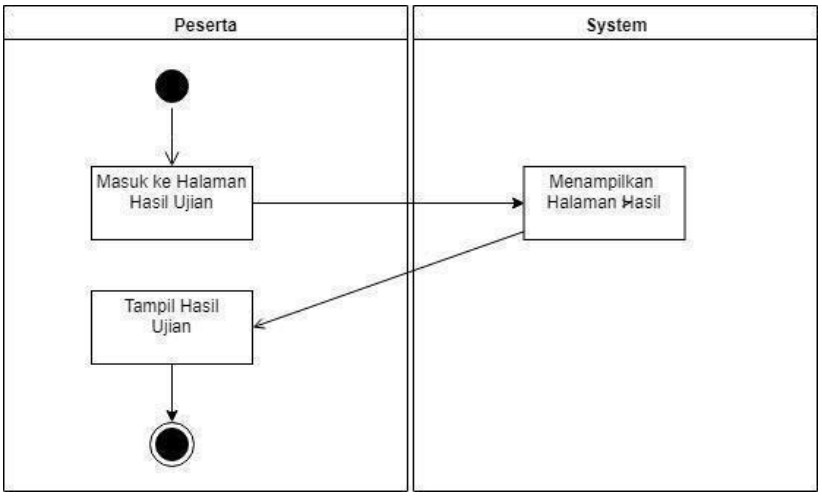
Diagram *use case* menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar, yang menjadi persoalan itu apa yang dilakukan bukan bagaimana melakukannya. Contoh *use case* pada Gambar 2.9.




**Gambar 2. 9Contoh use case Diagram**

1. *Activity* Diagram

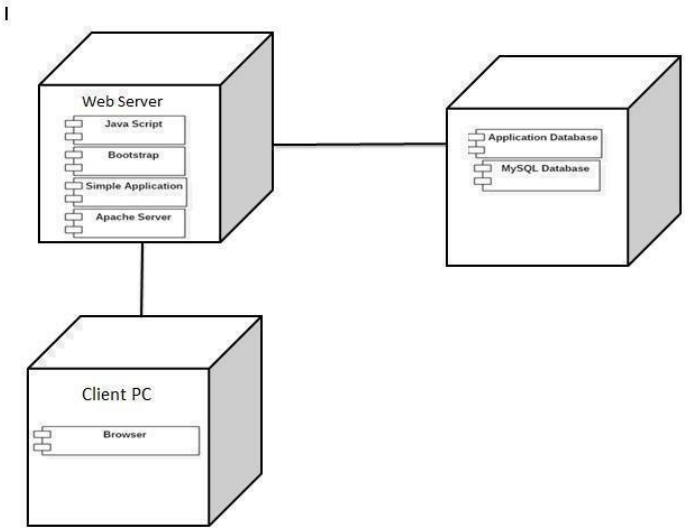
Diagram *activity* berfokus pada aktifitas-aktifitas yang terjadi yang terkait dalam suatu proses tunggal. Jadi dengan kata lain, diagram ini menunjukkan bagaimana aktifitas-aktifitas tersebut bergantung satu sama lain. Contoh *activity diagram* pada Gambar 2.10.




**Gambar 2. 10Contoh activity Diagram**

1. Deployment Diagram

Diagram Deployment menerangkan bahwa konfigurasi fisik *software* dan *hardware*. Contoh pada Gambar 2.11.




**Gambar 2. 11Contoh Deployment Diagram**

## 2.8. Coronavirus (Covid19)

Pada awal tahun 2020, dunia dikejutkan dengan mewabahnya pneumonia baru yang bermula dari Wuhan, Provinsi Hubei yang kemudian menyebar dengan cepat ke lebih dari 190 negara dan teritori. Wabah ini diberi nama coronavirus disease 2019 (COVID-19) yang disebabkan oleh *Severe Acute Respiratory Syndrome Coronavirus*-2 (SARS-CoV-2). Sumber penularan kasus ini pertama dikaitkan dengan pasar ikan di Wuhan. Penyebaran penyakit ini telah memberikan dampak luas secara sosial dan ekonomi. Coronavirus adalah suatu kelompok virus yang dapat menyebabkan penyakit pada hewan atau manusia. Beberapa jenis coronavirus diketahui menyebabkan infeksi saluran nafas pada manusia mulai dari batuk pilek hingga yang lebih serius seperti Middle East Respiratory Syndrome (MERS) dan Severe Acute Respiratory Syndrome (SARS). Coronavirus jenis baru yang ditemukan menyebabkan penyakit COVID-19.