BAB 2
TINJAUAN PUSTAKA

2.1 Profil Instansi

Pada bagian ini berisi uraian mengenai keadaan yang ada pada tempat penelitian meliputi informasi umum instansi, visi dan misi beserta struktur organisasinya.

2.1.1. Jabar Digital Service

Salah satu visi Gubernur Ridwan Kamil dalam pemerintahannya adalah menjadikan Jawa Barat sebagai provinsi digital. Beliau menekankan kekhawatirannya atas apa yang beliau pikir sebagai salah satu masalah provinsi yang paling mengkhawatirkan yaitu, kesenjangan digital antara desa dan kota. Kurangnya infrastruktur yang layak di daerah pedesaan menjadi kendala penduduk desa untuk menuai manfaat yang ditawarkan oleh teknologi digital [6].

Untuk memecahkan masalah tersebut, maka muncullah dorongan untuk membentuk Jabar *Digital Service* (JDS) atau Unit Layanan Digital, Data, dan Informasi Geospasial Jawa Barat. Sebuah unit teknis di bawah Dinas Komunikasi dan Informatika Jawa Barat, JDS bertujuan untuk mempersempit kesenjangan digital, meningkatkan efisiensi dan akurasi pembuatan kebijakan berdasarkan data dan teknologi, dan merevolusi dengan teknologi tata kelola dan mata pencaharian warga di Jawa Barat [6].

2.1.2. Logo Instansi

Logo merupakan sebuah lambang yang dimiliki oleh setiap perusahaan atau instansi. Pembuatan logo dimaksudkan untuk merepresentasikan identitas suatu perusahaan yang mencerminkan jiwa, visi dan misi suatu perusahaan/instansi. Logo Jabar Digital Service dapat dilihat pada Gambar 2.1.1.

Gambar 2.1.1 Logo Jabar *Digital Service*

2.1.3. Visi

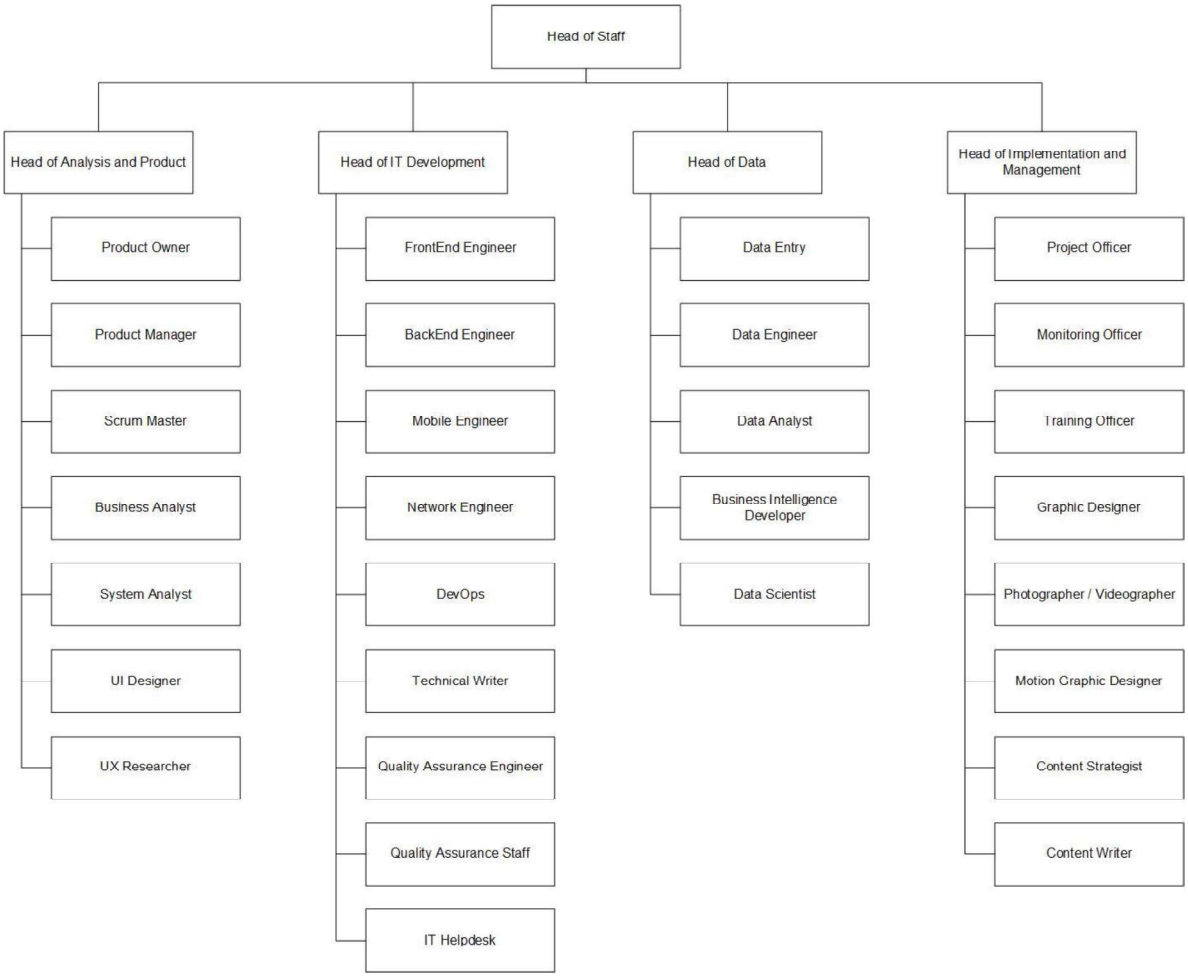
Terwujudnya Jawa Barat sebagai provinsi digital yang berbasis data dan teknologi, dalam mendukung pelayanan dan pengambilan kebijakan publik yang responsif, adaptif, dan inovatif.

2.1.4. Misi

1. Menjadikan data sebagai pendukung pengambilan keputusan dan perumusan kebijakan.
2. Mentransformasi layanan publik melalui pemanfaatan teknologi digital.
3. Mengakselerasi pembentukan ekosistem inovasi digital untuk kesejahteraan masyarakat.

2.1.5. Struktur Organisasi

Struktur organisasi pada suatu perusahaan sangatlah penting, karena dari struktur organisasi tersebut kita dapat melihat dan membedakan antara satu bidang dengan bidang yang lainnya.Struktur organisasi Jabar Digital Service dapat dilihat pada Gambar 2.1.2.



Gambar 2.1.2 Struktur Organisasi Jabar *Digital Service*

2.2 Landasan Teori

Pada bagian ini berisikan teori-teori yang berkaitan dengan topik penelitian yang digunakan sebagai acuan pada penelitian ini. Adapun teori-teori yang digunakan adalah sebagai berikut.

2.2.1. Visualisasi Data

Visualisasi data adalah proses penyajian data dalam bentuk grafik yang membuat informasi mudah dimengerti, hal ini membantu menjelaskan tentang fakta dan menentukan arah tindakan [7]. Definisi visualisasi data menjelaskan tentang pentingnya data dengan menempatkan data dalam konteks visual. Hal ini melibatkan penciptaan dan studi representasi visual dari data yang dikenal sebagai informasi. Visualisasi data memungkinkan pengguna untuk memperoleh

pengetahuan yang lebih banyak mengenai data mentah yang didapatkan dari berbagai sumber. Visualisasi dapat dilakukan dengan menggunakan *dashboard*, di mana teks, pola, dan korelasi yang tidak terdeteksi dapat dengan mudah divisualisasikan dengan menggunakan perangkat lunak visualisasi.

Visualisasi data tidak hanya mengubah data menjadi grafik visual, akan tetapi visualisasi data juga memerlukan perencanaan. Setiap jenis data memerlukan teknik visualisasi yang sesuai berdasarkan kebutuhannya. Berdasarkan tingkat kompleksitas data, untuk menghasilkan solusi yang berharga perlu melibatkan berbagai disiplin ilmu, seperti statistika, *data mining*, desain grafis, dan *information visualization* [5].

2.2.1.1. Proses Visualisasi Data

Proses memahami data dimulai dari beberapa pertanyaan, tidak semerta-merta dijawab begitu saja, akan tetapi terdapat langkah-langkah dalam menjawab pertanyaan berdasarkan data. langkah-langkah tersebut adalah sebagai berikut [5]:

1. *Acquire*

   Tahap ini adalah proses pengumpulan data dari berbagai sumber, baik dari file penyimpanan atau sumber melalui jaringan. Tahap Acquire hanya berfokus pada bagaimana data didapatkan, jika produk akhir akan didistribusikan melalu internet maka, data yang ada harus memiliki struktur dan dapat disimpan pada sebuah server.
2. *Parse*

   Tahap ini adalah proses penyesuian data kedalam sebuah format yang telah ditentukan yang kemudian akan dikategorikan kedalam beberapa kategori, agar data bapat dibaca dan bisa dibedakan satu dengan yang lainnya.
3. *Filter*

   Pada tahap ini adalah proses seleksi data dengan menghapus data yang tidak diperlukan. Beberapa data yang terdapat pada berkas, mungkin perlu diterjemahkan ke dalam model matematika atau dilakukan normalisasi terlebih dahulu.
4. *Mine*

Pada tahap ini adalah proses penerapan metode disiplin ilmu statistika dan data mining sebagai jalan untuk mencari pola atau dijabarkan pada konteks matematis.

5. *Represent*

   Pada tahap ini adalah proses pengubahan data dalam bentuk visual seperti bar *graph, tree, atau tree*. Tahap Represent menunjukkan bentuk dasar data yang akan diambil. Tahap ini merupakan tahap yang sangat penting dalam visualisasi data. Pemilihan model visualisasi yang tepat akan mempengaruhi kualitas dari produk yang dihasilkan

6. *Refine*

   Pada tahap ini adalah proses meningkatkan hasil representasi agar terlihat lebih menarik. Graphic design lebih banyak terlibat pada tahap ini. Poin-poin yang cukup penting pada visual grafik dibandingkan denga poin lainnya diberikan pembeda agar data mudah dibaca.

7. *Interact*

   Pada tahap ini adalah proses menambahkan metode untuk manipulasi data atau mengendalikan fitur yang terlihat dengan kata lain data bisa ditampilkan sesuai kehendak pengguna. Contoh interaksi antara pengguna dan data seperti *zoom-in, zoom-out*, merubah rentang data, melakukan *filtering*, dll.

2.2.1.2. Tipe-tipe Visualisasi Data

Tujuan visualisasi adalah untuk membantu pemahaman manusia terhadap data dengan memaksimalkan sistem penglihatan manusia yang bisa membedakan pattren, spot the trends, dan identifikasi outlier [8]. Tantangan dari visualisasi data adalah bagaimana membuat visualisasi yang effektif, menarik, dan tepat terhadap data yang dipakai [8]. Terdapat tujuh hal yang harus dipenuhi dalam melakukan abstraksi tingkat tinggi (*high-level abstraction*), semakin banyak hal yang disembunyikan, semakin banyak juga langkah-langkah yang harus dipenuhi, langkah tersebut yaitu [9] :

1. *Overview*: Melihat gambaran dari keseluruhan data.

2. *Zoom*: Memperbesar item yang terlihat menarik.
3. *Filter*: Melakukan penyaringan terhadap item yang dirasa kurang menarik.
4. *Details-on-demand*: Pilih satu item dari grup tertentu dan dapat melihat detail kapan saja.
5. *Relate*: Lihat relasi dari setiap item.
6. *History*: Dapat mengulang kembali atau kembali ke aksi sebelumnya.
7. *Extract*: Dapat melakukan ekstraksi dari parameter yang diberikan.

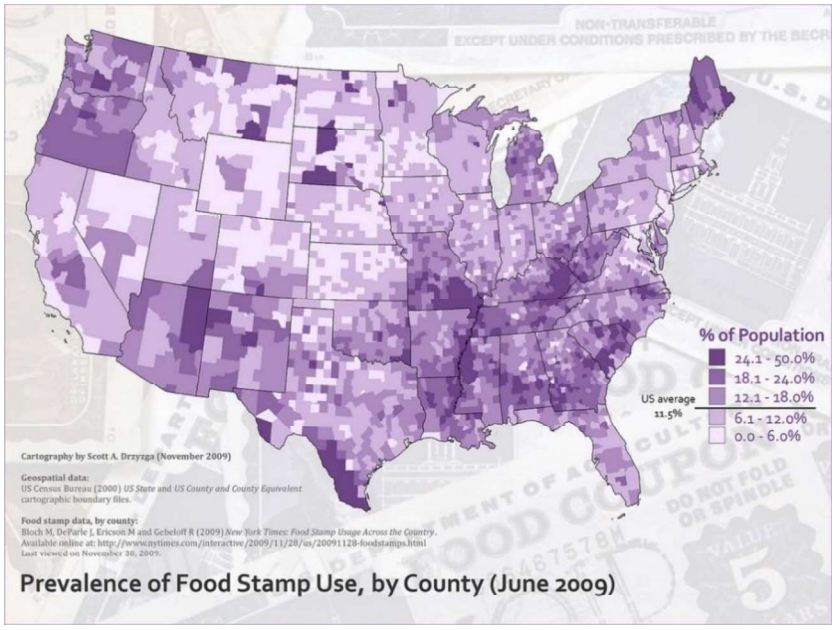
Berdasarkan taksonominya, grafik visual dibedakan menjadi [9]: 1D/Linear; 2D/Planar; 3D/Volumetric; Temporal; *Multidimensional*; *Tree/ Hierarchical*; dan *Network*.

2.2.1.2.1. 1D/Linear

Grafik 1-dimensi termasuk di dalamnya adalah tipe data tekstual, kode sumberprogram, dan huruf alfabet. Setiap *item* yang digambarkan memiliki elemen garis . Contoh dari grafik 1D seperti kode-kode DNA, perbedaan kode sumber, urutan alfabat dan lain-lain. (tidak divisualisasikan secara umum).

2.2.1.2.2. 2D/Planar

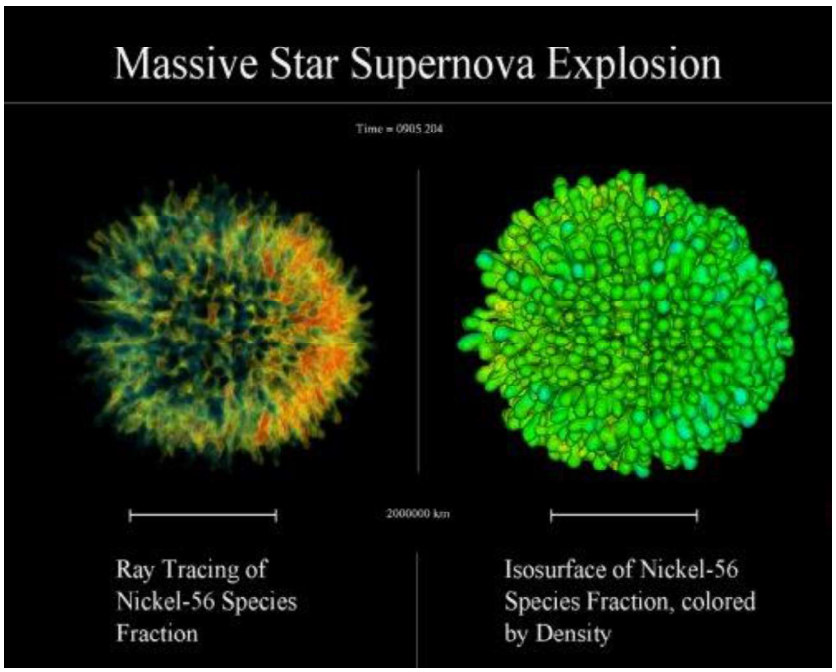
Grafik 2-dimensi termasuk di dalamnya peta geografis, denah rancangan, atau layout koran. Setiap item pada grafik 2-dimensi memiliki total area dan atribut (warna, ukuran, dll). Contoh grafik 2D/Planar dapat dilihat pada Gambar 2.2.1.



Gambar 2.2.1 Contoh Grafik 2D/Planart

2.2.1.2.3. 3D/Volumetric

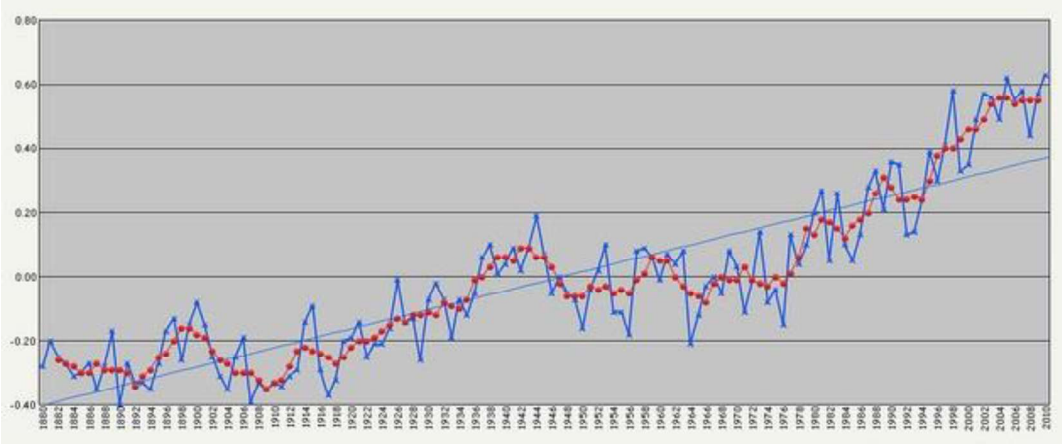
Grafik 3-dimensi adalah visual yang menggambarkan objek nyata, seperti tubuh manusia, bentuk bangunan, dll. Setiap item pada grafik 3-dimensi memiliki volume. Contoh grafik 3D/Volumetric dapat dilihat pada Gambar 2.2.2.



Gambar 2.2.2 Contoh Grafik 3D/Volumetric

2.2.1.2.4. *Temporal*

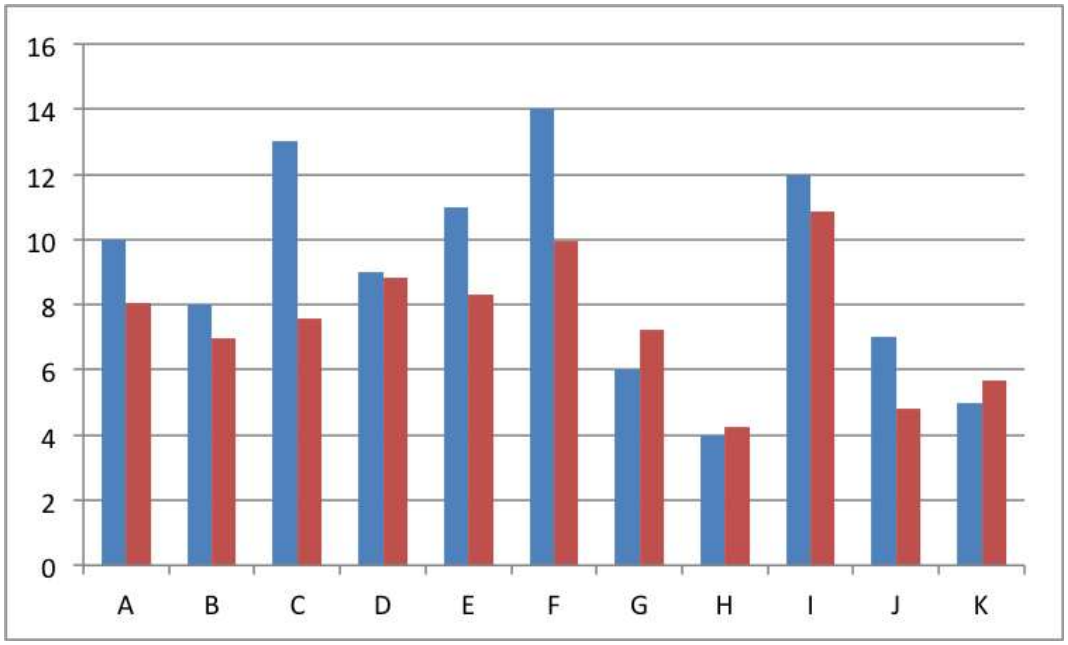
Grafik temporal adalah grafik yang berhubungan dengan waktu (time lines). Grafik ini menggambarkan persentasi historikal dari data 1-dimensi. Yang membedakan, grafik temporal memiliki item dengan waktu awal dan waktu akhir, atau periode tertentu. Contoh grafik temporal dapat dilihat pada Gambar 2.2.3.



Gambar 2.2.3 Contoh Grafik Temporal

2.2.1.2.5. Grafik *Multidimensional*

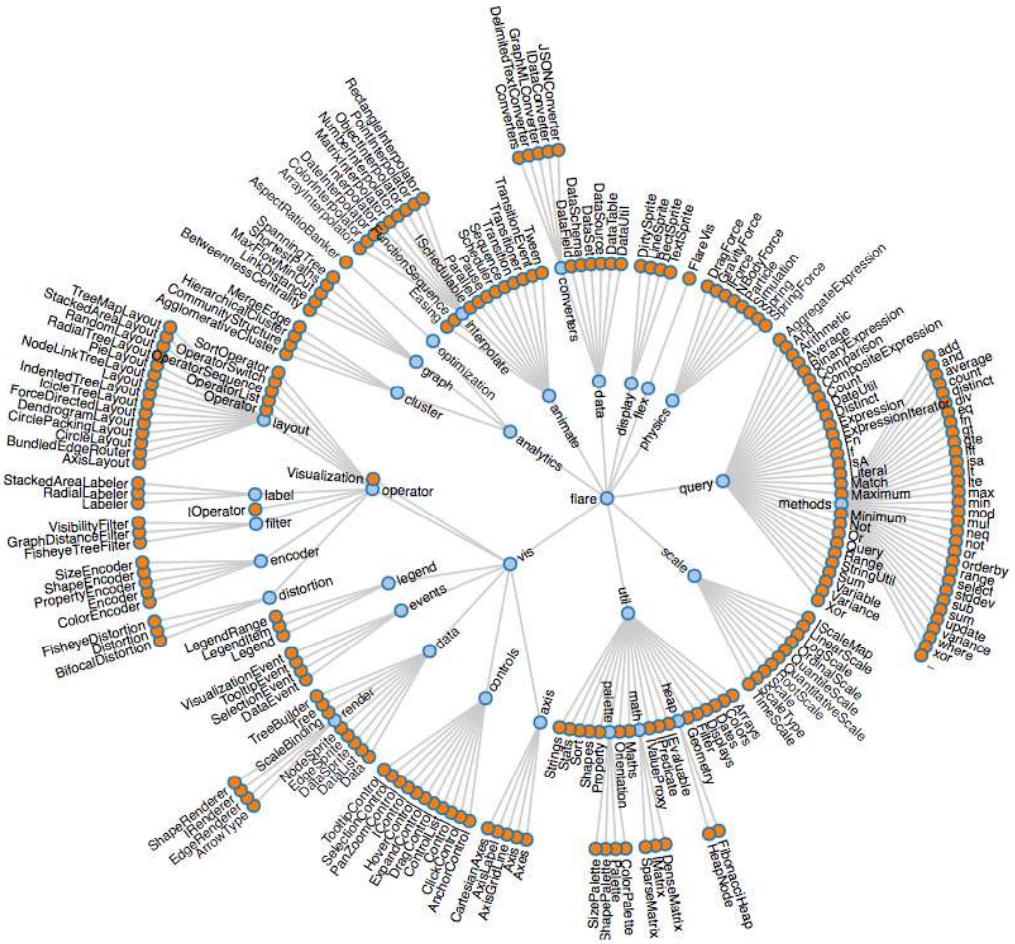
Grafik temporal didalamnya termasuk grafik-grafik yang dihasilkan dari manipulasi data dari disiplin ilmu statistika. Antarmuka representasi multidimensional adalah grafik 2-dimensi. Grafik multi-dimensi termasuk didalamnya grafik *pie*, histogram, *tag cloud, bubble cloud, bar, tree-map, scatter plot, bubble chart, line chart, step chart, heat-map, parallel sets, spider chart, box-plot, mosaic display, waterfall*, dan *tabular*. Contoh grafik multidimensional dapat dilihat pada Gambar 2.2.4.



Gambar 2.2.4 Contoh Grafik Multidimensional

2.2.1.2.6. *Tree/Hierarchial*

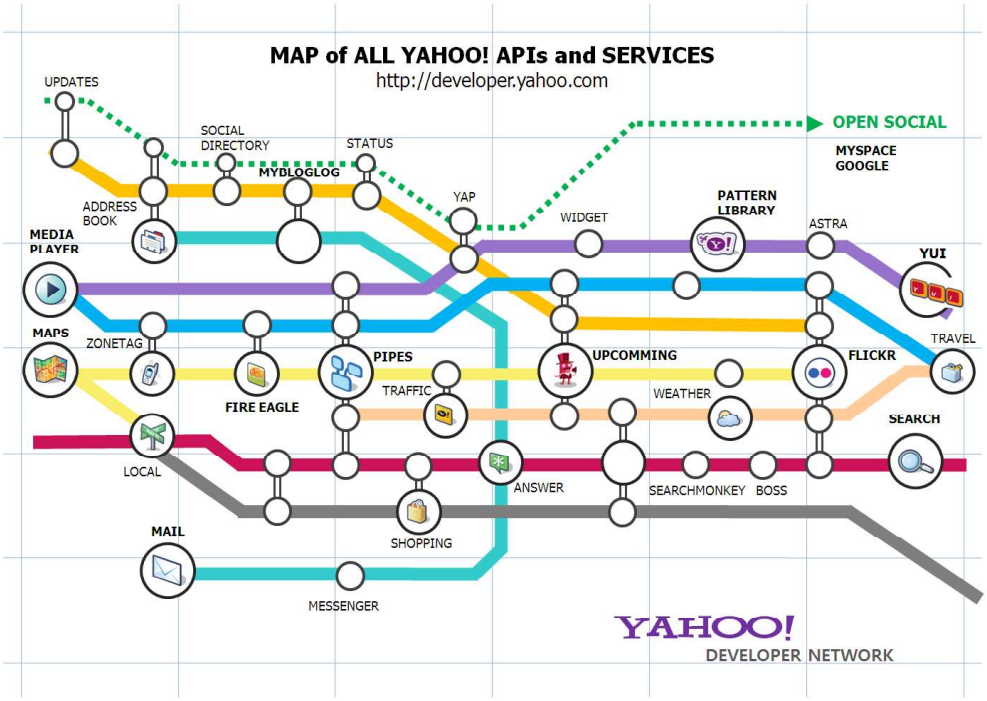
Tree adalah grafik herarkikal dari item-item yang memiliki hubungan satu dengan lainnya, atau yang memiliki induk (kecuali *root*). Setiap item antara induk dan anak bisa memiliki banyak atribut. Grafik tree termasuk didalamnya grafik *tree, dendorogram, radial-tree, hyperbolic-tree, tree-map*, dan *sunburst*. Contoh grafik Tree/Hierarchial dapat dilihat pada Gambar 2.2.5.



Gambar 2.2.5 Contoh Grafik Tree/Hierarchial

2.2.1.2.7. *Network*

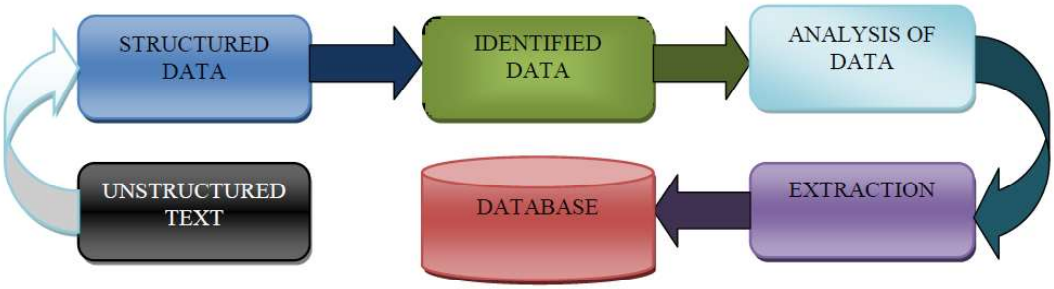
Network adalah grafik relasi yang tidak bisa digambarkan oleh grafik tree, network memvisualisasikan relasi-relasi item yang banyak (semena-mena). Grafik network termasuk di dalamnya grafik *matrix, node-diagram, dependency, hive-plot, alluvial diagram,* dan *subway*. Contoh grafik network dapat dilihat pada Gambar 2.2.6.



Gambar 2.2.6 Contoh Grafik *Network*

2.2.2. *Text Mining*

Text mining adalah proses penemuan akan informasi atau trend baru yang sebelumnya tidak terungkap dengan memproses dan menganalisa data tekstual dalam jumlah besar [10]. Yang membedakan antara *text mining* dengan *data mining* adalah pada *data mining* masukan data terstruktur sedangkan *text mining* masukan datanya tidak terstruktur [11]. Dalam menganalisa sebagian atau keseluruhan *unstructured text, text mining* mencoba untuk mengasosiasikan satu bagian teks dengan yang lainnya berdasarkan aturan-aturan tertentu. Hasil yang di harapkan adalah informasi baru atau "*insight*" yang tidak terungkap jelas sebelumnya. *Text mining* mengadopsi dan mengembangkan banyak teknik dan solusi dari berbagai disiplin ilmu, seperti *information retrieval, data mining, natural language processing*, statistik dan matematik, *machine learning* dan visualisasi [10]. Tahapan dasar yang dapat dilakukan dalam text mining dapat dilihat pada Gambar 2.2.7.



Gambar 2.2.7 Tahapan Dasar Dalam *Text Mining*

Langkah awal yang dilakukan adalah mengumpulkan data yang berupa data tidak terstruktur dari sumber yang telah ditentukan, untuk kemudian data yang tidak terstruktur tersebut akan diubah menjadi data yang terstruktur untuk memudahkan proses komputasi otomatis. Setelah data berubah menjadi data yang terstruktur pola dapat diidentifiaksi untuk selanjutnya akan analisis, untuk menemukan informasi yang berharga dan akan disimpan dalam *database* [10].

2.2.2.1. *Text Preprocessing*

Pengolahan dan pengambilan pengetahuan dari sebuah data tekstual tidak dapat diselesaikan dengan menggunakan teknik *data mining*. Diperlukan beberapa tahapan proses tambahan agar data tekstual yang merupakan data tidak terstruktur dapat dilakukan pengolahan dan pengambilan informasi [11]. Hal ini dikarenakan dalam *data mining*, data yang dapat diolah hanyalah data yang berupa nilai-nilai numerik. Proses ini sering disebut dengan *text preprocessing*, yaitu mengubah data tidak terstruktur menjadi data terstruktur, berikut ini adalah tahapan yang dilakukan dalam *text processing* [11].

1. *Case-folding*

   Case-folding adalah proses mengubah semua huruf dari data masukan menjadi huruf kecil (*lowercase*). Hal ini bertujuan untuk menyeragamkan semua huruf dari data masukan.

2. *Cleansing*

   Cleansing adalah proses menghapus semua karakter yang tidak dianggap tidak valid, seperti angka, tanda baca, karakter khusus, email, URL (*Uniform Resources Locator*), dan juga simbol.

3. *Tokenizing*

   Tokenzing adalah proses untuk memisahkan kata yang dipisahkan oleh spasi.

4. *Stemming*

   Stemming adalah proses untuk merubah berbagai kata berimbuhan menjadi kata dasarnya.

5. *Stopword-removal*

   Stopword-removal adalah proses untuk menghilangkan kata-kata yang kurang penting berdasarakan daftar stopword. *Stopword* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of words.*

2.2.2.2. *Text Representation*

Merupakan tahapan merubah data tekstual menjadi representasi yang lebih mudah untuk diproses. Pada tahapan ini, sebuah kalimat direpresentasikan sebagai objek dan katakata yang menyusunnya direpresentasikan sebagai fitur. Data tekstual akan membentuk sebuah ruang dengan jumlah objek sebanyak jumlah kalimat yang ada dan jumlah fitur sebanyak jumlah kata yang berbeda.

2.2.2.3. Teknik *Text Mining*

Merupakan tahapan utama pada proses Text Mining. Pada tahapan ini dilakukan penerapan teknik yang digunakan untuk pengambilan informasi dari data tekstual yang telah diproses sebelumnya. Terdapat beberapa teknik yang dapat digunakan diantaranya *summarization, information extraction, categorization, visualization, clustering, topic tracking, question, answering*, dan *sentiment analysis* [12]. Pemilihan teknik yang digunakan disesuaikan dengan jenis informasi yang ingin diambil dari data tekstual yang tersedia.

2.2.3. *Sentiment Analysis*

Sentiment analysis atau bisa disebut juga dengan opinion mining adalah riset komputasional dari opini, sentimen, dan emosi yang disampaikan secara

tekstual lalu diklasifikasikan menjadi kelompok sentimen positif dan negatif. Secara umum analisis sentimen dibagi menjadi 2 kategori [13] :

1. *Coarse-grained sentiment analysis*
   Kategori ini melakukan proses analisis pada level dokumen. Singkatnya adalah orientasi sebuah dokumen diklasifikasikan secara keseluruhan. Terdapat 3 kategori dalam orientasi ini yaitu positif, negatif, dan netral. Akan tetapi, ada juga yang menjadikan nilai orientasi ini bersifat kontinu atau tidak diskrit.
2. *Fined-grained sentiment analysis*
   Objek yang diklasifikasikan tidak pada level dokuman melainkan pada level kalimat dalam sebuah dokumen.

Sentiment analysis terdiri dari 3 subproses besar yaitu [14] :

1. *Subjectivity classification*
   Menentukan kalimat yang merupakan opini.
2. *Orientation detection*
   Mengklasifikasikan opini ke dalam kelas positif, negatif dan netral.
3. *Opinion holder and target detection*
   Menentukan bagian yang merupakan opinion holder (pemberi opini) dan bagian yang merupakan target.

Sentiment analysis kemudian akan membedakan teks menjadi dua kategori, yakni fakta dan opini. Fakta merupakan ekspresi objektif mengenai sesuatu. Sementara opini adalah ekpresi subjektif yang menggambarkan sentimen, perasaan, maupun penghargaan terhadap suatu hal.

2.2.4. Basis Data

Basis data data terdiri atas 2 kata, yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa, konsep dan sebagainya [15]. Basis data (*database*) dapat didefenisikan dalam sejumlah sudut pandang seperti :

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redudancy*) yang tidak perlu, untuk memenuhi kebutuhan.

Perangkat lunak yang dirancang untuk membantu pengguna dalam mengelola dan membuat basis data disebut DBMS(*Data Base Management System*) [15]. Fungsi utama dari DBMS adalah untuk menyimpan dan mengambil infomasi dari basis data secara lebih mudah dan efisien.

Sistem basis data didesain untuk mengelola informasi yang besar. Pengelolaan data termasuk mendefinisikan struktur informasi penyimpanan dan penyediaan mekanisme untuk melakukan manipulasi informasi. Selain itu, sistem basis data juga harus bisa memastikan bahwa informasi disimpan dengan aman, baik dari system crash atau akses yang tidak diizinkan. Jika data memang dikelola oleh beberapa pengguna, sistem harus menghindari sebisa mungkin hasil yang anomali.

Interaksi antara pengguna dan DBMS ditentukan melalui bahasa atau sintak khusus sesuai dengan jenis DBMS yang digunakan. Pada umumnya bahasa DBMS dikelompokan ke dalam dua bentuk, yaitu DDL (*data-definition language*) dan DML (*data-manipulation language*) [15].

1. DDL (*Data Definition Language*)

   DDL digunakan untuk membuat skema dan memberikan properti tambahan data, dengan kata lain DDL adalah bahasa untuk membuat tabel, membuat indeks, mengubah tabel, dan menentukan struktur penyimpanan tabel. Keluaran dari perintah DDL adalah kumpulan tabel yang disimpan dalam berkas khusus yang disebut *data dictionary*.

2. DML (*Data Manipulation Language*)

   DML adalah bahasa yang memungkinkan pengguna DBMS mengakses atau melakukan manipulasi data yang diorganisasikan dengan model data yang sesuai. Tipe akses tersebut yaitu :

a. Mengambil (*retrieve*) informasi dari basis data.
b. Memasukan (*insert*) informasi baru ke basis data.
c. Menghapus (*delete*) infromasi dari basis data.
d. Modifikasi (*update*) informasi yang terdapat dari basis data.

2.2.4.1. Basis Data NoSQL

Database NoSQL dibuat dengan tujuan khusus untuk model data spesifik dan memiliki skema fleksibel untuk membuat aplikasi modern. *Database* NoSQL dikenal secara luas karena kemudahan pengembangan, fungsionalitas, dan kinerja dalam berbagai skala. Database NoSQL menggunakan berbagai model data, termasuk dokumen, grafik, nilai kunci, dalam memori, dan pencarian [16]. Keunggulan NoSQL dibandingkan basis data relasi ada pada skabilitas. Selain skalabilitas, NoSQL memiliki keunggulan sebagai berikut [17] :

1. *Schemaless data representation*: Hampir semua model NoSQL merepresentasikan *schemaless*. Administrator basis data tidak perlu memikirkan bagaimana struktur dan model basis data yang berubah setiap waktu.
2. *Development time*: *Query* yang komplek pada basis relasi yang sudah besar, akan membuat proses penyajian data menjadi lambat (misalnya JOIN tabel pada basis data yang tersebar di banyak server). Pada NoSQL hal demikian tidak terjadi.
3. *Plan ahead for scalability*: Aplikasi yang memakai model NoSQL akan sangat elastis, NoSQL dapat menangani masalah lonjakan sumber daya.

2.2.5. *Naive Bayes Classifier*

Naive bayes classifier merupakan sebuah metode klasifikasi probabilistik sederhana yang menghitung sekumpulan probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari dataset yang diberikan. Algoritma mengunakan

teorema Bayes dan mengasumsikan semua atribut independen atau tidak saling ketergantungan yang diberikan oleh nilai pada variabel kelas [18]. Definisi lain mengatakan *Naive Bayes* merupakan klasifikasi dengan metode probabilitas dan statistik yang dikemukan oleh ilmuwan Inggris Thomas Bayes pada abad ke-18, yaitu untuk memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya [19].

Naive Bayes didasarkan pada asumsi penyederhanaan bahwa nilai atribut secara kondisional saling bebas jika diberikan nilai output. Dengan kata lain, diberikan nilai output, probabilitas mengamati secara bersama adalah produk dari probabilitas individu [20]. Keuntungan penggunaan *Naive Bayes* adalah bahwa metode ini hanya membutuhkan jumlah data pelatihan (*Training Data*) yang kecil untuk menentukan estimasi paremeter yang diperlukan dalam proses pengklasifikasian. *Naive Bayes* sering bekerja jauh lebih baik dalam kebanyakan situasi dunia nyata yang kompleks dari pada yang diharapkan [21].

$$P(H|X) = \frac{P(X|H).P(H)}{P(X)}$$

Keterangan :

X : Data dengan class yang belum diketahui

H : Hipotesis data merupakan suatu class spesifik

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

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

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

P(X) : Probabilitas X

Untuk menjelaskan metode Naive Bayes, perlu diketahui bahwa proses klasifikasi memerlukan sejumlah petunjuk untuk menentukan kelas apa yang cocok bagi sampel yang dianalisis tersebut. Karena itu, metode Naive Bayes di atas disesuaikan sebagai berikut :

$$P(C|F1..Fn) = \frac{P(C)P(F1..Fn|C)}{P(F1..Fn)}$$

Di mana Variabel C merepresentasikan kelas, sementara variabel *F1..Fn* merepresentasikan karakteristik petunjuk yang dibutuhkan untuk melakukan klasifikasi.

2.2.6. TF-IDF (*Term Frequency-Inverse Document Frequency*)

Metode *Term Frequency-Inverse Document Frequency* (TF-IDF) adalah suatu metode pemberian bobot hubungan suatu kata (*term*) terhadap dokumen. Pada dokumen tunggal tiap kalimat dianggap sebagai dokumen. Metode ini menggabungkan dua konsep untuk perhitungan bobot, yaitu Term frequency (TF) merupakan frekuensi kemunculan kata (t) pada kalimat (d). Document frequency (DF) adalah banyaknya kalimat dimana suatu kata (t) muncul [22].

Frekuensi kemunculan kata di dalam dokumen yang diberikan menunjukkan seberapa penting kata itu di dalam dokumen tersebut. Frekuensi dokumen yang mengandung kata tersebut menunjukkan seberapa umum kata tersebut. Bobot kata semakin besar jika sering muncul dalam suatu dokumen dan semakin kecil jika muncul dalam banyak dokumen [22]. Berikut ini adalah rumus penghitungan bobot (w) pada metode TF-IDF untuk masing-masing dokumen terhadap kata kunci :

$$W_{dt} = tf_{dt} * IDF_t$$

Keterangan :

d = dokumen ke-d

t = kata ke-t dari kata kunci

W = bobot dokumen ke-d terhadap kata ke-t

tf = banyaknya kata yang dicari pada sebuah dokumen

IDF = Inversed Document Frequency

IDF = log2 (D/df)

D = total dokumen

df = banyak dokumen yang mengandung kata yang dicari

Setelah bobot (W) masing-masing dokumen diketahui, maka dilakukan proses pengurutan dimana semakin besar nilai W, semakin besar tingkat similaritas dokumen tersebut terhadap kata kunci, begitupun sebaliknya.

Inverse Document Frequency memperhatikan kemunculan term pada kumpulan dokumen. Pada metode ini, term yang dianggap bernilai adalah term yang jarang muncul pada kumpulan dokumen. Berikut ini adalah persamaan IDF :

$$IDF(t) = log\frac{N}{df(t)}$$

Dimana df(t) adalah banyak dokumen yang mengandung term t. TF*IDF merupakan kombinasi metode TF dengan metode IDF. Sehinggan menghasilkan persamaan TF*IDF berikut ini:

$$TF * IDF(d,t) = TF(d,t) * IDF(t)$$

Perhitungan bobot *query relevance* merupakan bobot hasil perbandingan kemiripan antara *query* yang dimasukkan oleh pengguna terhadap keseluruhan kalimat. Sedangkan bobot kemiripan kalimat, merupakan bobot hasil perbandingan kemiripan antar kalimat.

2.2.7. Statistik

Statistik adalah sebuah ilmu yang digunakan untuk memecahkan suatu permasalahan dengan menggunakan beberapa tahapan yaitu pengumpulan data, pengolahan data, Analisis data dan intepretasi data serta kesimpulan dan keputusan yang diambil berdasarkan Analisis yang telah dilakukan [23].

Jenis statistik dibedakan menjadi dua, yaitu statistika deskriptif dan statistika inferensi [23].

a. Statistika Deskriptif yaitu statistika yang menggunakan metode numerik dan grafik untuk mencari pola dalam suatu kumpulan data, meringkas informasi yang terkandung dalam kumpulan data, dan menghadirkan informasi dalam bentuk yang diinginkan.

b. Statistika Inferensi yaitu statistik yang menggunakan data sampel untuk membuat estimasi, keputusan, prediksi, dan generalisasi terhadap kumpulan data yang lebih besar.

2.2.7.1. Distribusi Frekuensi

Tabel Frekuensi merupakan salah satu jenis penyajian data. Tabel Frekuensi adalah cara umum untuk menata atau menyusun data yang dimiliki dalam sebuah tabel yang menunjukkan sebaran atau distribusi frekuensi data dan tersusun atas frekuensi tiap-tiap kelas atau kategori yang telah ditetapkan. Frekuensi tiap kelas atau kategori menunjukkan banyaknya pengamatan dalam kelas yang sedang diamati [23].

Kelas	Frekuensi (*f*)
Bola Merah	16
Bola Biru	18
Bola Hijau	15
Bola Kuning	19
Bola Ungu	22
Total	90

Gambar 2.2.8 Contoh Distribusi Frekuensi

2.2.8. Persentase Pertumbuhan

Persentase Pertumbuhan adalah kenaikan suatu hal dalam bentuk angka persentase dari satu periode waktu ke periode waktu selanjutnya. Persentase pertumbuhan digunakan untuk berbagai kebutuhan, seperti kinerja perusahaan, pertumbuhan ekonomi negara, hingga tingkat pertumbuhan penduduk [24].

Berikut ini adalah rumus yang dapat digunakan untuk menghitung pertumbuhan :

$$Persentase\ pertumbuhan = \frac{(nilai_{akhir} - nilai_{awal})}{nilai_{awal}} x\ 100$$

2.2.9. Scikit-learn

Scikit-learn atau sklearn adalah pustaka machine learning yang mendukung *supervised learning* dan *unsupervised learning*, yang juga menyediakan berbagai utilitas seperti *model fitting, preprocessing data, model selection*, evaluasi dan beberapa utilitas lainnya. Sklearn dibangun diatas NumPy, SciPy, dan matplotlib. Paket ini berfokus pada membawa pembelajaran mesin kepada non-spesialis menggunakan bahasa tingkat tinggi untuk tujuan umum. Penekanan diberikan pada kemudahan penggunaan, kinerja, dokumentasi, dan konsistensi API. Paket ini didistribusikan di bawah lisensi BSD yang disederhanakan, mendorong penggunaannya dalam bidang akademik dan komersial [25].

Fitur-fitur yang terdapat pada sklearn meliputi *classification, regression, clustering, dimensionality reduction, model selection* dan *preprocessing*. Pengembangan sklearn dimulai pada tahun 2007 sebagai *Google Summer of Code project* oleh David Cournapeau. Pada tahun berikutnya Matthieu Brucher juga memulai pengembangannya pada sklearn sebagai bagian dari thesis-nya. Pada tahun 2010 Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort dan Vincent Michel dari INRIA mengambil kepemimpinan proyek dan melakukan rilis publik pertama pada 1 Februari 2010 [25].

2.2.10. Analisis Desain Berorientasi Objek

Analisis desain berorientasi objek (*Object Oriented Analysis and Design*) merupakan sebuah metode dalam memikirkan suatu masalah dengan menggunakan model yang dibuat menurut konsep. Yang dimaksud berorientasi objek adalah bahwa dalam mengorganisasikan perangkat lunak menggunakan kumpulan objek Yng bekerja sama antara informasi atau struktur data dan perilaku yang mengaturnya. Alasan digunakannya metode berorientasi objek adalah karena perangkat lunak bersifat dinamis, hal ini dikarenakan kebutuhan pengguna berubah dengan cepat [26].

Pendekatan berorientasi objek membawa pengguna kepada abstraksi atau istilah yang lebih dekat dengan dunia nyata, karena di dunia nyata yang sering pengguna lihat bukan fungsinya melainkan objeknya. Berbeda dengan pendekatan

terstruktur yang hanya mendukung abstraksi pada level fungsional. Adapun dalam pemrograman beroirentasi objek berbagai konsep yang ditekankan seperti: *Class, Object, Abstract, Encapsulation, Polymorphism, Inheritence,* dan *juga UML (Unified Modeling Languange)* [26].

Unified Modeling Language (UML) adalah sebuah bahasa pemodelan visual yang digunakan untuk memvisualisasikan, menspesifikasikan, membangun dan mendokumentasikan artefak sistem perangkat lunak. UML digunakan untuk memahami , mendesain, mengeksplorasi, mengkonfigurasi, memelihara, dan mengontrol informasi tentang sistem. Hal Ini dimaksudkan untuk digunakan dengan semua metode pengembangan, tahapan siklus hidup, domain aplikasi, dan media [26].

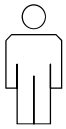
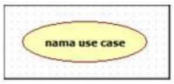
UML bukanlah bahasa pemrograman, melainkan alat yang dapat menyediakan generator kode dari UML ke dalam berbagai bahasa pemrograman serta membangun model rekayasa balik dari program yang ada. UML adalah bahasa pemodelan diskrit, yang dimaksudkan untuk menjadi bahasa pemodelan umum untuk sistem diskrit seperti yang terbuat dari perangkat lunak, firmware, atau logika digital [27].

UML diadopsi dengan suara bulat oleh keanggotaan *Object Management Group* (OMG) sebagai standar pada November 1997. OMG memiliki tanggung jawab untuk pengembangan lebih lanjut dari standar UML. Bahkan sebelum adopsi terakhir, sejumlah buku diterbitkan yang menguraikan hal-hal penting dari UML [27].

1. *Use Case Diagram*

Use case diagram dapat digunakan selama proses analisis untuk menangkap *requirements* sistem dan untuk memahami bagaimana sistem seharusnya bekerja. Selama tahap desain, use case diagram menetapkan perilaku (*behavior*) sistem saat diimplementasikan. Dalam sebuah model mungkin terdapat satu atau beberapa *use case diagram*. Tabel simbol use case diagram dapat dilihat pada Tabel 2.2.1.

Tabel 2.2.1 Simbol *Use Case Diagram*

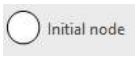
No	Simbol	Nama	Keterangan
1		*Actor*	Merupakan orang, proses atau sistem lain yang berinteraksi dengan sistem yang akan dibuat. Jadi walaupun simbol aktor dalam diagram usecase berbentuk orang, namun aktor belum tentu orang.
2		*Use case*	merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling berinteraksi atau bertukar pesan antar unit maupun aktor.
3		*Association*	Merupakan relasi yang digunakan untuk menggambarkan interaksi antara usecase dan aktor. Asosiasi juga menggambarkan berapa banyak objek lain yang bisa berinteraksi dengan suatu objek atau disebut *multiplicity* .
4		*Extend*	Menghubungkan antara satu objek dengan objek lain.
5		*Generalization*	Hubungan dimana objek berbagai perilaku dan struktur data dari objek yang ada di atasnya objek induk.
6		*Include*	Menspesifikasi bahwa *use case* sumber secara eksplisit.
7		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
8		*System Boundary*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

2. *Activity Diagram*

Activity diagram memodelkan alur kerja (*work flow*) sebuah proses bisnis dan urutan aktivitas dalam suatu proses. Diagram ini sangat mirip dengan sebuah *flowchart* karena user dapat memodelkan sebuah alur kerja dari satu aktivitas ke aktivitas lainnya atau dari satu aktivitas ke dalam keadaan sesaat (*state*). 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. Tabel simbol *activity diagram* dapat dilihat pada Tabel 2.2.2.

Tabel 2.2.2 Simbol *Activity Diagram*

No	Simbol	Nama	Keterangan
1		*Action*	State dari sistem yang mencerminkan suatu aksi.
2		*Decision*	Digunakan untuk menggambarkan suatu pengambilan keputusan / tindakan pada kondisi tertentu.
3		*Initial node*	Menunjukkan dimulainya suatu aktifitas.
4		*Final node*	Menunjukkan akhir dari aktifitas.
5		*Fork node*	Digunakan untuk menunjukkan aktifitas yang dilakukan secara paralel.
6		*Join node*	Digunakan untuk menunjukkan aktifitas yang digabungkan.
7		*Swimlane*	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi.