

BAB II

LANDASAN TEORI

2.1 Data Mining

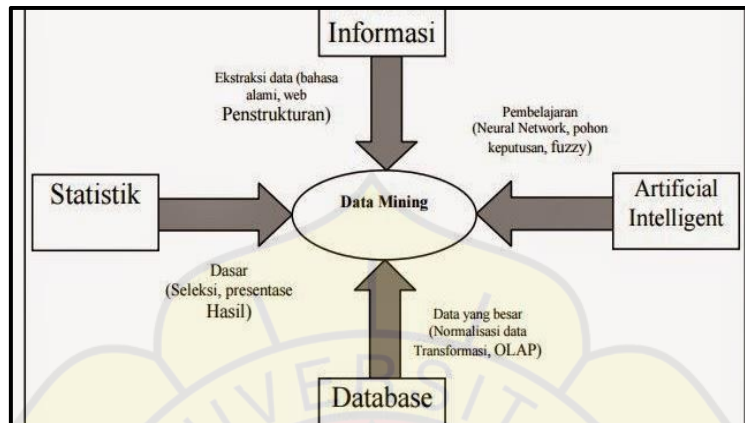
(Kurniawan, 2018) mengungkap penambangan data adalah langkah serta fase didalam mencari sesuatu susunan data-data. Susunan data-data ini bisa bermacam-macam jenisnya, juga peraturan, diagram atau jaring, pohonan (*trees*) atau samaan, dan beberapa lainnya. Dengan memanfaatkan penambangan data, maka dari situasi didapati diamati tren, susunan, dan prediksi yang akan datang. Penambangan data tersendiri punya banyaknya fase dan metode yang didapat diterapkan kedalam hidup sehari-hari.

(Parteek Bhatia, 2019) Penambangan data yaitu suatu progres yang menggunakan melebihi dari satu metode belajar komputer (*machine learning*) diperuntukan analisis dan mengekstrak kepengetahuan (*knowledge*) kesecaraan otomatisasi.

Penambangan data adalah rangkaian prosedur diperuntukan mengungkapkan penilaian tambahan satu set data dalam bentuk kepengetahuan yang sebelumnya tak ditemui secara manualan (Amalia, 2018).

Penambangan data adalah sebuah konsep yang dipergunakan diperuntukan memecahkan temuan kepengetahuan didalam basis data. *Data Mining* merupakan sebuah prosesi yang memanfaatkan Teknik Statistik, matematik, cerdasan tiruan, dan pembelajaran mesin diperuntukan mengekstraksikan dan mengidentifikasi keinformasian yang berguna dan kepengetahuan yang keterkaitan dari keberbagaian basis data yang megah. (Amalia, 2018).

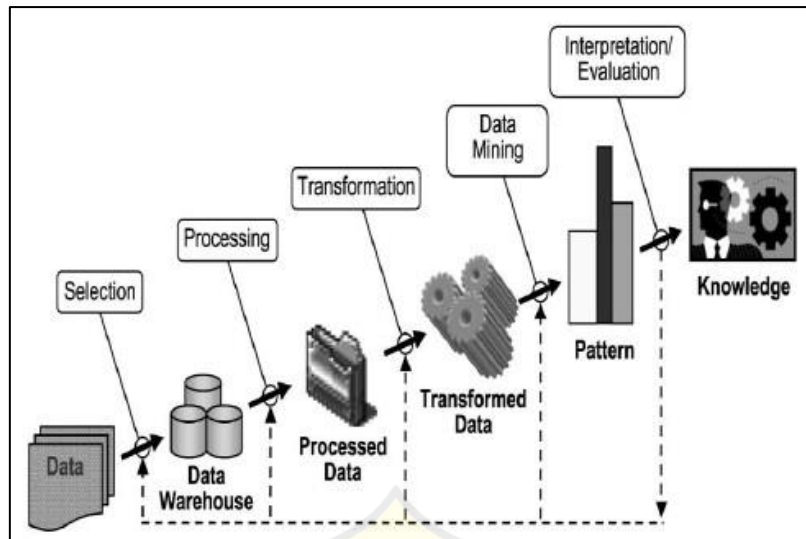
Gambar 2.1 menjelaskan tentang penambahan data adalah bidang yang menggabungkan banyak bidang ilmu yang menggabung teknik dari belajar mesin, pengenalan pola, statistik, database, dan visualisasi untuk mewujudkan masalah pengambilan keinformasian dari database yang megah.



Gambar 2. 1 Bidang Ilmu Data Mining (Larose, 2006)

2.2 Tahapan Proses Dalam Data Mining

Ada sejumlah prosesi tingkatan yang ada di data mining. Gambar 2.2 Menjelaskan fase yang awal yaitu mulai dari datanya bersumber dan diakhiri dengan keadannya informasian yang dihasilkan dari suatu nomor, yaitu:



Gambar 2. 2 Fase-fase Dalam Data Mining (Muchlisin Riadi, 2017)



Ketahapan dalam Penambangan data didapat diartikan berikut:

1. Seleksi Data

Pemilihan data dari kumpulan data operasional harus dilakukan sebelum memulai tahap pembelajaran informasi di KDD. Data terpilih yang digunakan untuk penambangan data disimpan dalam file terpisah dari database operasional.

2. *Pre-processing / Cleaning*

Sebelum memulai proses penambangan data, harus dilakukan proses pembersihan pada data yang menjadikan terfokus KDD. Proses pembersihan mencakupi diantaranya menghapus data yang sama, memeriksakan data yang tidak konsisten, dan memperbaiki salahan data.

3. *Transformation*

Merupakan proses transformasi data terpilih, agar data tersebut sesuai untuk proses penambangan data. Prosesi pemrograman didalam KDD merupakan proses kreatif dan sangat bergantung pada jenis atau pola informasi yang perlu dicari dalam database.

4. Data Mining

Penambangan data adalah proses menemukan pola atau informasi menarik dalam data yang telah dipilih dengan digunakan teknik atau metodologi tertentu. Metode, teknik, atau algoritma dalam penambangan data beragam. Pilihan algoritma atau metode yang sesuai tergantung pada tertujunya dan prosesi KDD secara keseluruhan.

5. *Interpretation / Evaluation*

Model informasi yang diperoleh dari proses data mining sangat diperlukan ditunjukkan didalam format hal ini dapat dimengerti oleh pihak-pihak yang

berkepentingan. Langkah ini merupakan bagian dari proses yang disebut KDD interpretasi. Langkah ini melibatkan pengecekan jika model atau informasi ditemukan bertentangan dengan fakta atau asumsi yang sudah ada sebelumnya.

2.3 Tugas Data Mining

Menurut Anjar Wanto (2020: 6), Ada beberapa tugas yang dilakukan oleh *Data Mining* dalam memproses pemecahan masalah dan mencari pengetahuan baru, diantaranya adalah sebagai berikut:

a. Klastering (*Clustering*)

Digunakan untuk mengelompokkan atau mengidentifikasi data yang memiliki karakteristik tertentu. Contoh algoritma: *K-Means*, *K-Medoids*, dan lainnya.

b. Klasifikasi (*Classification*)

Klasifikasi berguna untuk menemukan model/fungsi yang menjelaskan atau membedakan konsep dan juga kelas data, yang bertujuan untuk memperkirakan kelas dari suatu objek yang labelnya tidak diketahui. Contoh algoritmanya adalah: *C4.5*, *K-Nearest Neighbor (KNN)*, dan lain-lain.

c. Asosiasi (*Association*)

Asosiasi berguna mengatasi masalah bisnis yang tidak umum, yaitu dengan cara menganalisa tabel transaksi penjualan dan mengidentifikasi produk-produk yang sering dibeli bersamaan oleh pembeli, contohnya bila orang membeli sambal biasanya juga membeli kecap. Contoh algoritmanya adalah: *Apriori*, *Frequent Pattern Growth*, *FP-Growth*, atau *Eclat* yang dipakai di dalam penulisan ini.

d. Estimasi (*Estimation*)

Estimasi berguna untuk memperkirakan sesuatu hal yang belum pernah ada pada sebelumnya, dan disajikan dalam bentuk hasil kuantitatif atau angka. Contoh

algoritmanya adalah: *Regresi Linier*, *Confidence Interval Estimations*, dan lainnya.

e. *Prediksi (Prediction)*

Prediksi berguna untuk memperkirakan atau meramalkan suatu kejadian yang belum pernah terjadi. Contoh algoritmanya adalah: *Decision Tree*, *K-Nearest Neighbor* dan lainnya.

2.4 Metode Data Mining

Model *data mining* dibuat berdasarkan salah satu dari dua jenis pembelajaran *supervised* dan *unsupervised*. Fungsi pembelajaran *supervised* digunakan untuk memprediksi suatu nilai. Fungsi pembelajaran *unsupervised* digunakan untuk mencari struktur intrinsik, hubungan dalam suatu data yang tidak memerlukan kelas atau label sebelum dilakukan proses pembelajaran.

Contoh dari algoritma pembelajaran *unsupervised*, diantaranya *K-Means Clustering* dan Aturan Asosiasi Apriori. Contoh dari algoritma pembelajaran *supervised* yaitu Naïve Bayes untuk klasifikasi.

Metode *data mining* dapat diklasifikasikan berdasarkan fungsi yang dilakukan atau berdasarkan jenis aplikasi yang menggunakannya.:

- Klasifikasi (*supervised*)
- Clustering (*unsupervised*)
- Association Rules (*unsupervised*)
- Attribute Importance (*supervised*)

2.4.1 Klasifikasi (*supervised*)

Dalam hal klasifikasi, kita memiliki beberapa kasus (data sampel) dan ingin memperkirakan beberapa kelas yang ada dalam data sampel tersebut. Setiap data

instan berisi beberapa atribut, di mana setiap atribut memiliki satu dari beberapa nilai yang mungkin.

Hanya satu atribut di antara banyak atribut ini yang disebut atribut target, sementara atribut lainnya disebut atribut prediktor. Setiap nilai yang mungkin dari atribut target menunjukkan kelas yang diprediksi berdasarkan nilai-nilai atribut prediktor.

Klasifikasi digunakan untuk segmentasi pelanggan, pemodelan bisnis, analisis kartu kredit, dan banyak aplikasi lainnya. Sebagai contoh, perusahaan kartu kredit ingin memperkirakan pelanggan berdasarkan metode pembayaran.

2.4.2 Clustering (*unsupervised*)

Clustering merupakan teknik yang berguna untuk menjelajahi data. Digunakan ketika terdapat banyak kasus dan tidak ada pengelompokan yang alami. Dalam hal ini, algoritma data mining dapat digunakan untuk mencari pengelompokan yang ada dalam data.

Analisis Clustering mengidentifikasi kelompok yang ada dalam data. Kelompok adalah kumpulan objek data yang serupa satu sama lain. Metode pengelompokan yang baik menghasilkan kelompok yang berkualitas untuk memastikan kesamaan dalam data yang ada dalam satu kelompok. Model pengelompokan berbeda dari model prediktif karena dalam pengelompokan tidak perlu ada atribut target. Pengelompokan yang diorganisasi dalam struktur hirarkis akan menentukan taksonomi dari data. Dalam ODM, suatu kelompok dikarakterisasi oleh centroid, histogram atribut, dan model pengelompokan hirarkis. ODM membentuk pengelompokan hirarkis dengan menggunakan versi perbaikan dari algoritma K-Means dan O-Cluster.

2.4.3 Association Rules (*unsupervised*)

Fungsi Association Rules seringkali disebut dengan "market basket analysis", yang digunakan untuk menemukan relasi atau korelasi diantara himpunan item-item. Fungsi ini paling banyak digunakan untuk menganalisa data dalam rangka keperluan strategi pemasaran, desain katalog, dan proses pembuatan keputusan bisnis.

Tipe Association rule bisa dinyatakan sebagai misal: "70% dari orang-orang yang membeli mie, juice dan saus akan membeli juga roti tawar". Aturan asosiasi mengcapture item atau kejadian dalam data berukuran besar yang berisi data transaksi. Dengan kemajuan teknologi, data penjualan dapat disimpan dalam jumlah besar yang disebut dengan "basket data." Aturan asosiasi yang didefinisikan pada basket data, digunakan untuk keperluan promosi, desain katalog, segmentasi customer dan target pemasaran.

2.4.4 Attribute Importance (*supervised*)

Attribute Importance, disebut juga dengan feature selection, menyediakan solusi otomatis untuk meningkatkan kecepatan dan akurasi dari model klasifikasi yang dibangun pada table data yang memiliki jumlah atribut yang sangat banyak. Attribute Importance meranking atribut prediktif dengan melakukan eliminasi nilai yang redundant, tidak relevant atau tidak informative dan mengidentifikasi atribut predictor yang banyak paling berpengaruh dalam pengambilan keputusan.

Dengan menggunakan atribut yang lebih sedikit akan mereduksi waktu untuk membangun suatu model, juga dapat meningkatkan akurasi dari kemampuan prediksi. Jika terlalu banyak atribut yang dilibatkan maka akan banyak pula noise

yang terlibat yang akan berpengaruh terhadap model karena dapat menurunkan performansi dan akurasi.

2.5 Teknik Association Rules

Analisis asosiasi atau *association rule mining* adalah teknik data mining untuk menemukan aturan suatu kombinasi item (Pracoyo, 2016). Salah satu tahap analisis asosiasi yang menarik perhatian banyak peneliti untuk menghasilkan algoritma yang efisien adalah analisis pola frekuensi tinggi (*frequent pattern mining*).

Penting tidaknya suatu asosiasi dapat diketahui dengan dua tolak ukur, yaitu: *support* dan *confidence*. Support (nilai penunjang) adalah persentase kombinasi item tersebut dalam database, sedangkan *confidence* (nilai kepastian) adalah kuatnya hubungan antar item dalam aturan asosiasi.

2.6 Website

2.6.1 CSS (*Cascading Style Sheets*)

CSS (*Cascading Style Sheets*) adalah salah satu istilah teknis dalam pemrograman, CSS merupakan *styling language* (Bahasa desain), bagian dari markup language yang dapat mewarnaik atau mendesain suatu halaman *website* (Khairina F. Hidayati, 2021).

Menurut (Wahyudi, 2017), CSS adalah suatu bahasa pemrograman web yang digunakan untuk mengendalikan dan membangun berbagai komponen dalam web sehingga tampilan akan lebih rapi, terstruktur dan seragam.

2.6.2 Javascript

Menurut (Siahaan dan Rismon, 2020), yaitu “*Javascript* adalah sebuah bahasa script dinamis yang dapat dipakai untuk membangun interaktifitas pada halaman-

halaman *HTML* statis. Ini dilakukan dengan menamakan blok-blok kode *Javascript* di semua tempat pada halaman *web*.”*PHP (Hypertext Preprocessor)*.

2.6.3 PHP

Rohi Abdulloh (2018, h. 3) mengatakan bahwa “*PHP* berperan sebagai proses data pada sisi *server*, sesuai yang diminta oleh *client* menjadi informasi yang siap ditampilkan, juga sebagai penghubung aplikasi *web* dengan *database*.”.

2.6.4 Bootstrap

Zaenal A Rozi & SmitDev Community (2015, h. 1) mendiskusikan topik *bootstrap*. Dan mereka mengatakan bahwa “*Bootstrap* adalah paket aplikasi siap pakai untuk membuat *front-end* sebuah *website*. Bisa dikatakan, *bootstrap* adalah template desain *web* dengan fitur plus. *Bootstrap* diciptakan untuk mempermudah proses desain *web* bagi berbagai tingkat pengguna, mulai dari level pemula hingga yang sudah berpengalaman.”.

2.6.5 HTML (Hyper Text Markup Language)

Rohi Abdulloh (2018, h. 7) mengatakan bahwa “Bahasa standar *web* yang dikelola penggunaannya oleh W3C (*World Wide Web Consortium*) berupa tag-tag yang menyusun setiap elemen dari *website HTML*. Berperan sebagai penyusun struktur halaman *website* yang menempatkan setiap elemen *website* sesuai *layout* yang diinginkan.”.

2.6.6 Visual Studio Code (VS Code)

Ummy Gusti Salamah (2021, h. 1) mendefinisikan bahwa “*Visual Studio Code (VS Code)* ini adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi multiplatform, artinya tersedia juga untuk Linux, Mac, dan Windows. Teks editor *VSCode* juga bersifat open source, yang mana

kode sumbernya dapat kalian lihat dan kalian dapat berkontribusi untuk pengembangannya. Hal ini juga yang membuat *VSCode* menjadi favorit para pengembang aplikasi, karena para pengembang aplikasi bisa ikut serta dalam proses pengembangan *VSCode* ke depannya.

2.7 Database

Menurut (Hesananda et al., 2017), *Database* adalah suatu wadah untuk menampung sebuah data yang ada pada sistem. *Database* juga bisa diartikan sebagai kumpulan data. *Database* juga bisa dikenal sebagai formal dan tegas.

Database juga bisa diartikan dengan kumpulan data yang terintegrasi yang dapat dimanipulasi, diambil dan dicari secara cepat.

2.8 MySql

Yoga Ananda Putra, Sumijan, & Mardison. (2019) mendiskusikan bahwa “*MySQL* adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan *MySQL*, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial.

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya; *SQL* (*Structured Query Language*). *SQL* adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

2.9 XAMPP

Darman Umagapi & Arisandy Ambarita (2018) mendiskusikan bahwa “XAMPP adalah perangkat lunak (*free software*) bebas, yang mendukung untuk banyak sistem operasi, yang merupakan kumpulan dari beberapa program.”.

2.10 Algoritma Sistem

2.10.1 Algoritma Apriori

Algoritma apriori adalah Teknik data mining untuk menemukan aturan suatu kombinasi item. Dan merupakan algoritma yang paling terkenal untuk menemukan pola frekuensi tinggi (Astuti et al., 2016). Untuk mencari association rule dari suatu kumpulan data, tahap pertama yang harus dilakukan adalah mencari frequent itemset terlebih dahulu. Frequent itemset adalah sekumpulan item yang sering muncul secara bersamaan. Penting tidaknya suatu asosiasi dapat diketahui dengan dua tolak ukur, yaitu support dan confidence. Support adalah nilai penunjang atau persentase kombinasi sebuah item dalam database, sedangkan confidence adalah nilai kepastian yaitu kuatnya hubungan antar item dalam aturan asosiasi (Heroe Santoso, I Putu Hariyadi, Prayitno, 2016). Proses utama yang dilakukan dalam algoritma apriori untuk mendapat frequent itemset yaitu:

- *Join* (penggabungan). Proses ini dilakukan dengan cara pengkombinasian item dengan yang item lainnya hingga tidak bisa terbentuk kombinasi lagi.
- *Prune* (pemangkasan). Proses pemangkasan yaitu hasil dari item yang telah dikombinasikan kemudian dipangkas dengan menggunakan minimum support yang telah ditentukan.

2.10.2 Algoritma ECLAT (*Equivalent Class Transformation*)

Algoritma *Equivalent Class Transformation* (ECLAT) merupakan algoritma yang sangat sederhana untuk menemukan itemset yang paling sering muncul, pada dasarnya algoritma ECLAT melakukan pencarian secara depth-first search pada database dengan tata letak vertikal, jika database berbentuk horizontal maka harus dikonversikan ke bentuk vertikal terlebih dahulu. (Kaur & Grag, 2014)

Algoritma ECLAT mengurutkan kandidatnya dengan pencarian depth-first dan menggunakan perpotongan *Transaction Id List (TID-List)* antar itemnya, sehingga tidak perlu menghitung support semua itemset karena algoritma ini hanya menyimpan *frequent itemset* yang ditemukan. Algoritma ini juga direpresentasikan secara *vertical* dalam *database* (untuk setiap item, dibuat *list/array* transaksi) sehingga mengurangi pemakaian memori yang menyebabkan algoritma ini bekerja lebih cepat (Borgelt, 2003)

Proses pencarian dilakukan dari item yang paling sering muncul hingga yang paling jarang muncul tanpa harus memperhatikan urutan, sehingga proses pemindahan tidak perlu dilakukan secara berulang-ulang. Data setiap *itemset* disimpan disebut *Transaction Id List (TID List)*, kemudian *TID List* diurutkan berdasarkan transaksi yang mengandung *itemset* yang sama (*frequent itemset*). Selanjutnya *k-itemset* diatur ke dalam kelas-kelas berdasarkan kriteria tertentu yang terbentuk dengan mempartisi suatu himpunan (*equivalence class*), *(k+1)-itemset* bisa didapat dengan menggabungkan pasangan *frequent k-itemset* dari kelas yang sama. Dalam prosesnya, algoritma ini dilakukan secara rekursif, dimana pencarian *itemset* akan terus dilakukan sepanjang masih ada *itemset* yang tersisa (pencarian menyeluruh).

2.10.3 Association Rules

Association Rules merupakan suatu metode data mining untuk mencari pola asosiasi yang sering muncul dalam data. Metode ini menjadi populer karena sering digunakan untuk suatu data. *Association analysis* mengidentifikasi hubungan antara observasi dan variable dari suatu kumpulan data.

Hubungan ini dinyatakan oleh suatu kumpulan aturan yang menunjukkan kelompok items yang cenderung berhubungan dengan yang lainnya (Normah, 2020). Asosiasi terdapat istilah *antecedent* dan *consequent*. *Antecedent* untuk mewakili bagian “jika” dan *consequent* untuk mewakili bagian “maka”. Aturan asosiasi yang terbentuk adalah sebuah implikasi atau “if-then-rule” yang didukung oleh data. Bentuk dasar dari aturan asosiasi adalah $A \Rightarrow B$. didefinisikan jika A terjadi dalam transaksi, maka B pun terjadi dalam transaksi yang sama.

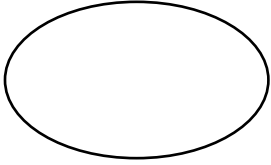


2.11 UML

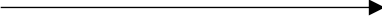

Menurut Sulianta (2017) dalam buku “Teknik Perancangan Arsitektur Sistem Informasi” mengatakan bahwa “Unified Modeling Language (UML) merupakan kumpulan diagram-diagram yang sudah memiliki standar untuk membangun perangkat lunak berbasis objek.”

Menurut Sukamto dan Shalahuddin (2018:133), mendiskusikan bahwa, “UML merupakan sebuah standar bahasa yang digunakan untuk menganalisis dan merancang serta menggambarkan arsitektur program dalam pemrograman object oriented.”

2.11.1 Use Case

Tabel 2. 1 Simbol *Use Case Diagram* (Yunahar, 2018)

Simbol	Deskripsi
<p data-bbox="480 445 603 477">Use Case</p> 	<p data-bbox="810 448 1385 770">Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar antar unit atau actor biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case.</p>
<p data-bbox="504 815 579 846">Actor</p> 	<p data-bbox="810 817 1358 1285">Orang proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari actor adalah gambar orang, biasanya dinyatakan menggunakan kata benda di awal frase nama actor.</p>
<p data-bbox="488 1330 595 1361">Asosiasi</p> 	<p data-bbox="810 1332 1358 1509">Komunikasi antar aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.</p>
<p data-bbox="491 1554 592 1585">Extensi</p> <p data-bbox="416 1626 667 1657">----<<extend>>----</p>	<p data-bbox="810 1556 1350 1800">Relasi use case tambahkan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan.</p>

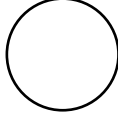
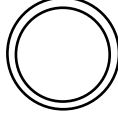
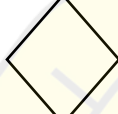

<p style="text-align: center;">Generalisasi</p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya.</p>
<p style="text-align: center;">Include</p> 	<p>Relasi use case tambahkan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini.</p>

Use Case Diagram merupakan diagram yang bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah sistem dengan suatu sistem tersendiri melalui sebuah cerita bagaimana sebuah sistem dipakai (T. Bayu Kurniawan dan Syarifuddin, 2020).

Use Case diagram terdiri dari aktor dan interaksi yang dilakukannya, aktor tersebut dapat berupa manusia, perangkat keras, system lain, atau mereka yang berinteraksi dengan sistem.

2.11.2 Activity Diagram

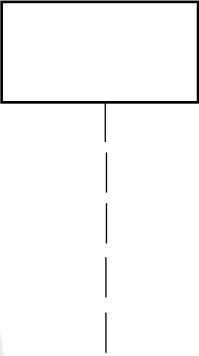

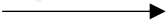
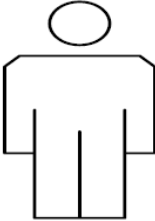
Tabel 2. 2 Simbol Activity Diagram (Yunahar, 2018)

Simbol	Deskripsi
	Simbol start untuk menyatakan awal dari suatu proses
	Simbol stop untuk menyatakan akhir dari suatu proses
	Simbol desticion digunakan untuk menyatakan kondisi dari suatu proses
	Simbol action menyatakan aksi yang dilakukan dalam suatu arsitektur sistem

Activity Diagram menggambarkan work flow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas dapat dilakukan oleh sistem” (Yunahar, 2018).

2.11.3 Sequence Diagram

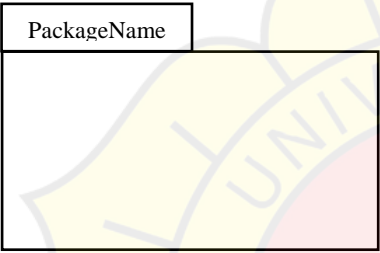
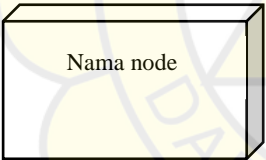

Tabel 2. 3 Simbol *Sequence Diagram* (Yunahar, 2018)


Nama Komponen	Keterangan	Simbol
Lifeline	Mengindikasikan keberadaan sebuah objek dalam basis waktu. Notasi untuk lifeline adalah garis putus-putus vertical yang ditarik dari sebuah objek	
Activation	Dinotasikan sebagai sebuah kotak segi empat Digambar pada sebuah lifeline mengidentifikasi sebuah objek yang akan melakukan sebuah aksi	
Message	Digambarkan dengan anak panah horizontal antara activation. Message mengidekasikan komunikasi antara objek-objek.	
Actor	Actor juga dapat berkomunikasi	

“Sequence Diagram adalah tool yang sangat populer dalam pengembangan sistem informasi secara object-oriented untuk menampilkan interaksi antar objek. Berdasarkan definisi tersebut, dapat disimpulkan bahwa Sequence Diagram adalah tool yang digunakan dalam pengembangan system” (Yunahar, 2018).

2.11.4 Deployment Diagram

Tabel 2. 4 Simbol Deployment Diagram (Rosa dan M. Shalahuddin, 2018)

Simbol	Deskripsi
<p data-bbox="438 763 549 792">Package</p> 	<p data-bbox="810 763 1385 869">Package merupakan sebuah bungkus dari satu atau lebih node</p>
<p data-bbox="459 1077 528 1106">Node</p> 	<p data-bbox="810 1077 1385 1630">Biasanya mangacu pada perangkat keras (Hardware), perangkat lunak yang tidak dibuat sendiri (Software), jika di dalam node disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen</p>
<p data-bbox="411 1740 576 1769">Dependency</p> 	<p data-bbox="810 1740 1385 1845">Kebergantungan antar node, arah panah mengarah pada node yang dipakai</p>

Link 	Relasi antar node
---	-------------------

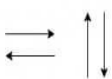







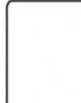



(Rosa dan M. Shalahuddin, 2018) mengemukakan bahwa “Deployment Diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi”. *Deployment Diagram* dapat digunakan untuk menggambarkan hal-hal sebagai berikut, yaitu:

1. Sistem tambahan (embedded system) yang menggambarkan rancangan device, node dan hardware.
2. Sistem *Client* atau *Server*.
3. Sistem terdistribusi murni.
4. Rekayasa ulang aplikasi.

Dapat disimpulkan bahwa *Deployment Diagram* merupakan diagram yang digunakan untuk menggambarkan konfigurasi pada *system*.

2.11.5 Flowchart

Menurut Wibawanto (2017:20) “*Flowchart* adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program.”

	<p>Flow</p> <p>Simbol yang digunakan untuk menggabungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga dengan Connecting Line.</p>		<p>Input/output</p> <p>Simbol yang menyatakan proses input atau output tanpa tergantung peralatan.</p>
	<p>On-Page Reference</p> <p>Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang sama.</p>		<p>Manual Operation</p> <p>Simbol yang menyatakan suatu proses yang tidak dilakukan oleh komputer.</p>
	<p>Off-Page Reference</p> <p>Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang berbeda.</p>		<p>Document</p> <p>Simbol yang menyatakan bahwa input berasal dari dokumen dalam bentuk fisik, atau output yang perlu dicetak.</p>
	<p>Terminator</p> <p>Simbol yang menyatakan awal atau akhir suatu program.</p>		<p>Predefine Proses</p> <p>Simbol untuk pelaksanaan suatu bagian (sub-program) atau prosedur.</p>
	<p>Process</p> <p>Simbol yang menyatakan suatu proses yang dilakukan komputer.</p>		<p>Display</p> <p>Simbol yang menyatakan peralatan output yang digunakan.</p>
	<p>Decision</p> <p>Simbol yang menunjukan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya dan tidak.</p>		<p>Preparation</p> <p>Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal.</p>

Gambar 2. 3 Simbol-simbol Flowchart (Rony Setiawan, 2021)



TEKNOLOGI INFORMASI

UNIVERSITAS DARMA PERSADA