

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Berdasarkan penelitian Mikha Dayan Sinaga dan Nita Sari Br. Sembiring (2016) dengan judul “Penerapan Metode *Dempster Shafer* Untuk Mendignosa Penyakit Dari Akibat Bakteri Salmonella” pada penelitian ini penyakit yang dapat diakibatkan oleh bakteri ini adalah foodborne, tifoid, paratifoid, penyakit tersebut dapat disebarkan melalui makanan yang menyebabkan sakit pada organ pencernaan. Sebelum menentukan hasil diagnosa, terlebih dahulu harus menyediakan data-data yang dibutuhkan yang diperoleh dari Dokter berdasarkan gejala umum dari pengalaman pakar. Untuk menegakkan diagnosis tentang penyakit yang diderita maka akan diuji kembali berdasarkan tes darah. Tes darah yang dimaksudkan disini bukan merupakan tes darah lengkap melainkan tes Titer Widal.

Berdasarkan penelitian Laila Septiana(2015) dengan judul “Metode *Dempster-Shafer* Untuk Sistem Pakar Deteksi kerusakan Mesin Cuci Berbasis *Web*” pada penelitian bertujuan untuk mempermudah perusahaan yang memiliki keterbatasan aset sumber daya manusia yang memiliki kemampuan kepakaran disuatu bidang dan adanya kemungkinan seorang ahli atau pakar tersebut berpindah tempat kerja, maka ada resiko bagi perusahaan tersebut kehilangan ahli untuk mendeteksi kerusakan mesin cuci.

Berdasarkan penelitian Ekka Pujo Ariesanto Akhmad dan M. Taufik(2010) dengan judul “Pengembangan Sistem Pakar Untuk Diagnosis Kerusakan Mesin Diesel” tujuan dari penelitian ini untuk mensubsitusikan pengetahuan pakar tentang mesin diesel ke dalam bentuk sistem, sehingga dapat digunakan oleh orang banyak dan membuat perangkat lunak yang akan membantu diagnosis pada sistem pelumasan, sistem penjalan (*start*), sistem pendinginan, sistem bahan bakar, dan sistem gas buang mesin diesel bila terdapat masalah atau kegagalan.

Berdasarkan penelitian Nita Sari Br. Sembiring dan Dayan Sinaga (2017) dengan judul ”Penerapan Metode *Dempster Shafer* Untuk Mendiagnosa Penyakit Dari Akibat Bakteri *Treponema Pallidum*” berdasarkan penelitian dan pembahasan yang dilakukan, maka sistem pakar yang dirancang dapat mendiagnosa penyakit yang diakibatkan oleh bakteri *Treponema Pallidum*. Kecepatan dalam mendiagnosa penyakit akan membantu praktisisi kesehatan di bawah Dokter untuk mendiagnosa penyakit melalui bantuan *rule-rule* yang disediakan, semakin banyak rule-rule yang dimasukkan maka tingkat keakurasian dan ketepatan hasil diagnosa semakin tepat. Metode *Dempster Shafer* bisa digunakan untuk melakukan perhitungan kemungkinan untuk mendapatkan hasil diagnosa penyakit berdasarkan gejala-gejala yang ada.

Berdasarkan penelitian Yudi (2018) dengan judul ”Sitem Pakar Diagnosa Penyakit Yang Disebabkan Oleh Hewan Ptozoa Metode *Dempster Shafer*” bertujuan untuk menerapkan bagian kecerdasan buatan yaitu sistem pakar untuk mengetahui penyakit yang disebabkan oleh hewan protozoa *Dempster Shafer*. Protozoa adalah makhluk ber-sel tunggal, beberapa diantaranya menyebabkan

penyakit pada tubuh manusia. Hampir semua manusia memiliki protozoa dalam tubuh mereka di beberapa titik dalam tubuh mereka. Namun, protozoa tertentu yang pernah penyakit ringan atau tanpa menjadi mengancam nyawa. Protozoa ini ditemukan pada paru-paru kebanyakan orang yang sehat. Namun, pada pasien dengan “AIDS” dapat menyebabkan pneumonia yang fatal. Sistem pakar ini nantinya diharapkan mampu memberi pengetahuan kepada masyarakat mengenai hewan protozoa dan penyakit yang disebabkan dan juga dapat mengetahui gejala-gejala yang ditimbulkannya, sehingga dapat ditanggulangi dengan cepat.

II.2. Landasan Teori

II.2.1. Sistem

Sistem adalah kumpulan objek seperti orang, sumber daya, konsep, dan prosedur yang dimaksudkan untuk melakukan suatu fungsi yang dapat diidentifikasi atau untuk melayani suatu tujuan. (Mikha Dayan Sinaga, 2018).

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran yang tertentu yang merupakan jaringan kerja dari prosedur lebih menekankan urutan-urutan operasi didalam sistem. (Japerson Hutahean, 2014).

II.2.2. Pakar

Pakar adalah ahli atau orang yang menguasai bidang tertentu. Seorang pakar harus memiliki kemampuan menyelesaikan permasalahan pada bidang

tertentu yang ditanganinya, kemudian memberikan penjelasan mengenai hasil dan kaitannya dengan permasalahan yang ada. Untuk meniru kepakaran seorang manusia, perlu dibangun sebuah sistem komputer yang menunjukkan seluruh karakteristik tersebut. Namun hingga saat ini, pekerjaan dibidang sistem pakar terfokus pada aktifitas penyelesaian masalah dan memberikan penjelasan mengenai solusinya. (Mikha Dayan Sinaga, 2018).

Pakar adalah orang yang mempunyai keahlian dalam bidang tertentu, yaitu pakar yang mempunyai knowledge atau kemampuan khusus yang orang lain tidak mengetahui atau mampu dalam bidang yang dimilikinya. Seorang pakar dengan sistem pakar mempunyai banyak perbedaan. (Mario Nur De Laurdes Making, dan Edi Iskandar, 2016).

II.2.3. Sistem Pakar

Secara umum, sistem pakar (*Expert System*) adalah sistem yang berupaya mengadopsi pengetahuan manusia ke dalam komputer, sehingga komputer dapat menyelesaikan masalah seperti yang biasanya dilakukan oleh para ahli [1]. Dengan sistem pakar ini, orang awam dapat memecahkan masalah yang cukup kompleks yang hanya bisa diselesaikan dengan bantuan para ahli. Tujuan pengembangan sistem pakar bukan untuk menggantikan peran para ahli, tetapi untuk mengimplementasikan pengetahuan para ahli dalam bentuk perangkat lunak, sehingga dapat digunakan oleh banyak orang dan tanpa biaya besar. (Wirhan Fahrozi, dkk, 2018).

Sistem pakar merupakan cabang dari *Artificial Intelligent* (AI). Implementasi sistem pakar banyak digunakan untuk kepentingan komersial karena sistem pakar dipandang sebagai cara penyimpanan pengetahuan pakar dalam bidang tertentu ke dalam program sehingga komputer dapat memberikan keputusan dan melakukan penalaran secara cerdas. Kecerdasan buatan sebagaimana telah diketahui, saat ini merupakan suatu inovasi baru dalam bidang ilmu pengetahuan. Kecerdasan buatan atau *Artificial Intelligence* merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. Teknologi kecerdasan buatan dipelajari dalam bidang-bidang, seperti: robotika, penglihatan komputer (*computervision*), jaringan saraf tiruan (*artificial neural system*), pengolahan bahasa alami (*natural language processing*), pengenalan suara (*speech recognition*) dan sistem pakar (*expert system*). (Mikha Dayan Sinaga, 2018).

II.2.4. Deteksi

Deteksi adalah suatu proses untuk memeriksa atau melakukan pemeriksaan terhadap sesuatu dengan menggunakan cara dan teknik tertentu. Deteksi dapat digunakan untuk berbagai masalah, misalnya dalam sistem pendeteksi suatu penyakit, dimana sistem mengidentifikasi masalah-masalah yang berhubungan dengan penyakit yang biasa disebut gejala. Tujuan dari deteksi adalah memecahkan suatu masalah dengan berbagai cara tergantung metode yang diterapkan sehingga menghasilkan sebuah solusi. (Titatjahya, 2014).

II.2.5. Boiler (Ketel Uap)

Boiler adalah bejana tertutup dimana panas pembakaran dialirkan ke air sampai terbentuk air panas atau steam. Air panas atau steam pada tekanan tertentu kemudian digunakan untuk mengalirkan panas ke suatu proses. Air adalah media yang berguna dan murah untuk mengalirkan panas kesuatu proses. Jika air dididihkan sampai menjadi steam, volumenya akan meningkat sekitar 1.600 kali, menghasilkan tenaga yang menyerupai bubuk mesiu yang mudah meledak, sehingga boiler merupakan peralatan yang harus dikelola dan dijaga dengan sangat baik. (Winanti.W,S dan T, Prayudi, 2006).

Boiler merupakan bejana tertutup yang terbuat dari baja yang berfungsi memindahkan panas yang dihasilkan pembakaran bahan bakar ke air yang pada akhirnya akan menghasilkan uap dan digunakan untuk proses diluar boiler itu sendiri, seperti pemanas, penggerak turbin, dan sebagainya. Pemakaian dan perawatan boiler yang baik akan membuat efisiensi boiler semakin tinggi dan menghemat biaya operasional secara umum. Berbagai usaha dapat dilakukan untuk menghemat biaya produksi uap diantaranya dengan penambahan peralatan guna memperbesar efisiensi dan bahkan adanya penggantian jenis bahan bakar yang digunakan. (Agus Sugiharto, 2010).

II.2.6. Metode *Dempster Shafer*

Dempster-Shafer adalah suatu teori matematika untuk pembuktian berdasarkan belief functions and plausible reasoning (fungsi kepercayaan dan pemikiran yang masuk akal), yang digunakan untuk mengkombinasikan potongan

informasi yang terpisah (bukti) untuk mengkalkulasi kemungkinan dari suatu peristiwa. Teori ini dikembangkan oleh Arthur P. Dempster dan Glenn Shafer. Secara umum teori Dempster-Shafer ditulis dalam suatu interval [Belief, Plausibility]. (Triara Puspitasari 2016).

II.2.7. Perhitungan Metode *Dempster Shafer*

Belief menunjukkan ukuran kekuatan evidence dalam mendukung suatu hipotesis. Plausibility menunjukkan keadaan yang bisa dipercaya. Keterkaitan antara plausibility dan belief dapat dituliskan.

$$Pl(H) = 1 - Bel(\neg H)$$

Adapun fungsi *belief* yang dapat diformulasikan sebagai berikut:

$$Bel(X) = \sum_{X \subseteq Y} m(Y) \dots \dots \dots (1)$$

Sedangkan, *Plausibility* (Pls) dinotasikan sebagai berikut:

$$Pls(X) = 1 - Bel(X') = 1 - \sum_{X \subseteq Y} m(X') \dots \dots \dots (2)$$

Dimana:

$$Bel(X) = Belief(X)$$

$$Pls(X) = Plausibility(X)$$

$$m(X) = \text{mass function dari } (X)$$

$$m(Y) = \text{mass function dari } (Y)$$

Plausibility juga bernilai 0 sampai 1, jika kita yakin akan X' maka dapat dikatakan $Belief(X') = 1$ sehingga dari rumus di atas nilai $Pls(X) = 0$.

Pada aplikasi sistem pakar dalam satu penyakit terdapat sejumlah evidence yang akan digunakan pada faktor ketidakpastian dalam pengambilan keputusan untuk diagnosa suatu penyakit. Untuk mengatasi sejumlah evidence tersebut pada teori Dempster-Shafer menggunakan aturan yang lebih dikenal dengan Dempster's Rule of Combination.

$$m_3(Z) = \sum m_1(X)m_2(Y) \dots \dots \dots (3)$$

Dimana:

$m_3(Z)$ = mass function dari evidence (Z)

$m_1(X)$ = mass function dari evidence (X)

$m_2(Y)$ = mass function dari evidence (Y)

Secara umum formulasi untuk Dempster's Rule of Combination adalah:

$$m_3(Z) = \frac{\sum_{X \cap Y = Z} m_1(X)m_2(Y)}{1 - k} \dots \dots \dots (4)$$

Dimana:

k = Jumlah evidential conflict.

Besarnya jumlah evidential conflict (k) dirumuskan dengan:

$$k = \sum_{X \cap Y = \emptyset} m_1(X)m_2(Y) \dots \dots \dots (5)$$

Sehingga bila persamaan (5) disubstitusikan ke persamaan (4) akan menjadi:

$$m_3(Z) = \frac{\sum_{X \cap Y = Z} m_1(X)m_2(Y)}{1 - \sum_{X \cap Y = \emptyset} m_1(X)m_2(Y)} \dots \dots \dots (6)$$

Dimana:

$m_3(Z)$ = mass function dari evidence (Z)

$m_1(X)$ = mass function dari evidence (X)

$m_2(Y)$ = mass function dari evidence (Y)

k = jumlah evidential conflict.

II.2.8. Website

Website adalah lokasi di internet yang menyajikan kumpulan informasi sehubungan dengan profil pemilik situs. *Website* adalah suatu halaman yang memuat situs-situs *web page* yang berada di internet yang berfungsi sebagai media penyampaian informasi, komunikasi, atau transaksi. *Web server* merupakan *software* yang memberikan layanan data yang berfungsi menerima permintaan HTTP atau HTTPS dari klien yang dikenal dengan *browser web* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman *web* yang umumnya berbentuk dokumen HTML. (Rulia Puji Hastanti, dkk, 2015).

II.3. PHP

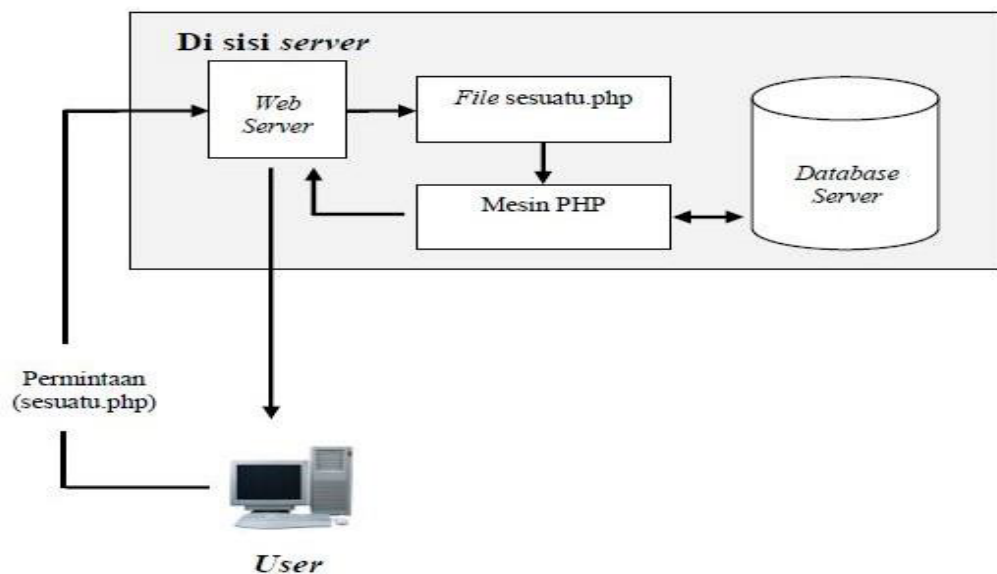
PHP merupakan bahasa *web server-side* yang bersifat *open source*. Bahasa PHP menyatu dengan *script* HTML yang sepenuhnya dijalankan pada *server*. *File* yang hanya berisi kode HTML yang dirancang tidak mendukung pembuatan aplikasi yang melibatkan *database* karena HTML dirancang untuk menyajikan informasi yang bersifat statis (tampilan yang isinya tetap hingga *webmaster* atau penanggung jawab *web* melakukan perubahan isi). Oleh karena itu, muncul pemikiran untuk membuat suatu perantara yang memungkinkan aplikasi bisa menghasilkan sesuatu yang bersifat dinamis dan berinteraksi dengan *database*.

Akhirnya lahirlah berbagai perantara seperti PHP, ASP dan JSP. (Andy Pratama Nugraha, dkk , 2015).

PHP *Hypertext Pre Processor* atau ada juga beberapa anggapan bahwa PHP juga merupakan kependekan dari “*Profesional Home Page*”. PHP dikenal sebagai bahasa *scripting* yang menyatu dengan *tag-tag* HTML yang dieksekusi disisi *server* (*server-side scripting language*), yang berfungsi untuk membuat sebuah *web* yang interaktif dan dinamis seperti halnya bahasa-bahasa *script* lainnya, misalnya *Active Server Pages* (ASP), *Java Server Pages* (JSP), dan *Allaire ColdFusion* (CFM). PHP termasuk jenis bahasa *server-side* yang didesain khusus untuk aplikasi *web*. PHP dapat disisipkan diantara bahasa HTML dan karena bahasa *server-side*, maka bahasa PHP akan dieksekusi di *server* sehingga yang dikirimkan *browser* adalah “hasil jadi” dalam bentuk *format* HTML, dan kode PHP tidak akan terlihat, sehingga dari segi keamanan lebih terjamin. PHP termasuk dalam *open source product*, jadi *source-code* nya dapat dirubah dan didistribusikan secara gratis. PHP juga dapat berjalan diberbagai *web server* contoh pada IIS, *Apache*, PWS dan lain-lain. Perintah dalam PHP dibuka dengan tanda ‘<?php’ dan ditutup dengan tanda ‘?>’, setiap akhir baris diakhiri dengan tanda ‘;’. *Tag* penutup ‘?>’ juga dapat berfungsi sebagai titik koma. Penulisan program PHP untuk *variable* menggunakan *case sensitive* yaitu antara huruf besar dan huruf kecil dibedakan, penulisan *variable* ditandai dengan ‘\$’. (Bagus Primantoro, dan Abdul Fadlil, 2014).

PHP merupakan bahasa *web server-side* yang bersifat *open source*. Bahasa PHP menyatu dengan *scriptHTML* yang sepenuhnya dijalankan pada *server*. *File*

yang hanya berisi kode HTML yang dirancang tidak mendukung pembuatan aplikasi yang melibatkan *database* karena HTML dirancang untuk menyajikan informasi yang bersifat statis (tampilan yang isinya tetap hingga *webmaster* atau penanggung jawab *web* melakukan perubahan isi). Oleh karena itu, muncul pemikiran untuk membuat suatu perantara yang memungkinkan aplikasi bisa menghasilkan sesuatu yang bersifat dinamis dan berinteraksi dengan *database*. Akhirnya lahirlah berbagai perantara seperti PHP, ASP dan JSP (Andy Pratama Nugraha, dkk , 2015).



Gambar II.1. Struktur PHP

(Sumber : Andy Pratama Nugraha, dkk , 2015)

II.3.1. MySQL

MySQL adalah *relational database management system* (RDBMS) yang didistribusikan secara gratis dibawah *license GPL (General Public License)*. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian *database* terutama untuk pemilihan/seleksi dan pemasukan data yng memungkinkan pengoperasian data dikerjakan dengan mudah dan secara otomatis. Keandalan suatu sistem *database* dapat diketahui dari cara kerja *optimizer* nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. Sebagai *database server*, MySQL dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query* MySQL dapat sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan *Interbase*. (Rina Candra Noor Santi, dan Sri Eniyati, 2015).

II.3.2. UML

Unified Modeling Language (UML) merupakan bahasa pemodelan yang berbentuk grafis yang digunakan untuk memvisualisasi, menspesifikasikan suatu sistem perangkat lunak. UML yang digunakan adalah *Use Case Diagram*. *Use case diagram* menjelaskan rancangan alur sistem yang nantinya akan berjalan. Sistem terdiri dari tiga *actor*, yaitu admin dan pengguna (*user*). Pengguna sistem nantinya dapat berkonsultasi mengenai gejala-gejala yang dialami, untuk memperoleh kesimpulan penyakit. (Soeheri, dan Andi Sanjaya, 2018).


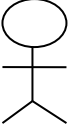
Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. Perancangan desain sistem yang akan dibangun menggunakan pemodelan *Unified Modelling System* (UML). Diagram-diagram yang digunakan *use case diagram, activity diagram, class diagram* dan *sequence diagram*. (Dicky Juliawan, dkk, 2017).

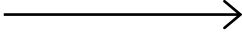
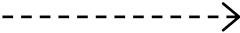
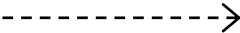
II.3.3. Bagian UML

a. Use Case Diagram

Dalam penyusunan suatu program diperlukan suatu model data yang berbentuk *diagram* yang dapat menjelaskan suatu alur proses sistem yang akan dibangun. Berikut ini merupakan simbol-simbol yang terdapat pada *Use Case Diagram*, dapat dilihat pada tabel II.1 sebagai berikut :

Tabel II.2. Simbol Use Case Diagram






Simbol	Deskripsi
<i>Use Case</i> 	Fungsionalitas yang disediakan <i>system</i> sebagai unit-unit yang saling bertukar pesan antar unit atau <i>actor</i> .
<i>Aktor/Actor</i> 	Orang, proses atau <i>system</i> lain yang berinteraksi dengan <i>system</i> informasi yang akan dibuat diluar <i>system</i> yang akan dibuat itu sendiri.

<i>Relationship/Association</i> 	Dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
<i>Extensi/Extend</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
<i>Include</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya.

b. Activity Diagram

Activity diagram adalah *diagram* yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Berikut ini merupakan simbol-simbol yang terdapat pada *Activity Diagram*, dapat dilihat pada tabel II.2 sebagai berikut :




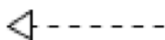
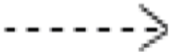

Tabel II.3. Simbol *Activity Diagram*

No	Gambar	Nama	Keterangan
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

c. *Class Diagram*

Class Diagram adalah deskripsi kelompok obyek -obyek dengan *property* perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya *class diagram* dapat memberikan pandangan *global* atas sebuah *system*. Hal tersebut tercermin dari *class-class* yang ada dan relasinya satu dengan yang lainnya. Berikut ini merupakan simbol-simbol yang terdapat pada *ClasssDiagram*, dapat dilihat pada tabel II.3 sebagai berikut :

Tabel II.4. Simbol *Class Diagram*

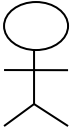
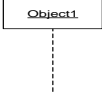


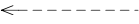
No	Gambar	Nama	Keterangan
1		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
2		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
3		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
4		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
5		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

a. *Sequence Diagram*

Sequence Diagram merupakan salah satu *diagraminteraction* yang menjelaskan bagaimana suatu operasi itu dilakukan, *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. *Diagram* ini diatur berdasarkan waktu. Obyek-obyek yang berkaitan dengan berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.

Berikut ini merupakan simbol-simbol yang terdapat pada *Sequence Diagram*, dapat dilihat pada tabel II.4 sebagai berikut :

Tabel II.5. Simbol *Sequence Diagram*

Simbol	Deskripsi
Aktor/ <i>Actor</i> 	Menggambarkan orang yang berinteraksi dengan <i>system</i>
<i>Object Lifeline</i> 	Menggambarkan interaksi antara satu atau lebih <i>actor</i> dengan <i>system</i> memodelkan bagian dari <i>system</i> yang bergabung ada pihak lain disekitarnya
<i>Message</i> (Masukan) 	Menyatakan bahwa sutau objek mengirimkan data atau masukan atau informasi ke objek lainnya
<i>Activation</i> 	Menyatakan objek dalam keadaan aktif dan berinteraksi, sebuah yang brhubungan dengan <i>activation</i> ini adalah sebuah tahapan yang dilakukan didalamnya
<i>Message</i> (Keluaran) 	Menyatakan bahwa suatu objek sudah menjalankan suatu operasi atau metode menghasilkan kembalian ke objek tertentu

III.3.4. Normalisasi

Normalisasi adalah suatu teknik untuk mengorganisasikan data ke dalam tabel-tabel untuk memenuhi kebutuhan pemakai di dalam suatu organisasi.

Tahapan Normalisasi adalah sebagai berikut :

1. Bentuk Tidak Normal

Bentuk ini merupakan kumpulan data yang akan direkam tidak ada keharusan mengikuti suatu format tertentu, dapat saja data tidak lengkap atau terduplikasi.

2. Bentuk Normal ke-1 (1NF)

Suatu relasi dikatakan dalam bentuk normal pertama jika dan hanya jika memenuhi syarat bahwa relasi tersebut memenuhi nilai-nilai *atomic* (tidak ada atribut yang berulang/ ganda).

3. Bentuk normal ke-2 (2NF)

Setiap atribut yang bukan *key* tergantung penuh secara fungsional pada *primary key* atau itu haruslah sudah ditentukan fungsi dan harus unik.

4. Bentuk Normal ke-3 (3NF)

Relasi haruslah dalam bentuk normal kedua dan semua atribut bukan kunci harus bergantung hanya pada *primary key*-nya.

Tujuan dari Normalisasi:

1. Untuk menghilangkan kerangkapan data.
2. Untuk mengurangi kompleksitas.
3. Untuk mempermudah pemodifikasian data.

Proses Normalisasi :

1. Data diuraikan dalam bentuk *table*, selanjutnya dianalisis berdasarkan persyaratan tertentu ke berapa tingkat.
2. Apabila tabel yang diuji belum memenuhi persyaratan tertentu, maka table tersebut perlu di pecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk optimal (Agus Dwi Putra, dan Tungga Buana Darma, 2017)

