

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Berikut ini beberapa penelitian terkait yang berhubungan dengan penelitian penulis, diantaranya:

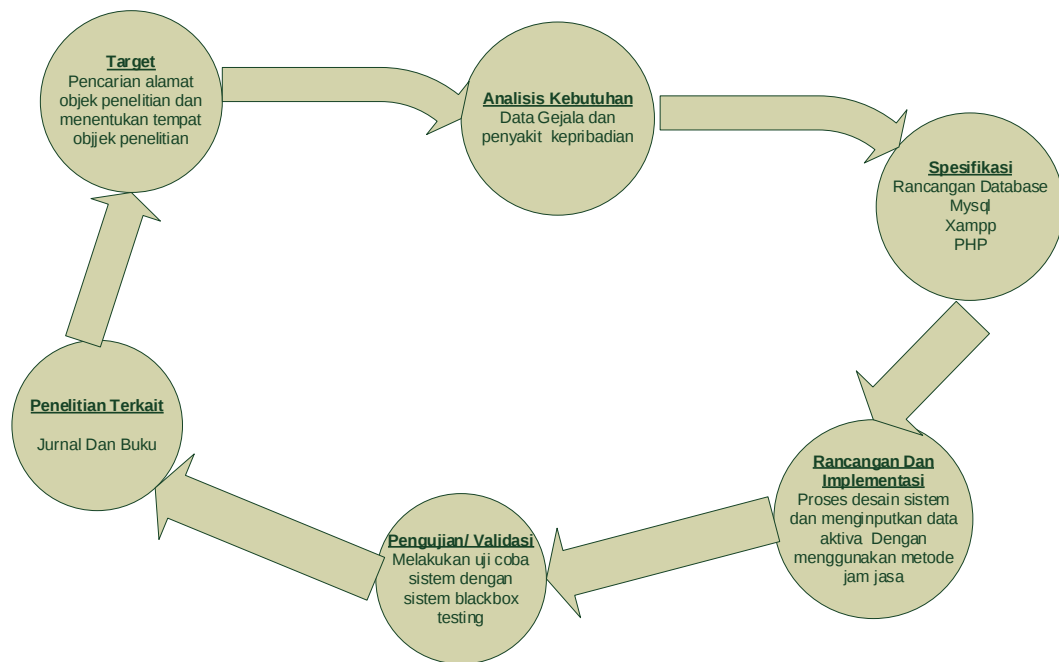
Berdasarkan penelitian dari Sari Murni (2019) dengan judul “Penerapan Metode *Teorema Bayes* Pada Sistem Pakar Untuk Mendiagnosa Penyakit Lambung”, Untuk diagnosa gejala sakit, melalui seorang pakar lambung akan diterapkan sistem pakar dengan metoda *Theorema Bayes*. Perhitungan metode Bayes dalam mendiagnosa penyakit lambung pada sistem pakar dirancang berdasarkan algoritma *Bayes* yaitu perhitungan sesuai dengan gejala-gejala penyakit yang diderita seseorang. Penyakit lambung diberi kode P01, P02, P03 ; gejala penyakit diberi kode G001 s.d G027. Dari perhitungan metode *Bayes* diperoleh Nilai Probabilitas dengan ada tidaknya penggunaan *Theorema Bayes* dalam setiap *range*. Sistem pakar juga memberi *advice* pengobatan medis berdasarkan diagnosa penyakit.

Berdasarkan penelitian dari Yus Hendra (2020) dengan judul “Sistem Pakar Untuk Diagnosa Penyakit Kejiwaan Dengan Menggunakan Metode *Teorema Bayes*” Sistem pakar diagnosa penyakit jiwa, dapat membantu operator klinik didalam penanganan cepat dalam mendiagnosa gejala penyakit kejiwaan Dengan adanya sistem pakar menggunakan metode *teorema bayes* dapat menghitung persentase besar kemungkinan jenis penyakit dari gejala gejala yang dimasukan.

Peneliti (Puji Sari Ramadhan, 2018) melakukan penelitian dengan judul Sistem Pakar Pendiagnosaan Dermatitis Imun Menggunakan *Teorema Bayes*, untuk mendiagnosis penyakit Dermatitis. Dalam penelitian ini, aplikasi sistem pakar diterapkan berbasis *web*. Kesimpulan dari hasil penelitian ini adalah penerapan Metode *Teorema Bayes*, digunakan untuk diagnosa penyakit Dermatitis dengan menurut dari gejala yang diinputkan, serta perhitungan nilai *probabilitas* penyakit dilakukan pada masing-masing penyakit atau hipotesa.

Peneliti (Hengki Tamando Sihotang, 2017) melakukan penelitian dengan judul Perancangan Aplikasi Sistem Pakar Diagnosa Diabetes Dengan Metode *Bayes*. Dalam penelitian ini, aplikasi sistem pakar diterapkan berbasis *web*. Kesimpulan dari hasil penelitian ini adalah penerapan Metode *Bayes*, digunakan untuk diagnosa penyakit Diabetes dengan menurut dari gejala yang diinputkan, serta perhitungan nilai *probabilitas* penyakit dilakukan pada masing-masing penyakit.

Menurut Bahar (2017) dengan judul “Penerapan *Teorema Bayes* Dalam Sistem Pakar Untuk Konsultasi Siswa Bermasalah” penelitian ini bertujuan untuk merancang suatu sistem pakar yang dapat membantu guru pendamping BK dalam memberikan konsultasi terhadap siswa bermasalah dengan menggunakan metode *teorema bayes*. *Teorema bayes* dapat digunakan untuk menghitung probabilitas dari berbagai gejala-gejala yang akan dipilih guru pendamping BK dan memiliki tingkat akurat yang tinggi dalam keberhasilan konseling penanganan. Aplikasi ini nantinya akan menghasilkan *output* berupa permasalahan, informasi masalah, saran atau konseling penanganan yang mana dapat digunakan guru pendamping BK untuk mengatasi masalah yang dihadapi para siswa.



Gambar II.1. Roadmap Penelitian

II.2. Studi Literatur

II.2.1. Data

Dalam kamus bahasa Inggris-Indonesia, data diterjemahkan sebagai istilah yang berasal dari kata “*datum*” yang berarti fakta atau bahan-bahan keterangan. Data merupakan deskripsi dari sesuatu dan kejadian yang kita hadapi. Data adalah fakta yang jelas lingkup, tempat, dan waktunya. (Sutopo et al., 2016).

Data adalah penggambaran suatu fakta atau keadaan. Informasi merupakan sekumpulan data yang telah diolah menjadi suatu bentuk sehingga berguna bagi penerima dan pemakai. Dalam dunia komputer *database* bisa dikategorikan bisa sangat spesial karena selalu menjadi hal utama dalam perancangan sistem suatu perusahaan, tentunya ada alasan tertentu mengapa *database* menjadi prioritas sendiri dalam kinerja manajemen perusahaan. (Herlina, 2016)

II.2.2 Sistem

Kata sistem berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*sustema*) yang artinya adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi. Secara umum sistem adalah kumpulan dari beberapa bagian tertentu yang saling berhubungan secara harmonis untuk mencapai suatu tujuan tertentu. Elemen-elemen yang mewakili suatu sistem secara umum adalah masukan (*Input*) pengolahan (*Processing*) dan keluaran (*Output*). (Sutopo et al., 2016)

Sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lainnya, yang berfungsi bersama-sama untuk mencapai tujuan tertentu. Sistem merupakan rangkaian komponen yang dikoordinasikan untuk mencapai serangkaian tujuan, yang memiliki karakteristik meliputi: komponen, atau sesuatu yang dapat dilihat, didengar atau dirasakan, proses, kegiatan untuk mengkoordinasikan komponen yang terlibat dalam sebuah sistem; tujuan, sasaran akhir yang ingin dicapai dari kegiatan koordinasi komponen tersebut. Dari kedua definisi tersebut, dapat disimpulkan bahwa sistem merupakan suatu komponen-komponen yang saling berhubungan dan berinteraksi untuk mencapai suatu tujuan tertentu. (David Ardian Cahyono, 2018)

Sistem dapat terdiri dari sistem-sistem bagian (*subsystem*). Sistem komputer dapat terdiri dari subsistem perangkat keras dan subsistem perangkat lunak. Masing-masing subsistem dapat terdiri dari subsistem-subsistem yang lebih kecil atau terdiri dari komponen-komponen. Subsistem perangkat keras (*hardware*) dapat terdiri dari alat masukan, alat pemroses, alat keluaran dan simpanan luar, dan

kemudian subsistem-subsistem tersebut akan berinteraksi sedemikian rupa sehingga dapat mencapai satu kesatuan yang terpadu. Dalam buku *Analisa dan Design Sistem Informasi* pendekatan terstruktur teori dan praktek aplikasi bisnis. Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu. (Deppi Linda, 2019).

II.2.2.1. Karakteristik Sistem

Untuk memahami atau mengembangkan suatu sistem, maka perlu membedakan unsur-unsur dari sistem yang membentuknya. Berikut adalah karakteristik sistem yang dapat membedakan suatu sistem dengan sistem lainnya yaitu:

1. Batasan (*Boundary*) adalah Penggambaran dari suatu elemen atau unsur mana yang termasuk di dalam sistem dan mana yang di luar sistem.
2. Lingkungan (*Environment*) adalah Segala sesuatu di luar sistem, lingkungan yang menyediakan asumsi, kendala, dan input terhadap suatu sistem.
3. Masukan (*Input*) adalah Sumber daya (data, bahan baku, peralatan, energi) dari lingkungan yang dikonsumsi dan dimanipulasi oleh suatu sistem.
4. Keluaran (*Output*) adalah Sumber daya atau produk (informasi, laporan, dokumen, tampilan layar Komputer, barang jadi) yang disediakan untuk lingkungan sistem oleh kegiatan dalam suatu sistem.

5. Komponen (*Component*) adalah Kegiatan-kegiatan atau proses dalam sistem yang mentransformasikan input menjadi bentuk setengah jadi (*output*). Komponen ini bisa merupakan subsistem dari sebuah sistem.
6. Penghubung (*Interface*) adalah Tempat di mana komponen atau sistem dan lingkungannya bertemu atau berinteraksi.
7. Penyimpanan (*Storage*) adalah Area yang dikuasai dan digunakan untuk penyimpanan sementara dan tetap dari informasi, energi, bahan baku, dan sebagainya. Penyimpanan merupakan suatu media penyangga di antara komponen tersebut bekerja dengan berbagai tingkatan yang ada dan memungkinkan komponen yang berbeda dari berbagai data yang sama (David Ardian Cahyono, 2018)

II.2.3. Informasi

Informasi dapat diibaratkan sebagai darah yang mengalir didalam tubuh manusia, seperti halnya informasi di dalam sebuah perusahaan yang sangat penting untuk mendukung kelangsungan perkembangannya, Sehingga terdapat alasan bahwa informasi sangat dibutuhkan bagi sebuah perusahaan. Akibat bila kurang mendapatkan informasi, dalam waktu tertentu perusahaan akan mengalami ketidakmampuan mengontrol sumber daya, sehingga dalam mengambil keputusan-keputusan strategis sangat terganggu, yang pada akhirnya akan mengalami kekalahan dalam bersaing dengan lingkungan pesaingnya. Disamping itu, sistem informasi yang dimiliki seringkali tidak dapat bekerja dengan baik. Masalah

utamanya adalah bahwa sistem informasi tersebut terlalu banyak informasi yang tidak bermanfaat atau berarti sistem terlalu banyak data. (Anwar et al., 2016)

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

1. Akurat (*Accurate*)

Berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya.

2. Tepat waktu (*Timelines*)

Berarti informasi yang datang pada si penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai logika karena informasi merupakan landasan dalam pengambilan keputusan.

3. Relevan (*Relevance*)

Berarti informasi tersebut bermanfaat bagi pemakainya (Han & goleman, daniel; boyatzis, Richard; Mckee, 2019).

II.2.4. Sistem Informasi

Sistem informasi adalah sejumlah komponen (manusia, komputer, teknologi informasi, dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi), dan dimaksudkan untuk mencapai suatu sasaran atau tujuan. Penggunaan sistem informasi telah banyak diterapkan diberbagai bidang termasuk dalam bisnis. Salah satu tujuan penerapan sistem informasi dalam bidang bisnis agar dapat meningkatkan keuntungan bisnis dengan menggunakan

kemampuan yang didapatkan dari sistem informasi. Ada beberapa kemampuan dari sistem informasi yang dapat mendukung dalam bidang bisnis. Kemampuan tersebut seperti pengurangan biaya, mempercepat pekerjaan, dapat meningkatkan kemudahan dalam pengambilan keputusan, dan peningkatan pelayanan terhadap pelanggan. (Nurlifa & Kusumadewi, 2017)

Sistem informasi adalah sekumpulan komponen-komponen sistem yang berada di dalam suatu ruang lingkup organisasi, saling berinteraksi untuk menghasilkan sebuah informasi yang bertujuan untuk pihak manajemen tertentu dan untuk mencapai tujuan tertentu. (I Made et al., n.d., 2016)

Faktor-faktor yang menentukan keandalan dari suatu sistem informasi atau informasi dapat dikatakan baik jika memenuhi kriteria-kriteria sebagai berikut :

a) Keunggulan (*Usefulness*)

Yaitu suatu sistem yang harus dapat menghasilkan informasi yang tepat dan relevan untuk mengambil keputusan manajemen dan personil operasi dalam organisasi.

b) Ekonomis

Kemampuan sistem yang mempengaruhi sistem harus bernilai manfaat minimal, sebesar biayanya.

c) Keandalan (*Reliability*)

Keluaran dari sistem harus mempunyai tingkat ketelitian tinggi dan sistem tersebut harus beroperasi secara efektif.

d) Pelayanan (*Customer Service*)

Yakni suatu sistem memberikan pelayanan yang baik dan efisien kepada para pengguna sistem pada saat berhubungan dengan organisasi.

e) Kapasitas (*Capacity*)

Setiap sistem harus mempunyai kapasitas yang memadai untuk menangani setiap periode sesuai yang dibutuhkan.

f) Sederhana dalam kemudahan (*Simplicity*)

Sistem tersebut lebih sederhana (umum) sehingga struktur dan operasinya dapat dengan mudah dimengerti dan prosedur mudah di ikuti.

g) Fleksibel (*Fleksibility*)

Sistem informasi ini harus dapat digunakan dalam kondisi yang bagaimana yang diinginkan oleh organisasi tersebut atau pengguna tertentu.

h) Komponen Sistem Informasi

Istilah dalam komponen sistem informasi adalah blok bangunan (building block) yang dapat dibagi menjadi enam blok yaitu :

a. Blok masukan (*Input block*)

Blok *input* merupakan data–data yang masuk ke dalam sistem informasi, yang dapat berupa *document-document* dasar yang dapat diolah menjadi suatu informasi tertentu.

b. Blok model (*Model block*)

Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan mengolah data *input* untuk menghasilkan suatu informasi yang dibutuhkan.

c. Blok keluaran (*Output block*)

Merupakan informasi yang menghasilkan sekumpulan data yang nantinya akan disimpan berupa data cetak laporan.

d. Blok teknologi (*Technology block*)

Blok teknologi merupakan penunjang utama dalam berlangsungnya sistem informasi. Yang memiliki beberapa komponen yaitu diantaranya alat memasukkan data (*input device*), alat untuk menyimpan dan mengakses data (*storage device*), alat untuk menghasilkan dan mengirimkan keluaran (*output device*) dan alat untuk membentuk pengendalian sistem secara keseluruhan (*control device*). Teknologi informasi terdiri dari 3 (tiga) bagian utama, yaitu teknisi (*humanware* atau *brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*).

e. Blok basis data (*Database block*)

Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Data perlu di simpan dan perlu di organisasi sedemikian rupa, supaya informasi yang dihasilkan berkualitas.

f. Blok kendali (*Control block*)

Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal–hal yang dapat merusak sistem dapat dicegah bila terlanjur terjadi.

(I Made et al., n.d., 2016)

II.2.5 Sistem Pakar

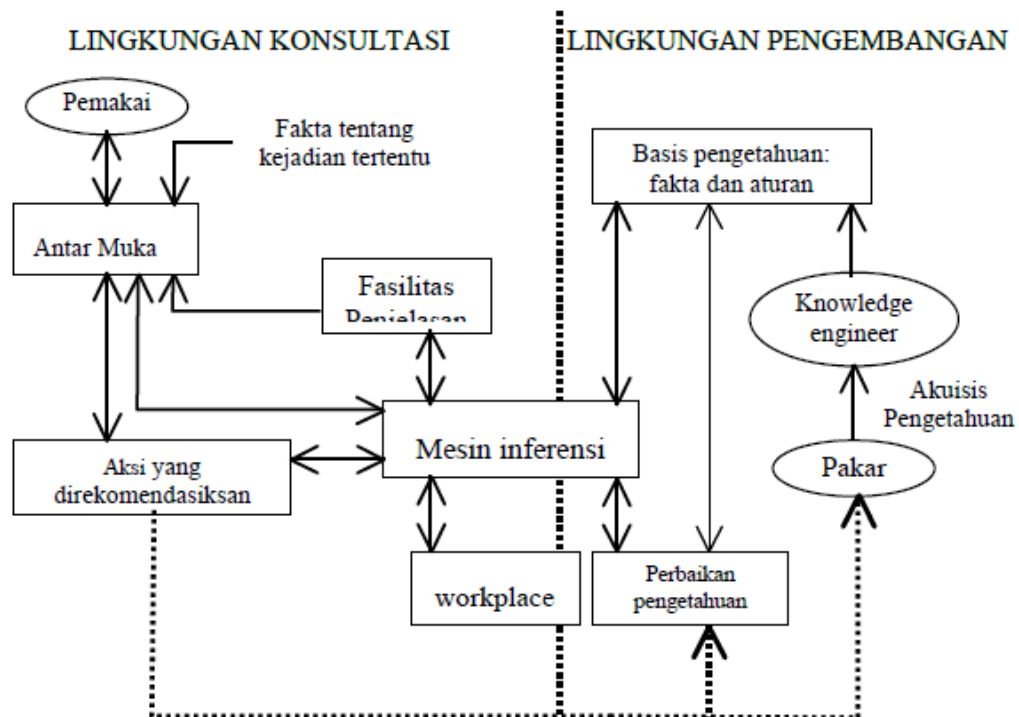
buatan yang mempunyai kemampuan khusus untuk menyelesaikan kondisi permasalahan yang ada. Dalam penelitian selanjutnya dikemukakan bahwa Sistem pakar adalah pengembangan dari sistem cerdas berbasis aplikasi. Pendapat lainnya yang mengemukakan, Sistem Pakar adalah hasil dari pengetahuan pakar dan teknik pencarian. Definisi lain yang menyebutkan Sistem Pakar untuk diimplementasikan dalam melakukan pemecahan masalah dan pengambilan kesimpulan dengan dasar pengetahuan pakar. Dalam referensi lainnya, mengemukakan Sistem Pakar adalah bagian yang terdapat dalam kecerdasan buatan yang diperuntukkan dalam pendagnosis kerusakan sistem dan sebagai solusi permasalahan. (Puji Sari Ramadhan : 2018)

Sistem pakar merupakan suatu bagian ilmu-ilmu artificial intelligence untuk dibuat suatu program aplikasi diagnosa penyakit pada manusia yang terkomputerisasi serta berusaha menggantikan dan menirukan proses penalaran dari seorang ahlinya atau pakar dalam memecahkan masalah spesifikasi yang dapat dikatakan duplikat dari seorang pakar karena pengetahuan ilmu tersebut tersimpan di dalam suatu sistem database. Secara umum sistem pakar adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer yang dirancang untuk memodelkan kemampuan menyelesaikan masalah seperti layaknya pakar. Sistem Pakar (*Expert System*) adalah salah satu cabang AI yang membuat penggunaan secara luas *knowledge* yang khusus untuk penyelesaian masalah tingkat manusia yang pakar. Komputer berbasis sistem pakar adalah program komputer yang mempunyai pengetahuan yang berasal dari manusia yang berpengetahuan luas atau

pakar dalam domain tertentu, di mana pengetahuan di sini adalah pengetahuan manusia yang sangat minim penyebarannya, mahal serta susah didapat. (Khairani Puspita : 2018)

II.2.5.1. Struktur Sistem Pakar

Adapun definisi Struktur Sistem Pakar dua bagian penting dari sistem pakar, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan digunakan oleh pembuat sistem pakar untuk membangun komponen-komponennya dan memperkenalkan pengetahuan dalam *knowledge base* (basis pengetahuan). Lingkungan konsultasi digunakan oleh pengguna untuk berkonsultasi sehingga pengguna mendapatkan pengetahuan dan nasihat dari sistem pakar layaknya berkonsultasi dengan seorang pakar (Januardi Nasir, 2018)



Gambar II.2. Struktur Sistem Pakar
Sumber : Januardi : 2018

II.2.5.2. Komponen Sistem Pakar

Sistem Pakar terdiri dari 3 komponen utama, yaitu: basis pengetahuan (*knowledge bases*), motor inferensi dan interface.

a. Representasi Pengetahuan (*Knowledge Base*)

Representasi pengetahuan adalah metode yang digunakan untuk mengkodekan pengetahuan dalam sebuah sistem pakar yang berbasis pengetahuan. Representasi pengetahuan dimaksudkan untuk menangkap sifat-sifat penting problema dan membuat informasi tersebut dapat diakses oleh prosedur pemecahan masalah.

b. Antarmuka Pengguna

Antarmuka pengguna adalah perangkat lunak yang menyediakan media komunikasi antara pengguna dengan sistem. *User Interface* merupakan mekanisme yang digunakan oleh pengguna dan sistem pakar untuk berkomunikasi. Antarmuka menerima informasi dari pemakai dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem. Selain itu, antarmuka menerima informasi dari sistem dan menyajikannya ke dalam bentuk yang dapat dimengerti oleh pemakai.

c. Mesin Inferensi (motor inferensi)

Mesin inferensi adalah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam *workplace* dan untuk memformulasikan kesimpulan. (eri Fahrur Rohman: 2018)

II.2.6. Metode *Teorema Bayes*

Dalam teori probabilitas dan statistika, *Teorema Bayes* adalah sebuah teorema dengan dua penafsiran berbeda. Dalam penafsiran *Bayes*, teorema ini menyatakan seberapa jauh derajat kepercayaan subjektif harus berubah secara rasional ketika ada petunjuk baru. Dalam penafsiran frekuensi teorema ini menjelaskan representasi invers probabilitas dua kejadian. (Puji Sari Ramadhan : 2019)

Teorema bayes dikemukakan oleh seorang pendeta Presbyterian inggris pada tahun 1763 yang bernama Thomas Bayes ini kemudian disempurnakan

Laplace. Teorema bayes digunakan untuk menghitung probabilitas terjadinya suatu peristiwa berdasarkan pengaruh yang didapat dari hasil observasi. Disamping ini metode *bayes* memanfaatkan data sampel yang diperoleh dari populasi juga memperhitungkan suatu distribusi awal yang disebut distribusi prior. Metode *bayes* juga memandang parameter sebagai *variable* yang menggambarkan pengetahuan awal tentang parameter sebelum pengamatan dilakukan dan dinyatakan dalam suatu distribusi yang disebut dengan distribusi prior. Setelah pengamatan dilakukan, informasi dalam distribusi prior dikombinasikan dengan data sampel melalui *teorema bayes*. (Trinanda Syahputra, 2017)

Misalkan kawan Anda bercerita dia bercakap-cakap akrab dengan seseorang lain di atas kereta api. Tanpa informasi tambahan, peluang dia bercakap-cakap dengan perempuan adalah 50%. Sekarang misalkan kawan Anda menyebut bahwa orang lain di atas kereta api itu berambut panjang. Dari keterangan baru ini tampaknya lebih boleh jadi kawan Anda bercakap-cakap dengan perempuan, karena orang berambut panjang biasanya wanita. Teorema Bayes dapat digunakan untuk menghitung besarnya peluang bahwa kawan Anda berbicara dengan seorang wanita, bila diketahui berapa peluang seorang wanita berambut panjang.

Seperti :

- a) W adalah kejadian percakapan dilakukan dengan seorang wanita.
- b) L adalah kejadian percakapan dilakukan dengan seorang berambut panjang
- c) M adalah kejadian percakapan dilakukan dengan seorang pria

Kita dapat berasumsi bahwa wanita adalah setengah dari populasi. Artinya peluang kawan Anda berbicara dengan wanita,

$$P(W) = 0,5 \dots\dots\dots(1)$$

Misalkan juga bahwa diketahui 75 persen wanita berambut panjang. Ini berarti bila kita mengetahui bahwa seseorang adalah wanita, peluangnya berambut panjang adalah 0,75. Kita melambangkannya sebagai:

$$P(L|W) = 0,75 \dots\dots\dots(2)$$

Sebagai keterangan tambahan kita juga mengetahui bahwa peluang seorang pria berambut panjang adalah 0,3. Dengan kata lain:

$$P(L|M) = 0,3 \dots\dots\dots(3)$$

Di sini kita mengasumsikan bahwa seseorang itu adalah pria atau wanita, atau $P(M) = 1 - P(W) = 0,5$. Dengan kata lain M adalah kejadian komplemen dari W.

Tujuan kita adalah menghitung peluang seseorang itu adalah wanita bila diketahui dia berambut panjang, atau dalam notasi yang kita gunakan, $P(W|L)$. Menggunakan teorema Bayes, kita mendapatkan:

$$P(W|L) = \frac{P(L|W)P(W)}{P(L|W)P(W) + P(L|M)P(M)} \dots\dots\dots(4)$$

Di sini kita menggunakan aturan peluang total. Dengan memasukkan nilai-nilai peluang yang diketahui ke dalam rumus di atas, kita mendapatkan peluang seseorang itu wanita bila diketahui dia berambut panjang adalah 0,714. Angka ini sesuai dengan intuisi awal kita, bahwa peluang kawan kita itu bercakap-cakap dengan wanita meningkat. (Puji Sari Ramadhan : 2018)

II.2.7. *Unified Modeling Language (UML)*

Unified Modelling Language (UML) merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek. UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan Bahasa pemodelan visual yang memungkinkan pengembang sistem membuat *blue print* atas visinya dalam bentuk yang baku. UML berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui jumlah elemen grafis yang bisa dikombinasikan menjadi.

Unified Modeling Language (UML) biasa digunakan untuk :

- a. Menggambarkan batasan sistem dan fungsi - fungsi sistem secara umum, dibuat dengan *use case* dan *actor*.
- b. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.
- c. Menggambarkan representasi struktur *static* sebuah sistem dalam bentuk *class diagrams*.
- d. Membuat model behavior “yang menggambarkan kebiasaan atau sifat sebuah sistem” dengan *state transition diagrams*.
- e. Menyatakan arsitektur implementasi fisik menggunakan *component and development*.
- f. Menyampaikan atau memperluas *fungsi* dengan *stereotypes*. (Alfina & Harahap, 2019)

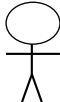
Pemodelan penggunaan UML merupakan metode pemodelan berorientasi objek dan berbasis visual. Karenanya pemodelan objek yang fokus pada pendefinisian struktur statis dan model sistem informasi yang dinamis daripada mendefinisikan data dan model proses yang tujuannya adalah pengembangan tradisional. UML menawarkan diagram yang dikelompokkan menjadi lima perspektif berbeda untuk memodelkan suatu sistem. Seperti satu set *blue print* yang digunakan untuk membangun sebuah rumah (Anwar et al., 2016)

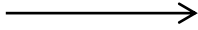
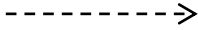
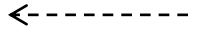
II.2.7.1. Use Case Diagram

Use case diagram merupakan pemodelan untuk sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Kawano et al., 2016)

Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.1 dibawah ini :

Tabel II.1 Simbol Use Case

Gambar	Keterangan	Deskripsi
	<i>Use case</i>	Menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor	Sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem.
	Asosiasi	Penghubung antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah

		yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi	Penghubung antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i>	Merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i>	Merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Ade Hendini, 2016)

II.2.7.2. Class Diagram

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi : Kelas (*Class*), Relasi *Associations*, *Generalization* dan *Aggregation*, atribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau *atribut*. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality* (Ade Hendini, 2016 : 111).

Tabel II.2. Multiplicity Class Diagram

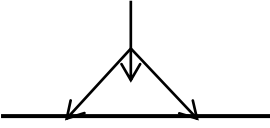
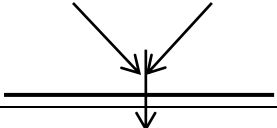
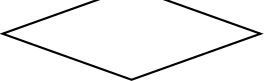
Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Ade Hendini, 2016 : 110)

II.2.7.3. Activity Diagram

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.3 dibawah ini:

Tabel II.3. Simbol Activity Diagram

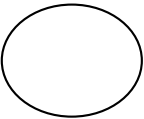
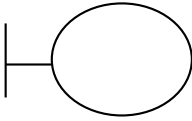
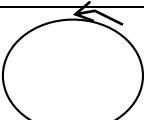
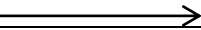
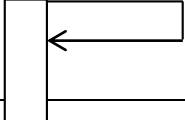
Gambar	Keterangan	Deskripsi
	<i>Start point</i>	Diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i>	Akhir aktifitas.
	<i>Activites</i>	Menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan).	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan)	Digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i>	Menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i>	Pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

(Sumber : Ade Hendini, 2016 : 109)

II.2.7.4. Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.4 dibawah ini :

Tabel II.4. Simbol *Sequence Diagram*

Gambar	Keterangan	Deskripsi
	<i>Entity Class</i>	Merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i>	Berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form entry</i> dan <i>form cetak</i> .
	<i>Control class</i>	Suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i>	Simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i>	Menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i>	Mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i>	Garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Ade Hendini, 2016 : 110)

II.2.8. Basis Data

Basis data atau *database* merupakan koleksi dari data-data yang terorganisir dengan rapi sehingga data dapat dengan mudah disimpan dan dimanipulasi. Kita dapat menjumpai pemanfaatan *database* dalam kehidupan sehari-hari, seperti penggunaan mesin ATM, sistem akademik universitas/sekolah, sistem informasi penjualan. Salah satu tujuan dari *database* adalah memberikan pengguna suatu pandangan abstrak dari data, yaitu sistem menyembunyikan rincian bagaimana data disimpan dan dipelihara. Sistem *database* harus dibuat semudah mungkin untuk dimengerti karena kebanyakan pengguna sistem *database* adalah orang-orang yang kurang terlatih di bidang teknologi. (Ni Kadek Ceryna : 2018 : 102)

II.2.9. PHP

PHP (singkatan rekursif: *PHP Hypertext Preprocessor*) merupakan script yang dapat disisipkan ke dalam HTML. PHP dipakai untuk membuat sistem berbasis web yang dinamis. PHP bahasa pemrograman yang *server-side* sehingga program tersebut akan dijalankan atau diproses oleh server. Hingga saat ini PHP sudah rilis versi 7.4.1 pada 18 desember 2019. Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti C, Java, dan Perl serta mudah untuk dipelajari. PHP merupakan bahasa *scripting server – side*, dimana pemrosesan datanya dilakukan pada sisi *server*. Sederhananya, *server* yang akan menerjemahkan skrip program, baru kemudian hasilnya akan dikirim kepada *client* yang melakukan permintaan. Adapun pengertian lain PHP adalah *akronim* dari *Hypertext Preprocessor*, yaitu suatu bahasa pemrograman berbasis kode – kode

(*script*) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke *web browser* menjadi kode HTML”. (Roberto kaban : 2019)

II.2.10. MYSQL

MySQL (*My Structured Query Language*) merupakan sebuah program pembuat *database* yang bersifat *Open Source*, artinya semua orang dapat menggunakannya dan dapat dijalankan pada semua *platform* baik *Windows* maupun *linux*. MySQL juga merupakan sebuah perangkat lunak sistem manajemen basis data *SQL* yang bersifat jaringan sehingga dapat digunakan untuk aplikasi multi *user*. MySQL juga sering dikenal dengan nama sistem manajemen *database* relasional. Suatu *database* relasional menyimpan data dalam tabel yang terpisah. Tabel-tabel tersebut terhubung oleh suatu relasi terdefinisi yang memungkinkan memperoleh kombinasi data dari beberapa tabel dalam suatu permintaan. Untuk administrasi *database*, seperti pembuatan *database*, pembuatan tabel, dan sebagainya dapat digunakan aplikasi berbasis web seperti *PHPMyAdmin* dengan aplikasi *XAMPP*. (Muhammad Faisal Widad : 2017)

II.2.11. Normalisasi

Normalisasi merupakan sebuah teknik dalam *logical* desain sebuah basis data / *database*, teknik pengelompokkan atribut dari suatu relasi sehingga membentuk struktur relasi yang baik tanpa redundansi. Tujuan normalisasi adalah mengorganisasikan data kedalam tabel-tabel untuk memenuhi kebutuhan pemakai,

menghilangkan kerangkapan data, mengurangi kompleksitas, mempermudah modifikasi data. (Lptk et al., 2016)

1. Proses Normalisasi

- a. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat.
- b. Apabila tabel yang diuji belum memenuhi persyaratan tertentu maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal.

2. Tahapan Normalisasi :

- 1) Bentuk tidak normal : Menghilangkan perulangan *grup*.

Tabel II.5. Contoh bentuk tidak normal (Unnormal)

No-Mhs	Nama Mhs	Jurusan	Kode-MK	Nama-MK	Kode Dosen	Nama Dosen	Nilai
2683	Welli	MI	M1350	Manajemen DB	B104	Ati	A
			M1465	Analisis Perc. Sistem	B317	Dita	B
5432	Bakti	AK	M1350	Manajemen DV	B104	Ati	C
			Akn201	Akuntansi	D310	Lia	B
			MKT300	Dasar Pemasaran	B212	Lola	A

Sumber : Mukhlis Fatih Latief : 2016

- 2) Bentuk Normal pertama (1NF) : Menghilangkan ketergantungan sebagian.

Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal kesatu bila setiap data bersifat atomik yaitu setiap irisan baris dan kolom hanya mempunyai satu nilai data.

Tabel II.6. Contoh Bentuk Normal Pertama (1NF)

No-Mhs	Nama Mhs	Jurusan	Kode-MK	Nama-MK	Kode Dosen	Nama Dosen	Nilai
2683	Welli	MI	M1350	Manajemen DB	B104	Ati	A
2683	Welli	MI	M1465	Analisis Perc. Sistem	B317	Dita	B
5432	Bakti	AK	M1350	Manajemen DV	B104	Ati	C
5432	Bakti	AK	Akn201	Akuntansi	D310	Lia	B
5432	Bakti	AK	MKT300	Dasar Pemasaran	B212	Lola	A

Sumber : Mukhlis Fatih Latief : 2016

3) Bentuk Normal kedua (2NF) : Menghilangkan ketergantungan transitif.

Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal kedua bila relasi tersebut sudah memenuhi bentuk normal kesatu dan atribut yang bukan *key* sudah tergantung penuh terhadap *key*-nya.

Tabel II.7. Contoh Bentuk Normal Kedua (2NF)

Kode-MK	Nama-MK	Kode Dosen	Nama Dosen
M1350	Manajemen DB	B104	Ati
M1465	Analisis Perc. Sistem	B317	Dita
M1350	Manajemen DV	B104	Ati
Akn201	Akuntansi	D310	Lia
MKT300	Dasar Pemasaran	B212	Lola

Sumber : Mukhlis Fatih Latief : 2016

4) Bentuk Normal ketiga (3NF) : Menghilangkan anomali-anomali hasil dari ketergantungan fungsional. Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal ketiga bila relasi tersebut sudah memenuhi bentuk normal kedua dan atribut yang bukan *key* tidak tergantung transitif terhadap *key*-nya.

Tabel II.8. Contoh Tabel Mahasiswa Dan Tabel Kuliah (3NF)

No_Mhs	Nama Mhs	Jurusan
2683	Welli	MI
5432	Bakti	AK

Sumber : Mukhlis Fatih Latief : 2016

Normalisasi adalah proses penyusunan *table-table* yang tidak redundan (*double*) yang dapat menyebabkan anomali pada saat terjadi manipulasi data seperti tambah, ubah dan hapus. Tujuan dari normalisasi adalah:

- a. Untuk menghilangkan kerangkapan data.
- b. Untuk mengurangi, kompleksitas.
- c. Untuk mempermudah pemodifikasian data. (Trisnawati, 2016)