BAB II
LANDASAN TEORI

2.1 Teori Dasar

Landasan teori perlu ditegakkan agar penelitian mempunyai dasar yang kokoh, dan bukan sekedar perbuatan coba-coba (*trial and error*). Adanya landasan teori merupakan ciri bahwa penelitian merupakan cara ilmiah untuk mendapatkan data. Kerlinger mengatakan bahwa teori adalah seperangkat konstruk (konsep), definisi, dan proposisi yang berfungsi untuk melihat fenomena secara sistematis melalui spesifikasi hubungan antarvariabel sehingga dapat berguna untuk menjelaskan dan meramalkan fenomena. Dengan kata lain, teori adalah generalisasi atau kumpulan generalisasi yang dapat digunakan untuk menjelaskan berbagai fenomena secara sistematis. (Sudaryono, 2015, p. 13)

Teori dasar ini bertujuan untuk mengkaji tentang teori-teori yang berhubungan dengan penelitian yang sedang dikerjakan. Teori adalah serangkaian bagian atau variabel, definisi, dan dalil yang saling berhubungan yang menghadirkan sebuah pandangan sistematis mengenai fenomena dengan menentukan hubungan antar variabel, dengan menentukan hubungan antar variabel, dengan maksud menjelaskan fenomena alamiah.

2.1.1 Kecerdasan Buatan atau *Artificial Intelligence* (AI)

Kecerdasan Buatan atau *Artificial Intelligence* (AI) merupakan bidang ilmu komputer yang mempunyai peran penting di era kini dan masa akan datang. Bidang ini telah berkembang sangat pesat di 20 tahun terakhir seiring dengan pertumbuhan kebutuhan akan perangkat cerdas pada industry dan rumah tangga. AI mencakup bidang yang cukup besar. Mulai dari yang paling umum hingga yang khusus. Dari *learning* atau *perception* hingga pada permainan catur, pembuktian teori matematika, menulis puisi, mengemudikan mobil, dan melakukan diagnosis penyakit. AI relevan dengan berbagai macam *task* kecerdasan, AI merupakan sebuah ilmu yang universal (Budiharto & Suhartono, 2014, p. 2). Ada beberapa bidang ilmu AI yaitu:

2.1.1.1 Logika Fuzzy

Logika fuzzy merupakan salah satu cabang dari bidang *soft computing*. Logika fuzzy merupakan suatu teori himpunan logika yang dikembangkan untuk mengatasi konsep nilai yang terdapat diantara kebenaran (*true*) dan kesalahan (*false*) (Irwansyah & Faisal, 2015, p. 31).

Ada 3 metode dalam logika *fuzzy*, yaitu:

1. Metode Mamdani

Sering dikenal sebagai metode *max-min*. Untuk mendapatkan *output* diperlukan beberapa tahapan, yaitu pembentukan himpanan *fuzzy,* aplikasi fungsi implikasi (aturan), komposisi aturan (Pusadan, 2014, p. 14).

2. Metode Sugeno

Metode Sugeno adalah penalaran dengan metode *output* (konsekuen) sistem tidak berupa himpunan *fuzzy*, melainkan berupa konstanta atau persamaan linear (Rofiq, 2013, p. 6).

3. Metode Tsukamoto

Pada metode ini, setiap aturan direpresentasikan menggunakan himpunan-himpunan *fuzzy*, dengan fungsi keanggotaan yang monoton. Untuk menentukan nilai *output crisp* atau hasil yang tegas, dicari dengan cara mengubah input menjadi suatu bilangan pada domain himpunan *fuzzy* tersebut (Sholihin, dkk. 2013, p. 501).

2.1.1.2 Jaringan Syaraf Tiruan

Jaringan syaraf tiruan (JST) sistem pemrosesn informasi yang memiliki karakteristik serupa dengan jaringan syaraf biologis dan terdiri dari sejumlah besar elemen pemrosesan sederhana yang disebut neuron, unit, sel, atau node. Ada 3 hal yang diadopsi dari sistem syaraf otak manusia oleh JST adalah Neuron, Topologi, Toleransi Kesalahan (*Fault Tolerant*) (Irwansyah & Faisal, 2015, p. 42). Menurut Russel dan Norvig (2003), JST memiliki proses belajar yang berbeda dengan

metode yang lainnya, sehingga *Nerual Network* memiliki struktur yang unik dimana metode pelatihannya dibagi ke dalam dua kelompok, yakni sebagai berikut:

1. *Feed-Forward Neural Network* (FFNN)

FFNN adalah jaringan syaraf tiruan dimana hubungan antara sinyal informasinya bergerak hanya satu arah saja dalam mengasosiasikan *input* dengan *output* yang ekstensif digunakan dalam pengenalan pola.

2. *Feed-Back Neural Network* (FBNN)

FBNN merupakan jaringan syaraf tiruan yang memiliki sinyal informasi yang bergerak dari kedua arah pada *layer* jarignannya.

2.1.1.3 Sistem Pakar atau *Expert System*

Sistem pakar adalah aplikasi berbasis komputer yang digunakan untuk menyelesaikan masalah sebagaimana yang dipikirkan oleh pakar. Pakar yang dimaksud di sini adalah orang yang mmepunyai keahlian khusus yang dapat menyelesaikan masalah yang tidak dapat diselesaikan oleh orang awam (Kusrini, 2008, p. 3).

Sistem pakar adalah sistem komputer yang ditujukan untuk meniru semua aspek kemampuan pengambilan keputusan seorang pakar. Sistem pakar memanfaatkan secara maksimal pengetahuan khusus selayaknya seorang pakar untuk memecahkan masalah (Rosnelly, 2012, p. 3).

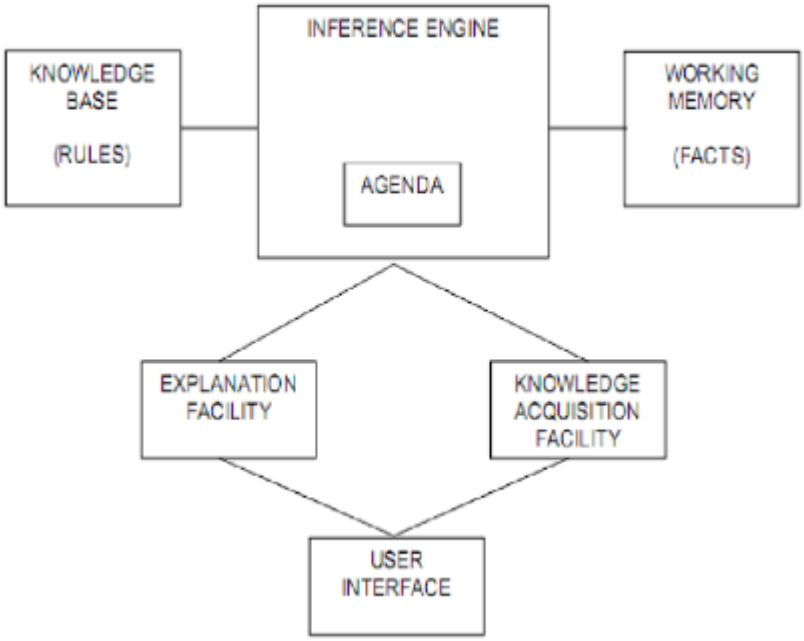
Pakar atau ahli didefinisikan sebagai seorang yang memiliki pengetahuan atau keahlian khusus yang tidak dimiliki oleh kebanyakan orang. Seorang pakar dapat memecahkan masalah yang tidak mampu dipecahkan kebanyakan orang. Dengan kata lain, dapat memecahkan suatu masalah dengan lebih efisien namun bukan berarti lebih murah. Pengetahuan yang dimuat ke dalam sistem pakar dapat berasal dari seorang pakar atau pun pengetahuan yang berasal dari buku, jurnal, majalah, dan dokumentasi yang dipublikasi lainnya, serta orang yang memiliki pengetahuan meskipun bukan ahli. Istilah sistem pakar sering disinonimkan dengan sistem berbasis pengetahuan.

A. Kelebihan Sistem Pakar

Sistem pakar memiliki beberapa fitur menarik yang merupakan kelebihannya, seperti meningkatkan ketersediaan, mengurangi biaya, permanen, keahlian multipel, meningkatkan kehandalan, penjelasan, respon yang cepat, stabil, pembimbing pintar dan basis data cerdas.

B. Struktur Sistem Pakar

Adapun struktur sistem pakar dapat dilihat pada gambar berikut.



Gambar 2.1 Gambar Struktur Sistem Pakar
Sumber: Rosnelly (2012: 13)

Komponen yang terdapat dalam struktur sistem pakar ini adalah *knowledge base (rules), inference engine, working memory, explanation facility, knowledge acquisition facility, user interface.*

1. *Knowledge Base* (Base Pengetahuan)

Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi dan dua elemen dasar, yaitu fakta dan aturan. Fakta merupakan informasi tentang objek dalam area permasalahan tertentu, sedangkan aturan merupakan informasi

tentang cara bagaimana memperoleh fakta baru dari fakta yang telah diketahui. Pada struktur sistem pakar diatas, *knowledge base* disini untuk menyimpan pengetahuan dari pakar berupa *rule*/aturan.

2. *Inference Engine* (Mesin Inferensi)

Mesin inferensi merupakan otak dari sebuah sistem pakar dan dikenal juga dengan sebutan *control structure* atau *rule interpreter*. Komponen ini mengandung mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan suatu masalah. Mesin inferensi disini adalah processor pada sistem pakar yang mencocokkan bagian kondisi dari rule yang tersimpan didalam *knowledge base* dengan fakta yang tersimpan di *working memory*.

3. *Working Memory*

Berguna untuk menyimpan fakta yang dihasilkan oleh *inference engine* dengan penambahan parameter berupa derajat kepercayaan atau dapat juga dikatakan sebagai global *database* dari fakta yang digunakan oleh *rule-rule* yang ada.

4. *Explanation Facility*

Menyediakan kebenaran dari solusi yang dihasilkan kepada user (reasoning chain).

5. *Knowledge Acquisition Facility*

Meliputi proses pengumpulan, pemindahan dan perubahan dari kemampuan pemecahan masalah seorang pakar atau sumber pengetahuan terdokumentasi ke program computer, yang bertujuan untuk memperbaiki atau mengembangkan basis pengetahuan.

6. *User Interface*

Mekanisme untuk memberi kesepakatan kepada *user* dan sistem pakar untuk berkomunikasi. Antar muka 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. *Inference Engine* atau Mesin Inferensi

Inferensi merupakan proses untuk menghasilkan informasi dari fakta yang diketahui atau diasumsikan. Inferensi adalah konklusi logis (*logical conclusion*) atau implikasi berdasarkan informasi yang tersedia. Dalam sistem pakar proses inferensi dilakukan dalam suatu modul yang disebut *Inference Engine* (Mesin Inferensi) (Kusrini, 2008, p. 8).

Ada dua metode inferensi yang penting dalam sistem pakar yaitu:

1. Runut Maju (*Forward Chaining*)

Runut maju berarti menggunakan himpunan aturan kondisi-aksi. Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan dijalankan, kemudian aturan tersebut dijalankan. Mungkin proses menambahkan data ke memori kerja. Proses diulang sampai ditemukan suatu hasil (Wilson, 1998).

Metode inferensi runut maju cocok digunakan untuk menangani masalah pengendalian (*controlling*) dan peramalan (*prognosis*) (Giarattano dan Riley, 1994).

2. Runut Balik (*Backward Chaining*)

Runut balik merupakan metode penalaran kebalikan dari runut maju. Dalam runut balik penalaran dimulai dengan tujuan kemudian merunut balik ke jalur yang akan mengarahkan ke tujuan tersebut (Giarattano dan Riley, 1994). Runut balik disebut juga sebagai *goal-driven reasoning*, merupakan cara yang efisien untuk memecahkan masalah yang dimodelkan sebagai masalah pemilihan terstruktur.

Tujuan inferensi adalah mengambil pilihan terbaik dari banyak kemungkinan. Metode inferensi runut balik ini cocok digunakan untuk memecahkan masalah diagnosi (Schnupp, 1989).

D. Karakteristik Sistem Pakar

Sistem pakar umumnya dirancang untuk memenuhi beberapa karakteristik umum, yaitu kinerja sangat baik, waktu respon yang baik, dapat diandalkan, dapat dipahami dan fleksibel.

2.1.2 Tabel Keputusan

Dalam proses pengambilan keputusan, semua informasi yang diperlukan disusun dalam bentuk ringkasan hasil yang disebut sebagai tabel hasil atau tabel keputusan. Tabel ini merupakan suatu matriks yang terdiri dari baris yang menunjukkan berbagai alternatif pilihan/keputusan dan kolom yang menunjukkan nilai harapan untuk setiap alternatif pilihan/keputusan pada berbagai keadaan atau situasi yang mungkin terjadi. (Herjanto & Herfan, 2008, p. 26).

Tabel 2.1 Contoh Tabel Keputusan

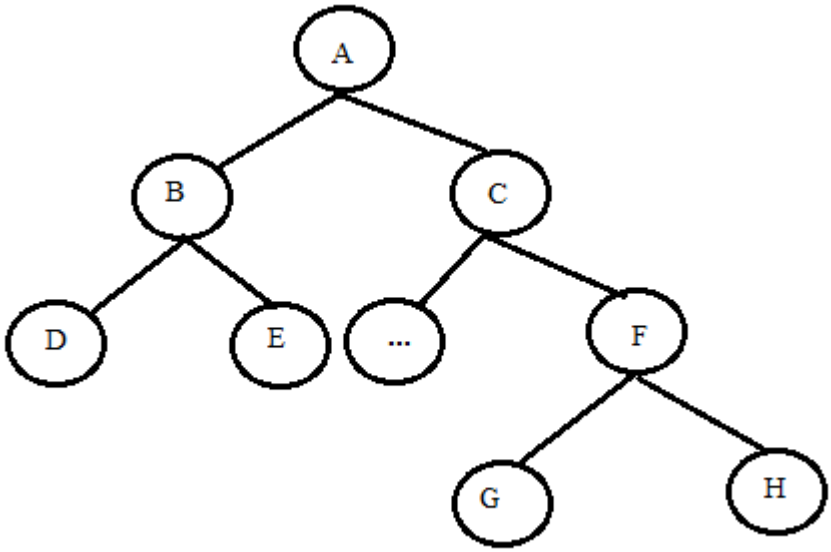
Hipotesa / **Indikator**	**Hipotesa1**	**Hipotesa2**	**Hipotesa3**	**Hipotesa4**
Indikator A	Ya	Ya	Tidak	Ya
Indikator B	Ya	Tidak	Ya	Tidak
Indikator C	Ya	Tidak	Ya	Tidak
Indikator D	Tidak	Ya	Tidak	Ya

Sumber: Data Penelitian (2017)

2.1.3 Pohon Keputusan

Sebuah pohon keputusan adalah sebuah struktur yang dapat digunakan untuk membagi kumpulan data yang besar menjadi himpunan-himpunan *record* yang lebih kecil dengan menerapkan serangkaian aturan keputusan. Dengan masing-masing rangkaian pembagian, anggota himpulan hasil menjadi mirip satu dengan yang lain (Berry & Linoff, 2004 dalam buku (Kusrini & Luthfi, 2009, p. 14)).

Sebuah pohon keputusan mungkin dibangun dengan saksama secara manual atau dapat tumbuh secara otomatis dengan menerapkan salah satu atau beberapa algortima pohon keputusan untuk memodelkan himpunan data yang belum terklasifikasi.



Gambar 2.2 Contoh Pohon Keputusan
Sumber: Data Penelitian (2017)

2.1.4 UML (*Unified Modeling Language*)

Salah satu pemodelan yang saat ini paling banyak digunakan adalah UML. UML (*Unified Modeling Language*) adalah salah standar bahasa yang banyak digunakan di dunia industry untuk mendefinisikan *requirement*, membuat analis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa & Shalahuddin, 2011, p. 113)

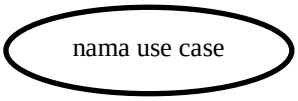
Berikut adalah beberapa jenis diagram UML:

2.1.4.1 Use Case Diagram

Use case diagram merupakan pemodelan untuk kelakukan sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Berikut adalah simbol-simbol yang ada pada diagram *use case:*

Tabel 2.2 Simbol *Use Case Diagram*

Simbol	Deskripsi
Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan meggunakan kata kerja di awal frase nama *use case*
Aktor / *actor*  nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang; tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
Asosiasi / *association* 	Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* yang memiliki interaksi dengan aktor
Ekstensi / *extend* <<extend>> 	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu; mirip dengan prinsip *inheritance* pada pemrograman berorientasi objek; biasanya *use case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambahkan
Generalisasi / *generalization* 	Hubungan generalisai dan spesialisasi (umum-khusu) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya
Menggunakan / *include* / *uses* <<include>>  <<uses>> 	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsina atau sebagai syarat dijalankan *use case* ini.

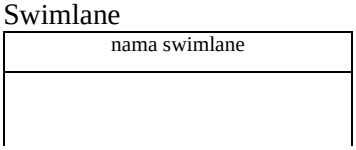
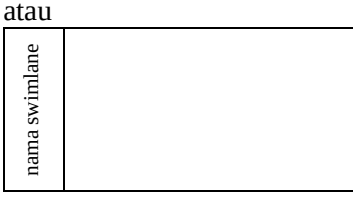
Sumber: Rosa, A. & Shalahuddin, M. (2011)

2.1.4.2 *Activity Diagram*

Activity diagram menggambarkan 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 *actor*, jadi aktivitas yang dapat dilakukan oleh sistem.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

Tabel 2.3 Simbol Activity Diagram

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / *decision* 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / *join*	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Swimlane  atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

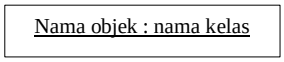
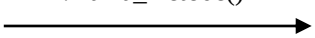
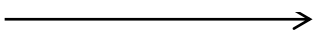
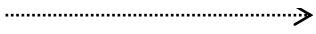
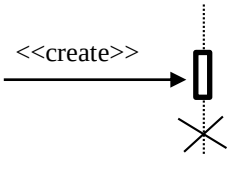
Sumber: Rosa, A. & Shalahuddin, M. (2011)

2.1.4.3 *Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antarobjek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

Tabel 2.4 Simbol *Sequence Diagram*

Simbol	Deskripsi
Aktor / *actor*  nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang; tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
Garis hidup / *lifeline* 	Menyatakan kehidupan suatu objek
Objek  	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi dengan pesan
Pesan tipe *create*  	Objek yang lain, arah panah mengarah pada objek yang dibuat
Pesan tipe *call*  	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi
Pesan tipe *send*  	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
Pesan tipe *return*  	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
Pesan tipe *destroy*  	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada *create* maka ada *destroy*

Sumber: Rosa, A. & Shalahuddin, M. (2011)

2.1.4.4 Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

Berikut adalah simbol-simbol yang ada pada diagram kelas:

Tabel 2.5 Simbol *Class Diagram*

Simbol	Deskripsi
Kelas nama_kelas +atribut +operasi()	Kelas pada struktur sistem
Antarmuka / *interface* nama_interface	Sama dengan konsep *interface* dalam pemrograman berorientasi objek
Asosiasi / *association*	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*
Asosiasi berarah / *directed association*	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity*
Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan / *dependency*	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi / *aggregation*	Relasi antar kelas dengan makna semua bagian (*whole part*)

Sumber: Rosa, A. & Shalahuddin, M. (2011)

2.2 Variabel

Variabel penelitian juga menjadi suatu bagian penting dalam penelitian. Kemampuan peneliti untuk memahami variabel penelitian sangat tergantung pada penguasaan konsep tentang penelitian terutama variabel penelitian. Selain itu, pengalaman peneliti dalam melaksanakan penelitian maupun menyusun proposal peneltian juga dapat menambah pemahaman tentang kemampuan mengidentifikasi variabel penelitian. Variabel adalah sebuah konsep yang dioperasionalkan. Lebih tepatnya, operasional properti dari sebuah objek agar dapat dioperasionalkan, diaplikasikan, dan menjadi properti dari objek (Swarjana, 2015, p. 42).

Variabel yang akan digunakan oleh peneliti dalam penelitiannya ada sebagai berikut:

2.2.1 Proyektor

Proyektor adalah alat yang digunakan untuk memproyeksikan tampilan dari komputer atau alat lain agar diperoleh gambar yang lebih besar. Jenis-jenis proyektor di antaranya proyektor CRT, proyektor LCD, proyektor DLP, proyektor *slide*, proyektor *film*, proyektor *overhead*, dan lain-lain (Priyatno, 2012, p. 16).

Fungsi-fungsi dari sebuah alat proyektor ada sebagai berikut:

a. Sebagai Alat Presentasi

Projector dapat membuat sebuah presentasi menjadi lebih hidup, karena dengan tampilan gambar atau tulisan itu kita dapat memberikan presentasi yang lebih dinamis dan atraktif.

b. **Sebagai Pemutar Video (*Home Theater*)**

Dengan Proyektor kita dapat menikmati bioskop di dalam rumah. Ini dikarenakan proses tampilan yang terjadi di bioskop bisa kita tampilkan di rumah, yaitu dengan proyeksi.

c. Sebagai Media Informasi

Karena Proyektor dapat menampilkan tampilan dengan layar besar, maka proyektor sangat efektif untuk dijadikan sebagai media informasi.

Proyektor yang diteliti oleh peneliti adalah proyektor Infocus IN2124 dimana jenis proyektor DLP (*Digital Light Processing*). Proyektor tersebut telah banyak digunakan oleh umum baik di perorangan maupun di perkantoran yang mana telah dijadikan alasan mengapa peneliti mengambil proyektor jenis tersebut. Pemrosesan Cahaya Digital adalah sebuah teknologi yang digunakan dalam proyektor dan televisi proyeksi. DLP awalnya dikembangkan oleh Texas Instruments, dan mereka tetap pembuat satu-satunya teknologi ini, meskipun banyak produk pasar berlisensi menggunakan chipset mereka (Utami, 2007, p. 73).

Dalam proyektor DLP, gambar diciptakan oleh kaca kecil mikroskopis yang disusun dalam sebuah *matrix* di atas chip semikonduktor, dikenal sebagai *Digital*

Micromirror Device (DMD). Setiap kaca mewakilkan satu *pixel* dalam gambar yang diproyeksikan. Jumlah kaca sama dengan resolusi gambar yang diproyeksikan: 800x600, 1024x768, dan 1280x720. *Matrix* adalah beberapa ukuran DMD yang umum. Kaca-kaca ini dapat diubah posisinya dengan cepat untuk merefleksikan cahaya melalui lensa atau ke sebuah *heatsink* (pembuangan cahaya dalam terminologi Barco).

Penyusunan posisi kaca-kaca ini pergantian antara ‘on’ dan ‘off’ secara cepat membuat DMD mengatur intenssitas cahaya yang direfleksikan melalui lensa, menciptakan efek abu-abu bertingkat sebagai tambahn untuk putih (kaca dalam posisi ‘on’), dan hitam (kaca dalam posisi ‘off’). Ada dua metode primer dimana sistem proyeksi DLP menciptakan sebuah gambar berwarna, yaitu dengan menggunakan proyektor DLP chip-tunggal dan proyektor tiga-chip.

2.3 *Software* Pendukung

Program pendukung yang digunakan untuk menjalakan sistem aplikasi tersebut adalah:

2.3.1 Bahasa Pemograman PHP

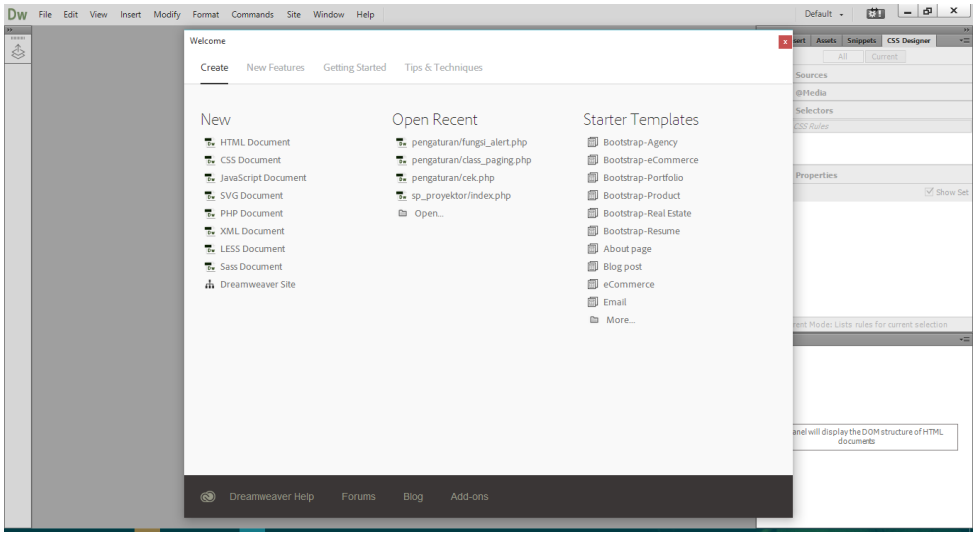
PHP merupakan bahasa pemrograman berbasis web yang dibuat secara khusus untuk membangun aplikasi berbasis web. Selain tersedia secara gratis, PHP juga mudah dipelajari oleh siapapun. Akan lebih mudah lagi, jika kita sudah pernah mempelajari Bahasa C atau C++. Banyak perintah PHP yang diturunkan atau mengadopsi perintah-perintah di Bahasa C (Solichin, 2016, p. 23).

Gambar 2.3 Logo PHP
Sumber: http://php.net/

2.3.2 Aplikasi Adobe Dreamweaver

Adobe Dreamweaver merupakan salah satu program aplikasi yang digunakan untuk membangun sebuah *website*, baik secara grafis maupun dengan menuliskan kode sumber secara langsung (Wahana Komputer n.d., p. 2).

Adobe Dreamweaver memudahkan pengembang website untuk mengelola halaman-halaman *website* dan asset-asetnya, baik gambar(*image*), animasi *flash*, video, suara dan lain sebagainya.

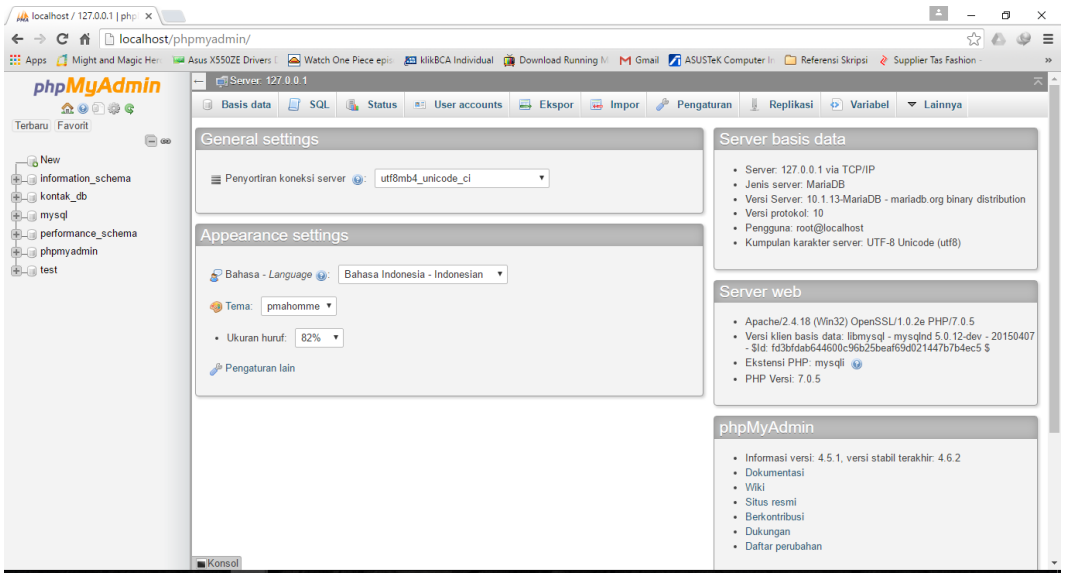


Gambar 2.4 Tampilan Adobe Dreamweaver CC 2015
Sumber: Data Penelitian (2017)

2.3.3 MySQL *Database*

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau yang dikenal dengan DBMS (*database management system*), *database* ini *multithread, multi-user* (Miftakhul Huda, 2010, p. 181).

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat *closed source* atau komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Stuctured Query Language*). SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.



Gambar 2.5 Tampilan *Database* MySQL
Sumber: Data Penelitian (201)

2.3.4 WampServer

WampServer adalah singkatan dari Windows, Apache, MySQL dan PHP. Wampserver adalah aplikasi yang menggabungkan antara Apache, Mysql, dan PHP. Sehingga, Wampserver akan mencukupi semua persyaratan yang diminta oleh Joomla. Dengan demikian, tidak perlu lagi meng-*install* masing–masing aplikasi Apache, Mysql, dan PHP (Meissa, 2009, p. 17).

Gambar 2.6 Logo *WampServer*
Sumber: http://www.wampserver.com/en/

2.3.5 StarUML

StarUML adalah *platform* pemodelan perangkat lunak yang mendukung UML (*Unified Modeling Language*). StarUML yang berbasiskan pada UML versi 1.4, menyediakan sebelas jenis *Diagram* yang berbeda, dan mendukung notasi UML 2.0 StarUML juga secara aktif mendukung pedekatan MDA (*Model Driven Architercture*) dengan mendukung konsep UML. StarUML unggul dalam hal kustomisasi lingkungan kerja pengguna, dan memiliki ekstensibilitas tinggi pada fungsionalitasnya. (Triandini & Suardika, 2012, p. 1)

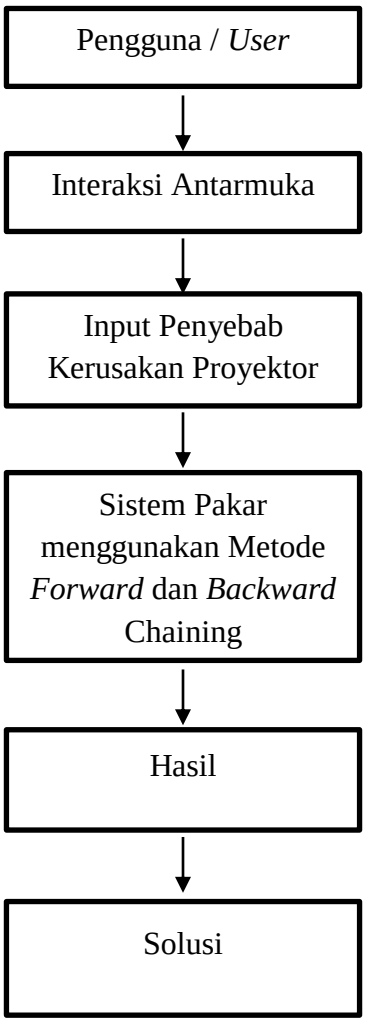
Gambar 2.7 *Logo StarUML*
Sumber: http://staruml.io/

2.4 Penelitian Terdahulu

Penulis	Judul	Metode	Hasil
Manaor dan Serasi (2013)	Perancangan Sistem Pakar Deteksi Kerusakan Printer Canon Berbasis *Web*	*Backward Chaining*	Kesimpulan dari Hasil Penelitian adalah metode tersebut layak digunakan.
Purwanto dan Putra (2016)	Sistem Pakar Diagnosa Kerusakan Pada Televisi LED Dengan Menggunakan Metode *Forward Chaining*	*Forward Chaining*	Kesimpulan dari Hasil Penelitian adalah metode tersebut layak digunakan.
Hartanto dan Herlawati (2015)	Sistem Pakar Pendeteksian Permasalahan Komputer Pada PT. Pasifik Satelit Nusantara Cikarang	*Forward Chaining*	Kesimpulan dari Hasil Penelitian adalah metode tersebut layak digunakan.
Saputra, dkk (2016)	Aplikasi Analisa Masalah Mesin Motor Bebek Menggunakan Metode *Backward Chaining*	*Backward Chaining*	Kesimpulan dari Hasil Penelitian adalah metode tersebut layak digunakan.
Iriani (2015)	Penerapan Metode *Backward Chaining* pada Sistem Pakar Diagnosa Penyakit Tulang Manusia	*Backward Chaining*	Kesimpulan dari Hasil Penelitian adalah metode tersebut layak digunakan.

2.5 Kerangka Berpikir

Penjelasan sementara terhadap suatu gejala yang menjadi objek permasalahan peneliti. Kerangka berpikir ini disusun dengan berdasarkan pada tinjauan pustaka dan hasil penelitian yang relevan atau terkait.



Gambar 2.8 Kerangka Berpikir Peneliti
Sumber: Data Penelitian (2017)