

BAB II KAJIAN PUSTAKA

2.1. Teori Dasar

Deskripsi teori dalam penelitian merupakan uraian sistematis tentang teori (bukan sekedar pendapat pakar atau penulis buku) dan hasil-hasil penelitian yang relevan dengan variabel yang diteliti. Berapa jumlah kelompok teori yang perlu dikemukakan atau dideskripsikan akan tergantung pada luasnya permasalahan dan juga, secara teknis, pada jumlah variabel yang diteliti (Dr.Sudaryono, 2015:15).

Teori adalah alur logika atau penalaran yang merupakan seperangkat konsep, definisi, dan proporsi yang disusun secara sistematis. Secara umum teori mempunyai tiga fungsi, yaitu menjelaskan (*explain*), meramalkan (*predict*), dan mengendalikan(*control*) suatu gejala (Dr.Sudaryono, 2015:14). Kecerdasan buatan atau *Artificial Intelligence* (AI) dan beberapa subdisplin ilmunya seperti *fuzzy logic*, jaringan saraf tiruan, dan sistem pakar (*expert system*).

2.1.1 Kecerdasan Buatan atau *Artificial Intelligence* (AI)

Menurut (Sutojo, Mulyanto, & Suhartono, 2011:1) kecerdasan buatan berasal dari bahasa inggris “*Artificial intelligence*” atau disingkat AI, yaitu *intelligence* adalah kata sifat yang berarti cerdas, sedangkan *artificial* artinya buatan. Kecerdasan buatan yang dimaksud merujuk pada mesin yang mampu

berpikir, menimbang tindakan yang akan diambil, dan mampu mengambil keputusan seperti yang dilakukan oleh manusia.

Berdasarkan definisi ini, maka kecerdasan buatan menawarkan media maupun uji teori tentang kecerdasan. Teori-teori ini nantinya dapat dinyatakan dalam bahasa pemrograman dan eksekusinya dapat dibuktikan pada komputer nyata. Program konvensional hanya dapat menyelesaikan persoalan yang diprogram secara spesifik, kecerdasan buatan memungkinkan komputer untuk berpikir atau menalar dan menirukan proses belajar manusia sehingga informasi baru dapat diserap sebagai pengetahuan, pengalaman, dan proses pembelajaran serta dapat digunakan sebagai acuan dimasa-masa yang akan datang (Sutojo, Mulyanto, & Suhartono, 2011:3).

2.1.2 Fuzzy Logic (Logika Fuzzy)

Menurut (Sutojo, Mulyanto, & Suhartono, 2011:211) logika *fuzzy* adalah metodologi sistem kontrol pemecahan masalah, yang cocok untuk di implementasikan pada sistem, mulai dari sistem yang sederhana, sistem kecil, *embedded system*, jaringan PC, *multi-channel* atau *workstation* berbasis akuisi data, dan *system control*. Metodologi ini dapat diterapkan pada perangkat keras, perangkat lunak ataupun kombinasi keduanya. Dalam logika klasik dinyatakan bahwa segala sesuatu bersifat biner, yang artinya adalah hanya mempunyai dua kemungkinan, “Ya atau Tidak”, “Benar atau Salah”, “Baik atau Buruk”.

Bila dibandingkan dengan logika konvensional, kelebihan logika *fuzzy* adalah kemampuannya dalam proses penalaran secara bahasa sehingga dalam perancangannya tidak memerlukan persamaan matematik yang rumit. Beberapa alasan yang dapat diutarakan mengapa kita menggunakan logika *fuzzy* di antaranya adalah mudah dimengerti, memiliki toleransi terhadap data-data yang tidak tepat, mampu memodelkan fungsi-fungsi *non linear* yang sangat kompleks, dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan, dapat bekerja sama dengan teknik-teknik kendali secara konvensional, dan didasarkan pada bahasa alami (Sutojo, Mulyanto, & Suhartono, 2011:212).

Untuk memahami logika *fuzzy*, sebelumnya perhatikan konsep himpunan *fuzzy*. Himpunan *fuzzy* memiliki 2 atribut, yaitu :

1. Linguistik, yaitu nama suatu kelompok yang mewakili suatu keadaan tertentu dengan menggunakan bahasa alami, misalnya dingin, sejuk, panas.
2. Numeris, yaitu suatu nilai yang menunjukkan ukuran dari suatu variabel, misalnya 10, 35, 40 dan sebagainya.

Beberapa metode yang digunakan dalam sistem inferensi *fuzzy* adalah (Sutojo, Mulyanto, & Suhartono, 2011:233-237)

1. Metode Tsukamoto

Dalam inferensinya, metode Tsukamoto menggunakan tahapan sebagai berikut:

- a. Fuzzifikasi
- b. Pembentukan basis pengetahuan *fuzzy* (*rule* dalam bentuk *IF...THEN*)

- c. Mesin inferensi menggunakan fungsi implikasi *MIN* (Minimum)
- d. Defuzzifikasi menggunakan metode Rata-rata (*Average*)

2. Metode Mamdani

Metode ini sering digunakan karena strukturnya yang sederhana. Pada metode ini, untuk mendapatkan *output* diperlukan 4 tahapan sebagai berikut:

- a. Fuzzifikasi
- b. Pembentukan basis pengetahuan *fuzzy* (*rule* dalam bentuk *IF...THEN*)
- c. Aplikasi fungsi implikasi menggunakan fungsi *MIN* (Minimum) dan komposisi antar-*rule* menggunakan fungsi *MAX*(Maximum) dengan menghasilkan himpunan *fuzzy* baru
- d. Defuzzifikasi menggunakan metode *Centroid* (Titik Tengah)

3. Metode Sugeno

Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985.

Dalam metode ini, *output* sistem berupa konstanta atau persamaan linier. Dalam inferensinya, metode Sugeno menggunakan tahapan sebagai berikut:

- a. Fuzzifikasi
- b. Pembentukan basis pengetahuan *fuzzy* (*rule* dalam bentuk *IF...THEN*)
- c. Mesin inferensi menggunakan fungsi implikasi *MIN* (Minimum)
- d. Defuzzifikasi menggunakan metode Rata-rata (*Average*)

2.1.3 Jaringan Syaraf Tiruan (JST)

Menurut (Sutojo, Mulyanto, & Suhartono, 2011:284) Jaringan saraf tiruan mempunyai kemampuan yang luar biasa untuk mendapatkan informasi dari data yang rumit atau tidak tepat, mampu menyelesaikan permasalahan yang tidak terstruktur dan sulit didefinisikan, dapat belajar dari pengalaman, mampu mengakuisisi pengetahuan walaupun tidak ada kepastian, mampu melakukan generalisasi dan ekstraksi dari suatu pola data tertentu, dapat menciptakan suatu pola pengetahuan melalui pengaturan diri atau kemampuan belajar (*self organizing*), mampu memilih suatu input data ke dalam kategori tertentu yang sudah ditetapkan (klasifikasi), mampu menggambarkan suatu proyek secara keseluruhan walaupun hanya diberikan sebagian data dari objek tersebut (asosiasi), mempunyai kemampuan mengolah data-data input tanpa harus mempunyai target (*self organizing*), serta mampu menemukan jawaban terbaik sehingga mampu meminimalisasi fungsi biaya (optimasi).

Kelebihan-kelebihan yang diberikan oleh jaringan saraf tiruan antara lain :

- a. Belajar *adaptive* : kemampuan untuk mempelajari bagaimana melakukan pekerjaan berdasarkan data yang diberikan untuk pelatihan atau pengalaman awal.
- b. *Self-Organization* : sebuah JST dapat membuat organisasi sendiri atau representasi dari informasi yang diterimanya selama waktu belajar.
- c. *Real Time Operation* : perhitungan JST dapat dilakukan secara paralel sehingga perangkat keras yang dirancang dan diproduksi secara khusus dapat mengambil keuntungan dari kemampuan ini.

2.1.4 Sistem Pakar (*Expert System*)

Sampai saat ini sudah banyak sistem pakar yang dibuat, seperti *MYCIN* untuk diagnosis penyakit, *DENDRAL* untuk mengidentifikasi struktur molekul campuran yang tak dikenal, *XCON* & *XSEL* ntuk membantu konfigurasi sistem komputer besar, *SOPHIE* untuk analisis sirkuit elektronik, *prospector* digunakan di bidang geologi untuk membantu mencari dan menemukan *deposite*, *FOLIO* digunakan untuk membantu memberikan keputusan bagi seorang manager dalam stok dan investasi, *DELTA* dipakai untuk pemeliharaan lokomotif listrik diesel, dan sebagainya. Istilah sistem pakar berasal dan istilah *knowledge-based expert system*. Istilah ini muncul karena untuk memecahkan masalah, sistem pakar menggunakan pengetahuan seorang pakar yang dimasukkan ke dalam komputer. Seseorang yang bukan pakar menggunakan sistem pakar untuk meningkatkan kemampuan pemecahan masalah, sedangkan seorang pakar menggunakan sistem pakar untuk *knowledge* (Sutojo, Mulyanto, & Suhartono, 2011:159-160).

Menurut (Sutojo, Mulyanto, & Suhartono, 2011:160) sistem pakar adalah sebuah *system* yang menggunakan pengetahuan manusia dimana pengetahuan tersebut dimasukkan kedalam sebuah komputer dan kemudian digunakan untuk menyelesaikan masalah-masalah yang biasanya membutuhkan kepakaran atau keahlian manusia.

2.1.4.1 Manfaat Sistem Pakar

Sistem pakar menjadi sangat populer karena sangat banyak kemampuan dan manfaat yang diberikannya, diantaranya (Sutojo, Mulyanto, & Suhartono, 2011:161).

1. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat daripada manusia.
2. Membuat seorang yang awam bekerja seperti layaknya seorang pakar.
3. Meningkatkan kualitas, dengan memberi nasehat yang konsisten dan mengurangi kesalahan.
4. Mampu menangkap pengetahuan dan kepakaran seseorang.
5. Dapat beroperasi di lingkungan yang berbahaya.
6. Memudahkan akses pengetahuan seorang pakar.
7. Andal, sistem pakar tidak pernah menjadi bosan dan kelelahan atau sakit.
8. Meningkatkan kapabilitas *system computer*. Integrasi *system* pakar dengan *system computer* lain membuat *system* lebih efektif dan mencakup lebih banyak aplikasi.
9. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti. Berbeda dengan *system computer* konvensional, *system* pakar dapat bekerja dengan informasi yang tidak lengkap. Pengguna dapat merespon dengan: “tidak tahu” atau “tidak yakin” pada satu atau lebih pertanyaan selama konsultasi dan *system* pakar tetap akan memberikan jawabannya.

10. Bisa digunakan sebagai media pelengkap dan pelatihan. Pengguna pemula yang bekerja dengan system pakar akan menjadi lebih berpengalaman karena adanya fasilitas penjelas yang berfungsi sebagai guru.
11. Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar.

2.1.4.2 Kekurangan Sistem Pakar

Selain manfaat, ada juga beberapa kekurangan yang ada pada sistem pakar, diantaranya :

1. Biaya yang sangat mahal untuk membuat dan memeliharanya.
2. Sulit dikembangkan karena keterbatasan keahlian dan ketersediaan pakar.
3. Sistem pakar tidak 100 % bernilai benar.

2.1.4.3 Ciri-Ciri Sistem Pakar

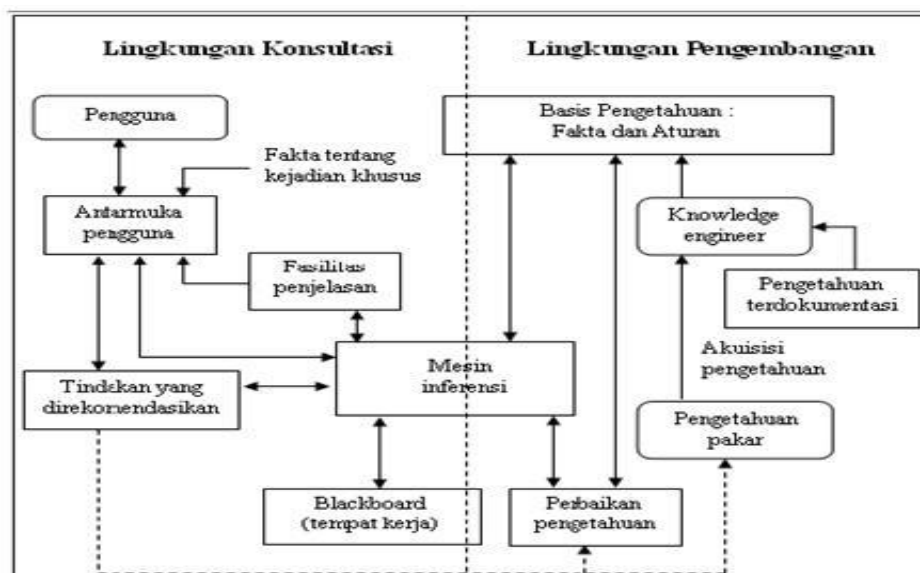
Ciri ciri dari sistem pakar (Sutojo, Mulyanto, & Suhartono, 2011:162) adalah sebagai berikut:

1. Terbatas pada domain keahlian tertentu.
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti.
3. Dapat menjelaskan alasan-alasan dengan cara yang dapat dipahami.
4. Bekerja berdasarkan kaidah/rule tertentu.
5. Mudah di modifikasi.

6. Basis pengetahuan dan mekanisme inferensi terpisah.
7. Keluarannya bersifat anjuran.
8. Sistem dapat mengaktifkan kaidah secara searah yang sesuai, dituntun oleh dialog dengan pengguna.

2.1.5 Struktur Sistem Pakar

Ada dua bagian penting dari sistem pakar, yaitu lingkungan pengembangan dan lingkungan konsultasi. Lingkungan pengembangan digunakan oleh pembuat sistem pakar untuk membangun ke dalam *knowledge base* (basis pengetahuan). Lingkungan konsultasi digunakan oleh pengguna untuk berkonsultasi sehingga pengguna mendapat pengetahuan dan nasihat dari sistem pakar layaknya berkonsultasi dengan seorang pakar. (Sutojo, Mulyanto, & Suhartono, 2011, p.166).



Gambar 2.1 Struktur Sistem Pakar

Sumber : (Sutojo, Mulyanto, & Suhartono, 2011:167)

Berikut penjelasan dari subsistem yang ada pada struktur sistem pakar pada Gambar 2.1:

1. Akuisisi Pengetahuan

Subsistem ini digunakan untuk memasukkan pengetahuan dari seorang pakar dengan cara merekayasa pengetahuan agar bisa diproses oleh komputer dan menaruhnya kedalam basis pengetahuan dengan format tertentu.

2. Basis pengetahuan

Basis pengetahuan mengandung pengetahuan yang diperlukan untuk memahami, memformulasikan, dan menyelesaikan masalah. Basis pengetahuan terdiri dari dua elemen dasar yaitu fakta dan *rule*.

3. Mesin Inferensi

Mesin inferensi adalah sebuah program yang berfungsi untuk memandu proses penalaran terhadap suatu kondisi berdasarkan pada basis pengetahuan yang ada, memanipulasi dan mengarahkan kaidah, model dan fakta yang disimpan dalam basis pengetahuan untuk mencapai solusi dan kesimpulan.

4. Daerah kerja (*Blackboard*)

Blackboard merupakan area pada memori yang berfungsi sebagai basis data. Tiga tipe keputusan yang dapat direkam pada *blackboard* yaitu rencana, agenda dan solusi.

5. Antarmuka pengguna (*User Interface*)

Digunakan sebagai media komunikasi antara pengguna dengan sistem pakar. Sistem perbaikan pengetahuan Kemampuan memperbaiki pengetahuan dari seorang pakar diperlukan untuk menganalisis pengetahuan, belajar dari kesalahan masa lalu, kemudian memperbaiki pengetahuannya sehingga dapat dipakai pada masa mendatang.

6. Pengguna (*user*)

Pada umumnya pengguna sistem pakar bukanlah seorang pakar (*non-expert*) yang membutuhkan solusi, saran, atau pelatihan (*training*) dari berbagai permasalahan yang ada. Pengetahuan yang dimasukkan ke dalam sistem pakar disajikan dalam bentuk yang dapat dimengerti dan diterima oleh sistem pakar, salah satunya adalah kaidah produksi.

2.1.6 Teknik Inferensi

Pada sistem pakar berbasis *rule*, domain pengetahuan di presentasikan dalam sebuah kumpulan *rule* berbentuk *IF-THEN*, sedangkan data di presentasikan dalam sebuah kumpulan fakta-fakta tentang kejadian saat ini. Mesin inferensi membandingkan masing-masing *rule* yang tersimpan dalam basis pengetahuan dengan fakta-fakta yang terdapat dalam database. Jika bagian *IF* (kondisi) dari *rule database* sebagai fakta baru yang ditambahkan, adapun metode yang digunakan adalah *backward chaining* (Sutojo, Mulyanto, & Suhartono, 2011:171).

2.1.6.1 *Backward Chaining*

Menurut (Sutojo, Mulyanto, & Suhartono, 2011:178) *backward chaining* adalah metode inferensi yang bekerja mundur ke arah kondisi awal. Proses diawali dari *Goal* (yang berada di bagian *THEN* dari *rule IF-THEN*), kemudian pencarian mulai dijalankan untuk mencocokkan apakah fakta-fakta yang ada cocok dengan premis-premis di bagian *IF*. Jika cocok, *rule* dieksekusi, kemudian hipotesis di bagian *THEN* ditempatkan di bagian basis data sebagai fakta baru. Jika tidak cocok, simpan premis di bagian *IF* ke dalam *stack* sebagai *subGoal*. Proses berakhir jika *Goal* ditemukan atau tidak ada *rule* yang bisa membuktikan kebenaran dari *subGoal* atau *Goal*.

2.1.7 Tabel dan Pohon Keputusan

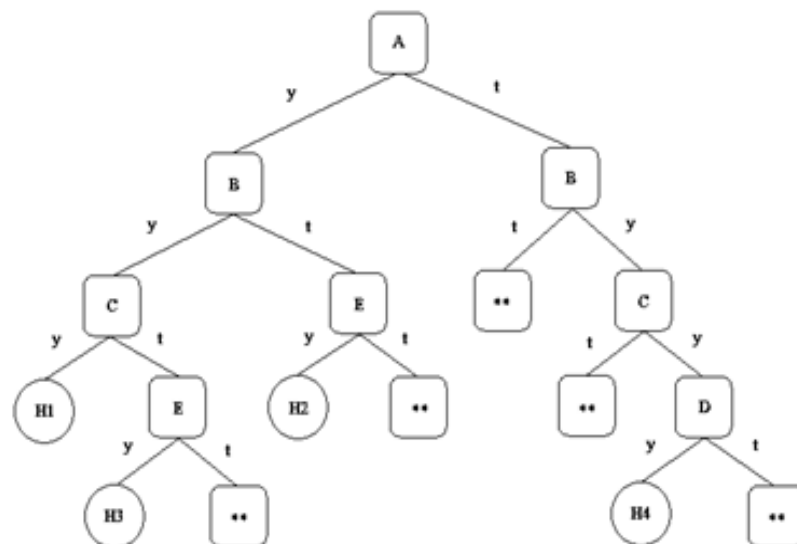
Tindakan mengacu pada kegiatan yang harus dilakukan sebelum hasil dapat diharapkan. Aksi mengacu pada kegiatan yang menyebabkan munculnya efek dari tindakan tersebut. Sebab mengacu pada keadaan tertentu yang menimbulkan akibat tertentu. Gejala mengacu pada keadaan yang menyebabkan adanya kerusakan atau keadaan tertentu yang mendorong adanya pemeriksaan.

Sebelum sampai pada bentuk kaidah produksi, pengetahuan yang didapatkan dalam domain tertentu yang disajikan pengetahuan dalam bentuk tabel keputusan kemudian dibuat dalam pohon keputusannya. Berikut adalah contoh penyajian dalam bentuk tabel keputusan dan pohon keputusan (Sri & Sari, 2008).

Tabel 2.1 Tabel Keputusan

Hipotesa Evidence	Hipotesa 1	Hipotesa 2	Hipotesa 3	Hipotesa 4
<i>Evidence A</i>	Ya	Ya	ya	Tidak
<i>Evidence B</i>	Ya	Tidak	ya	Ya
<i>Evidence C</i>	Ya	Tidak	tidak	Ya
<i>Evidence D</i>	Tidak	Tidak	tidak	Ya
<i>Evidence E</i>	Tidak	Ya	ya	Tidak

Sumber: Hartati & Iswanti, (2008)



Gambar 2.2 Pohon Keputusan
Sumber : Hartati & Iswanti(2008)

Keterangan:

A = evidence A, *H1 = hipotesa 1,* *y =ya*
B = evidence B, *H2 =hipotesa 2,* *t =tidak*
C = evidence C, *H3 =hipotesa 3,** = tidak menghasilkan hipotesa tertentu*
D = evidence D, *H4 = hipotesa 4*

Dari gambar 2.2 dapat diketahui bahwa *hipotesa* H1 terpenuhi jika memenuhi *evidence* A, B, dan C. *Hipotesa* H2 terpenuhi jika memiliki *evidence* A dan *evidence* E. *Hipotesa* H3 akan terpenuhi jika memiliki *evidence* A, B, dan E. *Hipotesa* H4 akan dihasilkan jika memenuhi *evidence* B, C, dan D. Notasi “y” mengandung arti memenuhi node (*evidence*) di atasnya, notasi “t” artinya tidak memenuhi.

2.1.8 WEB

Menurut (Anggiani, septima, 2012) *website* dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar, data animasi, suara, video dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang salingterkait dimana masing0masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*).

Ada beberapa hal yang dipersiapkan untuk membangun *web site* gratis, maka harus tersedia unsur-unsur pendukungnya sebagai berikut:

1. Nama *Domain* (*Domain name/URL – Uniform Resource Locator*).
2. Rumah *Website*.
3. *Content Management System (CMS)*

Perkembangan dunia *website* pada saat ini lebih menekankan pada pengelolaan *content* adalah sebuah *website*. Pengguna yang tidak bisa bahasa

pemrograman *website* pada saat ini bisa membuat *website* dengan memanfaatkan *CMS* tersebut.

2.1.8.1 Jenis-Jenis Web

Seiringan dengan perkembangan teknologi informasi yang begitu cepat, *website* juga mengalami perkembangan yang sangat berarti. Dalam pengelompokkan jenis *web*, lebih diarahkan berdasarkan kepada fungsi, sifat atau *style* dan bahasa pemrograman yang digunakan. Adapun Jenis-jenis *web* berdasarkan sifat atau *style* nya yaitu:

1. *Website* Dinamis, merupakan sebuah *website* yang menyediakan content atau isi yang selalu berubah-ubah setiap saat. Bahasa pemrograman yang digunakan antara lain *PHP*, *ASP*, *NET* dan memanfaatkan *database MySQL* atau *MS SQL*. Misalnya *website* www.artikel.com, www.detik.com, www.tecnomobile.co.cc dan lain-lain.
2. *Website* Statis, merupakan *website* yang contentnya sangat jarang diubah. Bahasa pemrograman yang digunakan adalah *HTML* dan belum memanfaatkan *database*. Misalnya yaitu *web profile* organisasi, dan lain-lain.

2.1.8.2 Fungsi Website

1. Personal *website*, *website* yang berisi informasi pribadi seseorang.
2. *Commercial website*, *website* yang dimiliki oleh sebuah perusahaan yang bersifat bisnis.
3. *Government website*, *website* yang dimiliki oleh instansi pemerintahan, pendidikan yang bertujuan memberikan pelayanan kepada pengguna.
4. *Non-Pro fit Organization website*, dimiliki oleh organisasi yang bersifat non-profit atau tidak bersifat bisnis.

2.2 Variabel Penelitian

Menurut (Dr.Sudaryono, 2015:17) variabel penelitian pada dasarnya adalah segala sesuatu yang berbentuk apa saja yang ditetapkan oleh peneliti untuk dipelajari sehingga diperoleh informasi dan kesimpulannya. Peneliti bekerja pada tingkat teoritis dan empiris. Pada tingkat teoritis, perhatiannya tercurah pada pengidentifikasian konsep dan hubungannya dengan proposisi. Pada tahap empiris, pernyataan riset akan di uji pada penelitian ini yang menjadi variabel adalah penyakit asma, penyakit asma ini juga mempunyai klasifikasi serangan pada anak yaitu asma episodik jarang, asma episodik ringan, asma persisten sedangkan serangan pada orang dewasa yaitu intermitten, persisten ringan, persisten sedang, persisten berat.

2.2.1 Penyakit Asma

2.2.1.1 Definisi Asma

Asma adalah suatu kelainan berupa inflamasi (peradangan) kronik saluran napas yang menyebabkan hipereaktivitas bronkus terhadap berbagai rangsangan yang ditandai dengan episodik berulang berupa mengi, batuk, sesak napas dan rasa berat di dada terutama pada malam dan atau dini hari yang umumnya bersifat *reversibel* baik dengan atau tanpa pengobatan.

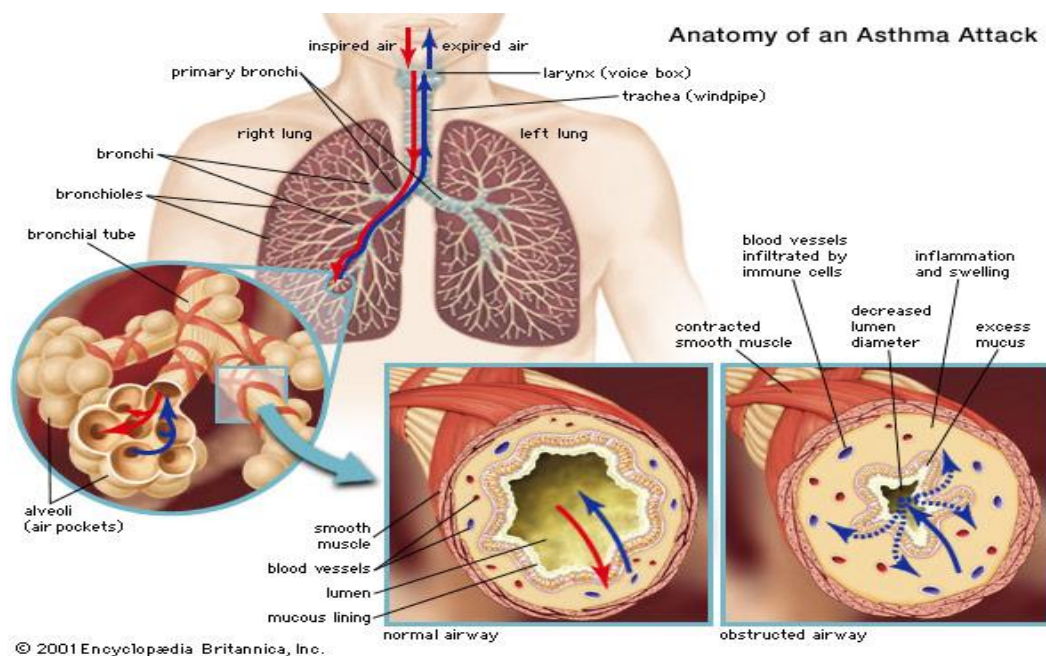
Asma bersifat *fluktuatif* (hilang timbul) artinya dapat tenang tanpa gejala tidak mengganggu aktifitas tetapi dapat eksaserbasi dengan gejala ringan sampai berat bahkan menimbulkan kematian. Pencetus serangan asma dapat disebabkan oleh sejumlah faktor antara lain alergen, virus, dan iritan yang dapat menginduksi respon inflamasi akut yang terdiri atas reaksi asma dini dan reaksi asma lambat. Setelah reaksi asma awal dan reaksi asma lambat, proses dapat terus berlanjut menjadi reaksi inflamasi sub-akut atau kronik (Depkes, 2009).

2.2.1.2 Patofisiologi dan Mekanisme Terjadinya Asma

Gejala asma, yaitu batuk sesak dengan mengi merupakan akibat dari obstruksi bronkus yang didasari oleh inflamasi kronik dan hiperaktivitas bronkus, hiperaktivitas bronkus merupakan ciri khas asma, besarnya hiperaktivitas bronkus ini dapat diukur secara tidak langsung. Berbagai cara digunakan untuk mengukur hiperaktivitas bronkus ini antara lain dengan uji provokasi beban kerja, inhalasi udara dingin, inhalasi antigen maupun inhalasi zat *non*-spesifik. Penyempitan

saluran napas yang terjadi pada asma merupakan suatu hal yang kompleks. Hal ini terjadi karena lepasnya mediator dari sel mast yang banyak ditemukan di permukaan mukosa bronkus, lumen jalan napas dan dibawah membran basal. Berbagai faktor pencetus dapat mengaktivasi sel mast selain sel mast, sel lain yang juga dapat melepaskan mediator adalah sel makrofag alveolar, eosinofil, sel epital jalan napas, netrofil, platelet, limfosit dan monosit.

Mediator inflamasi secara langsung maupun tidak langsung menyebabkan serangan asma, melalui sel efektor sekunder seperti eosinofil, netrofil, platelet dan limfosit. Sel-sel inflamasi ini juga mengeluarkan mediator yang kuat seperti lekoteriens tromboksan, PAF dan protein sitotoksis yang memperkuat reaksi asma. Keadaan ini menyebabkan inflamasi yang akhirnya menimbulkan hipereaktivitas bronkus (Depkes, 2009).



Gambar 2.3 Mekanisme Terjadinya Asma
Sumber: *AAIA Asthma* (2014)

Anatomi sistem pernapasan yang normal ketika kita menarik napas, udara melewati hidung dan mulut dan masuk ke dalam batang tenggorokan atau trakea. Trakea terbagi menjadi dua cabang sampai satu cabang masuk ke setiap paru-paru. Cabang-cabang ini disebut bronkus, terus dibagi menjadi bagian-bagian udara yang lebih kecil dan lebih kecil, bronchioles. Pada akhirnya dari cabang terkecil adalah kantung udara kecil, yang disebut alveoli. Alveoli mengambil oksigen dari udara yang kita hirup dan transfer ke aliran darah. Mereka juga menghilangkan karbondioksida dari darah, sehingga kita bisa menghirupnya.

Anatomi asma mengubah saluran udara paru-paru dengan membuatnya sangat sensitif dan mudah teriritasi. Selama suatu asma flare-up ada tiga faktor yang dapat menyebabkan saluran udara menjadi menyempit:

1. *Inflammation* (Peradangan) yaitu lapisan dalam tabung bronkus menjadi meradang dan bengkak.
2. *Bronchospasm* otot-otot di bagian luar bronkus menjadi kejang dan kencang.
3. *Excessive mucus* (lendir yang berlebihan), yaitu lendir yang berlebihan dihasilkan di dalam saluran udara.

2.2.2 Klasifikasi Asma

Berat-ringannya asma ditentukan oleh berbagai faktor, antara lain gambaran klinik sebelum pengobatan (gejala, eksaserbasi, gejala malam hari, pemberian obat inhalasi dan uji *feel* paru) serta obat-obat yang digunakan untuk mengontrol

asma. Tidak ada suatu pemeriksaan tunggal yang dapat menentukan berat-ringannya suatu penyakit. Asma di klasifikasi atas asma pada orang dewasa dan anak (Depkes, 2009).

2.2.2.1 Klasifikasi Derajat Asma pada Orang Dewasa

Pada umumnya penderita sudah dalam pengobatan; dan pengobatan yang telah berlangsung seringkali tidak adekuat. Dipahami pengobatan akan mengubah gambaran klinis bahkan *feel* paru, oleh karena itu penilaian berat asma pada penderita dalam pengobatan juga harus mempertimbangkan pengobatan itu sendiri. Tabel 2.2 menunjukkan bagaimana melakukan penilaian berat asma pada penderita yang sudah dalam pengobatan. Bila pengobatan yang sedang dijalani sesuai dengan gambaran klinis yang ada, maka derajat berat asma naik satu tingkat. Contoh seorang penderita dalam pengobatan asma persisten sedang dan gambaran klinis sesuai asma persisten sedang, maka sebenarnya berat asma penderita tersebut adalah asma persisten berat. Demikian pula dengan asma persisten ringan. Akan tetapi berbeda dengan asma persisten berat dan asma intermiten. Penderita yang gambaran klinis menunjukkan asma persisten berat maka jenis pengobatan apapun yang sedang dijalani tidak mempengaruhi penilaian berat asma, dengan kata lain penderita tersebut tetap asma persisten berat. Demikian pula penderita dengan gambaran klinis asma intermiten yang mendapat pengobatan sesuai dengan asma intermiten, maka derajat asma adalah intermiten (Heart, 2011).

Tabel 2.2 Klasifikasi Derajat Asma pada Orang Dewasa

Derajat asma	Gejala	Gejala malam	Faal paru
Intermitten	Bulanan		APE $\geq$ 80%
	- Gejala <1x/minggu. - Tanpa gejala diluar serangan. - Serangan singkat.	$\leq$ 2 kali sebulan	- VEP ₁ $\geq$ 80% nilai prediksi APE $\geq$ 80% nilai terbaik. - Variabiliti APE <20%.
Persisten ringan	Mingguan		APE >80%
	- Gejala >1x/minggu tetapi <1x/hari. - Serangan dapat mengganggu aktifitas dan tidur	>2 kali sebulan	- VEP ₁ $\geq$ 80% nilai prediksi APE $\geq$ 80% nilai terbaik. - Variabiliti APE 20-30%.
Persisten sedang	Harian		APE 60-80%
	- Gejala setiap hari. - Serangan mengganggu aktifitas dan tidur. - Membutuhkan bronkodilator setiap hari.	>2 kali sebulan	- VEP ₁ 60-80% nilai prediksi APE 60-80% nilai terbaik. - Variabiliti APE >30%.
Persisten berat	Kontinyu		APE 60 $\leq$ %
	- Gejala terus menerus - Sering kambuh - Aktifiti fisik terbatas	Sering	- VEP ₁ $\leq$ 60% nilai prediksi APE $\leq$ 60% nilai terbaik - Variabiliti APE >30%

Sumber : (Depkes, 2009)

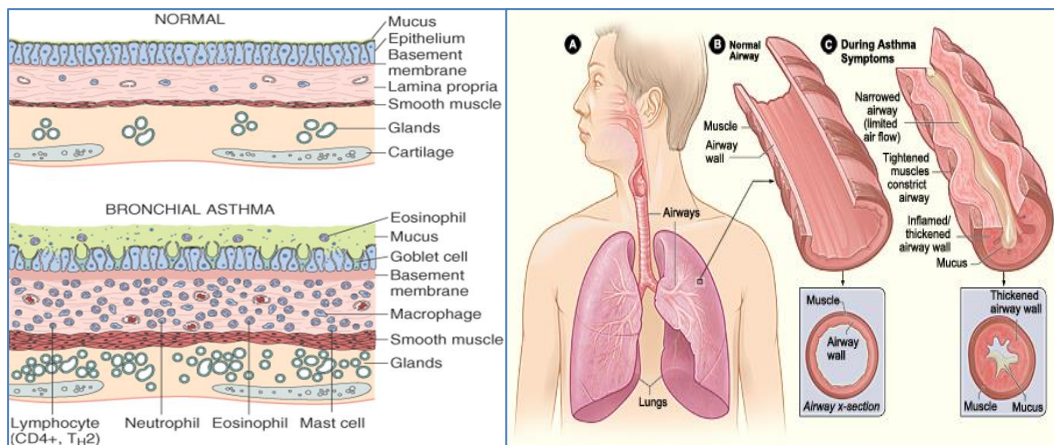
Klasifikasi derajat asma pada orang dewasa saat serangan dan tanpa serangan terbagi menjadi 4 derajat penyakit yaitu :

1. Intermitten termasuk pula dalam penderita alergi dengan pajanan alergen, asmanya kambuh tetapi di luar itu bebas gejala dan faal paru normal. Demikian pula penderita *exercise-induced asthma* atau kambuh hanya bila cuaca buruk, tetapi di luar pajanan pencetus tersebut gejala tidak ada dan faal paru normal.
2. Persisten ringan membutuhkan obat pengontrol setiap hari untuk mengontrol asmanya dan mencegah agar asmanya tidak bertambah berat; sehingga terapi utama pada asma *persisten* ringan adalah anti inflamasi setiap hari dengan glukokortikosteroid inhalasi dosis rendah dosis yang dianjurkan 200-400

ug BD/ hari atau 100-250 ug FP/hari atau ekivalennya, diberikan sekaligus atau terbagi 2 kali sehari.

3. Persisten sedang membutuhkan obat pengontrol setiap hari untuk mencapai asma terkontrol dan mempertahankannya. Idealnya pengontrol adalah kombinasi inhalasi glukokortikosteroid (400-800 ug BD/ hari atau 250-500 ug FP/ hari atau ekivalennya) terbagi dalam 2 dosis dan agonis beta-2 kerja lama 2 kali sehari. Jika penderita hanya mendapatkan glukokortikosteroid inhalasi dosis rendah (lebih besar 400 ug BD atau ekivalennya) dan belum terkontrol; maka harus ditambahkan agonis beta-2 kerja lama inhalasi atau alternatifnya. Jika masih belum terkontrol, dosis glukokortikosteroid inhalasi dapat dinaikkan. Dianjurkan menggunakan alat bantu/ *spacer* pada inhalasi bentuk IDT/MDI atau kombinasi dalam satu kemasan (fix combination) agar lebih mudah.

4. Persisten berat membutuhkan terapi pada keadaan ini yaitu untuk mencapai kondisi sebaik mungkin, gejala seringan mungkin, kebutuhan obat pelega seminimal mungkin, faal paru (APE) mencapai nilai terbaik, variabiliti APE seminimal mungkin dan efek samping obat seminimal mungkin. Untuk mencapai hal tersebut umumnya membutuhkan beberapa obat pengontrol tidak cukup hanya satu pengontrol. Terapi utama adalah kombinasi inhalasi glukokortikosteroid dosis tinggi (> 800 ug BD/ hari atau ekivalennya) dan agonis beta-2 kerja lama 2 kali sehari. Kadangkala kontrol lebih tercapai dengan pemberian glukokortikosteroid inhalasi terbagi 4 kali sehari daripada 2 kali sehari.



Gambar 2.4 Asma pada Orang Dewasa
Sumber: AAIA Asthma (2014)

2.2.2.2 Klasifikasi Derajat Asma pada Anak

Asma merupakan penyakit respiratorik kronis yang paling sering dijumpai pada anak. Prevalensi asma meningkat dari waktu ke waktu baik di negara maju maupun negara sedang berkembang. Peningkatan tersebut diduga berkaitan dengan pola hidup yang berubah dan peran faktor lingkungan terutama polusi baik indoor maupun outdoor. 1,2 Prevalensi asma pada anak berkisar antara 2-30%. Di Indonesia prevalensi asma pada anak sekitar 10% pada usia sekolah dasar,³ dan sekitar 6,5% pada usia sekolah menengah pertama.⁴ Patogenesis asma berkembang dengan pesat. Pada awal 60-an, bronkokonstriksi merupakan dasar patogenesis asma, kemudian pada 70-an berkembang menjadi proses inflamasi kronis, sedangkan tahun 90-an selain inflamasi juga disertai adanya remodelling. 5,6 Berkembangnya patogenesis tersebut berdampak pada tatalaksana asma secara mendasar, sehingga berbagai upaya telah dilakukan untuk mengatasi asma. Pada

awalnya pengobatan hanya diarahkan untuk mengatasi bronkokonstriksi dengan pemberian bronkodilator, kemudian berkembang dengan antiinflamasi. Pada saat ini upaya pengobatan asma selain dengan antiinflamasi, juga harus dapat mencegah terjadinya remodelling (Liansyah, 2014).

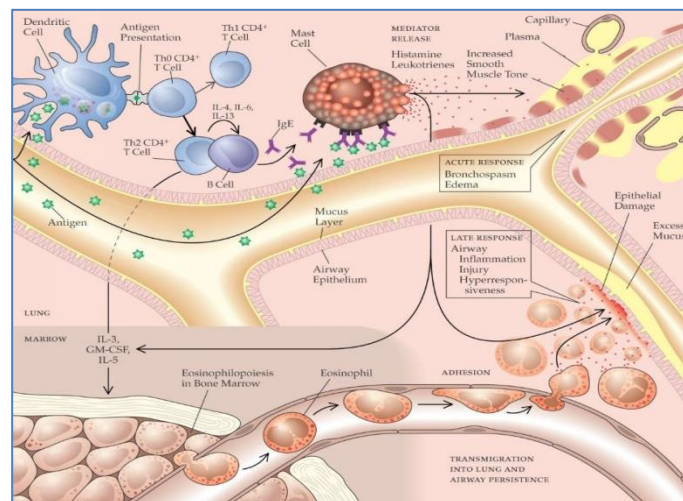
Tabel 2.3 Klasifikasi Derajat Asma pada Anak

Parameter klinis, kebutuhan obat dan faal paru asma	Asma episodik jarang	Asma episodik sering	Asma persisten
1 Frekuensi serangan	<1x/bulan	>1x/bulan	Sering
2 Lama serangan	<1minggu	>1minggu	Hampir sepanjang tahun, tidak ada periode bebas serangan
3 Intensitas serangan	Biasanya ringan	Biasanya sedang	Biasanya berat
4 Diantara serangan	Tanpa gejala	Sering ada gejala	Gejala siang dan malam
5 Tidur dan aktifitas	Tidak terganggu	Sering terganggu	Sangat terganggu
6 Pemeriksaan fisik diluar serangan	Normal (tidak ditemukan kelainan)	Mungkin terganggu (ditemukan kelainan)	Tidak pernah normal
7 Obat pengendali(anti inflamasi)	Tidak perlu	Perlu	Perlu
8 Uji faal paru(diluar serangan)	PEFatauFEV ₁ >80%	PEFatauFEV ₁ <60-80%	PEVatauFEV<60%
9 Variabilitas faal paru(bila ada serangan)	Variabilitas>15%	Variabilitas>30%	Variabilitas 20-30%. Variabilitas >50%

Sumber : (Depkes, 2009)

Klasifikasi derajat asma pada anak secara arbiteri Pedoman Nasional Asma Anak (PNAA) terbagi menjadi 3 derajat penyakit yaitu : Asma episodik jarang , Asma episodik sering , Asma *Persisten*. Dasar pembagian ini karena pada asma anak kejadian episodik lebih sering dibanding *persisten* (kronisitas). Dasar pembagian atau klasifikasi asma pada anak adalah frekuensi serangan, lamanya serangan, aktivitas diluar serangan dan beberapa pemeriksaan penunjang. Pedoman Nasional Asma Anak di dalam batasan operasionalnya menyepakatinya

kecurigaan asma apabila anak menunjukkan gejala batuk dan/atau mengi yang timbul secara episodik, cenderung pada malam hari/dini hari (nokturnal), musiman, setelah aktivitas fisik, serta adanya riwayat asma dan atopi pada penderita atau keluarganya.



Gambar 2.5 Asma pada Anak
Sumber: *AAIA Asthma* (2014)

2.3 *Software* Pendukung

Software adalah sebuah perangkat yang berfungsi sebagai pengatur aktivitas kerja komputer dan seluruh intruksi yang mengarah pada sistem komputer. *Software* dibuat dengan menggunakan bahasa pemrograman yang ditulis atau diciptakan oleh *programer* yang selanjutnya di komplikasi dengan aplikasi kompiller sehingga menjadi sebuah kode yang nantinya akan dikenali oleh mesin *hardware*.

2.3.1 HTML

2.3.1.1 Definisi HTML



Gambar 2.6 Logo HTML
Sumber : Pengolahan Data Penelitian (2017)

HTML merupakan singkatan dari *Hyper Text Markup Language*. HTML bisa disebut bahasa paling dasar dan penting yang digunakan untuk bahasa menampilkan dan mengelola tampilan pada halaman *website*. HTML digunakan untuk menampilkan berbagai informasi didalam sebuah penjelajah *web* dan formatting *hypertext* sederhana yang ditulis kedalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi (Saputra, 2012:1). secara manual. Namun, konsep *SGML* dari markah umum didasarkan pada unsur-unsur daripada hanya efek cetak, dengan pemisahan struktur dan markah juga; HTML telah semakin bergerak ke arah ini dengan CSS.

CSS (*cascading Style Sheet*) adalah *stylesheet language* yang digunakan untuk mendeskripsikan peyajian dari dokumen yang dibuat dalam *mark up language*. CSS merupakan sebuah dokumen yang berguna untuk melakukan pengaturan pada komponen halaman *web*, inti dari dokumen ini adalah

memformat halaman *web* yang memiliki kualitas yang lebih indah dan menarik (Diponegoro, 2012).

2.3.1.2 Cara Kerja HTML

Secara umum, dokumen HTML terbagi atas dua bagian, yaitu bagian *header* (kepala) dan *body* (badan). Bagian *header* diawali dengan *tag* `<head>` dan diakhiri dengan *tag* `</head>`, sedangkan bagian *body* diawali dengan *tag* `<body>` dan ditutup dengan *tag* `</body>`. Kedua bagian tersebut diapit oleh *tag* `<html>` dan `</html>`, yang digunakan untuk menandai bahwa dokumen yang kita buat adalah dokumen HTML.

1. Elemen HTML

Dokumen HTML disusun oleh elemen-elemen. “Elemen” merupakan istilah bagi komponen-komponen dasar pembentuk dokumen HTML. Beberapa contoh elemen adalah: *head*, *body*, *table*, *paragraph*, dan *list*. Elemen dapat berupa *teks* murni, atau bukan *teks*, atau keduanya (Sidik, 2014:10).

2. Tag HTML

Untuk menandai berbagai elemen dalam suatu dokumen HTML, kita menggunakan *tag*. *Tag* html terdiri atas sebuah kurung sudut kiri (`<`, tanda lebih kecil), sebuah nama *tag*, dan sebuah kurung sudut kanan (`>`, tanda lebih besar). *Tag* umumnya berpasangan (misalnya `<H1>` dengan `</H1>`), *tag* yang menjadi pasangan selalu diawali dengan karakter garing (`/`, garis miring). *Tag* yang pertama menunjukkan *tag* awal yang berarti awal elemen,

dan yang kedua menunjukan *tag* akhir, berarti akhir elemen (Sidik, 2014:10).

3. *Atribut Tag*

Tag dapat mempunyai *atribut*. *Atribut* menyatakan sesuatu tentang *tag* tersebut. *Atribut* digunakan untuk mengubah *default* pemformatan dokumen dengan *tag* yang bersangkutan

2.3.2 *Personal Home Page (PHP)*

2.3.2.1 Definisi PHP



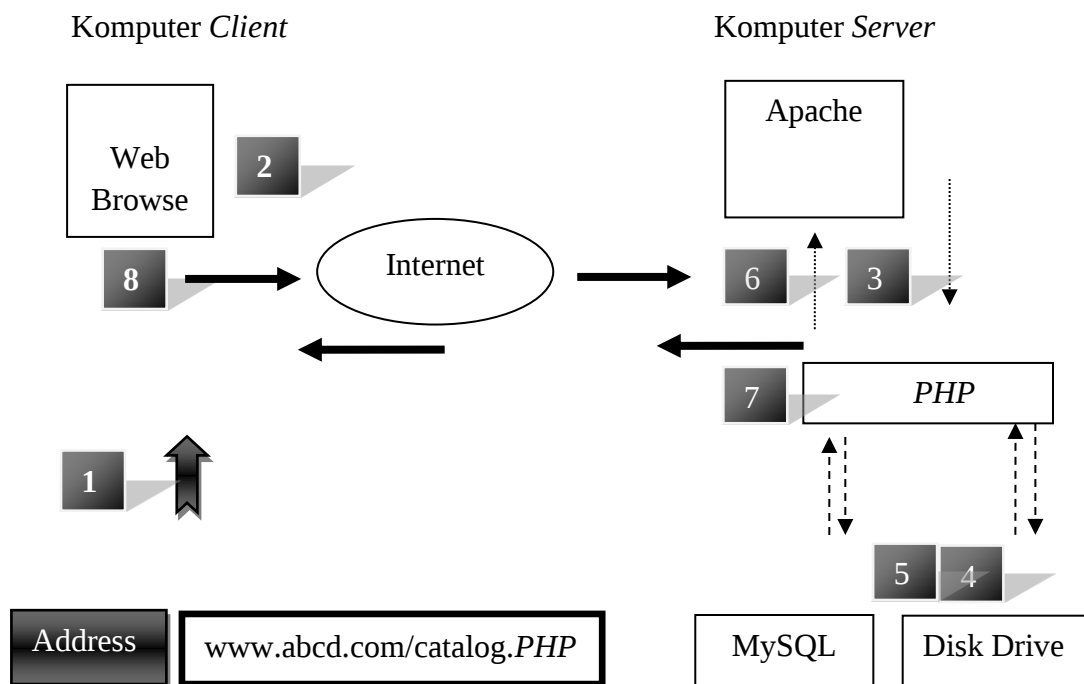
Gambar 2.7 Logo PHP
Sumber: Pengolahan Data Penelitian (2017)

PHP merupakan salah satu bahasa pemrograman berbasis *web* dimana sistem yang diterapkan adalah pada sisi *server side*. PHP dapat disisipkan diantara skrip-skrip bahasa HTML dan arena bahasa *server side* lainnya, dengan itu maka PHP akan dieksekusi secara langsung pada *server*. Sedangkan *browser* akan mengeksekusi halaman *web* tersebut melalui *server* yang kemudian akan menerima tampilan “hasil jadi” dalam bentuk HTML, sedangkan kode PHP itu sendiri tidak akan dapat terlihat. PHP juga mendukung komunikasi dengan layanan lain melalui protokol *IMAP*, *SNMP*, *NNTP*, *POP3* dan *HTTP*. Fungsi-

fungsi yang ada di PHP tidak *case sensitive* tetapi variabelnya *case sensitive* (membedakan huruf besar dan kecil). Kode PHP diawali dengan tanda lebih kecil (). Konsep kerja *HTML* diawali dengan permintaan suatu halaman web oleh browser. (Haryana, 2008).

2.3.2.2 Cara Kerja PHP

Menurut (Haryana, 2008) cara kerja aplikasi web yang ditulis dengan *PHP* dapat diilustrasikan dengan gambar dibawah ini.



Gambar 2.8 Cara Kerja PHP
Sumber: Raharjo (2014)

Berikut adalah keterangan dari gambar diatas:

1. User menulis `www.abcd.com/catalog.php` kedalam *address bar* dari web browser (IE, Mozilla Firefox, Opera, dll).

2. *Web browser* mengirimkan pesan di atas ke komputer server (*www.abcd.com*) melalui *internet*, meminta halaman *catalog.php*.
3. *Web server* (misalnya *Apache*), program yang berjalan di komputer server, akan menangkap pesan tersebut, lalu meminta *interpreter* PHP (program lain yang juga berjalan di komputer server) untuk mencari *file catalog.php* dalam *disk drive*.
4. *Interpreter* PHP membaca *file catalog.php* dari *disk drive*.
5. *Interpreter* PHP akan menjalankan perintah-perintah atau kode PHP yang ada dalam *file catalog.php*. Jika kode dalam *file catalog.php* melibatkan akses terhadap *database* (misalnya MySQL) maka *interpreter* PHP juga akan berhubungan dengan MySQL untuk melaksanakan perintah-perintah yang berkaitan dengan *database*.
6. *Interpreter* PHP mengirimkan halaman dalam bentuk HTML ke *Apache*.
7. Melalui *internet*, *Apache* mengirimkan halaman yang diperoleh dari *interpreter* PHP ke komputer *user* sebagai respon atas permintaan yang diberikan.
8. *Web browser* dalam komputer *user* akan menampilkan halaman yang dikirim oleh *Apache*.

2.3.3 XAMPP

2.3.3.1 Definisi XAMPP



Gambar 2.9 Logo Xampp
Sumber : Pengolahan Data Penelitian (2017)

Menurut (Putra Yoka 2015:25) *server web* adalah komputer yang digunakan untuk menyimpan dokumen-dokumen *web* yang akan melayani permintaan dokumen *web* dari kliennya. XAMPP adalah sebuah perangkat lunak *web server apache* yang didalamnya sudah tersedia database server *MySQL* dan dapat mendukung pemrograman *PHP*. XAMPP merupakan *software* yang mudah digunakan, gratis, dan mendukung instalasi di *Linux* dan *Windows*. Keuntungan lainnya adalah cukup dengan menginstal XAMPP sudah tersedia *Apache Web Server*, *MySQL Database Server*, *PHP support (PHP 4 dan PHP 5)* dan beberapa modul lainnya.

XAMPP adalah satu paket *software web server* yang terdiri dari *Apache*, *MySQL*, *PHP* dan *PHPMyAdmin*. Mengapa menggunakan XAMPP?, karena XAMPP sangat mudah penggunaannya, terutama jika seorang pemula. Proses instalasi XAMPP sangat mudah, karena tidak perlu melakukan konfigurasi

Apache, PHP dan MySQL secara Manual, XAMPP melakukan instalasi dan konfigurasi secara otomatis (Madcoms 2009:1).

2.3.3.2 *APACHE*

Manurut Winarto ada baiknya kita mengenali aplikasi yang dikemas oleh *XAMPP*.

1. *Apache HTTP Server*

Apache HTTP Server merupakan aplikasi untuk *server web* terpopuler di dunia, dengan pangsa pasar pada bulan Maret 2009 mencapai 46%, atau dengan kata lain 46% dari seluruh *website* yang ada di *internet* dijalankan menggunakan *Apache*. Dan pada tahun 2009 ini, *Apache HTTP server* tercatat sebagai aplikasi *server web* pertama yang menembus angka penggunaan mencapai 100 juta situs *web* di seluruh dunia.

Pada Skema tersebut, halaman *web* statis yang diminta *client* akan langsung dikirimkan begitu saja oleh *server*, begitu pula jika *client* mengklik sebuah link untuk meminta halaman lain. Namun pada *Apache* yang dipaketkan oleh XAMPP ini, sudah terdapat dua modul pengolah pemrograman di sisi *server* (*server-side scripting*), yaitu PHP dan *Perl*. Hal ini memungkinkan kita memanfaatkan *web server* untuk menginstall beberapa aplikasi berbasis *web* yang sudah disertakan di dalam CD, atau untuk mempelajari pembuatan *website* dinamis menggunakan bahasa pemrograman tersebut di *server* lokal.

Sebagaimana disebutkan sebelumnya, *Apache* memberikan kemampuan sebuah *web server* pada komputer kita, dan PHP memungkinkan kita menjalankan sebuah *website* dinamis yang menggunakan bahasa pemrograman PHP. Namun aplikasi berbasis *web* yang terdapat pada CD tidak bisa diinstall jika kita belum menyiapkan sebuah *database server* atau *server basis data* yang sesuai. *Database server* dibutuhkan untuk menyediakan mekanisme penyimpanan data secara terstruktur, efektif, dan efisien. MySQL yang dipaketkan dalam XAMPP merupakan aplikasi *server database* yang mumpuni, dan banyak digunakan pada aplikasi berbasis *web*. Bahkan banyak *website* besar dengan trafik yang tinggi memanfaatkan MySQL untuk penyimpanan basis datanya. Sebut saja *Flickr*, *Facebook*, *Wikipedia*, *Google*, *Nokia* dan *YouTube* yang secara resmi telah membeberkan bahwa *website* mereka menggunakan MySQL sebagai *database server*.

2.3.3.3 PHP MyAdmin



Gambar 2.10 Logo *phpMyAdmin*
Sumber : Pengolahan Data Penelitian (2017)

Menurut (e-Media Solusindo 2008:57) salah satu alat paling ciamik untuk mengelola *database MySQL* adalah *PhpMyAdmin*. Memanfaatkan *phpMyAdmin*, proses pengaturan *previlege* dan manipulasi *database MySQL* lebih mudah dilakukan menggunakan antarmuka *web* oleh orang awam sekalipun.

2.3.4 MySQL Database

2.3.4.1 Definisi MySQL



Gambar 2.11 Logo MySQL
Sumber : Pengolahan Data Penelitian (2017)

MySQL merupakan salah satu *database* kelas dunia yang sangat cook bila dipadukan dengan bahasa pemograman php. Mysql bekerja menggunakan bahasa sql yang merupakan bahasa standar yang digunakan untuk manipulasi *database*. Pada umumnya, perintah yang paling sering digunakan dalam mysql adalah *select*, *insert*, *update*, dan *delete*. Selain itu sql menyediakan perintah untuk membuat *database*, *field*, ataupun *index* untuk menambah atau menghapus data (agus saputra, 2012:77) .

2.3.4.2 Cara Kerja MySQL

Menurut Raharjo (2014:421) Elemen penting yang perlu diketahui dalam mempelajari SQL adalah sebagai berikut:

1. *Data Definition Language* (DDL), yaitu *statemen-statemen* yang berhubungan dengan pembuatan objek (misalnya tabel) dan pengelolaan strukturnya.
2. *Data Manipulation Language* (DML), yaitu *statemen-statemen* yang berhubungan dengan manipulasi data di dalam tabel.
3. *Data Control Language* (DCL), yaitu *statemen-statemen* kontrol seperti *GRANT* dan *REVOKE*.
4. *Transactional Control Language* (TCL), yaitu *statemen* yang digunakan untuk mengatur transaksi data seperti *START TRANSACTION*, *SAVEPOINT*, *COMMIT*, dan *ROLLBACK*.
5. *Data Query Language* (DQL), yaitu *statemen* yang mengacu ke seleksi data seperti *SELECT* dan *SHOW*.

Menurut Raharjo (2014:215) Untuk memanipulasi data pada tabel-tabel yang terdapat di dalam suatu *database*, anda perlu mempelajari perintah-perintah SQL dari MySQL. Berikut ini perintah-perintah yang perlu anda pelajari secara lebih detail .

1. *SELECT*: digunakan untuk mengambil data dari *database*.
2. *DELETE*: digunakan untuk menghapus data dari *database*.
3. *INSERT*: digunakan untuk memasukan data baru ke dalam *database*.

4. *REPLACE*: digunakan untuk mengganti data di dalam *database*. Jika terdapat *record* yang sama dalam suatu tabel, perintah ini akan menimpa *record* tersebut dengan data baru.
5. *UPDATE*: digunakan untuk mengubah data di dalam tabel.

Perintah-perintah diatas hanya digunakan untuk memanipulasi data. Untuk memanipulasi struktur objek *database*, gunakan perintah-perintah berikut:

1. *CREATE*: digunakan untuk membuat *database*, tabel, atau *indeks*.
2. *ALTER*: digunakan untuk memodifikasi struktur dari suatu tabel.
3. *DROP*: digunakan untuk menghapus *database*, tabel, atau *indeks*.

2.3.5 *StarUML*

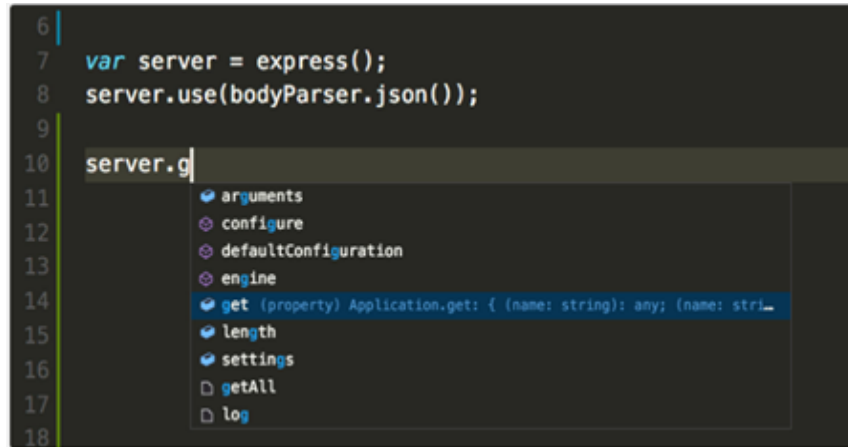
2.3.5.1 Definisi *StarUML*



Gambar 2.12 Logo *StarUML*
Sumber : Pengolahan Data Penelitian (2017)

StarUML merupakan salah satu *CASE* (*Computer-Aided Software Engineering*) *tools* atau perangkat pembantu berbasis komputer untuk rekayasa perangkat lunak yang mendukung alur hidup perangkat lunak (*life cycle support*). *StarUML* termasuk ke dalam kelompok *upper CASE tools* yang mendukung perencanaan strategis dan pembangunan perangkat lunak

2.3.6 Visual Studio Code



Gambar 2.13 Logo *Visual Studio Code*
 Sumber : Pengolahan Data Penelitian (2017)

Visual Studio Code adalah editor kode sumber yang dikembangkan oleh *Microsoft* untuk *Windows*, *Linux* dan *macOS*. Ini termasuk dukungan untuk *debugging*, kontrol *Git* tertanam, penyorotan sintaks, penyelesaian kode cerdas, cuplikan, dan *refactoring* kode. Ini juga dapat disesuaikan, sehingga pengguna dapat mengubah tema editor, pintasan keyboard, dan preferensi. muncul dengan dukungan built-in untuk *JavaScript*, *TypeScript* dan *Node.js* dan memiliki ekosistem ekstensi yang kaya untuk bahasa lain (seperti *C++*, *C#*, *Java*, *Python*, *PHP*, *Go*) dan runtimes (seperti *.NET* dan *Unity*).

2.3.7 Unified Modelling Language (UML)

Unified Modelling Language atau UML diketahui sebagai suatu bahasa yang sudah menjadi standar pada visualisasi, perancangan dan juga pendokumentasian sistem *software*. Saat ini UML sudah menjadi bahasa standar dalam penulisan *blueprint software*.

Menurut (A.S & M.Shalahuddin, 2011, p. 113) UML (*unified Modelling Language*) adalah salah standart bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requitment*, membuat analisis, dan desain serta menggambarkan arsitektur dalam pemogramman berorientasi objek. Pada perkembangan teknik pemogramman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemogramman berorientasi objek, yaitu *Unified Modelling Language (UML)*. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan sebuah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

2.3.7.1 Use Case Diagram

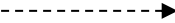
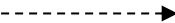
Use case diagram diketahui sebagai salah satu jenis diagram pada UML yang menggambarkan interaksi antara sistem dan aktor, *use case diagram* juga dapat mendeskripsikan tipe interaksi antara si pemakai sistem dengan sistemnya.

Use case atau diagram *use case* merupakan diagram pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang buat. secara kasar *use case* digunakan untuk mengetahui fungsi apa saja yang didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol aktor adalah gambar orang, tapi aktor belum tentu merupakan orang. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit atau aktor. Berikut simbol-simbol yang ada pada diagram *use case* (A.S & M.Shalahuddin, 2011, p. 113).

Tabel 2.4 *Use case Diagram*

	<i>ACTOR</i> : Sebuah entitas manusia/ mesin yang berinteraksi dengan sistem
	<i>USE CASE</i> : Merepresentasikan fungsionalitas dari suatu sistem / pekerjaan tertentu pada sistem
	<i>ASOSIASI</i> : Menjelaskan hubungan antara <i>use case</i> yang berupa pertukaran informasi

<< include >> 	DEPENDENCY INCLUDE : Menunjukkan suatu <i>use case</i> harus dipenuhi
<< Extend >> 	DEPENDENCY EXTEND : Menunjukkan suatu <i>use case</i> akan di laksanakan bersifat <i>optional</i>

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

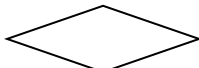
2.3.7.2 Activity Diagram

Diagram aktivitas atau *Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (A.S & M.Shalahuddin, 2011, p. 113).

Activity diagram atau diagram aktivitas merupakan suatu jenis diagram khusus dari *statechart diagram* yang mempresentasikan *state-state* dan transisi-transisi yang terjadi pada akhir operasi-operasi. Diagram aktivitas dapat digunakan untuk merepresentasikan *state-state* serta transisi-transisi yang bersifat sinkron (Nugroho, 2009).

Tabel 2.5 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 / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu		
	Lebih dari satu aktivitas digabungkan menjadi satu		
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir		
Swimlan <table border="1" data-bbox="319 996 641 1149"><tr><td>Nama swimlance</td></tr><tr><td></td></tr></table>	Nama swimlance		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi
Nama swimlance			

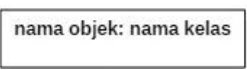
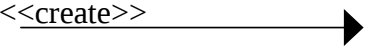
Sumber: (Rosa A.S dan M.Shalahudin, 2014:161).

2.3.7.3 Sequence Diagram

Menurut (A.S & M.Shalahuddin, 2011, p. 113) *sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu, untuk menggambarkan *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyaknya diagram sekuen yang harus digambarkan adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri tau yang penting semua *use case* yang telah didefenisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semangkin banyak *use case* yang didefenisikan maka diagram sekuen yang harus dibuat juga semangkin banyak. Berikut simbol simbol yang ada pada diagram sekuen:

Tabel 2.6 *Sequence Diagram*

Simbol	Deskripsi
Aktor/actor 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama actor
Garis hidup/lifeline 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor
Objek 	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya. Aktor tidak memiliki waktu aktif
Pesan tipe <i>create</i> <<create>> 	Menyatakan suatu objek membuat objek yang lain. Arah panah mengarah pada objek yang dibuat

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

2.3.7.4 Class Diagram

Menurut (A.S & M.Shalahuddin, 2011:141–146) Diagram kelas atau *class* diagram menggambarkan struktur segi dari pendefenisian kelas-kelas yang akan dibuat untuk membangun *system*. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas sedangkan operasi atau metode adalah fungsi yang dimiliki oleh suatu kelas.

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik sebaiknya memiliki jenis jenis kelas berikut:

1. Kelas *main*

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2. Kelas yang memiliki tampilan sistem

Kelas yang mendefenisikan dan mengatur tampilan pemakai

3. Kelas yang diambil dari pendefenisian *use case*

Kelas yang menangani fungsi-fungsi yang harus diambil dari pendefenisian *use case*.

4. Kelas yang diambil dari pendefenisian data

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun yang akan disimpan ke basis data.

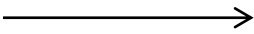
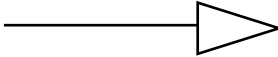
Jenis-jenis kelas diatas juga dapat digabungkan satu sama lain sesuai dengan pertimbangan yang dianggap baik asalkan fungsi-fungsi yang sebaiknya ada pada struktur kelas tetap ada. Susunan kelas juga dapat ditambahkan kelas utilitas seperti kenoeksi kebasis data, membaca *file teks*, dan lain sebagainya sesuai kebutuhan.

Dalam mendefenisikan metode yang didalam kelas perlu memperhatikan apa yang disebut dengan *cohesion dan coupling*. *Cohesion* adalah ukuran seberapa dekat keterkaitan instruksi didalam sebuah metode terkait satu sama lain sedangkan *coupling* adalah ukuran seberapa dekat keterkaitan instruksi didalam sebuah metode yang satu dengan metode yang lain dalam sebuah kelas.

Berikut adalah simbol-simbol yang ada pada diagram kelas:

Tabel 2.7 Diagram Class

Simbol	Deskripsi
	Kelas pada struktur sistem
Antarmuka / <i>interface</i> Nama_interface 	Sama dengan konsep interface dalam pemograman berorientasi objek
Asosiasi / <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi juga disertai dengan <i>multiplicity</i>
Assosiasi berarah / <i>directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai

	dengan <i>multiplicity</i>
Generalisasi 	Relasi antar kelas dengan makna generalisasi–spesialisasi(umum khusus)
Kebergantungan / <i>dependency</i>>	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi / <i>aggregation</i>	Relasi antar kelas dengan makna

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

2.4 Penelitian Terdahulu

Sebagai bahan pertimbangan dalam penelitian ini, maka penulis mencantumkan beberapa penelitian yang diambil dari beberapa jurnal ilmiah, yaitu:

1. Nama Penulis : Widayati Tri Yohana

Judul Jurnal : Sistem pakar untuk mendiagnosis penyakit asma

Volume / ISSN : 2013 : 01

Kesimpulan :Sistem pakar untuk mendiagnosa penyakit asma berdasarkan gejala yang dialami oleh penderita asma itu sendiri dengan gejala-gejala yang telah di derita, jadi dengan adanya sistem ini dapat mempermudah para penderita asma untuk mengetahui tingkat keparahan penyakit asma yang di derita. Sistem pakar untuk mendiagnosa penyakit asma ini hanya merupakan alternatif lain untuk membantu meringankan para

penderita asma dalam mempelajari dan memahami gejala dan penanggulangannya (Widayati, Komputer, Asma, & Penyakit, 2013).

2. Nama Penulis : Istri sulistyowati dan Arif soeleman

Judul Jurnal : Sistem pakar berbasis *web* untuk mendiagnosa penyakit dalam pada manusia

Volume / ISSN : 9 Nomor 1, April 2013 / 1414-9999

Kesimpulan : Diperoleh fakta bahwa tujuan utama permasalahan sistem pakar untuk mendiagnosis penyakit dalam pada manusia yang muncul adalah terbatasnya jumlah, waktu dan tenaga dari seorang dokter sehingga untuk melakukan konsultasi ketika berhalangan hadir akan menyulitkan pasien dan untuk mempermudah pasien, penulis membuat sistem yang mampu untuk menentukan jenis penyakit dan pengobatannya berdasarkan gejala-gejalanya (Sulistyowati, Istri & Soeleman, 2013).

3. Nama Penulis : Maurits Muhammad dkk

Judul Jurnal : Sistem pakar untuk mendiagnosa penyakit asma berbasis *forward chaining*.

Volume / ISSN : 3 No.1, April 2014 / 465-526

Kesimpulan : Diperoleh fakta bahwa tujuan permasalahan sistem pakar mendiagnosa penyakit asma berbasis *forward chaining* diharapkan akan membantu penyakit asma untuk menentukan

jawaban dan mengetahui penyakit mereka serta akan menghemat tenaga dan pikiran karena proses dijalankan oleh komputer (Arnie & Maurits, 2014).

4. Nama Penulis : Mutsaqof Noor Aniq Ahmad dkk

Judul Jurnal : Sistem pakar untuk mendiagnosis penyakit infeksi menggunakan *forward chaining*.

Volume / ISSN : 4 No.1, Juni 2015 / 2301-7201

Kesimpulan : Diperoleh fakta bahwa tujuan permasalahan sistem pakar untuk mendiagnosis penyakit infeksi menggunakan forward chaining diharapkan sistem pakar akan membantu tenaga medis *non* dokter untuk mendeteksi penyakit serta mendapat informasi solusi pengobatannya sejak dini sehingga dapat dilakukan penanganan secepatnya (Aniq, Mutsaqof, & Suryani, 2015).

5. Nama Penulis : Joan Angelina Widiyans & Ikrimah Hidayati

Judul Jurnal : Sistem Pendiagnosa Penyakit Asma Pada Anak Dengan Metode *Certainty Factor*

Volume / ISSN : 2302-3805

Kesimpulan : Asma merupakan penyakit dengan keadaan yang menunjukkan respon abnormal saluran napas terhadap berbagai rangsangan yang menyebabkan penyempitan jalan

napas yang meluas. Sistem pendidiagnosa penyakit asma pada anak ini dibuat dengan metode *Certainty Factordengan* inferensi runut maju atau *Forward Chaining* berdasarkan gejala-gejala yang dialami pasien usia 0 hingga 13 tahun untuk menentukan jenis penyakit asma yang diderita. Hasil penelitian ini berupa informasi mengenai kondisi pasien yang terkena asma beserta solusi pencegahan dan penyembuhannya (Joan Angelina Widiens, 2016).

2.5 Kerangka Berpikir

Kerangka pemikiran yang baik akan menjelaskan secara teoritis peraturan antara variabel yang akan di teliti. Serta secara teoritis dapat di jelaskan antara variabel *dependen* dan variabel *independen* serta variabel *intervening* dan *moderating*. (Sugiono, 2014: 60). Berdasarkan dukungan teori dasar dan penelitian terdahulu yang diperoleh, maka dapat disusun kerangka pemikiran sebagai berikut:



Gambar 2.14 Kerangka Berpikir
Sumber : Pengolahan Data Penelitian (2017)

Data-data yang dibutuhkan berkaitan dengan penyakit asma dianalisis terlebih dahulu oleh *user* agar lebih sederhana atau mudah dilakukan proses

pengolahan datanya. Data-datanya tersebut kemudian di input dan diolah menggunakan sistem pakar menggunakan metode *backward chaining*. Sistem pakar yang menggunakan metode *backward chaining* ini menggunakan bahasa pemograman *PHP* dan *database MySQL* yang dapat digunakan untuk mendiagnosa penyakit asma dan menghasilkan solusi cara mendiagnosa penyakit asma.