

BAB II KAJIAN PUSTAKA

2.1. Teori Dasar

Menurut (Dr.Sudaryono, 2015, p. 15) Deskripsi teori dalam suatu penelitian merupakan uraian sistematis tentang teori (dan bukan sekedar pendapat pakar atau penulis buku) dan hasil penelitian yang relevan dengan *variable* yang diteliti. Berapa jumlah kelompok teori yang dikemukakan atau dideskripsikan akan tergantung pada luasnya permasalahan dan secara teknis tergantung pada jumlah *variable* yang diteliti. Bila dalam suatu penelitian terdapat tiga *variable* independen dan satu dependen, maka kelompok teori yang perlu dideskripsikan ada empat kelompok teori, yaitu kelompok teori yang berkenaan dengan *variable* independen dan satu dependen , oleh karena itu semakin banyak *variable* yang diteliti, maka akan semakin banyak teori yang dikemukakan.

2.1.1 Kecerdasan Buatan atau *Artificial Intelligence (AI)*

Menurut (Sutojo, Mulyanto, & Suhartono, 2011, p. 1) kecerdasan buatan berasal dari bahasa inggris “*Artificial intelligence*” atau disingkat AI, yaitu *inteligence* 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, p. 3).

2.1.2 Fuzzy Logic (Logika Fuzzy)

Menurut (Sutojo, Mulyanto, & Suhartono, 2011, p. 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, p. 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, p. 233)

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, p. 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* untuk 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 dari 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, p. 159).

Menurut (Sutojo, Mulyanto, & Suhartono, 2011, p. 160) sistem pakar adalah sebuah *system* yang menggunakan pengetahuan manusia dimana pengetahuan tersebut dimasukkan ke dalam 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, p. 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

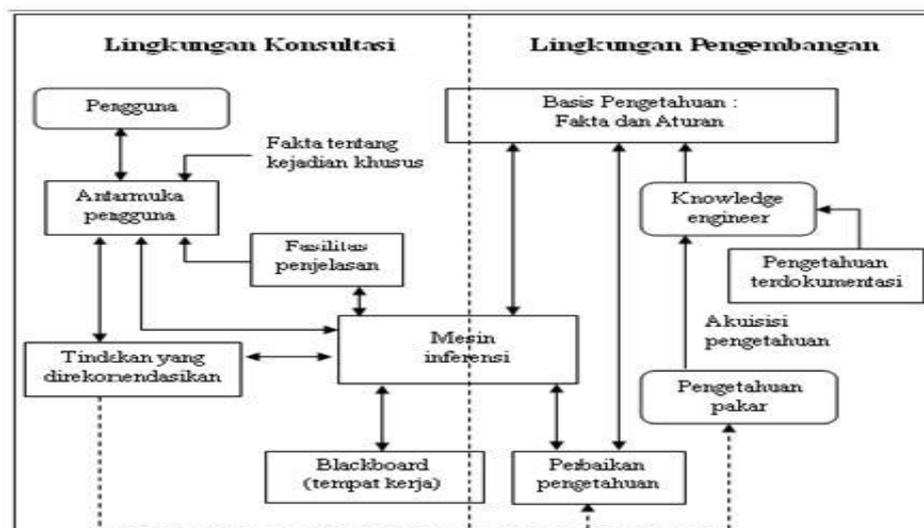
Ciri ciri dari sistem pakar (Sutojo, Mulyanto, & Suhartono, 2011:162) adalah sebagai berikut:

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, p. 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, p. 171).

2.1.6.1 Backward Chaining

Menurut (Sutojo, Mulyanto, & Suhartono, 2011, p. 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* di tempatkan 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

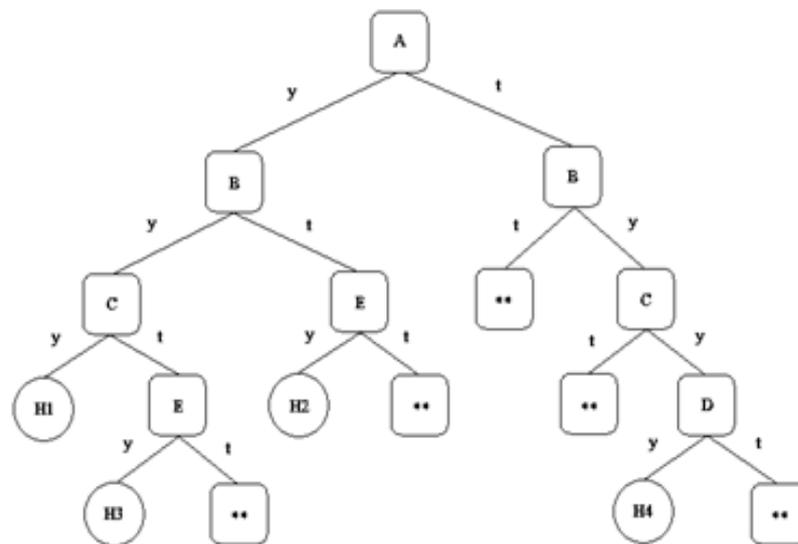
Tindakan mengacu pada kegiatan yang harus dilakukan sebelum hasil dapat diharapkan. Aksi mengacu pada kegiatan yang menyebabkan munculnya efek dari tindakan tersebut. Sebab mengacu pada keadaan tertentu yang menimbulkan akibat tertentu. Gejala mengacu pada keadaan yang menyebabkan adanya kerusakan atau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

2.2 Variabel Penelitian

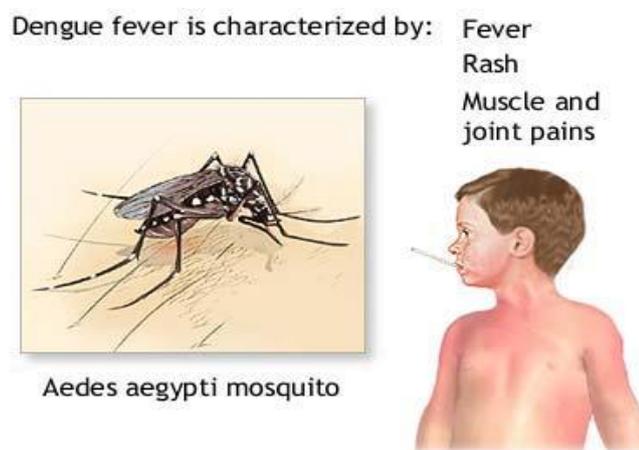
Menurut (Dr.Sudaryono, 2015, p. 17) variabel penelitian pada dasarnya adalah segala sesuatu yang berbentuk apa saja yang ditetapkan oleh peneliti untuk di pelajari sehingga diperoleh informasi dan kesimpulannya. Peneliti bekerja pada tingkat teoritis dan empiris. Pada tingkat teoritis, perhatiannya tercurah pada pengidentifikasian konsep dan hubungannya dengan proposisi. Pada tahap empiris, pernyataan riset akan di uji pada penelitian ini yang menjadi variabel adalah penyakit pada balita, jenis penyakit pada balita yaitu *Denque haermorrhagic fever*, kejang demam, infeksi saluran pernapasan, demam *thypoid* dan diare.

1. Penyakit *Denque haermorhagic fever*

Penyakit *Denque haermorhagic fever* merupakan penyakit yang disebabkan oleh virus denque sejenis virus yang tergolong arbovirus dan masuk kedalam tubuh penderita gigitan nyamuk aedes aegypti (betina).

a. Tanda dan Gejala

- Suhu tinggi
- Nyeri pada otot seluruh tubuh
- Nyeri dibelakang kepala
- Suara serak
- Batuk
- Epitaksis
- Disuria
- Anoreksia



Gambar 2.3 *Denque Haermorhagic Fever*
Sumber : Data Penelitian (2018)

b. Solusi dan Penanganannya

- Tirah baring
- Diet makan lunak
- Minum banyak
- Pemberian cairan intra vena
- Monitor tanda-tanda vital tiap 3 jam
- Periksa HB, HT, dan trombosit setiap hari
- Pemberian obat anti piretik
- Monitor tanda-tanda pendarahan lebih lanjut
- Pemberian anti biotik
- Bila kejang dapat diberi obat penenang

2. Penyakit Kejang Demam

Kejang demam terjadi pada saat suhu meningkat disebabkan oleh suatu proses ekstrakranium. Kejang adalah pembebasan listrik yang tidak terkontrol dari sel saraf cortex serebral yang ditandai dengan serangan yang tiba-tiba.

a. Tanda dan Gejala

- Ketidakseimbangan kimiawi
- Demam
- Patologis otak
- Ekslampsia
- Idiopatik

- Obat-obatan



Gambar 2.4 Penyakit Kejang Demam
Sumber : Data Penelitian (2018)

b. Solusi dan Penanganannya

- Pemberian Diazepam : dosis awal 0,3-0,5 mg/kg BB/dosis IV 9perlahan, bila kejang belum berhenti dapat diulang dengan dosis ulangan setelah 20 menit
- Turunkan demam : memberi anti piretik parasetamol atau salisilat 10 mg/kg BB/dosis, kompres air biasa
- Penanganan suportif : Bebaskan jalan nafas, beri zat asam, jaga keseimbangan cairan dan elektrolit, Pertahankan tekanan darah.

3. Penyakit ISPA

Infeksi Saluran Pernapasan atas adalah penyakit infeksi yang menyerang salah satu bagian dan atau lebih dari saluran napas, mulai dari hidung (saluran atas) hingga alveoli (saluran bawah) termasuk jaringan adneksanya, seperti sinus, rongga telinga tengah dan pleura.

a. Tanda dan Gejala

- Rinitis
- Nyeri tenggorokan
- Batuk dengan dahak kuning atau putih kental
- Nyeri retrosternal dan konjungvitis
- Suhu badan meningkat antara 4-7 hari disertai malas
- Mialgia
- Nyeri kepala
- Anoreksia
- Mual
- Muntah dan insomnia



Gambar 2.5 Penyakit ISPA
Sumber : Data Penelitian (2018)

b. Solusi dan Penangananya

- Menjaga keadaan gizi agar tetap baik

- Immunisasi
- Menjaga kebersihan perorangan dan lingkungan
- Mencegah anak berhubungan dengan penderita ISPA
- Pemberian obat antibiotik

4. Penyakit Demam Thypoid

Tifus abdominalis (demam tifioid, enteric fever) ialah penyakit infeksi akut yang biasanya terdapat pada saluran pencernaan dan gangguan kesadaran.

a. Tanda dan Gejala

- Perasaan tidak enak
- Lesu
- Nyeri kepala
- Pusing dan tidak bersemangat
- Demam : demam berlangsung selama 3 minggu
- Gangguan pada saluran pencernaan
- Gangguan kesadaran
- Panas tinggi yang kontinua
- Kesadaran menurun sampai spoor, koma atau delerium
- Dehidrasi, asidosis, peritonitis, bronkopneumonia
- Gizi yang buruk



Gambar 2.6 Penyakit Thypoid
Sumber : Data Penelitian (2018)

b. Solusi dan Penanganannya

- Isolasikan penderita dan desinfeksi pakaian
- Lakukan perawatan yang baik untuk menghindari komplikasi, mengingat sakit yang lama, lemah dan anoreksia dan lain-lain
- Selama masih demam harus istirahat, sampai dengan 2 minggu normal kembali yaitu istirahat total, seminggu kemudian boleh duduk dan selanjutnya boleh berdiri dan berjalan
- Diet makanan yang mengandung cairan, kalori dan tinggi protein
- Pemberian antoibiotik dan kloramfenikol

5. Penyakit Diare

Diare akut yang awalnya mendadak dan berlangsung singkat dalam beberapa jam sampai 7 atau 14 hari. Proses terjadinya diare akut dipengaruhi dua hal pokok yaitu konsistensi fases dan motilitas usus, umumnya terjadi akibat pengaruh keduanya. Gangguan proses mekanik dan ensimatik, disertai gangguan mukosa, akan mempengaruhi pertukaran air dan elektrolit, sehingga

mempengaruhi konsistensi fases yang terbentuk. Peristaltik saluran cerna yang teratur akan mengakibatkan proses cerna secara ensimatik yang akan mempengaruhi pola defikasi.

a. Tanda dan Gejala

- Mengalami nause
- Muntah
- Nyeri perut sampai kejang perut
- Demam
- Diare
- Kekurangan cairan menyebabkan pasien merasakan haus, lidah kering, tulang pipih menonjol dan turgor kulit menurun
- Suara menjadi serak



Gambar 2.7 Penyakit Diare
Sumber : Data Penelitian (2018)

b. Solusi dan Penanganannya

- Perilaku sehat : Pemberian ASI, Makanan pendamping ASI, Mencuci tangan, menggunakan air bersih yang cukup, menggunakan jamban, membuang tinja yang benar, pemberian imunisasi campak
- Penyehatan lingkungan : penyediaan air bersih, pengolahan sampah, sarana pembuangan air limbah
- Pemberian obat oralit, diberikan cairan ringer laktat, bila tak tersedia di berikan cairan NaCl isotonik ditambah satu ampul Na bikarbonat 7,5 % 50 ml
- Pemberian obat anti diare bersifat simtomatik
- Pemeriksaan darah tepi langka
- Pemeriksaan urine lengkap

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 pemograman yang ditulis atau diciptakan oleh *programer* yang selanjutnya di komplikasi dengan aplikasi kompuler sehingga menjadi sebuah kode yang nantinya akan dikenali oleh mesin *hardware*.

2.3.1 *Personal Home Page* (PHP)

2.3.1.1 Definisi PHP



Gambar 2.8 Logo PHP
Sumber: Data Penelitian (2018)

PHP merupakan suatu bahasa *scripting* khususnya digunakan untuk web *delevopment*. Karena sifatnya yang *server side scripting*, maka untuk menjalankan PHP harus menggunakan *web server*. PHP juga dapat diintegrasikan dengan HTML, JavaScript, JQuery, Ajax. Pada umumnya PHP lebih banyak digunakan bersamaan dengan file bertipe HTML. Dengan menggunakan PHP membuat *website powerful* yang dinamis disertai manajemen *database* (Hidayatullah & Kawistara, 2017:224).

2.3.1.2 Kelebihan PHP

PHP berbasis *server side scripting*. PHP dapat melakukan tugas-tugas yang dilakukan dengan mekanisme CGI seperti mengambil, mengumpulkan data dari database, *meng-generate* halaman dinamis, bahkan menerima dan mengirim cookie seperti dibawah ini:

1. PHP Berbasis *Server Side Scripting*
2. *Command Line Scripting*
3. PHP dapat Membuat Aplikasi Dekstop
4. Digunakan untuk Berbagai Macam *Platform Os*
5. Mendukung Berbagai Macam *Web Server*
6. *Object Oriented Programming* atau *Procedural*
7. Mendukung Banyak RDMS
8. Mendukung Banyak Komunikasi
9. Pengolahan Teks yang Sangat Baik
10. Membuat Program PHP

2.3.2 XAMPP

2.3.2.1 Definisi XAMPP



Gambar 2.9 Logo *Xampp*
Sumber : Data Penelitian (2018)

Menurut (Hidayatullah & Kawistara, 2017:123) *Web Server* merupakan tempat dimana menyimpan aplikasi *web*, kemudian mengaksesnya melalui

internet. *Web server* yang sangat terkenal adalah *apache*. *Apache web server* bisa digunakan di windows, linux dan MAC.

2.3.2.2 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*.

2.3.2.3 PHP MyAdmin



Gambar 2.10 Logo *phpMyAdmin*
Sumber : Data Penelitian (2018)

Salah satu alat paling ciamik untuk mengelola *database MySQL* adalah *PhpMyAdmin*. Memanfaatkan *phpMyAdmin*, proses pengaturan *privilege* dan manipulasi *database MySQL* lebih mudah dilakukan menggunakan antarmuka *web* oleh orang awam sekalipun.

2.3.3 MySQL

2.3.3.1 Definisi MySQL



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

MySQL merupakan salah satu DBMS yang sudah sangat banyak digunakan oleh pemrograman aplikasi *web* seperti *PostgreSQL*, *SQL Server*, *MS Access* dari *Microsoft*, *DB2* dari *IBM*, *Oracle* dan *Oracle Corp*, *Dbase*, *FoxPro*. Kelebihan dari MySQL adalah gratis, handal, selalu di *update* dan banyak forum yang memfasilitasi para pengguna jika memiliki kendala. MySQL juga menjadi DBMS yang sering di bundling dengan *web server* sehingga proses instalasinya menjadi lebih mudah (Hidayatullah & Kawistara, 2017, p. 175).

2.3.3.2 Cara Kerja MySQL

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*.

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*.

2.3.4 *StarUML*

2.3.4.1 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

2.3.5 *Notepad++*

***Notepad++* adalah sebuah *text editor* yang sangat berguna bagi setiap orang dan khususnya bagi para *developer* dalam membuat program. *Notepad++* menggunakan komponen *Scintilla* untuk dapat menampilkan dan**

menyuntingan teks dan berkas kode sumber berbagai bahasa pemrograman yang berjalan di atas sistem operasi *Microsoft Windows*. Selain manfaat dan kemampuannya menangani banyak bahasa pemrograman, *Notepad++* juga dilisensikan sebagai perangkat *free*. Jadi, setiap orang yang menggunakannya tidak perlu mengeluarkan biaya untuk membeli aplikasi ini karena *sourceforge.net* sebagai layanan yang memfasilitasi *Notepad++* membebaskannya untuk digunakan. Beberapa daftar bahasa program yang didukung oleh *Notepad++* adalah C, C++, Java, C#, XML, HTML, PHP, Javascript dan masih banyak lagi bahasa program yang didukung. *Notepad++* sangat ringan untuk digunakan, jadi sekalipun komputer yang Anda miliki dengan spesifikasi rendah tetap bisa menggunakannya karena seperti yang Anda ketahui beberapa program untuk menulis kode sekaligus *compiler*-nya biasanya membutuhkan komputer dengan spesifikasi tertentu.

2.3.6 *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 pemrograman berorientasi objek. Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman 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.

1. *Use Case Diagram*

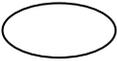
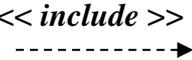
Use case diagram diketahui sebagai salah satu jenis diagram pada UML yang menggambarkan interaksi antara sistem dan aktor, *use case* diagram juga dapat mendeskripsikan tipe interaksi antara si pemakai sistem dengan sistemnya.

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).

Tabel 2.2 *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
	<i>DEFENDENCY INCLUDE</i> : Menunjukkan suatu <i>use case</i> harus dipenuhi
	<i>DEFENDENCY EXTEND</i> : Menunjukkan suatu <i>use case</i> akan di laksanakan bersifat <i>optional</i>

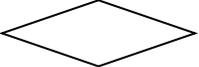
Sumber:(A.S & M.Shalahuddin, 2011)

2. *Activity Diagram*

Diagram aktivitas atau *Activty 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).

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

Tabel 2.3 *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 1568 646 1713"> <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).

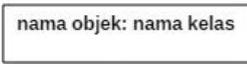
3. *Sequence Diagram*

Menurut (A.S & M.Shalahuddin, 2011) *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 semakin banyak *use case* yang didefenisikan maka diagram sekuen yang harus dibuat juga semakin banyak. Berikut simbol simbol yang ada pada diagram sekuen:

Tabel 2.4 *Sequence Diagram*

Simbol	Deskripsi
<p data-bbox="327 1485 480 1518">Aktor/<i>actor</i></p> 	<p data-bbox="743 1485 1343 1957">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</p>

<p>Garis hidup/<i>lifeline</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor</p>
<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan</p>
<p>Waktu aktif</p> 	<p>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</p>
<p>Pesan tipe <i>create</i></p> 	<p>Menyatakan suatu objek membuat objek yang lain. Arah panah mengarah pada objek yang dibuat</p>

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

4. *Class Diagram*

Menurut (A.S & M.Shalahuddin, 2011:141–146) Diagram kelas atau *class* diagram menggambarkan struktur segi dari pendefeniann 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 mendefinisikan dan mengatur tampilan pemakai

3. Kelas yang diambil dari pendefenisian *use case*

Kelas yang menangani fungsi-fungsi yang harus diambil dari pedefenisian *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.

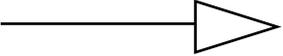
Jenis-jenis kelas diatas juga dapat digabungkan satu sama lain sesuai dengan pertimbangan yang dianggap baik asalkan fungsi-fungsi yang sebaiknya ada pada struktur kelas tetap ada. Susunan kelas juga dapat ditambahkan kelas utilitas seperti kenoeksi kebasis data, membaca *file teks*, dan lain sebagainya 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 2.5 *Diagram Class*

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

1. Nama Penulis : Magdalena Simanjuntak

Judul Jurnal :Sistem pakar diagnosa penyakit diare (gastroentritis) dengan menggunakan forward chaining

Volume / ISSN : Vol .2 No. 3 / 2548-429X

Kesimpulan : Sistem pakar diagnosa penyakit diare ini merupakan suatu sistem dimana masyarakat umum dengan mudah mendiagnosa penyakit diare secara cepat tanpa dibatasi waktu. Sistem pakar diagnosa penyakit diare sangat dibutuhkan oleh masyarakat umum karena dapat mendiagnosa penyakit diare dan mengetahui sebab, solusi, obat dalam penyembuhan diare yang diderita (simanjuntak, 2017).

2. Nama Penulis : Mhd Ridhon, Solikhun dkk

Judul Jurnal : Sistem Pakar diagnosa awal penyakit akibat virus pada anak berbasis *mobile* dengan *Forward chaining*.

Volume / ISSN : 2/ 2540-7600 2540-7597

Kesimpulan : Sistem ini dapat mendiagnosa gejala awal penyakit yang disebabkan virus yang menyerang pada anak-anak khususnya penyakit campak berdasarkan gejala-gejala dan aplikasi sistem pakar ini dapat melakukan diagnosa awal terhadap suatu penyakit yang disebabkan virus serta memberikan informasi mengenai definisi, pengobatan serta pencegahannya, sehingga dapat membantu orangtua dalam mengenali gejala serta jenis-jenis penyakit akibat virus pada anak-anak (Mhd Ridhon, Solikhun).

3. Nama Penulis : Evi Dewi Sri Mulyani, Irna Nur Restianie

Judul Jurnal : Aplikasi Sistem Pakar untuk mendiagnosa penyakit anak (balita) dengan menggunakan metode *forward chaining*.

Volume / ISSN : 2302-3805

Kesimpulan : Aplikasi sistem pakar ini dapat mendiagnosa suatu penyakit anak (balita) yang sering diderita berdasarkan gejala yang dialami, kemudian dapat memberikan informasi penyakit tersebut beserta solusinya. Dengan diberikan *interface yang user friendly* maka aplikasi sistem pakar ini dapat mudah dipelajari dan mudah digunakan untuk kalangan orang awam, sehingga dapat mempermudah dalam mendiagnosa penyakit (Dewi, Mulyani, & Restianie, 2016).

4. Nama Penulis : Joko Trianto

Judul Jurnal : Penerapan metode *forward chaining* untuk diagnosa penyakit diare pada anak usia 3-5 tahun berbasis *mobile android*.

Volume / ISSN : 2541-1004

Kesimpulan : Tampilan sistem aplikasi yang dibuat masih sederhana, untuk penelitian selanjutnya diharapkan sistem yang dirancang dan dibuat dapat menyempurnakan penelitian sebelumnya agar dapat menghasilkan tampilan yang lebih menarik dan inovatif. Sistem aplikasi yang dibuat belum cukup akurat dalam mendiagnosa penyakit diare, untuk penelitian selanjutnya sistem diharapkan dapat dikembangkan lagi untuk menghasilkan hasil diagnosa yang lebih akurat dibandingkan penelitian sebelumnya. Pembuatan sistem aplikasi ini masih beroperasi secara offline, sebaiknya pengembangan kedepannya dapat beroperasi secara *online* (Trianto, Informatika, Pamulang, & Selatan-indonesia, 2018).

5. Nama Penulis : Bagus fery yanto dkk

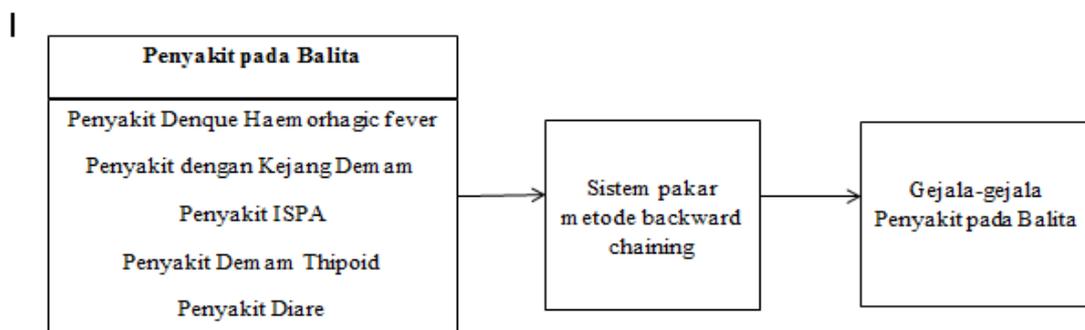
Judul Jurnal : Aplikasi sistem pakar diagnosa penyakit pada anak bawah lima tahun menggunakan metode *forward chaining*.

Volume / ISSN : 3 No.1 / 61-67

Kesimpulan : Pembuatan aplikasi sistem pakar diagnosa penyakit pada Balita menggunakan metode *forward chaining* dapat dilakukan dengan tiga tahap. Tahap pertama adalah pengumpulan data dan informasi dari MTBS dan wawancara. Tahap kedua adalah pembuatan *rule* berdasarkan 18 penyakit menggunakan metode *forward chaining*. Tahap ketiga adalah implementasi aplikasi sistem pakar berbasis *android* dengan fitur diagnosa penyakit, riwayat diagnosa dan kumpulan penyakit. Hasil evaluasi menunjukkan tingkat akurasi sistem yang telah dibuat sebesar 82% dengan menggunakan 50 data uji coba (Yanto et al., 2017).

2.5 Kerangka Berpikir

Kerangka pemikiran yang baik akan menjelaskan secara teoritis peraturan antara variabel yang akan diteliti. Serta secara teoritis dapat dijelaskan 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 2.13 Kerangka Berpikir

Sumber: Data Penelitian (2018)

Berdasarkan Gambar 2.14 kerangka berpikir input yang pertama adalah jenis-jenis penyakit pada balita kemudian sistem pakar dengan metode *backward chaining*, kemudian jenis-jenis penyakit pada balita yang didapat dari wawancara seorang pakar dan diolah dengan metode *backward chaining* untuk dimasukkan kedalam sebuah *web* kemudian *output* atau keluaran dari penyakit tersebut adalah gejala-gejala dari penyakit pada balita tersebut.