



PERANCANGAN SISTEM PAKAR KERUSAKAN JARINGAN DENGAN METODE FORWARD CHAINING

Disusun dan diajukan untuk memenuhi persyaratan Ujian Akhir memperoleh
Gelar Sarjana Komputer pada Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

OLEH

**NAMA : TOGAR MANATAR SILABAN
NPM : 1514370411
PROGRAM STUDI : SISTEM KOMPUTER**

**PROGRAM STUDI SISTEM KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
2019**

ABSTRAK

TOGAR MANATAR SILABAN

Perancangan Sistem Pakar Kerusakan Jaringan

Dengan Metode Forward Chaining

2019

Perkembangan teknologi informasi dan komunikasi yang sangat pesat sebagai penunjang kegiatan manusia dalam melaksanakan pekerjaan sehari-hari membuat penggunaan informasi dan komunikasi semakin meningkat pesat. Penggunaan informasi dan komunikasi yang tinggi bisa menyebabkan kerusakan jaringan. Sejalan dengan hal ini, maka kebutuhan akan pakar komputer semakin dibutuhkan. Pakar komputer yang dimaksud adalah seseorang yang memiliki keahlian khususnya dalam menangani masalah yang berkaitan dengan masalah kerusakan jaringan komputer. Pada Penelitian ini penulis ingin membuat sebuah sistem yang dapat membantu seorang teknisi dalam mendeteksi kerusakan jaringan. Sistem pakar adalah merupakan salah satu cabang kecerdasan buatan yang mempelajari bagaimana meniru cara berpikir seorang pakar dalam menyelesaikan suatu permasalahan, membuat keputusan maupun mengambil kesimpulan sejumlah fakta. Metode yang digunakan adalah *forward chaining* yaitu pencarian atau teknik pelacakan kedepan yang dimulai dengan informasi yang ada dan penggabungan *rule* untuk menghasilkan suatu kesimpulan atau tujuan. Diharapkan dengan adanya perancangan sistem pakar ini, mampu membantu seorang teknisi dalam mendeteksi kerusakan jaringan komputer didalam suatu perusahaan penyedia layanan internet.

Kata kunci : *Sistem Pakar, Jaringan Komputer, Forward Chaining*

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya, sehingga penulis dapat merampungkan skripsi dengan judul: “Perancangan Sistem Pakar Kerusakan Jaringan Dengan Metode Forward Chaining”. Ini untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar Sarjana Komputer Strata Satu pada Program Studi Sistem Komputer Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.

Dalam kesempatan ini, penulis mengucapkan terima kasih yang sebesar-besarnya kepada pihak yang telah membantu dalam penyusunan Tugas Akhir ini. Penulis ingin mengucapkan terima kasih kepada :

1. Orangtua tercinta Bapak J. Silaban dan R. Banjarnahor yang telah mencurahkan segenap cinta dan kasih sayang semoga Tuhan Yang Maha Kuasa memberikan berkat melimpah, kesehatan serta umur yang panjang
2. Dekan Fakultas Sains dan Teknologi, Ibu Sri Shindi Indri, ST, MSc
3. Ketua Program Studi Sistem Komputer, Bapak Eko Hariyanto, S.Kom.,M.Kom
4. Dosen Pembimbing I, Bapak Hafni, S.Kom.,M.Kom
5. Dosen Pembimbing II, Bapak Raja Narsul Fuad, S.Kom.,M.Kom
6. Pemimpin PT Bank Sumut Cabang Sidikalang, Bapak Efendi Karokaro
7. Wina Elda Sari Siregar, Amd.Kes yang telah senantiasa memberikan dukungan, semangat dan motivasi sehingga penulis mampu berjuang untuk menyelesaikan skripsi ini
8. Teman – teman Kelas KK II yang sudah mendukung dalam menyelesaikan skripsi ini

Penulis juga menyadari bahwa penyusunan Tugas Akhir ini belum sempurna baik dalam penulisan maupun isi disebabkan keterbatasan kemampuan penulis. Oleh karena itu, penulis mengharapkan kritik dan saran yang sifatnya membangun dari pembaca untuk kesempurnaan isi Tugas Akhir ini.

Medan, Juli 2019
Penulis

TOGAR M SILABAN
NPM. 1514370411

DAFTAR GAMBAR

Gambar 2.1 Sistem Pakar	8
Gambar 2.2 Hierarki Pengetahuan	12
Gambar 2.3 StrukturmKerja metode forward chaining	16
Gambar 3.1 Paradigma	37
Gambar 3.2 Use Case diagram	47
Gambar 3.3 Tampilan awal	48
Gambar 3.4 Tampilan halaman utama	48
Gambar 3.5 Tampilan menu cek masalah.....	49
Gambar 4.1 Tampilan XAMPP	54
Gambar 4.2 Tampilan database	55
Gambar 4.3 Tampilan awal	56
Gambar 4.4 Tampilan cek masalah	56
Gambar 4.5 Solusi	57

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR GAMBAR	iv
DAFTAR TABEL	v
DAFTAR LAMPIRAN	vi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah	2
1.3. Batasan Masalah	2
1.4. Tujuan Penelitian	2
1.5. Manfaat Penelitian	3
BAB II LANDASAN TEORI	4
2.1. Sistem Pakar	4
2.1.1. Komponen Sistem Pakar	4
2.1.2. Elemen Manusia pada Sistem Pakar	9
2.1.3. Representasikan Pengetahuan dan Penalaran	11
2.2. Metode <i>Forward Chaining</i>	15
2.3. Jaringan Komputer	17
2.4. Alat Bantu Perancangan Sistem	18
2.4.1. <i>Unified Modelling Language</i>	18
2.4.2. <i>Use Case</i> Diagram	19
2.4.3. Class Diagram	21
2.4.4. Sequence Diagram	22
2.4.5. <i>Communication</i> Diagram	25
2.4.6. <i>Statechart</i> Diagram	26
2.4.7. <i>Activity</i> Diagram	27
2.4.8. <i>Deployment</i> Diagram	29
2.5. Bagan Alir (<i>flowchart</i>)	30
2.6. Web	31
2.7. PHP	32
2.8. MYSQL	34
2.9. XAMPP	35
BAB III METODE PENELITIAN	36
3.1. Metode Penelitian	36
3.1.1. Metode Pengumpulan Data	36
3.1.2. Metode Perancangan Sistem	37
3.2. Analisa Sistem	39
3.2.1. Kebutuhan Sistem	39
3.3. Analisa Data	40
3.4. Perancangan UML	46
3.4.1. Use Case Diagram	46
3.5. Perancangan Interface Program	47
3.6. Perancangan Database	49
BAB IV HASIL DAN PEMBAHASAN	51

4.1.	Hasil	51
4.1.1.	Kebutuhan Hardware	52
4.1.2.	Kebutuhan Software	52
4.2.	Implementasi Program	52
4.2.1.	Proses Instalasi XAMPP 1.7.4.	52
4.2.2.	Tampilan Database Sistem Pakar Kerusakan Jaringan	54
4.3.	Pembahasan	55
BAB V	KESIMPULAN DAN SARAN	58
5.1.	Kesimpulan	58
5.2.	Saran	59
DAFTAR PUSTAKA		
BIOGRAFI PENULIS		
LAMPIRAN-LAMPIRAN		

LAMPIRAN – LAMPIRAN

Lampiran 1. Lembar Permohonan Pra Pengajuan Judul	L-1
Lampiran 1. Lembar Permohonan Pengajuan Judul	L-2
Lampiran 2. Lembar Bimbingan Skripsi	L-3
Lampiran 3. Lembar Permohonan Meja Hijau	L-4
Lampiran 4. Surat Keterangan Riset.....	L-5
Lampiran 4. Coding Program	L-5

DAFTAR TABEL

Tabel 2.1 Hubungan Pemakai, kepentingan pemakai dan fungsi Sistem Pakar	11
Tabel 2.2 Tabel Keputusan	14
Tabel 2.3 Simbol – simbol <i>use case</i> diagram	19
Tabel 2.4 Simbol – simbol <i>Class</i> diagram	21
Tabel 2.5 Simbol – simbol <i>sequence</i> diagram	22
Tabel 2.6 Simbol – simbol <i>communication</i> diagram	25
Tabel 2.7 Simbol – simbol <i>statechart</i> diagram	26
Tabel 2.8 Simbol – simbol <i>activity</i> diagram	27
Tabel 2.9 Simbol – simbol <i>deployment</i> diagram	28
Tabel 2.10 Simbol – simbol <i>flowchart</i>	29
Tabel 3.1 Indikator pada masalah jaringan	42
Tabel 3.2 Basis aturan kerusakan jaringan	43
Tabel 3.4 Tabel User	50

BAB I

PENDAHULUAN

1.1. Latar Belakang

Kebutuhan akan informasi saat ini sangatlah meningkat pesat dimana semua orang harus terhubung ke internet agar dapat mendapatkan berita terbaru informasi dan komunikasi dengan rekan kerja atau bisnis, dengan semakin banyak pengguna semakin banyak juga kendala yang terjadi pada masalah jaringan. Untuk itu, pihak yang menyediakan jasa jaringan atau internet sangatlah membutuhkan SDM yang terlatih dan terampil dalam instalasi atau perawatan jaringan.

Permasalahan sekarang banyak teknisi yang belum terampil bahkan masih pemula dikarenakan dalam perekrutan karyawan banyak yang tidak sesuai dalam bidang keahliannya, dengan perkembangan teknologi masalah ini bisa di selesaikan dengan sistem pakar dimana bisa menyimpan dan menalar seperti seorang ahli.

Metode yang dapat digunakan dalam sebuah sistem pakar salah satunya metode *forwardchaining*, Metode *Forward Chaining* adalah metode pencarian atau teknik pelacakan kedepan yang dimulai dengan informasi yang ada dan penggabungan rule untuk menghasilkan suatu kesimpulan atau tujuan. Pelacakan maju ini sangat baik jika bekerja dengan permasalahan yang dimulai dengan rekaman informasi awal dan ingin dicapai penyelesaian akhir, karena seluruh proses akan dikerjakan secara berurutan maju (Verina, 2015).

Berdasarkan uraian diatas, maka penelitian ini berjudul “**Perancangan Sistem Pakar Kerusakan Jaringan dengan metode *Forward Chaining***”

1.2. Perumusan Masalah

Berdasarkan latar belakang di atas, maka didapatkan beberapa rumusan masalah sebagai berikut :

1. Bagaimana membuat sistem pakar mendeteksi kerusakan jaringan komputer
2. Bagaimana penerapan sistem pakar dengan metode *forward chaining*
3. Bagaimana sistem pakar ini dapat membantu teknisi dalam masalah jaringan komputer

1.3. Batasan Masalah

Permasalahan yang diangkat pada penelitian ini memiliki batasan masalah sebagai berikut:

1. Pengambilan data atau rule masalah pada jaringan yaitu masalah hardware dan software
2. Metode yang digunakan dalam sistem pakar menggunakan metode *forward chaining*

1.4. Tujuan Penelitian

Dalam penulisan ini ada beberapa tujuan yang akan dicapai, diantaranya :

1. Menerapkan metode *forward chaining* dalam sistem pakar

2. Dengan sistem pakar dapat membantu teknisi dalam menganalisa masalah pada jaringan
3. Membuat sebuah sistem yang dapat menirukan atau menyimpan informasi seperti ahli

1.5. Manfaat Penelitian

1. Digunakan sebagai alat bantu dalam menganalisa permasalahan pada jaringan komputer
2. Membantu teknisi dalam menambah ilmu dalam bidang jaringan komputer.
3. Menambah wawasan penulis sendiri tentang sistem pakar dengan metode *forward chaining*

BAB II

LANDASAN TEORI

2.1. Sistem Pakar

Sistem pakar adalah sistem yang menirukan apa yang dikerjakan oleh seorang pakar ketika mengatasi permasalahan yang rumit, berdasarkan pengetahuan yang dimilikinya. Pengetahuan sistem pakar dibentuk dari kaidah atau pengalaman tentang perilaku elemen dari domain bidang pengetahuan tertentu. Pengetahuan pada sistem pakar diperoleh dari orang yang mempunyai pengetahuan pada suatu bidang tertentu, buku-buku, jurnal ilmiah maupun dokumentasi yang tercetak lainnya. Pengetahuan-pengetahuan tersebut dipresentasikan dalam format tertentu, dan dihimpun dalam suatu basis pengetahuan. Basis pengetahuan ini selanjutnya dipakai sistem pakar untuk menentukan penalaran atas problema yang dihadapinya. Sistem pakar mencoba mencari penyelesaian yang memuaskan, yaitu sebuah penyelesaian yang cukup bagus agar sebuah pekerjaan dapat berjalan walaupun itu bukan penyelesaian yang optimal (**Hartati dan Iswanti, 2016**).

2.1.1. Komponen Sistem Pakar

Sistem pakar sebagai sebuah program yang difungsikan untuk menirukan pakar manusia harus melakukan hal-hal yang dapat dikerjakan oleh seorang pakar. Untuk membangun sistem yang seperti itu maka komponen-komponen yang harus dimiliki adalah sebagai berikut (**Hartati dan Iswanti, 2016**):

- 1) Antar muka pengguna (*User nterface*)

Sistem pakar menggantikan seorang pakar dalam suatu situasi tertentu, maka sistem harus menyediakan pendukung yang diperlukan oleh pemakai yang tidak memahami masalah teknis. Sistem pakar juga menyediakan komunikasi antar sistem dan pemakainya, yang disebut sebagai antar muka. Antar muka yang efektif dan ramah pengguna (*user-friendly*) penting sekali terutama bagi pemakai yang tidak ahli dalam bidang yang diterapkan pada sistem pakar.

2) Basis pengetahuan (*Knowledge Base*)

Basis pengetahuan merupakan kumpulan pengetahuan bidang tertentu pada tingkatan pakar dalam format tertentu. Pengetahuan ini diperoleh dari akumulasi pengetahuan pakar dan sumber-sumber pengetahuan lainnya seperti yang telah disebutkan sebelumnya. Basis pengetahuan bersifat dinamis, bisa berkembang dari waktu ke waktu. Perkembangan ini disebabkan karna pengetahuan selalu bertambah, *terupdate*. Pada sistem pakar basis pengetahuan terpisah dari mesin inferensi, pemisahan ini bermanfaat untuk pengembangan sistem pakar secara leluasa disesuaikan dengan perkembangan pengetahuan pada suatu domain. Penambahan dan pengurangan dapat dilakukan pada basis pengetahuan ini tanpa mengganggu mesin inferensi.

3) Mesin inferensi (*Inference Machine*)

Mesin inferensi merupakan otak dari sistem pakar, berupa perangkat lunak yang melakukan tugas inferensi penalaran sistem pakar, biasanya dikatakan sebagai mesin pemikir (*Thinkin machine*). Pada prinsipnya

mesin inferensi inilah yang akan mencari solusi dari suatu permasalahan. Konsep yang biasanya digunakan untuk mesin inferensi adalah runut balik (*top-down*), yaitu proses penalaran yang berawal dari tujuan yang kita inginkan, menelusuri fakta-fakta yang mendukung untuk mencapai tujuan. Selain itu dapat juga menggunakan runut maju (*bottom-up*), yaitu proses penalaran yang bermula dari kondisi yang diketahui menuju tujuan yang diinginkan.

Mesin inferensi sesungguhnya adalah program komputer yang menyediakan metodologi untuk melakukan penalaran tentang informasi pada basis pengetahuan dan pada memori kerja, serta untuk merumuskan kesimpulan-kesimpulan. Komponen ini menyajikan arahan-arahan tentang bagaimana menggunakan pengetahuan dari sistem dengan membangun agenda yang mengelola dan mengontrol langkah-langkah yang diambil untuk menyelesaikan masalah ketika dilakukan konsultasi. Didalam mesin inferensi ini terdapat agenda, yaitu daftar prioritas aturan yang dibuat oleh mesin inferensi, yang polanya dipenuhi oleh fakta atau obyek dalam memori kerja.

4) Memory kerja

Merupakan bagian dari sistem pakar yang menyimpan fakta-fakta yang diperoleh saat dilakukan proses konsultasi. Fakta-fakta inilah yang nantinya akan diolah oleh mesin inferensi berdasarkan pengetahuan yang disimpan dalam basis pengetahuan untuk menentukan suatu

keputusan pemecahan masalah. Konklusinya bisa berupa hasil diagnosa, tindakan dan akibat.

Sistem pakar memiliki beberapa keunggulan dalam penggunaannya, yaitu:

(Kemala, 2015)

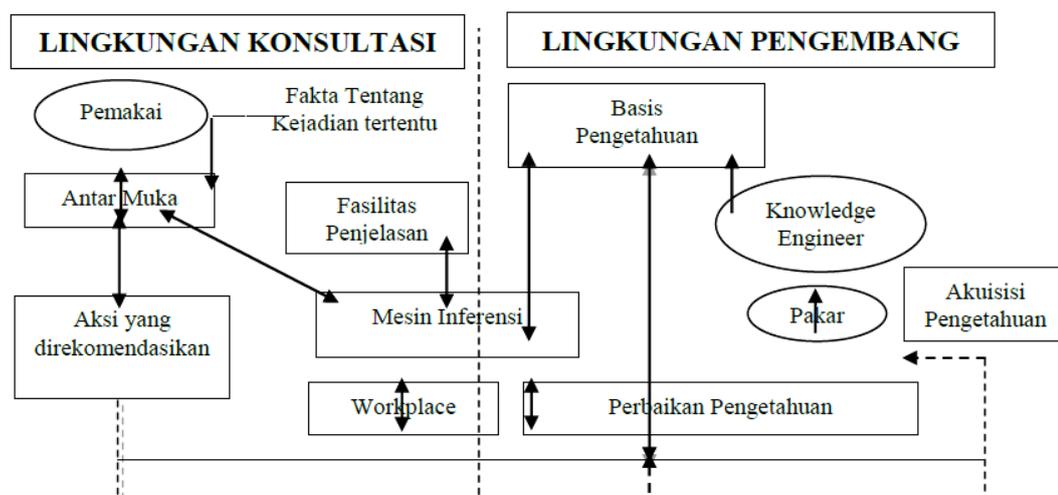
- 1) Membuat seorang awam pada bidang tertentu dapat bekerja layaknya seorang pakar pada bidang tersebut.
- 2) Memudahkan pengoperasian peralatan yang kompleks karena sistem pakar dapat melatih pekerja yang tidak berpengalaman.
- 3) Dapat diandalkan karena sistem pakar tidak pernah lelah atau bosan serta konsisten dan perhatian penuh dalam memberi jawaban.
- 4) Dapat bekerja dengan informasi yang tidak lengkap atau tidak pasti.
- 5) Meningkatkan kualitas, karena sistem pakar menyediakan nasehat yang konsisten serta mengurangi tingkat terjadinya kesalahan. Selain keunggulan, sistem pakar juga memiliki kelemahan dibandingkan dengan seorang ahli.

Sistem pakar memiliki beberapa ciri dan karakteristik yang berbeda dengan sistem yang lain adalah sebagai berikut: **(Kemala, 2015)**

- 1) Pengetahuan sebuah sistem pakar adalah suatu konsep yang bersifat fakta dan aturan, sedangkan proses pengolahan datanya bersifat numerik.
- 2) Informasi dalam sistem pakar tidak selalu lengkap, subjektif, tidak konsisten (berubah tergantung pada kondisi lingkungan), dapat dimodifikasi sehingga keputusan yang diambil bersifat tidak pasti namun mengikuti aturan tertentu.

- 3) Solusi permasalahan yang akan ditangani oleh sistem pakar bervariasi dan memiliki banyak pilihan jawaban yang dapat diterima semua faktor karena memiliki ruang masalah yang luas dan tidak pasti. Sistem fleksibel dalam menangani kemungkinan solusi dari berbagai masalah.
- 4) Perubahan pengembangan pengetahuan sistem pakar terjadi setiap saat dan sepanjang waktu. Modifikasi sistem untuk menampung jumlah pengetahuan semakin besar dan bervariasi.
- 5) Keakuratan solusi yang diberikan sangat penting, karena itu dibutuhkan informasi yang cukup untuk menunjang solusi tersebut
- 6) Output tergantung dari dialog dengan user.
- 7) Terbatas pada bidang spesifik

Dengan adanya fasilitas ini pada sistem, maka seorang pakar akan dengan mudah menambahkan pengetahuan ataupun kaidah baru pada sistem pakar. Untuk menjamin bahwa pengetahuan pada sistem pakar ini *up to date* dan valid, maka fasilitas akuisisi pengetahuan hanya bisa diakses oleh pakar.



Gambar 2.1. Sistem Pakar

Sumber : (Verina, 2015)

2.1.2. Elemen Manusia Pada Sistem Pakar

Menurut (Hartati dan Iswanti, 2016) “Sistem pakar tidak lepas dari elemen manusia yang terkait di dalamnya”. Personil yang terkait dalam sistem pakar ada 4, yaitu:

1) Pakar

Pakar adalah seorang individu yang memiliki pengetahuan khusus, pemahaman, pengalaman dan metode-metode yang digunakan untuk memecahkan masalah persoalan dalam bidang tertentu. Seorang pakar memiliki kemampuan kepakar, yaitu:

- a) Dapat mengenali dan merumuskan suatu masalah
- b) Menyelesaikan masalah dengan cepat dan tepat
- c) Menjelaskan solusi dari suatu masalah
- d) Restrukturisasi pengetahuan
- e) Belajar dari pengalaman
- f) Memahami batas kemampuan

Selain itu, pakar juga memiliki kemampuan untuk mengaplikasikan pengetahuannya dan memberikan saran serta pemecahan masalah pada domain tertentu. Ini merupakan pekerjaan pakar, memberikan pengetahuan tentang bagaimana seseorang melaksanakan tugas untuk menyelesaikan masalah.

2) Pembangun / Pembuat Pengetahuan

Pembangun pengetahuan memiliki tugas utama menerjemahkan dan merepresentasikan pengetahuan yang diperoleh dari pakar, baik berupa pengalaman pakar dalam menyelesaikan masalah maupun sumber

terdokumentasi lainnya ke dalam bentuk yang bisa diterima oleh sistem pakar. Dalam ini pembangun pengetahuan (*knowledge engineering*) menginterpretasikan dan merepresentasikan pengetahuan yang diperoleh dalam bentuk jawaban-jawaban atas pertanyaan-pertanyaan yang diajukan pada pakar atau pemahaman penggambaran analogis, sistematis, konseptual yang diperoleh dari membaca beberapa dokumen cetak seperti *text book*, jurnal, makalah dan sebagainya.

3) Pembangun / Pembuat Sistem

Pembangun sistem adalah orang yang bertugas untuk merancang antar muka pemakai sistem pakar, merancang pengetahuan yang sudah diterjemahkan oleh pembangun pengetahuan ke dalam bentuk yang sesuai dan dapat diterima oleh sistem pakar dan mengimplementasikannya ke dalam mesin inferensi. Selain hal tersebut pembangun sistem juga bertanggung jawab apabila sistem pakar akan diintegrasikan dengan sistem komputerisasi lain. Alat pembangun (*tool builder*) dapat dipakai untuk menyajikan atau membangun tool yang spesifik. Penjual (*vendor*) dapat memberikan *tool* dan saran, staf pendukung dapat memberikan saran dan bantuan secara teknis dalam proses pembangunan sistem pakar.

4) Pengguna

Banyak sistem berbasis komputer mempunyai susunan pengguna tunggal. Hal ini berbeda jauh dengan sistem pakar yang kemungkinan mempunyai beberapa kelas pengguna. Tabel dibawah menunjukkan beberapa contoh

hubungan antar kelas pengguna, kepentingan pengguna dan fungsi dari sistem pakar.

Tabel 2.1 Hubungan Pemakai, Kepentingan Pemakai dan Fungsi Sistem Pakar

Pemakai	Kepentingan	Fungsi Sistem Pakar
Klien bukan pakar	Mencari saran/nasehat	Konsultan atau penasehat.
Mahasiswa	Belajar	
Pembangun sistem dan pengetahuan.	Memperbaiki/menambah basis pengetahuan, merancang sistem	Partner
Pakar.	Membantu analisis rutin atau proses komputasi, mengklasifikasikan informasi, alat bantu diagnosa	Rekan kerja atau asisten.

(Sumber : Hartati dan Iswanti, 2016)

Pengguna mungkin tidak terbiasa dengan komputer dan mungkin pada domain masalah. Bagaimanapun juga, banyak solusi permasalahan menjadi lebih baik dan kemungkinan lebih murah dan keputusan yang cepat bila menggunakan sistem pakar. Pakar dan pembangun sistem harus mengantisipasi kebutuhan-kebutuhan pengguna dan membuat batasan-batasan ketika mendesain sistem pakar.

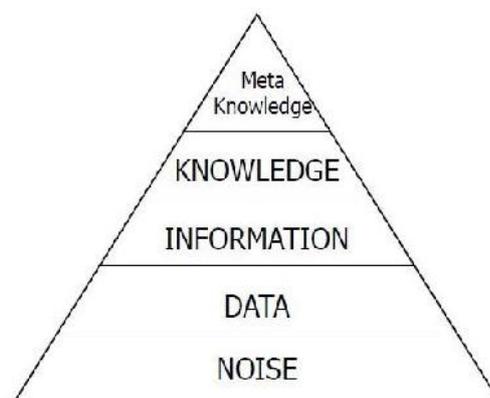
2.1.3. Representasi Pengetahuan dan Penalaran

Setelah menentukan bidang kepakaran yang akan dibuatkan sistem pakarnya, maka selanjutnya melakukan representasi pengetahuan dengan cara mengumpulkan pengetahuan yang sesuai dengan domain keahlian tersebut. Pengetahuan yang dikumpulkan bisa dari media cetak, elektronik maupun pengetahuan dari pakar keahlian dan pengalamannya. Pengetahuan yang

dikumpulkan tidak bisa dimasukkan begitu saja ke dalam suatu komputer, harus mengikuti format yang bisa dimengerti komputer. Untuk itu pengetahuan harus terlebih dahulu direpresentasikan atau disajikan ke dalam format tertentu dan akan dihimpun dalam suatu basis pengetahuan (**Hartati dan Iswanti, 2016**).

1) Pengetahuan

Pemrosesan yang dilakukan oleh sistem pakar merupakan pemrosesan pengetahuan, bukan pemrosesan data seperti yang dikerjakan dengan pemrograman secara konvensional yang kebanyakan dilakukan oleh sistem informasi. Pengetahuan (*knowledge*) yang digunakan pada sistem pakar merupakan serangkaian informasi mengenai gejala-diagnosa, sebab-akibat, aksi-reaksi tentang suatu domain tertentu. Pengetahuan merupakan bagian dari suatu hierarki seperti yang terlihat pada gambar dibawah ini:



Gambar 2.2. Hierarki Pengetahuan
(Sumber : Hartati dan Iswanti, 2016)

Noise adalah suatu item data yang tidak mempunyai maksud (*interest*), atau data yang masih kabur atau tidak jelas tidak ada artinya. Tingkat berikutnya adalah data yaitu item yang mempunyai makna potensial.

Data diolah menjadi informasi. *Meta knowledge* adalah pengetahuan tentang keahlian. Keahlian merupakan pemahaman yang luas dari pengetahuan spesifik yang diperoleh dari hasil pembelajaran, pelatihan dan pengalaman.

2) Kaidah produksi

Kaidah menyediakan cara formal untuk merepresentasikan rekomendasi, arahan atau strategi. Kaidah produksi dituliskan dalam bentuk jika-maka (*if-then*). Kaidah *if-then* menghubungkan antesenden dengan konsekuensi yang diakibatkannya. Berbagai struktur kaidah *if-then* yang menghubungkan obyek atau atribut sebagai berikut:

IF premis THEN konklusi

IF masukan THEN keluaran

IF kondisi THEN tindakan

IF antesenden THEN konsekuensi

IF data THEN hasil

IF tindakan THEN tujuan

IF aksi THEN reaksi

IF sebab THEN akibat

IF sebab THEN akibat

IF gejala THEN diagnosa

Premis mengacu pada fakta yang harus benar sebelum konklusi tertentu diperoleh. Masukan mengacu pada data yang harus tersedia sebelum keluaran dapat diperoleh. Kondisi mengacu pada keadaan yang harus

berlaku sebelum tindakan dapat diambil. Antesenden mengacu situasi yang terjadi sebelum konsekuensi dapat diamati. Data mengacu pada informasi yang harus tersedia sehingga sebuah hasil dapat diperoleh. Tindakan mengacu pada kegiatan yang harus dilakukan sebelum hasil dapat diharapkan. Aksi mengacu pada kegiatan yang menyebabkan munculnya efek dari kegiatan tersebut. Sebab mengacu pada keadaan tertentu yang menimbulkan akibat tertentu. Gejala mengacu pada keadaan menyebabkan adanya kerusakan atau keadaan tertentu yang mendorong adanya pemeriksaan.

3) Tabel keputusan dan Pohon keputusan

Tabel keputusan merupakan suatu cara untuk mendokumentasikan pengetahuan. Tabel keputusan merupakan matrik kondisi yang dipertimbangkan dalam pendeskripsian kaidah. Berikut contoh tabel keputusan:

Tabel 2.2.Tabel Keputusan

Kondisi 1	√	
Kondisi 2	√	√
Kondisi 3		√

(Sumber: Hartati dan Iswanti, 2016)

Kaidah yang disajikan dalam bentuk kaidah produksi disusun dari tabel keputusan. Kaidah secara langsung dapat dihasilkan dari tabel keputusan tetapi untuk menghasilkan kaidah yang efisien terdapat suatu langkah yang harus ditempuh yaitu membuat pohon keputusan. Pohon keputusan

dapat diketahui atribut atau kondisi yang dapat direduksi sehingga menghasilkan kaidah yang efisien dan optimal. Pohon keputusan yang dibuat mengacu pada tabel keputusan dan dapat digunakan sebagai acuan untuk mereduksi atribut-atribut yang sebenarnya dapat dihilangkan dalam proses identifikasi suatu sistem.

2.2. Metode *Forward Chaining*

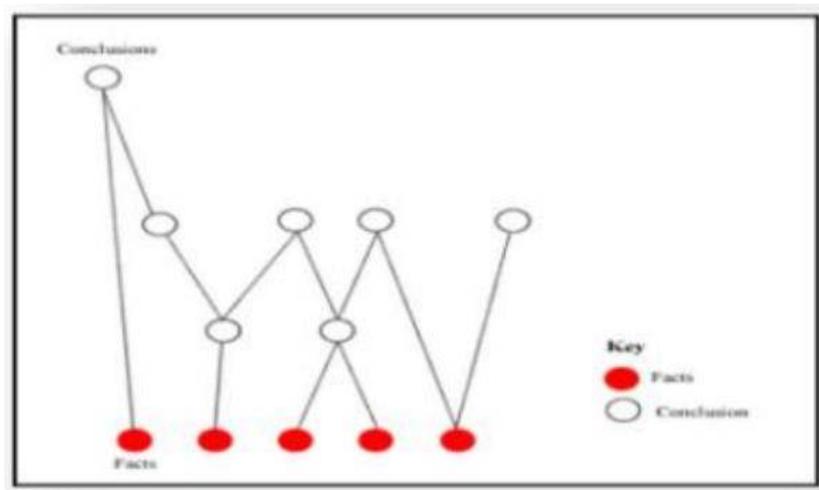
Metode *Forward Chaining* adalah metode pencarian atau teknik pelacakan kedepan yang dimulai dengan informasi yang ada dan penggabungan *rule* untuk menghasilkan suatu kesimpulan atau tujuan. Pelacakan maju ini sangat baik jika bekerja dengan permasalahan yang dimulai dengan rekaman informasi awal dan ingin dicapai penyelesaian akhir, karena seluruh proses akan dikerjakan secara berurutan maju (**Verina, 2015**).

Forward Chaining adalah tehnik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta – fakta tersebut dengan bagian IF dari rules IF – THEN. Bila ada fakta yang cocok dengan bagian IF, maka rule tersebut dieksekusi. Bila sebuah rule dieksekusi, maka sebuah fakta baru (bagian THEN) ditambahkan ke dalam database.

Langkah – langkah dalam membuat sistem pakar dengan menggunakan metode *forward chaining* yaitu: (**Irmayani, 2015**)

- 1) Pendefinisian masalah dimulai dengan pemilihan domain masalah dan akuisi pengetahuan.
- 2) Pendefinisian data input, yaitu sistem *forward chaining* memerlukan data awal untuk memulai inferensi.

- 3) Pendefinisian struktur pengendalian data, yaitu aplikasi yang kompleks memerlukan aturan tambahan untuk membantu pengaktifan aturan.
- 4) Penulisan kode awal, yaitu untuk menentukan efektifitas pengetahuan sistem dalam struktur aturan yang baik.
- 5) Pengujian sistem, yaitu dilakukan dengan beberapa aturan untuk menguji sejauh mana sistem berjalan dengan benar.
- 6) Perancangan antarmuka, dibuat bersamaan dengan pembuatan basis *knowledge*.
- 7) Pengembangan sistem, yaitu meliputi penambahan antar muka dan pengetahuan sesuai dengan prototipe sistem.
- 8) Evaluasi sistem, pada tahap ini dilakukan pengujian sistem dengan masalah yang sebenarnya. Jika sistem belum berjalan dengan baik maka akan dilakukan pengembangan kembali.



Gambar 2.3. Struktur Kerja Metode *Forward Chaining*
(Tutik dkk. 2009)

2.3. Jaringan Komputer

Jaringan komputer adalah jaringan telekomunikasi yang memungkinkan antar komputer untuk saling bertukar data. Tujuan dari jaringan komputer adalah agar dapat mencapai tujuannya, setiap bagian dari jaringan komputer dapat meminta dan memberikan layanan (*service*). Pihak yang meminta/ menerima layanan disebut klien (*client*) dan yang memberikan/ mengirim layanan disebut pelayan (*server*). Desain ini disebut dengan sistem *client-server*, dan digunakan pada hampir seluruh aplikasi jaringan komputer.

Dua buah komputer yang masing-masing memiliki sebuah kartu jaringan, kemudian dihubungkan melalui kabel maupun nirkabel sebagai medium transmisi data, dan terdapat perangkat lunak sistem operasi jaringan akan membentuk sebuah jaringan komputer yang sederhana. Apabila ingin membuat jaringan komputer yang lebih luas lagi jangkauannya, maka diperlukan peralatan tambahan seperti *Hub*, *Bridge*, *Switch*, *Router*, *Gateway* sebagai peralatan interkoneksinya (Dony Arius, 2014).

1. TCP/IP

Beberapa materi yang digunakan adalah *subnetting* atau pembagian kelas IP. Subnetting adalah suatu proses untuk memecah suatu jaringan IP jaringan ke Sub Jaringan yang lebih kecil atau juga dapat diartikan sebagai metode yang dilakukan untuk membagi blok setiap alamat IP address menjadi beberapa blok IP address. TCP/IP membagi IP menjadi lima kelas, yaitu:

- a. Kelas A8 bit pertama merupakan *bit network* sedangkan 24 bit terakhir merupakan *bit host*.
- b. Kelas B16 bit pertama merupakan *bit network* sedangkan 16 bit terakhir merupakan *bit host*.
- c. Kelas C24 bit pertama merupakan *bit network* sedangkan 8 bit terakhir merupakan *bit host*.
- d. Kelas D digunakan untuk *multicast address*, yakni sejumlah komputer yang memakai bersama suatu aplikasi. Penggunaan *multicast address* yang sedang berkembang saat ini adalah aplikasi *real-time video conference* yang melibatkan lebih dari dua *host* (*multipoint*), menggunakan *Multicast Backbone* (MBone).
- e. Kelas E (4 bit pertama adalah 1111 atau sisa dari seluruh kelas). Pemakaiannya dicadangkan untuk kegiatan eksperimental.

2.4. Alat Bantu Perancangan Sistem

Langkah yang dilakukan pada tahap merancang suatu sistem dan program adalah membuat usulan pemecahan masalah secara logika. Dalam merancang sebuah sistem juga dibutuhkan beberapa alat bantu perancangan sistem agar analisa dan hasil yang diinginkan dapat dicapai dengan maksimal.

2.4.1. Unified Modelling Language (UML)

Unified Modelling Language (UML) merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan

pemodelan. Jadi penggunaan UML tidak terbatas pada metode tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek (Rosa dan Shalahuddin, 2013).

2.4.2. Use Case Diagram

Activity diagram digunakan untuk memodelkan perilaku di dalam suatu bisnis. Activity diagram dapat dilihat sebagai sebuah sophisticated data flow diagram (DFD) yang digunakan pada analisis structural. Akan tetapi, berbeda dengan DFD, activity diagram mempunyai notasi untuk memodelkan aktivitas yang berlangsung secara paralel, bersamaan, dan juga proses pengambilan keputusan yang kompleks (Suryasari, 2014).

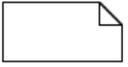
Tabel 2.3 Simbol – Simbol Use Case Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).

Tabel 2.3 Simbol – Simbol Use Case Diagram (lanjutan)

NO	GAMBAR	NAMA	KETERANGAN
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

Tabel 2.3 Simbol – Simbol Use Case Diagram (lanjutan)

NO	GAMBAR	NAMA	KETERANGAN
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Sumber : (Suryasari, 2014)

2.4.3. Class Diagram

Class diagram adalah visualisasi kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Diagram ini memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam model desain dalam *logical view* dari suatu sistem. Kelas memiliki 3 area utama yaitu nama, atribut, dan operasi. Nama berfungsi untuk member identitas pada sebuah kelas, atribut fungsinya adalah untuk menunjukkan karakteristik pada data yang dimiliki suatu objek di dalam kelas, sedangkan operasi fungsinya adalah memberikan sebuah fungsi ke sebuah objek. (Anwar, 2014)

Tabel 2.4. Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung padanya
3		<i>Extend</i>	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.

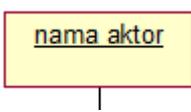
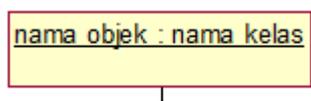
Sumber : (Aprianti, 2016)

2.4.4. *Sequence Diagram*

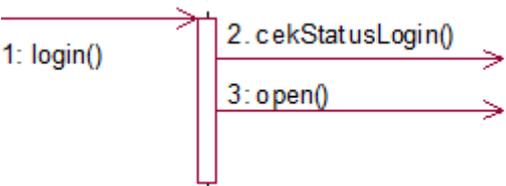
Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirim dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu (Rosa dan Shalahuddin, 2013).

Berikut ini adalah simbol-simbol yang ada pada *sequence diagram* (Rosa dan M. Salahuddin, 2013):

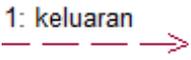
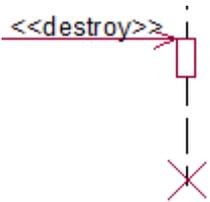
Tabel 2.5 Simbol-simbol sequence diagram

Simbol	Deskripsi
<p>Actor</p>  <p>nama aktor</p> <p>Atau</p> 	<p>Orang, proses, atau system lain yang berinteraksi dengan system informasi yang akan dibuat di luar system informasi yang akan dibuat sendiri, jadi walaupun symbol dari actor adalah gambar orang, tapi actor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor.</p>
<p>Garis hidup / lifeline</p> 	<p>Menyatakan kehidupan suatu objek</p>
<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan</p>

Tabel 2.5 Simbol-simbol sequence diagram (lanjutan)

Simbol	Deskripsi
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif yang berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>  <p>Maka cekStatusLogin() dan open() dilakukan di dalam metode login() Actor tidak memiliki waktu aktif</p>
<p>Pesan tipe create</p> <p><code><<create>></code></p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
<p>Pesan tipe call</p> <p>1 : nama_metode</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p>  <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi</p>
<p>Pesan tipe send</p> <p>1: masukan</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah yang mengarah pada objek yang dikirim</p>

Tabel 2.5 Simbol-simbol sequence diagram (lanjutan)

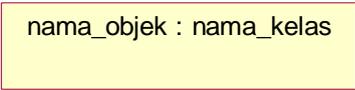
Simbol	Deskripsi
Pesan tipe return 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
Pesan type destroy 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy

2.4.5. *Communication Diagram*

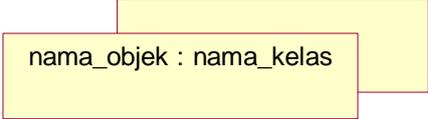
Communication diagram adalah diagram kolaborasi tetapi dibuat untuk setiap sekuen. Dan menggambarkan interaksi antar objek/bagian dalam bentuk urutan pengiriman pesan (Rosa dan Shalahuddin, 2013).

Berikut ini adalah simbol-simbol yang ada pada *Communication diagram* (Rosa dan M. Salahuddin, 2013):

Tabel 2.6 Simbol-simbol *communation diagram*

Symbol	Deskripsi
Objek 	objek yang melakukan interaksi pesan

Tabel 2.6 Simbol-simbol *commutation diagram* (lanjutan)

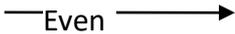
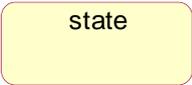
Symbol	Deskripsi
Link 	relasi antar objek yang menghubungkan objek satu dengan lainnya atau dengan dirinya sendiri 
Arah Pesan 	arah pesan yang terjadi, jika pada suatu link ada dua arah pesan berbeda maka arah juga digambarkan dua arah pada dua sisi link 

2.4.6. Statechart Diagram

Statechart diagram digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem atau objek. Jika diagram *sequence* digunakan untuk interaksi antar objek maka *statechart* diagram digunakan untuk interaksi di dalam sebuah objek (Rosa dan Shalahuddin, 2013).

Berikut ini adalah simbol yang digunakan dalam *statechart diagram* (Rosa dan M. Salahuddin, 2013):

Tabel 2.7 Simbol-simbol statechart diagram

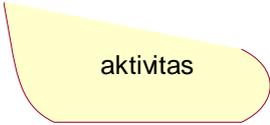
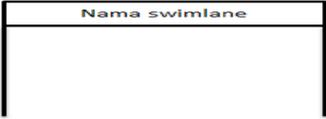
Simbol	Deskripsi
Start / status awal (initial state) 	<i>Start</i> atau <i>Initial State</i> adalah <i>state</i> atau keadaan awal pada saat sistem mulai hidup.
End / status Akhir (Final State) 	<i>End</i> atau <i>final state</i> adalah keadaan akhir dari daur hidup suatu sistem.
Event 	<i>Event</i> adalah kegiatan menyebabkan berubahnya status mesin.
State 	State atau status Keadaan sistem pada waktu tertentu. State dapat berubah jika pada event tertentu yang memicu perubahan tersebut.

2.4.7. Activity Diagram

Activity diagram menggambarkan *work flow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. yang perlu diperhatikan disini adalah bahwa diagram kreativitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (**Rosa A.S dan Shalahuddin, 2013**).

Berikut ini adalah simbol-simbol yang ada pada *activity diagram* (**Rosa A. S dan M. Salahuddin, 2013**):

Tabel 2.8 Simbol-simbol *activity diagram*

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

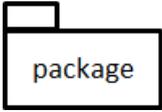
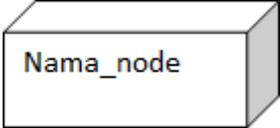
2.4.8. Deployment Diagram

Deployment/physical diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan hal-hal berikut: (Rosa dan Shalahuddin, 2013).

1. Sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node*, dan *hardware*.
2. Sistem *client/server*.
3. Sistem terdistribusi murni.
4. Rekayasa ulang aplikasi.

Berikut ini adalah simbol yang digunakan dalam *deployment diagram* (Rosa dan M. Salahuddin, 2013):

Tabel 2.9 Simbol-simbol deployment diagram

Simbol	Deskripsi
Package 	Package Merupakan sebuah bungkus dari satu atau lebih <i>node</i>
Node 	Biasanya mengacu pada perangkat keras, perangkat lunak yang tidak dibuat sendiri, jika dalam <i>node</i> disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen.

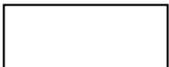
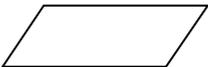
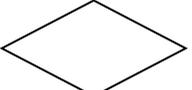
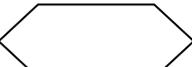
Tabel 2.9 Simbol-simbol deployment diagram (lanjutan)

Simbol	Deskripsi
Kebergantungan / dependency 	Kebergantungan antar node, arah panah mengarah pada <i>node</i> yang dipakai
Link 	Relasi antar <i>node</i>

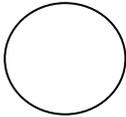
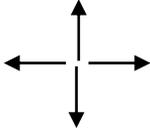
2.5. Bagan Alir (*Flowchart*)

Flowchart adalah skema atau bagan yang menggambarkan arus dan urutan suatu kegiatan program kerja secara keseluruhan dari system secara logika mulai dari awal hingga akhir. Simbol-simbol yang digunakan adalah sebagai berikut:

Tabel 2.10 Simbol-simbol *flowchart*

No	Simbol	Deskripsi
1		Terminal, untuk memulai atau mengakhiri suatu program
2		Proses suatu simbol yang menunjukkan setiap pengolahan yang dilakukan
3		Input-output untuk memasukan data ataupun menunjukkan hasil dari suatu proses
4		Decesion, suatu kondisi yang akan menghasilkan beberapa kemungkinan jawaban atau pilihan
5		Preparation, proses suatu simbol yang menyediakan tempat-tempat pengolahan dalam storage

Tabel 2.10 Simbol-simbol *flowchart* (lanjutan)

No	Simbol	Deskripsi
6		Conector, suatu prosedur akan masuk atau keluar melalui simbol ini dalam lembar yang sama
7		Off-page Conector, merupakan simbol masuk atau keluarnya suatu prosedur pada lembar kertas lainnya
8		Flow, arus dari pada prosedur yang dapat dilakukan atas kebawah dan bawah keatas, dari kiri kekanan ataupun dari kanan kekiri
9		Stored data, penyimpanan data secara sementara
10		Predifined process, untuk menyatakan sekumpulan langkah proses yang ditulis sebagai procedure

(Sumber :Rosa A.s, 2014)

2.6. Web

Awal perkembangan web dimulai pada bulan Maret 1989 saat tim berneer-lee yang bekerja di laboratorium fisika partikel Eropa atau yang dikenal dengan nama CERN (*conseil european pour la recherche nuclaire*) yang terletak di Geneva Swiss, mengajukan protokol (bahasa atau prosedur yang digunakan untuk menghubungkan antara komputer yang satu dengan lainnya) sistem distribusi

informasi internet yang digunakan untuk berbagai informasi di antara para fisikawan. Protokol inilah yang selanjutnya dikenal sebagai protokol *world wide web* dan dikembangkan oleh *world wide web consortium* (w3c). w3c adalah konsorsium dari sejumlah organisasi yang berkepentingan dalam pengembangan berbagai standar yang berkaitan dengan web.

Sumber daya yang ada di Internet jumlahnya sangat banyak, seperti *Chatting*, *E-mail*, *Milis*, dan sebagainya. Salah satu sumber daya internet yang perkembangannya sangat pesat adalah *www* (*world wide web*) atau sering disebut dengan istilah web saja. *Web* didistribusikan dengan menggunakan pendekatan *hypertext*. Dimana hanya dengan menggunakan suatu teks yang tidak terlalu banyak/ singkat bisa dijadikan acuan untuk membuka dokumen yang lain. Melalui pendekatan *hypertext* ini seorang *user* dapat memperoleh informasi yang diinginkan dengan cepat. Caranya bisa berpindah dari suatu dokumen ke dokumen yang lain. Dokumen-dokumen yang diperlukan informasinya tersebut dapat terletak dilokasi manapun, asalkan terletak pada jaringan internet.

2.7. PHP

PHP diciptakan pertama kali oleh Rasmus Lerdorf pada tahun 1994. Awalnya PHP Digunakan untuk mencatat jumlah serta untuk mengetahui siapa saja pengunjung pada homepage-nya. Rasmus Lerdorf adalah seorang pendukung open source. Oleh karena itu, ia mengeluarkan *Personal Home Page Tools* versi 1.0 secara gratis, kemudia menambahkan kemampuan PHP 1.0 dan meluncurkan PHP 2.0.

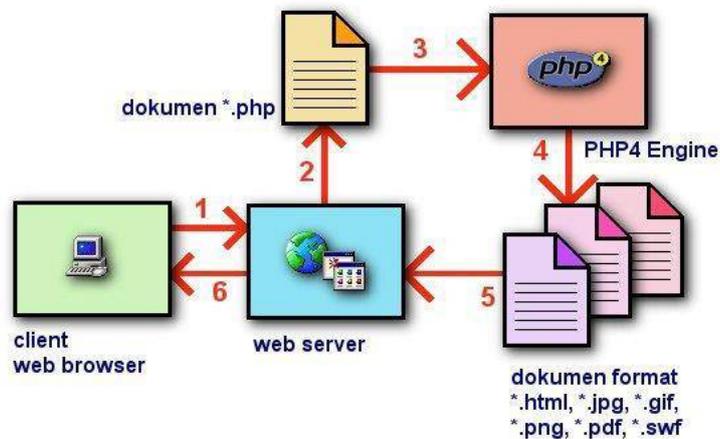
Pada tahun 1996, PHP telah banyak digunakan dalam website didunia. Sebuah kelompok pengembang software yang terdiri dari Rasmus, Zeew Suraski, Andi Gutman, Stig Bakten, Shane Caraveo, dan Jim Winstead bekerja sama untuk menyempurnakan PHP 2.0. Akhirnya, pada tahun 1998, PHP 3.0 diluncurkan. Penyempurnaan terus dilakukan sehingga pada tahun 2000 dikeluarkan PHP 4.0. Tidak berhenti sampai disitu, kemampuan PHP terus ditambah, dan sampai saat ini versi terbaru PHP yang telah dikeluarkan adalah PHP 5.0.x.

PHP memiliki banyak kelebihan yang tidak dimiliki oleh bahasa script sejenis. PHP difokuskan pada pembuatan *script server-side*, yang bisa melakukan apa saja yang dapat dilakukan oleh CGI, seperti mengumpulkan data dari form, menghasilkan isi halaman web dinamis, dan kemampuan mengirim serta menerima *cookies*, bahkan lebih dari pada kemampuan CGI.

PHP dapat digunakan pada semua sistem operasi, antara lain Linux, Unix (termasuk variannya HP-UX, Solaris, dan Open BSD), Microsoft Windows, Mac OS, RISC OS. PHP juga mendukung banyak Web Server, seperti Apache, Microsoft Internet Information Server (MIIS), Personal Web Server (PWS), Netcape and iPlanet servers, O'Reilly Website Pro Server, audium, Xitami, OmniHTTPd, dan masih banyak lagi lainnya, bahkan PHP dapat bekerja sebagai suatu CGI Processor. PHP tidak terbatas pada hasil keluaran HTML (*Hyper Text Markup Language*). PHP juga memiliki kemampuan untuk mengolah keluaran gambar, file PDF, dan movie flash. PHP juga dapat menghasilkan text seperti XHTML dan file XML lainnya.

Salah satu fitur yang dapat diandalkan oleh PHP yakni dukungannya terhadap banyak database seperti Adabas D, dBase, Direct MS-SQL, Empress, FrontBase, Hyperwave, IBM DB2, Informix, Ingres, Interbase, MSQL, MySQL, ODBC, Oracle, Ovrimos, PostgrSQL, Solid, Sybase, Unix DBM dan Velocis. Umumnya database MySQL digunakan untuk bekerja sama dengan PHP.

(Kusuma Ardhana, 2017)



2.8. Mysql

Mysql adalah sebuah perangkat lunak sistem manajemen basis data SQL (*database management system*) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. Beberapa kelebihan yang dimiliki oleh MySQL sebagai berikut :

1. Gratis

Sama dengan PHP, MySQL bersifat opensource, semua orang bebas menggunakannya tanpa harus membayar sepeser pun

2. Cross Platform

MySQL dapat digunakan under windows, ataupun under linux.

3. Lengkap dan Cepat

Pasangan yang cocok dengan PHP. Wajar jika banyak hosting saat ini mendukung adanya PHP dan MySQL karena kecepatan, gratis, dan dapat di jalankan di sistem operasi manapun.

2.9. Xampp

XAMPP adalah perangkat lunak gratis yang bebas digunakan. XAMPP berfungsi sebagai server yang berdiri sendiri (localhost), yang terdiri dari Apache HTTP Server, MYSQL database dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. XAMPP dikembangkan oleh perusahaan apache friends yang memiliki kelebihan bisa berperan sebagai server web apache untuk simulasi pengembang sebsite.

XAMPP banyak mendukung sistem operasi, merupakan kompilasi dari beberapa program. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU *General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis. Untuk mendapatkannya dapat mendownload langsung dari web resminya.

BAB III

METODE PENELITIAN

3.1. Metode Penelitian

Secara umum metode penelitian diartikan sebagai cara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu. Cara ilmiah berarti kegiatan penelitian itu didasarkan pada ciri-ciri keilmuan, yaitu rasional, empiris, dan sistematis. Rasional berarti kegiatan penelitian itu dilakukan dengan cara-cara yang masuk akal, sehingga terjangkau oleh penalaran manusia. Empiris berarti cara-cara yang dilakukan itu dapat diamati oleh indera manusia, sehingga orang lain dapat mengamati dan mengetahui cara-cara yang digunakan. Sistematis artinya, proses yang digunakan dalam penelitian itu menggunakan langkah-langkah tertentu yang bersifat logis (Sugiyono, 2013). Berdasarkan pendapat di atas dapat disimpulkan data yang diperoleh melalui penelitian itu adalah data rasional, empiris (teramati) dan sistematis yang mempunyai kriteria tertentu yaitu valid. Valid menunjukkan derajat ketepatan antara data yang sesungguhnya terjadi pada obyek dengan data yang dapat dikumpulkan oleh peneliti.

3.1.1 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini adalah metode deskriptif. Adapun teknik pengumpulan data dilakukan dengan cara sebagai berikut:

- 1) Observasi

Pengumpulan data dengan melakukan tanya jawab kepada Bapak

Romulus Simangunsong Pimpinan Seksi Operasional PT Bank Sumut Cabang Sidikalang yang membawahi bagian IT untuk mengetahui kerusakan jaringan komputer.

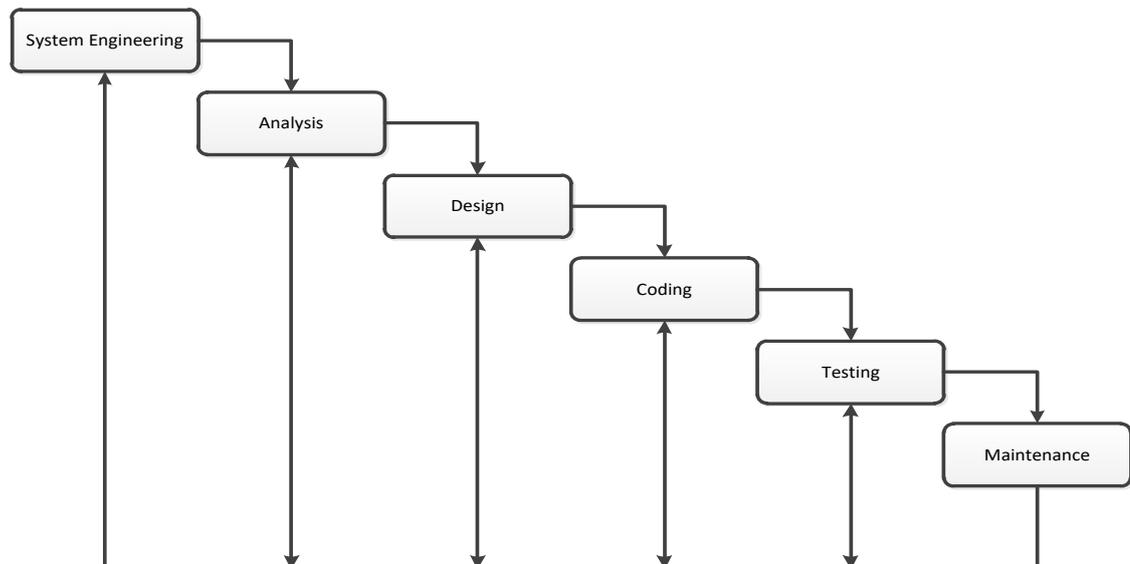
2) Studi Pustaka

Pengumpulan data dengan menggunakan atau mengumpulkan sumber-sumber tertulis, dengan cara membaca, mempelajari dan mencatat hal-hal penting yang berhubungan dengan masalah yang sedang dibahas guna memperoleh gambaran secara teoritis.

3.1.2 Metode Perancangan Sistem

Metodelogi yang digunakan untuk membangun sistem ini adalah Model Waterfall. Model ini terdiri dari beberapa tahapan, yaitu: *System Engineering, Analysis, Design, Coding, Testing dan Maintenance*.

Paradigma Waterfall dapat dilihat pada gambar berikut:



Gambar 3.1 Paradigma *Waterfall (Classic Life Cycle)*
(Sumber: Iqbal, 2017)

Penjelasan Metodologi *Waterfall*:

- 1) *System Engineering* adalah Permodelan ini diawali dengan mencari kebutuhan dari keseluruhan sistem yang akan diaplikasikan ke dalam bentuk *software*.
- 2) *Analisis* adalah tahap menganalisa hal-hal yang diperlukan dalam pelaksanaan proyek pembuatan *software*. Dalam hal ini dilakukan dengan menganalisa dokumen-dokumen yang digunakan dalam informasi data.
- 3) *Design* adalah tahap penterjemah dari keperluan-keperluan yang dianalisis dalam bentuk yang lebih mudah dimengerti oleh pemakai.
- 4) *Coding* adalah tahap penterjemah data/pemecahan masalah *software* yang telah dirancang dalam bahasa pemograman *Android C#* dan *SQLite Database*.
- 5) *Testing* adalah tahap pengujian terhadap program yang telah dibuat. Pengujian ini dimulai dengan membuat suatu uji kasus untuk setiap fungsi pada perangkat lunak untuk memastikan tidak ada kesalahan dan semua berjalan dengan baik dan input yang diberikan hasilnya sesuai.
- 6) *Implementation* adalah menguji *software* aplikasi yang telah dibuat dan dirancang dengan membuat aplikasi *software* menjadi sebuah file, dan diimplementasikan sesuai pembahasan tersebut.
- 7) *Maintenance* adalah perangkat lunak yang telah dibuat dapat mengalami perubahan. Pemeliharaan dapat dilakukan jika ada

permintaan tambahan fungsi sesuai dengan keinginan pemakai ataupun adanya pertumbuhan dan perkembangan baik perangkat lunak maupun perangkat keras.

3.2. Analisa Sistem

Analisis sistem dilakukan bertujuan untuk mengetahui masalah yang terjadi pada sistem pakar, sebagai dasar melakukan pengembangan sistem dari data-data yang telah diperoleh dari teori-teori yang telah dipelajari. Analisis sistem dilakukan agar pengembangan dan perancangan sistem yang baru dapat dilakukan dengan baik dan sesuai dengan kebutuhan *user* tersebut.

Dalam melakukan analisis terhadap sistem yang sedang berjalan, dalam sistem pakar ini, penulis menganalisa apakah aplikasi berjalan secara efektif pada saat ini. Banyaknya permasalahan dalam pengambilan keputusan yang dilakukan secara manual menurut beberapa pertimbangan yang telah ditentukan, serta dengan mengenal aplikasi sistem pakar ini diharapkan dapat menghasilkan calon-calon anggota baru yang berjiwa pemimpin, cerdas, serta dapat bertanggung jawab.

3.2.1. Kebutuhan Sistem

Dalam proses perancangan aplikasi sistem sistem pakar ini di butuhkan perangkat-perangkat dasar maupun perangkat-perangkat pendukung dalam bentuk software dan hardware. Adapun perangkat-perangkat tersebut adalah :

1. Perangkat Keras (*Hardware*)

Satu unit Personal Komputer, dengan spesifikasi minimum sebagai berikut:

- a. Processor : Intel (R) Celeron (R) CPU n2840 @2.16GHz (2 CPUs),
2.2GHz
- b. Memory : 2048MB RAM
- c. Monitor : *LCD 14 Inch*

2. Perangkat Lunak (*Software*)

- a. Sistem Operasi Windows 7.
- b. Bahasa pemograman PHP
- c. XAMPP adalah *software* untuk lokal web server.
- a. MySQL adalah perangkat lunak untuk menyimpan data.

3.3. Analisa Data

Permasalahan pada jaringan biasa disebabkan oleh beberapa faktor. Faktor tersebut adalah oleh alat/komponen jaringan itu sendiri yang dikarenakan oleh tidak berfungsi atau mati. Untuk mengetahui adanya permasalahan dalam jaringan, kita dapat melihat indikator yang muncul pada komponen jaringan. Indikator-indikator tersebut memberikan isyarat jika terjadi kerusakan atau tidak berfungsinya komponen. Indikasi kerusakan pada masing masing komponen dapat diuraikan sebagai berikut:

1. Server : Server adalah komputer yang berisikan daftar user yang diperbolehkan masuk ke server tersebut. Jadi apabila komputer server mengalami kerusakan atau gangguan secara otomatis seluruh jaringan tidak berfungsi karena server merupakan pintu masuk dan sebagai pusat jaringan tersebut.

2. Workstation : Workstation adalah komputer yang memanfaatkan jaringan untuk menghubungkan komputer tersebut dengan komputer lain. Apabila terjadi kerusakan pada komputer workstation berarti komputer yang tersebut tidak dapat masuk dalam jaringan.
3. Hub/switch : Hub/switch merupakan pembagi signal data bagi kartu jaringan (Network Card). Jika Hub mengalami kerusakan berarti seluruh jaringan juga tidak dapat berfungsi untuk berkomunikasi. Apabila terjadi kerusakan pada Hub dapat dilihat pada lampu indikator power dan lampu indikator untuk masing-masing workstation.
4. Network Interface Card (Kartu jaringan) : Apabila terjadi gangguan atau kerusakan pada kartu jaringan berakibat pada komputer tersebut tidak dapat masuk dalam sistem jaringan. Indikator yang dapat dilihat dalam kerusakan kartu jaringan adalah lampu indikator NIC.
5. Kabel dan konektor : Kabel dan konektor merupakan media penghubung antar komputer.

Permasalahan pada jaringan dapat digolongkan menjadi 2 yaitu kerusakan pada hardware dan software diantaranya yaitu:

1. Kerusakan pada hardware :
 - a. Jika terjadi masalah indikator dapat dilihat melalui lampu NIC, sedangkan untuk melihat secara software dapat dilihat melalui Device Manager
 - b. Kabel dan konektor

1. Kabel *Coaxial*. Masalah yang muncul adalah kabel terbuka (open), konektor longgar, kabel short, resistor pada terminating konektor, short antara kabel dan plug konektor, dan longgar pada *male* konektor.
 2. Kabel UTP. Masalah yang muncul adalah konektor longgar, kabel short, dan kabel terbuka, serta masalah setting susunan kabel.
2. Kerusakan pada software :
- a. Kesalahan setting konfigurasi jaringan
 - b. Kesalahan protokol yang digunakan
 - c. Kesalahan pengalamatan IP
 - d. Kesalahan identifikasi *client* dan *server*
 - e. Kesalahan *networkservice*
 - f. Kesalahan *securitysystem*
 - g. Kerusakan file program

Tabel 3.1 Indikator masalah pada jaringan

Kode	Indikator Masalah	Permasalahan			
		Konektor M1	Hardware M2	Software M3	Solusi S4
C1	Apakah dalam pemasangan konektor ada yang longgar...?	√			
C2	Apakah terlihat kabel koneksi yang rusak atau dalam kondisi tidak baik...?	√			
C3	Apakah ada kerusakan – kerusakan file program...?		√		
C4	Apakah ada kesalahan network service komputer anda...?		√		
C5	Apakah lampu indikator router tidak menyala...?			√	
C6	Apakah pemasangan indikator router/hub kurang pas...?			√	

Berikut ini menunjukkan bagaimana cara kerja metode inferensi runut maju dalam kasus permasalahan yang sering terjadi pada sebuah jaringan komputer dapat dilihat dibawah ini :

M1: IF C1 and C2, THEN S1

M2 : IF C83and C4 and, THEN S2

M3 :IF C65and C6THEN S3

Berikut ini adalah tabel *rule-based knowledge* atau pembentukan aturan kerusakan jaringan dengan memakai inferensi *forward chaining*.

Tabel 3.2 Basis Aturan Kerusakan Jaringan

Aturan	Diagnosa		Keterangan
1	IF	Indikator Client menyala	Y
		Address Server Sudah Betul	Y
		Ip Address Client Statis	Y
		Penulisan Account Sudah Benar	Y
		Penulisan Password Sudah Benar	Y
Aturan	Diagnosa		Keterangan
2	IF	Indikator Client menyala	T
		Address Server Sudah Betul	T
		Ip Address Client Statis	T
		Penulisan Account Sudah Benar	T
		Penulisan Password Sudah Benar	T
		Client Rusak	

Tabel 3.2 Basis Aturan Kerusakan Jaringan (lanjutan)

Aturan	Diagnosa		Keterangan
3	IF	Indikator Client menyala	T
		Address Server Sudah Betul	Y
		Ip Address Client Statis	Y
		Penulisan Account Sudah Benar	Y
		Penulisan Password Sudah Benar	Y
Aturan	Diagnosa		Keterangan
4	IF	Indikator Client menyala	Y
		Address Server Sudah Betul	T
		Ip Address Client Statis	Y
		Penulisan Account Sudah Benar	Y
		Penulisan Password Sudah Benar	Y
		Kesalahan Penulisan IP	
Aturan	Diagnosa		Keterangan
5	IF	Indikator Client menyala	Y
		Address Server Sudah Betul	Y
		Ip Address Client Statis	T
		Penulisan Account Sudah Benar	Y
		Penulisan Password Sudah Benar	Y
		Terdapan IP Address Yang Sama	

Tabel 3.2 Basis Aturan Kerusakan Jaringan (lanjutan)

Aturan	Diagnosa		Keterangan
6	IF	Indikator Client menyala	Y
		Address Server Sudah Betul	Y
		Ip Address Client Statis	Y
		Penulisan Account Sudah Benar	T
		Penulisan Password Sudah Benar	Y
		Kesalahan Penulisan Account	
Aturan	Diagnosa	Keterangan	
7	IF	Indikator Client menyala	Y
		Address Server Sudah Betul	Y
		Ip Address Client Statis	Y
		Penulisan Account Sudah Benar	Y
		Penulisan Password Sudah Benar	T
		kesalahan Penulisan Password	
Aturan	Diagnosa		Keterangan
8	IF	Indikator Client menyala	Y
		Address Server Sudah Betul	T
		Ip Address Client Statis	T
		Penulisan Account Sudah Benar	T
		Penulisan Password Sudah Benar	T

Tabel 3.2 Basis Aturan Kerusakan Jaringan (lanjutan)

Aturan	Diagnosa		Keterangan
9	IF	Indikator Client menyala	Y
		Address Server Sudah Betul	Y
		Ip Address Client Statis	T
		Penulisan Account Sudah Benar	T
		Penulisan Password Sudah Benar	T
		Human Error	

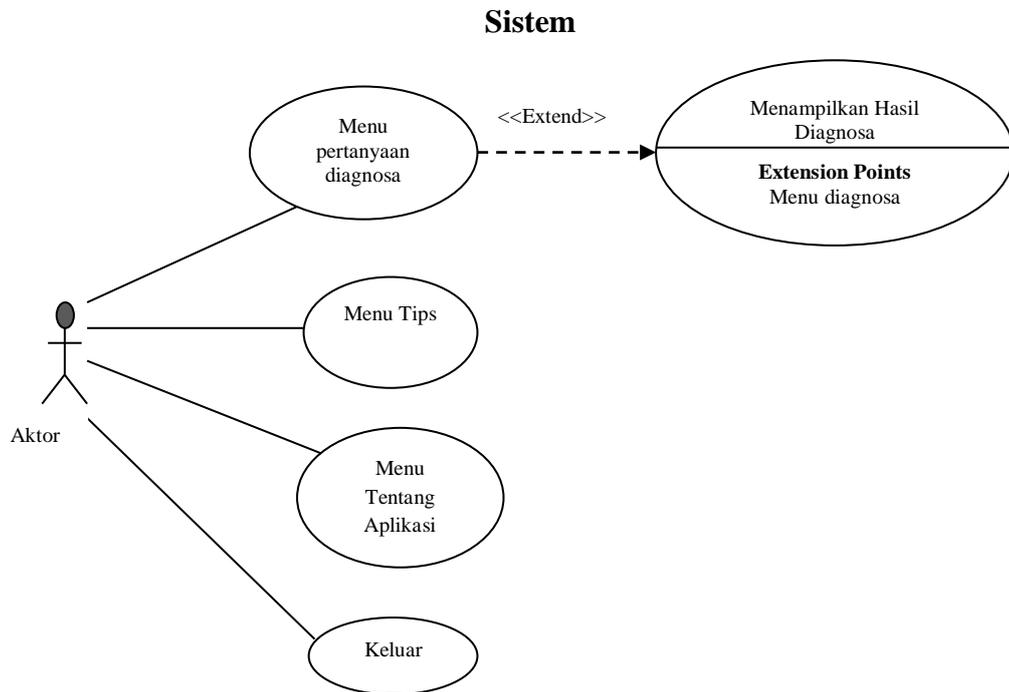
(Sumber: Sudirman Jonas,2019)

3.4. Perancangan UML

Perancangan UML (*Unified Modelling Language*) adalah untuk menentukan cara kerja program sistem pakar. Adapun perancangan UML (*Unified Modelling Language*) untuk perancangan sistem pakar diagnosa kerusakan pada jaringan komputer adalah sebagai berikut :

3.4.1. Use Case Diagram

Use case menggambarkan bagaimana seseorang akan menggunakan atau memanfaatkan sistem, sedangkan aktor adalah seseorang atau sesuatu yang berinteraksi dengan sistem. *Use case* diagram menggambarkan bagaimana proses-proses yang dilakukan oleh aktor terhadap sistem.

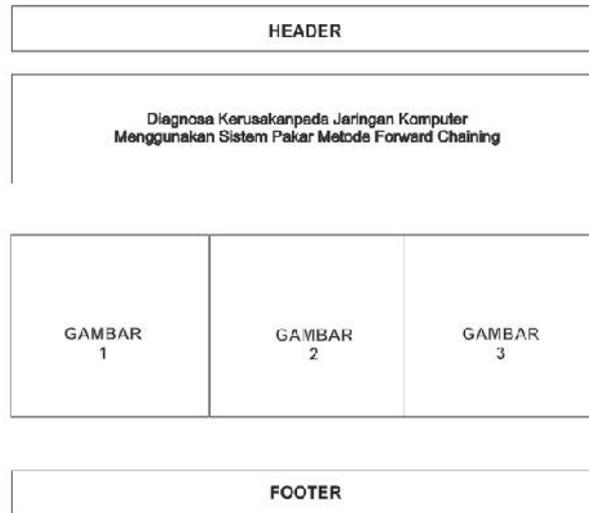


Gambar 3.2 Use Case Diagram

3.5. Perancangan Interface Program

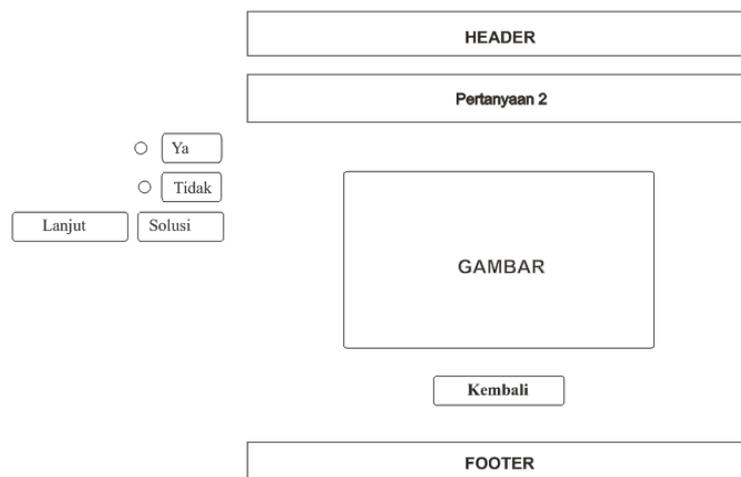
Perancangan antar muka adalah bentuk rancangan tampilan sementara dari pembuatan aplikasi sistem pakar. Perancangan ini dibuat untuk memberikan penjelasan tentang tampilan yang dihadapkan pada aktor di saat menggunakan aplikasi sistem pengambilan keputusan pakar ini. Sehingga dapat mempermudah dalam mengimplementasikan aplikasi serta akan memudahkan pembangunan aplikasi yang memenuhi prinsip perancangan anta muka yang baik. Perancangan aplikasi sistem pengambilan keputusan yang akan dibangun adalah sebagai berikut :

Perancangan tampilan awal aplikasi adalah perancangan menu yang akan ditampilkan kepada *user* di saat pertama membuka aplikasi.



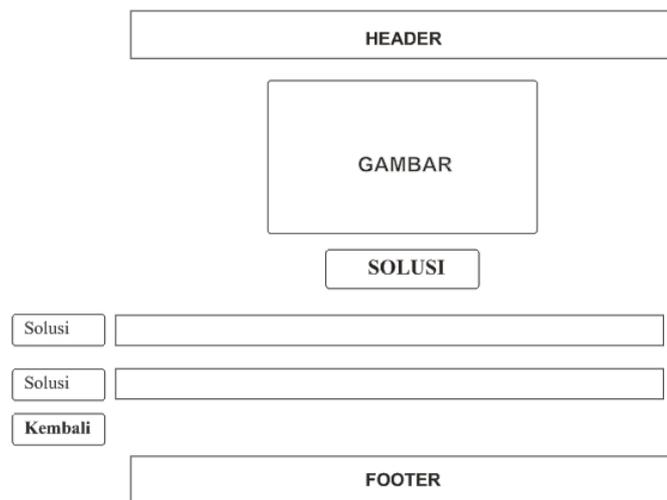
Gambar 3.3 Tampilan Awal Aplikasi

Halaman beranda atau utama merupakan halaman depan dari Aplikasi sistem pakar kerusakan pada jaringan komputer. Dan tampilannya dapat dilihat pada gambar berikut :



Gambar 3.4 Tampilan Halaman Utama

Halaman cek masalah merupakan halaman yang menampilkan indikator kerusakan jaringan ketika terdapat gejala yang di temukan user dapat meceklis pada cek box dan pada bagian bawah terdapat hasil analisa serta cara penanggulangan nya. Dan tampilannya dalam dilihat pada gambar berikut :



Gambar 3.5 Tampilan Menu Cek Masalah

3.6. Perancangan database

Rancangan *database* adalah gambaran tentang file file dalam table sehingga dapat dilihat bentuk bentuk file tersebut, tipe data nya serta ukuran data tersebut. Adapun struktur table yang ada pada *database* MySQL dari sistem pakar kerusakan jaringan yang dibuat dapat digambarkan sebagai berikut.

a. Struktur Tabel

Tabel merupakan tempat penyimpanan informasi dari sebuah aliran data dalam sebuah sistem. Berikut merupakan struktur dari beberapa tabel sistem yang akan dibangun.

Tabel 3.4. Tabel User

Nama	Type	Size	Keterangan
Username	Varchart	10	
Password	Varchar	50	
Email	Varchar	30	

Tabel 3.5. Tabel Gejala

Nama	Type	Size	Keterangan
Kode_kerusakan	Int	10	Primary key
Nama_kerusakan	Varchar	60	
Solusi	Int	15	

Primary key : id_gejala

BAB IV

HASIL DAN PEMBAHASAN

4.1. Hasil

Implementasi sistem merupakan bagian dari siklus pengembangan sistem itu sendiri. Untuk melakukan sebuah implementasi maka diperlukan aplikasi perancangan *interface* dan penulisan *coding* sesuai dengan sistem yang dirancang ataupun yang telah dianalisa.

Perancangan sistem disini menggunakan model UML (*Unified Modelling Language*) yang digunakan untuk menggambarkan cara kerja sistem pengambilan keputusan. Setelah perancangan sistem selesai maka di lakukan implementasi kedalam program yakni dengan menggunakan bahasa pemrograman PHP dan *database* MySQL.

Implementasi sistem dilakukan setelah tahapan perancangan sistem dan pengujian dilakukan. Implementasi sistem dapat dilakukan setelah sistem yang dibuat dapat berjalan sebagaimana mestinya. Perancangan *interface* dilakukan untuk interaksi *user* dengan sistem yang telah dibuat.

Adapun kebutuhan *hardware* ataupun *software* dalam pengujian sistem ini adalah sebagai berikut:

4.1.1. Kebutuhan *Hardware*

Dalam implementasi ini menggunakan bantuan dari perangkat keras (*hardware*), dimana perangkat keras yang digunakan adalah sebagai berikut:

1. Processor : Intel (R) Core (TM) i3-2310M CPU 2.10GHz
2. Memory : 2048MB RAM
3. Monitor : Dell 21 inc

4.1.2. Kebutuhan *Software*

Dalam implementasi sistem ini juga menggunakan bantuan dari berbagai perangkat lunak, diantara perangkat lunak yang digunakan adalah sebagai berikut:

- a. Sistem operasi windows 7.
- b. Bahasa pemograman PHP dengan menggunakan editor *notepad++* sebagai media pembuatan aplikasi.
- c. XAMPP adalah *software* untuk lokal web server.
- d. MySQL adalah perangkat lunak untuk menyimpan data.

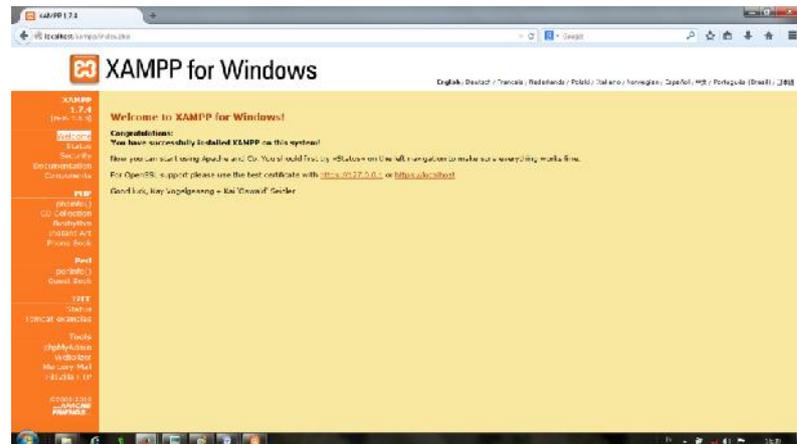
4.2. Implementasi Program

Implementasi program merupakan realisasi dari hasil perancangan yang telah dibuat dengan UML. Perancangan sistem yang sudah dibuat dikonversikan ke dalam bentuk program aplikasi. Untuk mengimplementasikan program sistem pengambilan keputusan kerusakan jaringan, dibutuhkanlah aplikasi pendukung yang harus diinstal terlebih dahulu dan dikonfigurasi dengan benar agar program yang dibuat dapat berjalan dan berfungsi sebagaimana mestinya. Berikut ini aplikasi pendukung.

4.2.1. Proses Instalasi XAMPP 1.7.4

Langkah awal yang harus dilakukan adalah men-*download* file XAMPP .
Kemudian lakukan instalasi. Langkah-langkah instalasinya adalah sebagai berikut :

1. Jalankan *file* xampp-win32-1.7.4-4-VC6-installer.exe, sebagai contoh penulis menggunakan xampp versi 1.8.2.
2. Ketika proses instalasi dimulai klik next.
3. Selanjutnya pilih lokasi penginstalan XAMPP, kemudian klik next.
4. Anda bisa memilih opsi sesuai dengan kebutuhan. Klik install untuk memulai proses instalasi.
5. Tunggu beberapa saat sampai proses instalasi XAMPP selsesai.
6. Instalasi selesai. Klik finish.
7. Sampai tahan ini berarti kita sudah menginstal XAMPP. Selanjutnya kita akan menggunakan XAMPP sebagai server lokal.
8. Jalankan XAMPP Control Panel yang ada di desktop. Atau melalui start menu -> all program -> apachefriends -> XAMPP Control Panel. Kemudian klik tombol Start pada modul Apache dan MySQL.
9. Setelah berhasil menjalankan apache dan MySQL. Selanjutnya buka web browser, lalu ketikkan <http://localhost> pada address bar. Jika berhasil maka akan muncul halaman XAMPP seperti dibawah ini, berarti XAMPP sudah terinstal dengan benar.

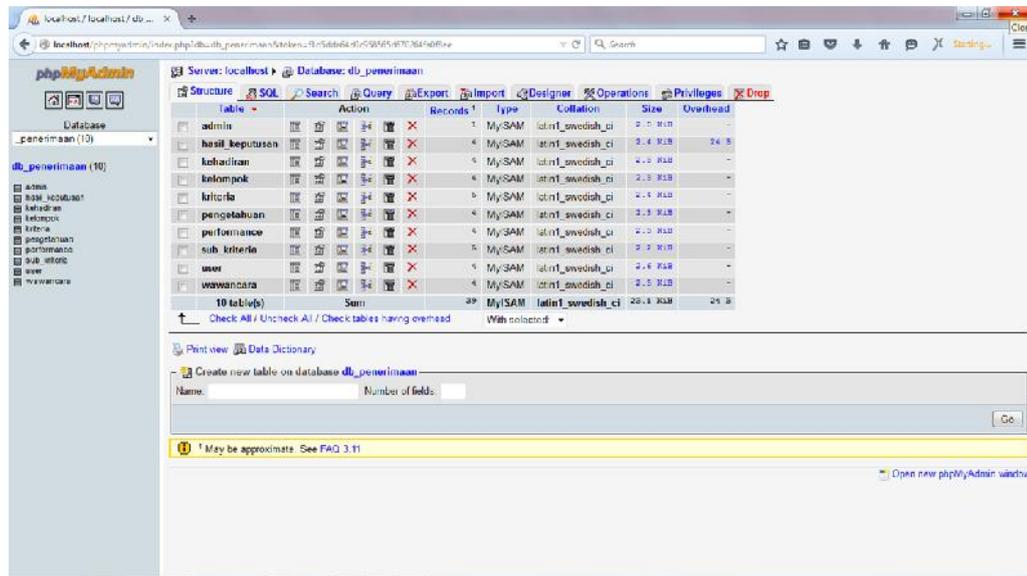


Gambar 4.1 Tampilan XAMPP

10. Selanjutnya kita bisa menggunakan XAMPP untuk menjadikan website yang telah dibuat dan mengelola database MySQL melalui menu phpmyadmin.

4.2.2. Tampilan *Database* Sistem Pakar Kerusakan Jaringan

Berikut ini merupakan gambar dari *database* sistem pakar yang diakses oleh *admin* melalui PHP Myadmin, *database* db_penerimaan yang dibuat oleh *admin* untuk menampung data-data yang berhubungan segala data yang diperlukan.



Gambar 4.2 Tampilan *Database db_penerimaan*

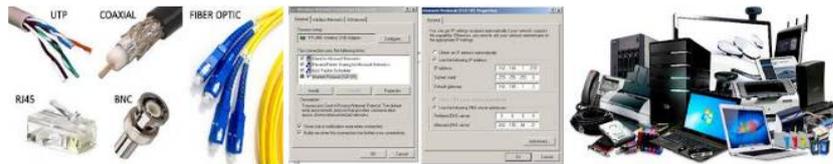
4.3. Pembahasan

Pengujian dan implementasi sistem bertujuan untuk melihat apakah sistem yang dirancang sudah sesuai dengan apa yang diinginkan atau belum, setelah dilakukannya pengujian dan implementasi, kualitas sebuah sistem akan terlihat. Berikut ini adalah implementasi dari perancangan sistem pengambilan sistem pakar *forward chaining*.

1. Halaman Awal Website.

Halaman awal website merupakan halaman yang akan ditampilkan ketika pertama kali membuka aplikasi sistem pakar kerusakan jaringan dengan metode *forward chaining*.

Diagnosa Kerusakan pada Jaringan Komputer Menggunakan Sistem Pakar



@togar

Gambar 4.3 Tampilan Awal

2. Halaman Cek Masalah

Halaman cek masalah merupakan halaman yang menampilkan gambar hardware dan *software* yang mengalami kerusakan. Setelah memilih gambar hardware dan *software* yang akan kerusakan akan muncul pertanyaan dan user dapat menceklis box sesuai indikasi yang muncul. Tampilannya dapat dilihat pada gambar berikut :



Gambar 4.4 Tampilan Cek Masalah

3. Halaman Solusi

Halaman solusi merupakan halaman yang menampilkan hasil dari cek masalah dan pada bagian bawah terdapat hasil analisa serta cara penanggulngannya. Dan tampilannya dalam dilihat pada gambar berikut :



Solusi :

solusi 1. untuk permasalahan yang sering terjadi pada switch di atas adalah anda harus sering-sering mereset switch, pastikan suhu ruangan di mana switch diletakkan harus cukup agar switch tidak panas dan hang
solusi 2. sesuaikan beban koneksi dan kualitas yang dipikul oleh switch sehingga tidak keberatan dan tidak melampaui kemampuannya

Go Back

Gambar 4.5 Solusi

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil dari analisa, perancangan aplikasi sistem pakar deteksi kerusakan jaringan dengan menggunakan metode *forward chaining*, maka dapat diambil kesimpulan, yaitu :

1. Sistem yang dibuat dengan menggunakan metode *forward chaining* mampu mendeteksi kerusakan jaringan yang dialami oleh pengguna.
2. Sistem yang dibangun dapat membantu *user* untuk mendapatkan solusi dari mendeteksi kerusakan jaringan serta jenis perbaikan dari jaringan.
3. Sistem ini juga bisa memberikan informasi mengenai jaringan, sehingga dapat segera dilakukan pencegahan sejak awal.
4. Aplikasi ini dapat mempermudah *user* untuk mendeteksi kerusakan jaringan tanpa harus datang langsung ke ahli jaringan terlebih dahulu.

5.2. Saran

Adapun saran dari penulis untuk pengembangan sistem aplikasi kedepannya adalah :

1. Perlu dikembangkan lebih lanjut dengan menambahkan gejala yang sesuai dengan mendeteksi kerusakan jaringan, sehingga sistem pakar ini dapat mendeteksi kerusakan jaringan lebih banyak.
2. Sistem ini banyak kekurangan dan keterbatasan dalam memperoleh data *user* sehingga diharapkan kepada para pembaca dapat mengembangkan sistem ini lebih baik lagi.

DAFTAR PUSTAKA

- Andrian, Yudhi, and Purwa Hasan Putra. "Analisis Penambahan Momentum Pada Proses Prediksi Curah Hujan Kota Medan Menggunakan Metode Backpropagation Neural Network." *Seminar Nasional Informatika (SNIF)*. Vol. 1. No. 1. 2017.
- Azmi, Fadhillah, and Winda Erika. "Analisis keamanan data pada block cipher algoritma Kriptografi RSA." *CESS (Journal of Computer Engineering, System and Science)* 2.1: 27-29.
- Barus, S., Sitorus, V. M., Napitupulu, D., Mesran, M., & Supiyandi, S. (2018). Sistem Pendukung Keputusan Pengangkatan Guru Tetap Menerapkan Metode Weight Aggregated Sum Product Assesment (WASPAS). *Jurnal Media Informatika Budidarma*, 2(2).
- Batubara, S., Wahyuni, S., & Hariyanto, E. (2018, September). Penerapan Metode Certainty Factor Pada Sistem Pakar Diagnosa Penyakit Dalam. In *Seminar Nasional Royal (SENAR)* (Vol. 1, No. 1, pp. 81-86).
- Dhany, H. W., Izhari, F., Fahmi, H., Tulus, M., & Sutarman, M. (2017, October). Encryption and decryption using password based encryption, MD5, and DES. In *International Conference on Public Policy, Social Computing and Development 2017 (ICOPOSDev 2017)* (pp. 278-283). Atlantis Press.
- Erika, Winda, Heni Rachmawati, and Ibnu Surya. "Enkripsi Teks Surat Elektronik (E-Mail) Berbasis Algoritma Rivest Shamir Adleman (RSA)." *Jurnal Aksara Komputer Terapan* 1.2 (2012).
- Fachri, Barany. Perancangan Sistem Informasi Iklan Produk Halal Mui Berbasis Mobile Web Menggunakan Multimedia Interaktif. *Jurasik (Jurnal Riset Sistem Informasi dan Teknik Informatika)*, 2018, 3: 98-102.
- Fuad, R. N., & Winata, H. N. (2017). Aplikasi keamanan file audio wav (waveform) dengan terapan algoritma RSA. *InfoTekJar: Jurnal Nasional Informatika dan Teknologi Jaringan*, 1(2), 113-119.
- Ginting, G., Fadlina, M., Siahaan, A. P. U., & Rahim, R. (2017). Technical approach of TOPSIS in decision making. *Int. J. Recent Trends Eng. Res*, 3(8), 58-64.
- Gideon Abram Filando Suwarso, Gregorius Satia Budhi, Lily Puspa Dewi. *Sistem Pakar untuk Penyakit Anak Menggunakan Metode Forward Chaining*.7. <https://ejournal.unsrat.ac.id>
- Hindayati Mustafidah, Hendrik Prawijaya, Dwi Aryanto.(2011). *Sistem Pakar untuk Mendiagnosa Kerusakan Komputer*.6.<https://jurnalnasional.umpc.ac.id>

- Henri Septanto. (2014). *Permasalahan, penanganan masalah dan perawatan Jaringan komputer client-server pada Kejuruan operator komputer Di PPKD Jakarta Timur*.14. <https://ejournalbinainsani.ac.id>
- Irvan Maulana Yusup, Mochammad, Tedy Putra, Muhammad Andika Usrahmawan, Nurrochmat Saptoaji, Putri Dewi Wulandari, Rina Apriani. (2018).
- Januar Lestari. (2016). *Analisis sistem deteksi kerusakan komputer Dengan menggunakan metode forward chaining*.9.<https://jurnal.acba.ac.id>
- Peti Savitri.(2018).*Implementasi Metode Forward Chaining Dalam Sistem Pendeteksi Kerusakan Hardware Pada Komputer Dan Laptop Berbasis Android*.10. <https://jurnalnasional.umk.ac.id>
- Rancang Bangun Sistem Pakar Diagnosa Kerusakan Pada Komputer Berbasis Desktop Dengan Metode Forward Chaining*.13.12. <https://zenodo.org/record/1312/files>
- Saiful Rizal, Rini Agustina.*Sistem Pakar Diagnosa Kerusakan Komputer Dengan Metode Forward Chaining Dan Certainty Factor*.6.media.neliti.com.
- Sitorus, Z., & Siahaan, A. P. U. (2016). Heuristic Programming in Scheduling Problem Using A Algorithm. IOSR J. Comput. Eng, 18(5), 71-79.
- Sumartono, I., Siahaan, A. P. U., & Mayasari, N. (2016). An overview of the RC4 algorithm. IOSR J. Comput. Eng, 18(6), 67-73.
- Supiyandi, S., Hermansyah, H., & Sembiring, K. A. (2020). Implementasi dan Penggunaan Algoritma Base64 dalam Pengamanan File Video. Jurnal Media Informatika Budidarma, 4(2), 340-346.
- Syahputra, Rizki, and Hafni Hafni. "Analisis kinerja jaringan switching clos tanpa buffer." journal of science and social research 1.2 (2018): 109-115.
- Tasril, V. (2018). Sistem Pendukung Keputusan Pemilihan Penerimaan Beasiswa Berprestasi Menggunakan Metode Elimination Et Choix Traduisant La Realite. INTECOMS: Journal of Information Technology and Computer Science, 1(1), 100-109.
- Wahyuni, Sri. "Implementasi Rapidminer Dalam Menganalisa Data Mahasiswa Drop Out." Jurnal Abdi Ilmu 10.2 (2018): 1899-1902.
- Windah Supartini, Hindarto.(2016).*Sistem Pakar Berbasis Web Dengan Metode Forward Chaining Dalam Mendiagnosis Dini Penyakit Tuberkulosis di JawaTimur*.8. <https://kinetik.umm.ac.id>
- Yasmiyati.*Sistem Pakar Diagnosis Penyakit Pada Perokok Dengan Metode Forward Chaining Berbasis Web*.21.