

BAB II

LANDASAN TEORI

2.1. Landasan Teori

Supaya penelitian ini mempunyai dasar yang kuat dibutuhkan landasan teori. Landasan teori ini suatu konsep dengan ungkapan yang tersusun dengan rapi dan terstruktur, teori tentang deskripsi terhadap *variable* yang diteliti melalui pendefenisian yang jelas lengkap dan bermacam *referensi*. Dalam penelitian ini akan menguraikan beberapa teori dasar kecerdasan buatan.

3.1.1. Kecerdasan Buatan (Artificial Intelligence)

Artificial inteligen atau di singkat dengan AI yang artinya kecerdasan buatan. Menurut beberapa tokoh mengartikan kecerdasan buatan adalah kecerdasan yang mengarah kepada mesin yang mampu berfikir dan melakukan tindakan layaknya seperti manusia yang dibuat ke dalam sebuah sistem yang mampu mengambil atau menyelesaikan masalah. Kemajuan teknologi AI yang sangat cepat hingga pada saat ini sudah menjamah hampir di setiap kehidupan manusia. Manusia dapat menjadi pintar dalam melakukan segala permasalahan di dunia ini dikarenakan manusia memiliki pengetahuan dan di bidang pengalaman (Azmi and Yasin 2017).

Kecerdasan buatan bisa di katakan bidang ilmu komputer yang terpenting di zaman sekarang ini dan masa depan untuk melaksakan sistem komputer yang sangat cerdas. lingkungan ini sudah berkembang dengan sangat luas pada usia 20 tahun

terakhir seiringnya dengan keperluan perangkat cerdas pada industri dan rumah tangga. Dibawah ini ada berbagai pendapat tentang kecerdasan buatan yaitu:

1. Alan Turing, ahli matematika berkebangsaan inggris yang dijuluki Bapak Komputer. Modern dan membongkar sandi Nazi dalam era Perang Dunia ke II 1950, menetapkan defenisi *artificial inteligent* jika mesin dapat membuat seseorang percaya bahwa dirinya mampu berkomunikasi dengan orang lain, maka dapat disebut bahwa mesin tersebut sangat cerdas atau pun layaknya seperti manusia.
2. Jhon Mc.Carthy dari Stanford mendefenisikan yaitu sistem komputer yang dapat mempresentasikan dan memahami maksud dan tujuan seorang pakar yang digunakan dalam menyelesaikan masalah.
3. Right and Knight mengartikan bahwa kecerdasan buatan hanya sebuah sandi tentang bagaimana menjadikan komputer dapat melakukan hal-hal pada saat dapat dilakukan lebih baik oleh manusia.

Perkembangan teknologi, kecerdasan buatan sudah merambat dan dapat di fungsikan sebagai alat kesehatan maupun di bidang pendidikan, dalam kecerdasan buatan dapat disebutkan yaitu:

1. Sistem pakar (*Expert System*)

Sistem pakar dapat disimpulkan sistem yang berbasis teknologi komputer yang dipakai untuk membereskan suatu masalah, dilakukan sama seperti yang dipikirkan

oleh seorang pakar. Dengan demikian komputer dapat menggabungkan suatu pangkalan pengetahuan base dengan sistem inferensi untuk menirukan keahlian yang dimiliki para ahli.

2. Pengenalan ucapan (*Speech Recoignitiong*)

Pengenalan ucapan adalah sistem komputer yang di kembangkan supaya sebuah komputer dapat memahami input yang berupa suara.

3. Pengolahan Bahasa Alami

Melalui pengolahan bahasa alami (*natural languae processing*) yaitu penyusunan salah satu program yang memilikii keahlian yang mampu mengartikan bahasa manusia.

4. Robotika dan sistem sensor (*Robotics and sensory system*)

Robotika dan sistem sensor ini dapat diartikan sebagai pengetahuan teknologi yang dapat mendefenisikanm dalam sebuah gambar atau tujuan melalui komputer, aplikasi dan laporan *structural robotika* dapat berhubungan dengan elektronik mekanika dan perangkat lunak.

5. Visi komputer

Computer vision adalah ilmu komputer sebagai pengenalan bentuk yang mau diamati supaya dapat memberikan sebuah informasi yang bermanfaat.

6. *Intelligent Computer_Aidedd Intruction*

Adalah ilmu komputer yang dapat menyelesaikan suatu pekerjaan manusia dalam bidang pendidikan pengajar.

7. *Game Playing*

Adalah bermain dengan komputer berbentuk animasi yang menarik yaitu permainan manusia melawan mesin yang mempunyai pengetahuan berfikir.

Kecerdasan buatan dikelompokkan menjadi empat metode yaitu:

1. Pencarian (*Searcing*) teknik ini dipergunakan sebagai pencarian arah optimum mengarahkan seorang diperjalanan. Contoh: komputer yang digunakan dengan dilengkapi *Global Positioning System* (GPS). Si penumpang cukup mengatakan tempat dan tujuannya, kemudian sipengemudi akan menginput tujuan tersebut ke dalam komputer mendadak itu juga dapat di lihat rute jalan yang harus jalani.
2. Penalaran (*Reasoning*) dapat digunakan dalam melakukan penakaran sebuah masalah yang dapat dialami manusia. Dalam menggunakan teknik ini pengetahuan menjadi basis utamanya.
3. Perencanaan (*Planning*) ini dipergunakan sebagai membuat suatu perencanaan terhadap sebuah masalah yang mau diselesaikan. Dalam dunia *manufacture* dan *robotic*, teknik *planning* ini mempunyai peranan yang sangat penting.

Selain itu kecerdasan buatan memiliki ilmu yang sangat populer yaitu:

3.1.2. Logika Fuzzy (Fuzzy Logic)

Sistem pemecah permasalahan untuk dapat mengimplementasikan pada sistem disebut logika fuzzy. Pada logika fuzzy terdapat rentang nilai derajat keanggotaan dari hingga dengan satu, yang di fungsikan sebagai penerjemah besaran memakai bahasa.

3.1.3. Jaringan Syaraf Tiruan

Jaringan syaraf tiruan dapat diartikan sebagai otak yang dibuat pada sebuah cerita atau karangan ilmiah. Otak buatan tersebut mampu bekerja bisa di bilang layaknya manusia, dan dapat menanggapi sesuatu potongan informasi yang dapat diperoleh. Pandangan manusia itu dapat membuat pengguna-pengguna dalam memperhatikannya. Komputer dibuat supaya dapat menjalankannya layaknya manusia.

3.1.4. Sistem Pakar

Sistem pakar ini salah satu sistem yang berbasis teknologi komputer yang dapat dipakai dalam membereskan sebuah masalah dilakukan sama seperti yang dipikirkan oleh pakar. Sistem pakar ini juga disebut sebagai sistem yang berjuang merebut kemampuan manusia ke sebuah komputer, sehingga dapat diselesaikannya sebuah masalah seperti yang sering dilakukan oleh para ahli. Sistem pakar ini dapat di harapkan, pemakai mampu menuntaskan masalah tertentu dan tanpa meminta pertolongan dari para ahli dalam bagian tersebut (Azmi and Yasin 2017).

Berikut pendefinisian sistem menurut para ahli yaitu:

1. Sistem yang berupa kecerdasan yang memiliki kesamaan dengan manusia yang dipergunakan dalam mengolah informasi komputer sehingga permasalahan seseorang dapat terselesaikan dengan bantuan seorang ahli”(Turban-2001).
2. Menurut Jackson sistem yang komputer dapat mempresentasikan dan mengerti maksud dari seorang pakar yang digunakan dalam menyelesaikan permasalahan.
3. Komputer yang berbasis pengetahuan dapat memberikan solusi sehingga dapat menambah spesifikasi kualitas seorang pakar(Luger & Stubbefield-1993).

3.1.5. Manfaat Sistem Pakar

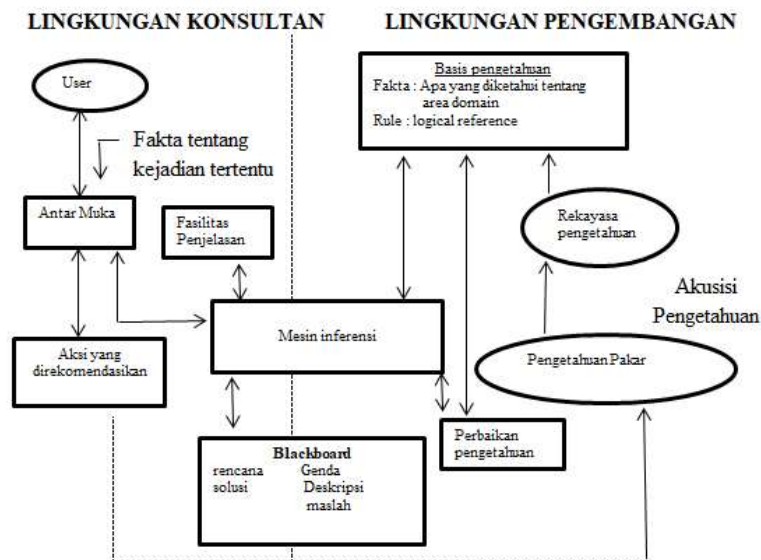
Manfaat sistem pakar memiliki kemampuan yaitu:

1. Jumlah data yang besar dapat dipertemukan
2. Data tersebut dapat disimpan dengan waktu yang sangat lama dalam bentuk yang ditentukan.
3. Melakukan pekerjaan perhitungan dengan cara cepat waktu serta dapat mencari data yang terselip dengan waktu yang cepat.
4. Meningkatkan kualitas pemula
5. Mengembangkan kelebihan dengan cara memberikan nasehat dengan konsisten serta kesalahan dapat dikurangi.
6. Membuat sebuah sistem yang dapat digunakan seorang yang awam untuk melakukan pekerjaan yang berhubungan dengan seorang pakar.

7. Pengetahuan dan kepakaran seseorang dapat ditangkap
8. Dapat memudahkan akses pengetahuan seorang pakar
9. Mampu berinteraksi dilingkungan yang sangat berbahaya

3.1.6. Struktur Sistem Pakar

Sistem pakar mempunyai komponen yang terpenting dan dimanfaatkan oleh pembentuk sistem sebagai komponen mempromosikan dalam ilmu pengetahuan. Pengembangan lingkungan dapat digunakan para pembuat sistem pakar sehingga komponen-komponen yang dibangun dan pengetahuan dapat diperkenalkan ke dalam *knowledge base (basis pengetahuan)*.



Gambar 2.1 Komponen dalam sebuah pakar
Sumber : Zulfian Azmi & Verdi Yasin (2017)

Keterangan:

1. Akuisisi pengetahuan Sub sistem ini dapat di pakai untuk menginput sebuah informasi yang berasal salah satu pakar menggunakan cara memanipulasi pengetahuan supaya komputer dapat memprosesnya dan meletakkannya ke suatu basis pengetahuan berdasarkan format yang ditentukan dengan bentuk representasi pengetahuan.
2. Basis pengetahuan (*Knowledge Base*) mengandung pengetahuan yang sangat diperlukan dengan cara menguasai, memformulasikan dan menyelesaikan masalah.
3. Mesin inferensi (*Inference Engine*) salah satu program yang berguna memandu nalar sebagai keadaan bisa mengoperasikan sesuai dengan aturan. Penyeleewengan dan mengarahkan petunjuk ,model,dan fakta yang disimpan dalam basis pengetahuan sehingga dapat solusi atau kesimpulan dapat tercapai.
4. Daerah kerja (*Blackboard*) sistem pakar membutuhkan *blackboard* untuk mencatat hasil sementara yang akan diambil dalam menguraikan masalah yang terjadi saat itu. Adapun tipe-tipe ketentuan yang dapat di tulis pada *blackboard* yaitu:
 - a. Rancangan : cara menghadapi berbagai kegiatan
 - b. Agenda : aktivitas yang digunakan untuk menanti waktu dieksekusi
 - c. Solusi : aktivitas yang akan dibangun
5. Antarmuka Pemakai(*User Inference*) Difungsikan sebagai alat pengguna alat untuk berkomunikasi dengan pengguna dan sistem pakar dengan menggunakan bahasa yang alami (*natural langue*).

6. Sub Sistem (*Explanation Subsystem/Justifier*) Berfungsi memberikan uraian terhadap pemakai sehingga memperoleh informasi seorang pakar melalui proses pengalihan informasi dan menyelesaikan masalah. Dalam mengetahui proses pemindahan keahlian pakar maupun pemecahan masalah penting bagi pengguna memiliki kemampuan.
7. Sistem Perbaikan Pengetahuan (*Knowledge Refinising System*) adalah orang yang mampu menyelesaikan suatu kesalahan dari salah seorang pakar dalam menganalisis pengetahuan.
8. Pada umumnya pengguna disebut juga dengan pemakai sistem yang dibentuk oleh beberapa ahli ketika menyelesaikan masalah, solusi, saran, atau pelatihan dari beberapa masalah yang tertentu.

Pakar ini mempunyai kemampuan yang berfungsi untuk dapat merancang dan mengembangkan cara kerjanya.

3.1.7. Keuntungan dan Kelemahan Sistem Pakar

Menurut (Azmi and Yasin 2017) Beberapa keuntungan sistem pakar jika menggunakannya, diantaranya adalah:

1. Membuat kemudahan bagi seseorang untuk mendapatkan saran dan jawaban
2. Meningkatkan produktivitas dan hasil yang baik
3. Kemampuan dan keahlian para pakar dapat di simpan
4. Penyelesaian masalah yang khusus dapat dikembangkan

5. Meningkatkan keterjaminan
6. Dapat memberikan jawaban dengan lebih cepat
7. Menjadikan sistem pakar yang *intelligent* (cerdas)
8. Mampu bekerja dengan informasi yang kurang dan mengandung ketidakpastian.
9. Dapat dipakai dalam membuka basis data dengan cerdas

Sistem pakar sama halnya dengan sistem yang lainnya, sistem pakar ini juga memiliki kekurangan seperti:

1. Pengetahuan tidaklah gampang dalam mendapatkannya. Dikarenakan sistem pakar yang kita buat dari masalah yang kita buat kadang kala berbeda.
2. Sistem pakar sangat menghabiskan biaya yang tinggi untuk membuat sistem yang berkualitas.
3. Sistem pakar bukanlah 100% bermanfaat, oleh sebab itu. Manusia sangat berperan dan sangat dibutuhkan sebelum menggunakannya.

Pada kelemahan dan kesukaran dari sebuah sistem pakar tersebut sama sekali bukanlah tidak dapat di atasi. Namun demi terus menerus melaksanakan pengalihan dan pengolahan yang didasarkan pengalaman yang telah ada.

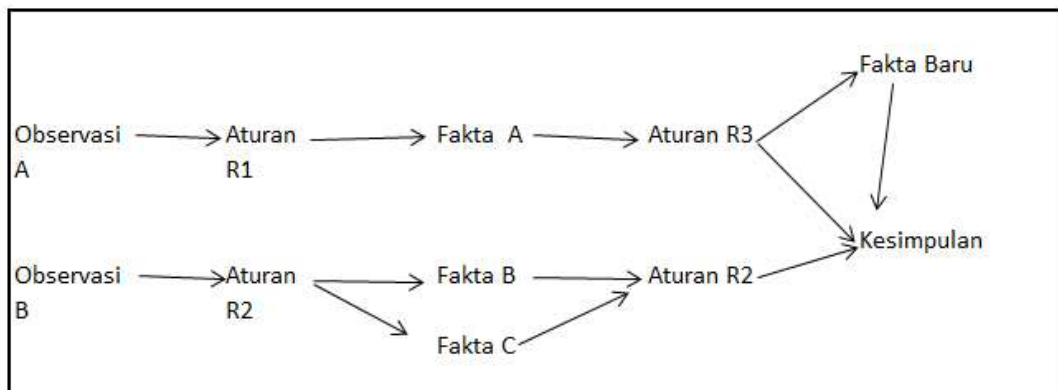
3.1.8. Inferensi

Topik ini memiliki peran yang sangat penting dalam sebuah sistem pakar karena penalarannya merupakan teknik yang umum untuk menyelesaikan masalah dalam sistem pakar. Secara umum sistem pakar digunakan apabila suatu algoritma yang ada

tidak ada jalan keluarnya dan cuma penalaran yang memungkinkan untuk solusi tersebut.

3.1.8.1. Forward Chaining

Forward chaining dapat diartikan sebagai sebuah metode yang diawali dengan penalaran sebuah data dari berbagai fakta yang tersedia mengarah pada kesimpulan. *Forward Chaining* juga dapat dikatakan suatu cara dalam pengambilan keputusan yang dimulai dengan sebuah fakta sehingga mengarah pada kesimpulan akhir. Pemeriksaan kedepannya diawali dari masukan informasi, dan selanjutnya mencoba menggambarkan kesimpulan. Sebuah gaya pelacakan pertama dengan kebenaran sudah di ketahui kemudian membuat pencocokan bukti-bukti dengan faktor IF dari Rules IF_THEN.



Gambar 2.2 *Forward Chaining*

Sumber : (Zulfian Azmi & Verdi Yasin,2017)

3.1.8.2.Backward Chaining

Backward Chaining dapat dikatakan sebuah metode yang memiliki cara kerja dengan cara kembali pada posisi semula. Proses berawal sebuah Goal (yang berada dibagian THEN dan rule IF-THEN), lalu dijalankan penelusuran untuk mencocokkan apakah fakta-fakta yang ada cocok dengan premis-premis di bagian IF. Jika cocok,*rule* dieksekusi,kemudian hipotesis dibagian THEN ditempatkan dibasis data sebagai fakta baru. Jika tidak cocok,simpan di bagian IF ke dalam *stack* sebagai *sub Goal*.

2.1.9.1. Representasi Pengetahuan

Pengetahuan (*Knowledge*) adanya reaksi yang terbentuk dalam jiwa dan pikiran seseorang. Dan representase pengetahuan adalah langkah dalam mengatakan pengetahuan untuk dapat ditemukan pada sebuah diagram yang ditentukan, maka relasi diantara pengetahuan lain bisa di pakai dalam menguji keabsahan penalaran untuk dapat di ketahui.

2.1.9.2 Jaringan Semantik

Representasi pengetahuan yang dipakai dalam melaksanakan data maupun informasi dalam memperlihatkan hubungan diantara objek merupakan pengertian dari jaringan semantik. Objek dapat berbetuk dengan benda fisik seperti mobil, rumah atau konsep berupa konsep pikiran atau tindakan.

2.1.9.3. Logika Dan Himpunan

Selain pada aturan, pengetahuan jaringan semantik dapat juga direpresentasikan dengan menggunakan simbol logika, yang studi aturannya merupakan bagian penalaran yang benar. Logika ini juga memiliki peran penting dalam sebuah sistem pakar untuk menarik sebuah kesimpulan dari fakta ke kesimpulan. Pada awalnya logika dibesarkan oleh Filosoft Aristoteles Yunani Aristoteles pada abad ke 4 sebelum masehi. Dia mendapatkan 14 tipe dan 5 lagi tipe ditemukan pada pertengahan waktu. Silogisme mempunyai 2 premis 2 premis dan 1 satu lagi konklusi yang di simpulkan dari tumpukan.

Premis : Semua laki-laki adalah makhluk hidup

Premis : Manusia merupakan makhluk hidup

Kesimpulan : Manusia adalah makhluk hidup

Pada integritas, premis dapat dikatakan fakta dari kesimpulan yang wajib di ikuti. Integritas salah satu cara untuk mempresentasikan pengetahuan.

3.2.Variable

Variable adalah sebuah karakter atau nilai dari seseorang, obyek serta aktifitas yang di tentukan oleh peneliti sehingga di pahami dan dapat di tarik kesimpulannya (Sugiyono, 2012).

3.2.1. Nyeri punggung akibat sendi facet

Nyeri punggung ini terjadi karena artritis di sendi facet. Hal ini yang paling sering terjadi pada pasien yang berusia 50 tahun ke atas. kebalikan nyeri punggung akibat bantalan tulang belakang, nyeri punggung sendi facet ini tambah parah saat melakukan ekstensi punggung dan berkurang saat membungkuk (Toni 2016).

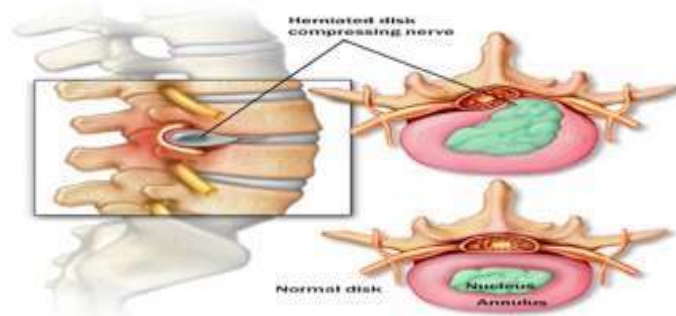


Gambar 2.3 Nyeri Punggung Akibat Sendi Facet

Sumber : (Toni 2016)

3.2.2. Nyeri Punggung Akibat Proses Denegratif

Proses denegratif ini menyerang sendi facet atau *sacroiliac* serta bantalan tulang belakang. Nyeri ini bisa datang dan hilang tanpa penyebab yang jelas, terkadang parah, terkadang ringan. Inilah karakteristik dari sakit pinggang akibat proses denegratif. (Toni 2016)

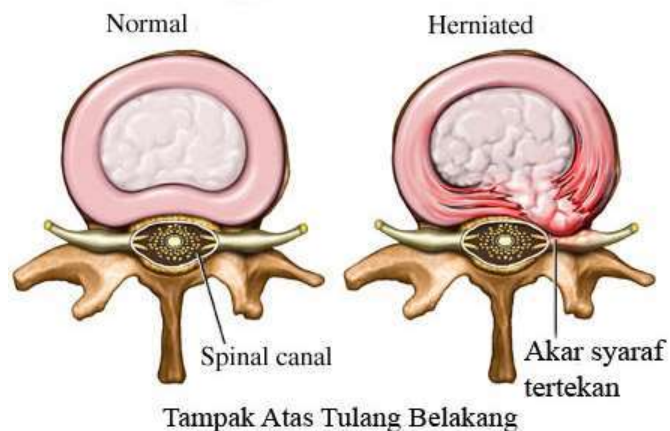


Gambar 2.4 Nyeri Punggung Akibat Proses Degenratif

Sumber : (Toni 2016)

3.2.3. Nyeri Punggung Karena HNP (*Herniated Nucleus Pulposus*)

HNP (*Herniated Nucleus Pulposus*) terjadi karena jika ada robekan di *annulus fibrosus* sehingga *jelly* keluar dan menjepit saraf. HNP terjadi secara tiba tiba. Penyebab utamanya adalah mengangkat barang berat dengan posisi membungkuk. Dalam postur ini, tekanan dalam bantalan tulang belakang sangat hebat sehingga menyebabkan *jelly* keluar.

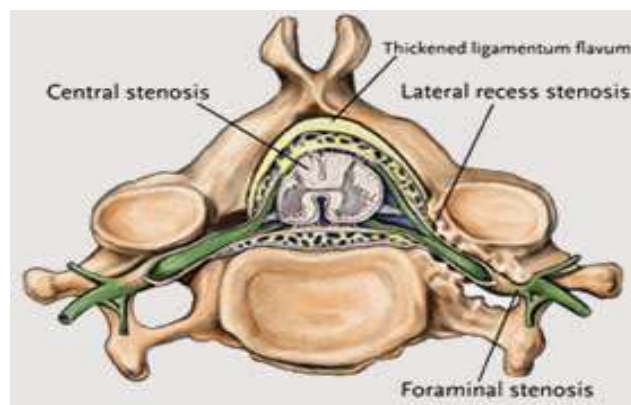


Gambar 2.5 Nyeri Punggung Karena HNP

Sumber : (Toni 2016)

3.2.4. Nyeri Punggung Karena *Spinal Stenosis*

Spinal stenosis ini adalah penyempitan rongga saraf yang terjadi karena proses denegratif. Dalam proses ini, bantalan tulang belakang menipis dan *bulging* sehingga terjadi pengapuran dan membesar di sendi *facet* serta ligamen menebal. Semua ini menyebabkan rongga saraf menyempit dan saraf terjepit.

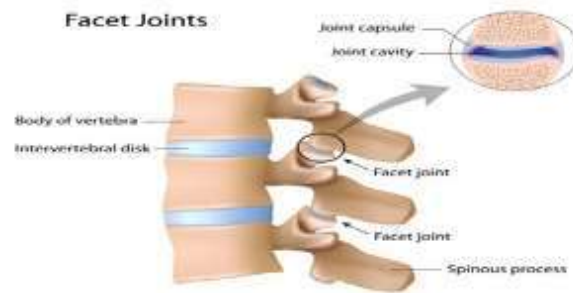


Gambar 2.6 Nyeri Punggung Karena *Spinal Stenosis*

Sumber : (Toni 2016)

3.2.5. Nyeri Punggung Akibat Sendi Facet Yang Terkunci

Terkadang punggung kita terasa kaku dan tidak tidak nyaman. Yang menjadi salah satu penyebabnya adalah sendi facet yang terkunci. Hal ini sangat lah sering terjadi pada pasien muda. Selain kaku punggung, pasien juga mengeluh nyeri di punggung, pantat, atau paha. Otot punggung mengalami *spasme* atau kaku sehingga sendi facet tidak bisa bergerak dengan bebas.

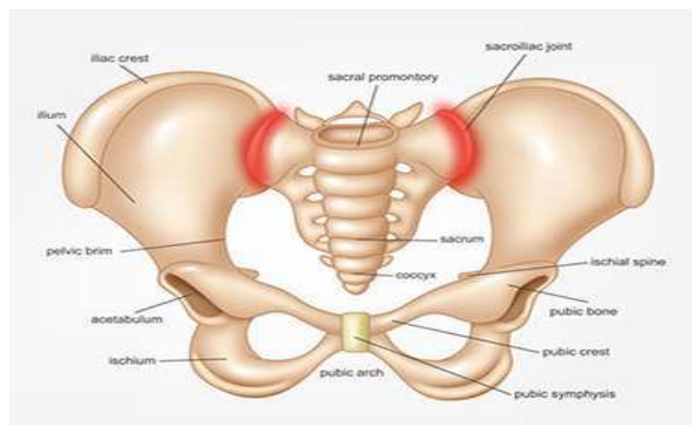


Gambar 2.7 Nyeri Akibat Sendi Facet Yang Terkunci

Sumber : (Toni 2016)

3.2.6. Nyeri Facet Akibat Radang di Sendi *Sacroiliac*

Nyeri sendi akibat sendi sacroiliac (*sacroiliac joint dysfunction*) terjadi pada orang muda dan tua. Lokasi nyeri sendi sacroiliac terletak di bawah lokasi nyeri sendi facet. Terkadang nyeri ini menjalar ke pantat dan pangkal paha bagian depan.



Gambar 2.8 Nyeri Facet Akibat Radang di Sendi Sacroiliac

Sumber : (Toni 2016)

3.2.7. Nyeri Punggung Akibat Radang di Bantalan Tulang belakang

Nyeri ini akan memburuk saat membungkuk dan berkurang saat ekstensi punggung. Hal ini bisa terjadi akibat robekan di lapisan pembungkus bantalan tulang belakang. Nyeri punggung bisa terjadi langsung atau beberapa jam setelah kejadian yang menyebabkan nyeri punggung. Tidak jarang juga radang di bantalan tulang belakang terjadi penyebab yang jelas (Toni 2016).



Gambar 2.9 Nyeri Punggung Akibat Radang di Bantalan Tulang Belakang

Sumber : (Toni 2016)

1.3. Software Pendukung

1.3.6. UML

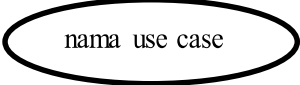


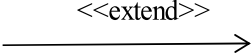
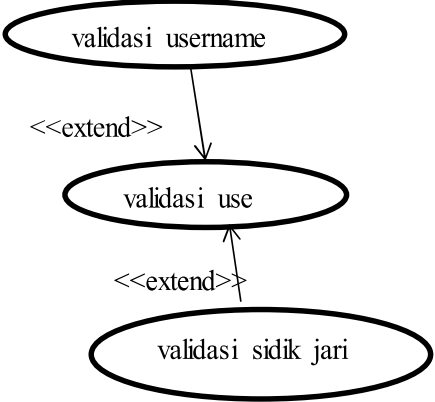
Dalam masa kemajuan teknologi perangkat lunak, sangat dibutuhkan bahasa dalam menampilkan sebuah perangkat lunak yang akan dirancang, dibutuhkan adanya sebuah standar supaya orang-orang dimanapun berada bisa memahami model dari perangkat lunak ini. UML (*Unified modeling Langue*) adalah bahasa yang memiliki kelengkapan notasi dan sebuah standar visualisasi, menganalisis dan

mendesain, serta arsitektur dapat digambarkan dalam pemrograman berorientasi. (Rosa and M.Shalauddin 2018).

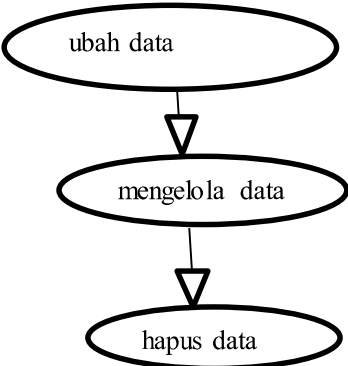
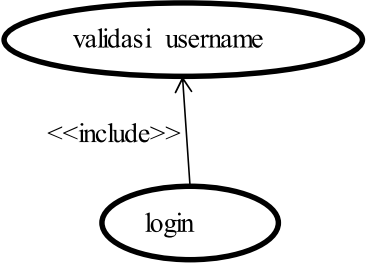
1.3.6.1. *Use Case Diagram*

Use case diagram yang memiliki fungsi yang memberikan gambaran dari beberapa pelaku untuk mengenali sistem yang diperintah. *Use case* menguraikan sebuah interaksi dan beberapa aktor dengan sistem informasi yang nantinya dirancang. *Use case* ini juga dapat dimanfaatkan sebagai alat dalam mendapatkan informasi dari fungsi yang ada pada sistem informasi dan siapa saja user yang mampu menggunakannya. (Rosa and M.Shalauddin 2018).

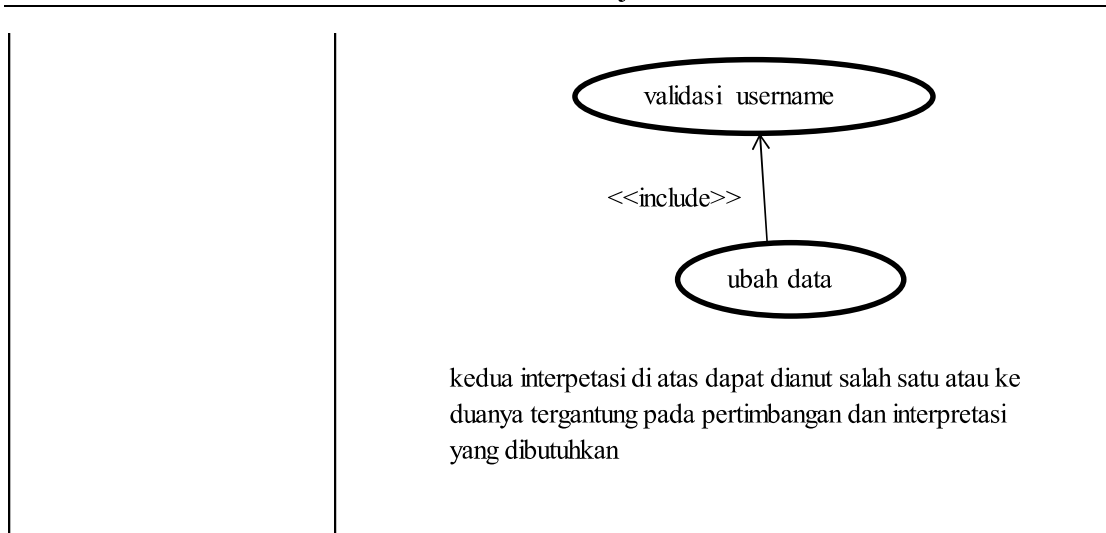
Tabel 2.1 *Use Case*

Simbol	Deskripsi
Use case 	fungsionalitas yang di sediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
Aktor/actor  nama aktor	orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat sendiri, jadi walaupun simbol dari aktor adalah orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
Asosiasi/association 	komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
Ekstensi/ <i>extend</i> 	<p>relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambah dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> dan pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang ditambahkan, misal</p>  <pre> graph TD A([validasi username]) -- "<<extend>>" --> B([validasi use]) C([validasi sidik jari]) -- "<<extend>>" --> B </pre> <p>arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>

Tabel 2.1 Lanjutan

<p>Generalisasi/ <i>generalitation</i></p>	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misal:</p>  <pre> graph TD A(ubah data) --> B(mengelola data) B --> C(hapus data) </pre> <p>arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>
<p>Menggunakan / include/ <i>uses</i></p> <p><<include>> →</p> <p>"uses" →</p>	<p>relasi <i>use case</i> tambahan ke sebuah <i>use case</i> diman <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini</p> <p>ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i></p> <p>include berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan; misal pada kasus berikut:</p>  <pre> graph TD A(login) -- "<<include>>" --> B(validasi username) </pre> <p>include ini berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tmbahan dijalankan , misal pada kasus berikut:</p>

Tabel 2.1 Lanjutan



Sumber : (Rosa and M.Shalauddin 2018)

2.4.1.2. Class Diagram


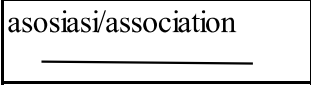
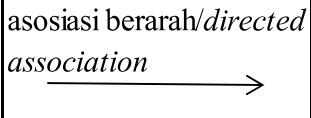
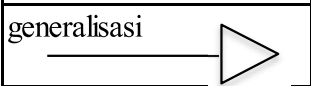

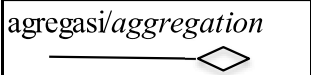
Diagram *class* yaitu suatu situasi yang berorientasi obyek. Kelas tersebut mempunyai atribut-atribut dan metode atau operasi. Pada tahap utama dalam *class diagram* dapat dilihat dalam tabel dibawah ini.

1. Memiliki atribut yang dimana suatu kelas mempunyai variabel.
2. Beberapa fungsi yang mempunyai sebuah kelas adalah metode atau operasi.

Tabel 2.2 Class Diagram

Simbol	Deskripsi
Kelas <div style="border: 1px solid black; padding: 5px; width: fit-content;"> nama_kelas +atribut +operasi() </div>	Kelas pada struktur sistem

Tabel 2. 2 Lanjutan



antarmuka/ <i>interface</i> 	sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
asosiasi/ <i>association</i> 	relasi antar kelas dengan makna umum asosiasi biasanya juga di sertai dengan <i>multiplicity</i>
asosiasi berarah/ <i>directed association</i> 	relasi antar kelas dengan makna kelas yang satu digunakan oleh yang lain, asosiasi biasanya juga di sertai dengan <i>multiplicity</i>
generalisasi 	relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
ketergantungan kelas 	relasi antar kelas dengan makna kebergantungan kelas
agregasi/ <i>aggregation</i> 	relasi antarkelas dengan makna semua-bagian(<i>unwhole-part</i>)

Sumber : (Rosa and M.Shalauddin 2018)

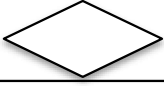


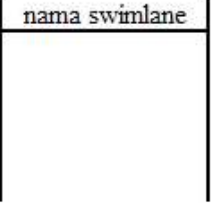
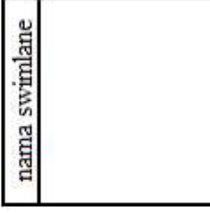
2.4.1.3. Activity Diagram

Diagram aktivitas menggambarkan proses kerja dalam perangkat lunak. Dapat juga dikatakan sebagai aliran kerja bisnis dan menu yang terdapat dalam perangkat lunak. Perlu diperhatikan bahwa *activity diagram* dapat digambarkan pada aktivitas sistem dan aktor tidak melakukannya. (Rosa and M.Shalauddin 2018).

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

Tabel 2.3 Lanjutan


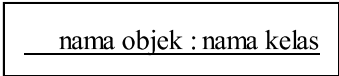
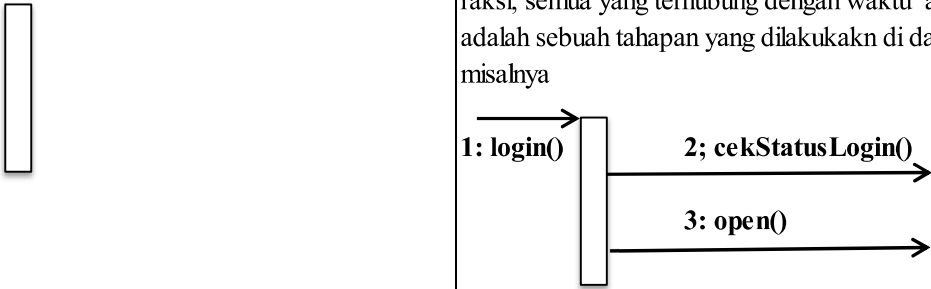
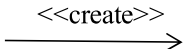
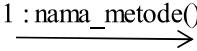
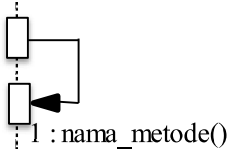
percabangan/decision 	asosiasi percabgan diman jika ada pilihan aktivitas lebih dari satu
penggabungan/join 	asosiasi penggabungan diman lebis dari satu aktivitas digabungkan menjadi satu
status akhir 	status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
swimlane  atau 	memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber : (Rossa And M.Shalahuddin,2018)

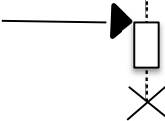
2.4.1.4.Sequence Diagram

Kelakuan objek *sequence diagram* dapat digambarkan dalam *use case* seperti menguraikan pesan yang nantinya dikirim dan diterima dalam objek. Sehingga dalam menggambarkan *sequence diagram* beberapa objek menjadi ikut dalam suatu *use case* bersama dengan beberapa metode yang dipunyai kelas. Dan objek yang diintanisasi harus diketahui. (Rosa and M.Shalauddin 2018).

Tabel 2.4 *Sequence Diagram*

Simbol	Deskripsi
<p>Garis hidup / <i>lifeline</i></p> 	menyatakan kehidupan suatu objek
<p>Objek</p> 	menyatakan objek yang berinteraksi pesan
<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, misalnya</p> <p>maka cekstatusLogin() dan open() dilakukan dalam metode login()</p> <p>Aktor tidak memiliki waktu aktif</p>
<p>Pesan tipe create</p> 	menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
<p>pesan tipe call</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 ia memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang dikirim</p>

Tabel 2.4 Lanjutan

<p>pesan tipe send 1 : masukan →</p>	<p>menyatakan bahwa suatu objek mengirimkan data/ masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>
<p>Pesan tipe return 1 : keluaran →</p>	<p>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</p>
<p>Pesan tipe destroy <<destroy>> </p>	<p>menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang di akhiri, sebaiknya jika ada create maka ada destroy</p>

Sumber : (Rosa and M.Shalauddin 2018)

1.3.7. Website

Website merupakan kumpulan dari beberapa halaman yang sering dikunjungi pengguna ketika menjelajah di internet apakah untuk mencari informasi, mencari hiburan ataupun berbelanja secara *online* di marketplace. Disaat akan mengakses sebuah halaman *website* tentu pengguna harus terhubung terlebih dahulu ke *internet*. Anda dapat memasukkan *url* berupa nama domain yang mewakili sebuah *website* agar nantinya diarahkan ke halaman *website* yang disetujui. *Website* adalah kumpulan dari beberapa halaman atau page yang saling terhubung dan mempunyai informasi yang saling berkaitan. Pendapat lain menjelaskan *website* merupakan sekumpulan halaman pada suatu domain di *internet* dan dibuat dengan sengaja untuk bertujuan tertentu, antar halaman utama dengan memasukkan URL pada sebuah *browser*. (Nopriadi and Sestri 2020).

1.3.8. HTML (*Hypertext Markup Language*)

Menurut (Abdulloh 2018) HTML *Hypertext Markup Language* yaitu bahasa yang secara umum *web* yang dijalankan penggunaanya oleh W3C (*World Wide Web Consortium*) yang dapat mengatur setiap elemen dari *website* berupa *tag-tag*. Peran HTML ini juga dapat memberikan pengetahuan dalam menjelajah *web internet* serta *formatting hypertext* yang sederhana. HTML memiliki aturan-aturan yang disusun secara terstruktur dalam proses pembuatannya. Dalam menuliskan *skrip* pada HTML cukup memerlukan *text* editor seperti *Notepad* sebagai bentuk yang paling sederhana atau editor khusus yang mampu menandai setiap unsur HTML dan menampilkannya dengan *Notepad++*, *Sublime Text*, dan masih banyak lagi aplikasi lain yang sejenisnya.

HTML terbagi dari banyak unsur sebagai pembentuk struktur skrip HTML, seperti: *tag*, atribut, serta elemen.

1. Tag

Tag sebagai simbol individual yang berbentuk dua karakter “<” dan “>” yang menekan suatu *text* sebagai nama tag.

HTML memiliki karakteristik sebagai berikut:

1. Tag dapat ditulis dengan huruf besar (kapital), kecil atau campuran, dimana sifatnya berupa *incase sensitive*. Contohnya: <BODY>, <body> atau <Body>

dimana seluruhnya mempunyai nilai yang serupa. Untuk hasil yang semakin bagus dapat dipakai huruf kecil.

2. Tag diperbolehkan memiliki tag yang lain. Tag dapat dibuka pertama dan ditutup diakhir, dan sebaliknya. Akan tetapi jika ada sesuatu yang salah pada struktur ini maka akan diperoleh *error*, Misalnya:

```
<p>I like <b>HTML</b></p>
```

2. Atribut

Atribut mampu menyelesaikan elemen tag yang nantinya ditampilkan. Atribut juga ada yang mempunyai nilai dan ada juga yang tidak. Nilai pada sebuah atribut dicatat dengan tanda petik dan dipisahkan menggunakan *symbol* sama dengan pada *atribut name*. Misalnya: `<p align="center">`.

Dibawah ini merupakan karakteristik dalam atribut HTML:

1. Atribut dapat dicatat dalam tag pembuka.
2. Hanya yang memiliki atribut yang berbeda untuk dapat diterapkan dalam tagnya.
3. Global atribut yaitu sebagian atribut yang diterapkan pada seluruh tag, seperti: *id*, *style* dan sebagainya.
4. Suatu tag biasanya dicatat dengan satu atribut, banyak atribut, maupun tanpa memakai atribut.
5. Bila pada tag dicatat banyak atribut, maka setiap atribut harus dipisah dengan memakai spasi.

3. Element

Element merupakan suatu bidang dari skrip HTML dimana terdapat tag pembuka, elemen isi, dan tag penutup. Bila suatu elemen diperlihatkan dalam browser, maka yang terlihat dalam browser yaitu hanya isinya. Misalnya:

```
<p> Saya suka memakai HTML</p>
```

Bila terlihat dalam browser, menjadi “Saya suka memakai HTML”. Maka itulah yang akan muncul pada *browser*.

1.3.9. Bahasa Pemrograman PHP

PHP singkatan dari *Hypertext Preprocessor* yang merupakan bahasa yang mampu di temukan dan dapat disisipkan dalam *skrip* HTML. Bahasa pemrograman ini memiliki tujuan untuk memprogram situs *web* dinamis. Dan Php ini dapat menolong pengguna dalam menciptakan sistem yang berbasis *web*. (Abdulloh 2018).

Agar dapat menjalankan PHP harus menyediakan perangkat lunak berikut:

1. *Web server* (Apace,IIS, Personal Web Server/PWS)
2. PHP Server
3. Database Server (MySQL,Interbase, MS SQL,dll)

Fungsi PHP dalam pemrograman *web* diantaranya sebagai berikut:

1. Mewujudkan halaman yang dinamis pada *website*

2. Membuat, membuka, menulis, membaca, menghapus dan menutup *file* pada *server*.
3. Melakukan proses data yang dikirim dari *form*.
4. Melakukan pengiriman dan pengaksesan persamaan
5. Data dalam database dapat di modifikasi
6. Akses *user* dapat di kontrol
7. Mengkripsi data.

1.3.10. Php MyAdmin

Php MyAdmin merupakan aplikasi memiliki orientasi objek atau sebuah aplikasi yang mempermudah management database *MySQL*. Peneliti menggunakan *Php Myadmin* yang ditemukan di paket XAMPP yang digunakan dalam membuat database. Dengan memiliki keunggulan-keunggulan yang ada, sehingga para pengguna awam tidak mampu untuk dapat memahami sintaks-sintaks SQL dalam perancangan *database* dan *table* (Nugroho 2019).



Gambar 2.10 Logo Php Myadmin

1.3.11. MySQL Database

Menurut (Abdulloh 2018) Sebelum lebih jauh membahas *database*, ada baiknya kenali dulu istilah yang sering digunakan pada pembahasan database, yaitu sebagai berikut:

1. DBMS, yang mempersiapkan *service* bagi pengguna dalam menciptakan sebuah sistem perangkat lunak, mengakses dan mengontrol *database*.
2. Gabungan informasi yang menggunakan komputer dengan cara sistematis disebut dengan *database* sampai informasi diperoleh dengan cara memeriksa menggunakan program komputer.
3. Table, merupakan kumpulan data yang diorganisasikan dalam baris (*record*) dan kolom (*field*). Dalam satu database biasanya terdiri dari beberapa tabel.
4. Field, merupakan bagian pada table yang mempunyai ukuran dan memiliki tipe data yang tertentu.

5. *Record*, adalah baris data yang mempunyai nilai masing-masing *field* dalam tabel. Satu tabel biasanya terdapat banyak *record*.
6. *Key*, merupakan suatu *field* (kolom) yang menjadikan sebagai kunci pada operasi tabel.
7. SQL, *Structured Query Langue* merupakan bahasa yang digunakan dalam mengakses dan manipulasi database.



Gambar 2.11 Logo PHP MySQL

1.3.12. CSS (*Cascading Style Sheet*)

CSS bertujuan memisahkan komponen penting dengan komponen lainnya yang bentuknya lebih spesifik. Karena strukturnya yang terlalu mudah maka penggunaanya tidak mengira bahwa CSS ini tidak termasuk salah satu bahasa pemrograman. Hanya berupa gabungan aturan yang mengatur *style* elemen HTML (Abdulloh 2018).

1.3.13. XAMPP

XAMPP singkatan dari X Apache MySQL PHP Perl,X adalah sistem operasi (*Windows,Linux,Unix*), merupakan paket *software* yang terdiri dari *serverweb*(Apache), database(MySQL – MariaDb), dan pengembangan aplikasi (PHP dan Perl); disebut juga sebagai software Stack. XAMPP dikembangkan oleh group pengguna *server web* Apache – *ApacheFriends.org*. XAMPP merupakan *software* stack untuk pengembangan aplikasi berbasis *web* yang paling populer, masih banyak paket *software* yang mirip, seperti LAMP,MAMP,atau WAMP; yang berbasis pada *server web* Apache juga (Betha 2020).



Gambar 2.12 Logo XAMPP

1.3.14. Notepad ++

Notepad ++ merupakan program freeware yang berfungsi untuk pengganti editor Notepad *windows* bawaan. Notepad ++ disalin ke dalam bahasa C++ yang memiliki eksekusi lebih cepat baik dan memiliki ukuran program yang kecil.



Gambar 2.13 Logo Notepade++

1.4. Peneliti Terdahulu

1. Pada jurnal penelitian (Izman Husna, Arifin, Gunung Setiabudi, 2021) dengan judul “Usia dan posisi kerja pengrajin payet berpengaruh terhadap keluhan *low back pain*” dengan : ISSN 1829-9407 membahas tentang masalah kesehatan akibat pekerja terlalu lama duduk sehingga menimbulkan penyakit *low back pain*.
2. Pada jurnal penelitian (Naufal Afif, Siswi jayanti, Ida Wahyuni 2012) dengan judul “Hubungan postur kerja, durasi mengemudi dengan keluhan nyeri punggung bawah (NPB) pada sopir truk barang antar kota di cv semeru putra semarang” dengan ISSN : 2715-5617 membahas tentang kurangnya diperhatikan kesehatan dalam bekerja di dalam perusahaan.
3. Pada jurnal penelitian (Kardinati Hayati dan tati Devi 2020) dengan judul “Efektifitas terapi *ice message* dan *back message* terhadap perubahan intensitas nyeri pada pasien *low back pain* di rumah sakit granmed lubuk pakam” dengan ISSN: 2655-0830 membahas tentang cara menamngami pasien dengan cara melakukan terapi atau pun dengan cara pijat pada punggung tersebut.

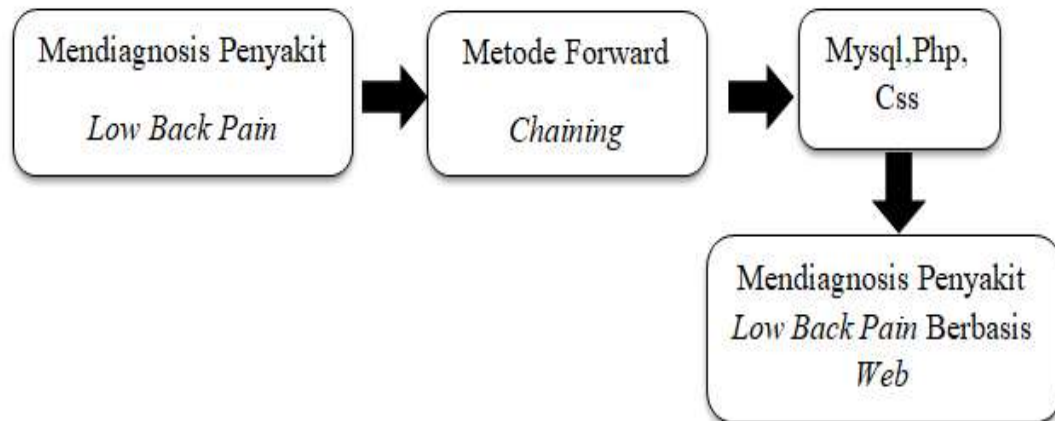
4. Pada jurnal penelitian (Ricca Sahara, Terry Y.R. Pristya, 2020) dengan judul “Faktor resiko yang berhubungan dengan kejadian *low back pain* pada pekerja” dengan ISSN : 1412-2804 membahas tentang resiko penyakit *low back pain* pada pekerja yang berlebihan sehingga dapat menimbulkan kelelahan dan mengakibatkan *low back pain*.
5. Pada jurnal penelitian (Edriyani Yonlafado Br. Simajuntak, Erwin Silitonga, Novita Aryani, 2020) dengan judul “latihan fisik dalam upaya pencegahan *low back pain*” dengan ISSN : 2721-9224 membahas tentang mengatasi kondisi *low back pain* dengan cara latihan fisik yang rutin sehingga sendi-sendi tulang belakang dapat bergerak untuk mencegah terjadinya *low back pain*.
6. Pada jurnal penelitian (Ibnu Akil, 2017) dengan judul “analisa efektivitas metode *forward chaining* dan *backward chaining* pada sistem pakar” dengan ISSN : 1978-1946 & 2527-6514 membahas tentang penggunaan algoritma *forward chaining* dan *backward chaining* dalam pencarian untuk mencocokkan fakta fakta yang menuju kesimpulan.
7. Pada jurnal penelitian (Rahmi Ras Fanny, Nelly Astuti Hasibuan, Efori Buuololo, 2017) dengan judul “Perancangan sistem pakar diagnosa penyakit Asidosis tubulus renalis menggunakan metode *certainty factor* dengan penelusuran *forward chaining*” dengan ISSN : 2548-8368 membahas tentang sistem pakar yang bertujuan dengan teknik pelacakan cara maju dengan cara rekaman informasi untuk mencapai penyelesaian akhir.

8. Pada jurnal penelitian (Aprilia Wulanduri,2020) dengan judul “Terapi Chiropratics (Spinal Manipulation) terhadap *low back pain* dengan ISSN : 2715-9728 membahas tentang nyeri pada pasien kronis yang seringkali mengobati dengan cara alternatif, jurnal ini membahas tentang cara penanggulangan penderita penyakit *low back pain* dengan cara terapi atau disebut juga Spinal Manipulation.
9. Pada jurnal penelitian (Yulisman, Monalisa 2019) dengan judul “sistem pakar mendiagnosa penyakit demam berdarah dengan menggunakan metode *forward chaining* berbasis web” dengan ISSN : E-2597- 3918 membahas tentang sistem pakar berbasis *web* metode *forward chaining* dengan menggunakan bahasa pemrograman PHP dan MySql sebagai basis data dalam pembuatan dan pengembangan sistem pakar.
10. Pada penelitian ini (Refli Noviardi, 2020) dengan judul “sistem pakar berbasis *web* menggunakan metode *forward chaining* dalam menganalisa kerusakan mesin fotokopi dan penanggulangannya (studi kasus di-El Copier *service center* dan distributor” dengan ISSN 2550-1201 membahas tentang bagaimana menciptakan aplikasi sistem pakar dengan metode *forward chaining* untuk menganalisa kerusakan mesin fotokopi.

1.5. Kerangka Berpikir

Kerangka pemikiran dalam perancangan ini dimulai dengan identifikasi masalah dan perumusan masalah yang dapat dibuat sistem pakar. Selanjutnya basis

pengetahuan yang menentukan dan mengimplementasikan metode *forward chaining* yang dapat digunakan pada aplikasi sistem pakar yang dibuat sehingga dapat di implementasikan (Nita and Deni 2018).



Gambar 2.14 Kerangka Berpikir

Sumber : Data Penelitian (2021)