

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan dunia otomotif serta pemasarannya pada era globalisasi sekarang ini telah berkembang menjadi begitu kompleks dan begitu penuh dengan inovasi-inovasi yang dilakukan oleh para pelakunya. Khususnya pada perkembangan industri sepeda motor di Indonesia mengalami perkembangan yang signifikan dengan hadirnya berbagai macam merek yang digunakan oleh perusahaan yang merupakan produsen. Di Indonesia, sepeda motor dianggap dapat memenuhi kebutuhan masyarakat golongan ekonomi menengah kebawah, disamping keunggulan dalam kemampuan bermanuver disela-sela kemacetan. Sepeda motor juga memberikan efisiensi dalam biaya perjalanan.

Sepeda motor Beat merupakan jenis sepeda motor matic yang banyak digemari oleh masyarakat. Honda Beat adalah sepeda motor yang berjenis skuter matic dan dalam peluncuran perdananya sudah langsung menarik konsumen karena honda matic mempunyai kualitas mesin terbaik.

Salah satu sepeda motor yang sangat diminati dan masih banyak penggunaanya Honda Beat. Banyaknya peminat Motor Honda Beat tak luput dari desainnya yang irit, mudah dimodifikasi, elegan dan minimalis, selain itu harga jenis motor Honda satu ini bisa dikatakan murah.

Dikarenakan masih banyaknya pengguna Honda Beat yang menggunakan sistem Karbu, tentu banyak sekali kendala yang terjadi, yang jarang diketahui oleh

pengguna awam. Kerusakan yang sering terjadi pada sistem Mesin Honda Beat adalah sebagai berikut (1) Kerusakan Sistem Bahan Bakar, (2) Kerusakan Sistem Pengapian, dan (3) Kerusakan kompresi Mesin. Dari permasalahan tersebut tentunya dibutuhkan suatu sistem yang dapat dengan mudah digunakan oleh pengguna sepeda motor Honda Beat yang tidak begitu mengerti tentang sistem Mesin. Sistem yang mampu untuk mendeteksi kerusakan Mesin Honda Beat tersebut adalah sistem pakar.

Sistem Pakar merupakan sebuah sistem yang mampu mengidentifikasi sebuah permasalahan dengan menggunakan keahlian seorang pakar yang telah ditanamkan kedalam sebuah sistem dengan menggunakan algoritma tertentu. Dalam jurnal Edik Informatikan dikatakan bahwa “Sistem pakar adalah sebuah sistem yang dibangun dengan berbasis komputer yang menggunakan beberapa pengetahuan, fakta dan teknik penalaran dalam memecahkan suatu permasalahan yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut. Implementasi sistem pakar ini sangat banyak digunakan untuk kepentingan komersial karena sistem pakar dapat dipandang sebagai cara penyimpanan pengetahuan pakar dalam bidang tertentu kedalam program komputer dan dirancang sedemikian rupa sehingga dapat memberikan keputusan dan melakukan penalaran secara cerdas” (Boekittinggi, et al., 2017).

Sistem pakar tidak dapat digunakan dalam menyelesaikan permasalahan tanpa sebuah algoritma atau algoritma, oleh sebab itu untuk membantu dalam mendeteksi kerusakan mesin pada Honda Beat, dipilihlah *K-Nearest Neighbor (KNN)*. Karakteristik algoritma ini adalah merepresentasikan

derajat kepercayaan suatu fakta atau aturan. *K-Nearest Neighbor (KNN)* adalah suatu sistem yang mampu memecahkan ketidakpastian (Butsiarah Markani, 2019).

Dari pembahasan penelitian ini diharapkan perangkat lunak yang dirancang dapat membantu teknisi ataupun pemilik sepeda motor Honda Beat untuk mengetahui kerusakan sistem Mesin motornya dengan mudah. Berdasarkan deskripsi masalah di atas maka diangkatlah judul penelitian “**SISTEM PAKAR DALAM MENDETEKSI KERUSAKAN SISTEM MESIN SEPEDA MOTOR HONDA BEAT DENGAN MENGGUNAKAN ALGORITMA *K-NEAREST NEIGHBOR (KNN)*”**”.

1.2 Rumusan Masalah

Adapun rumusan masalah yang akan di bahas pada penelitian ini yaitu sebagai berikut:

1. Bagaimana mendeteksi kerusakan Mesin Sepeda Motor Honda Beat menggunakan sistem pakar untuk mengetahui jenis kerusakan yang ada pada mesin sehingga dapat diterapkan pada algoritma *K-Nearest Neighbor (KNN)* di sistem pakar ?
2. Bagaimana merancang program sistem pakar yang dapat mendeteksi Kerusakan Mesin pada sepeda motor Honda Beat dengan Algoritma *K-Nearest Neighbor (KNN)*?
3. Bagaimana menerapkan algoritma *K-Nearest Neighbor (KNN)* kedalam program untuk mendeteksi Kerusakan Sistem Mesin pada sepeda motor Honda Beat?

4. Bagaimana menguji aplikasi sistem pakar dalam mendeteksi Kerusakan Sistem Mesin pada sepeda motor Honda Beat dengan Algoritma *K-Nearest Neighbor (KNN)*?

1.3 Batasan Masalah

Agar penelitian yang dilakukan ini sesuai dengan permasalahan yang di bahas maka penelitian ini memiliki beberapa batasan yaitu sebagai berikut:

1. Permasalahan yang dibahas pada penelitian ini adalah kerusakan Sistem Mesin pada sepeda motor Honda Beat.
2. Solusi dalam memecahkan permasalahan dalam sistem pakar untuk mendeteksi kerusakan Sistem Mesin pada sepeda motor Honda Beat adalah algoritma *K-Nearest Neighbor (KNN)*.
3. Data pendukung dalam perancangan sistem pakar ini diambil dari Bahagia Service dan didukung oleh pakar bernama Gatot Kuswari yang ahli dalam bidang mesin Honda Beat.
4. Program yang digunakan dirancang berbasis web.

1.4 Tujuan Penelitian

Tujuan penelitian merupakan suatu hal yang akan dicapai pada penelitian ini. Adapun tujuan dari penelitian ini adalah :

1. Untuk mendeteksi kerusakan Mesin Sepeda Motor Honda Beat menggunakan sistem pakar untuk mengetahui jenis kerusakan yang ada pada mesin sehingga dapat diterapkan pada algoritma *K-Nearest Neighbor (KNN)* di sistem pakar.
2. Untuk merancang program sistem pakar yang dapat mendeteksi Kerusakan Mesin pada sepeda motor Honda Beat dengan Algoritma *K-Nearest Neighbor (KNN)*.

3. Untuk menerapkan algoritma *K-Nearest Neighbor (KNN)* kedalam program untuk mendeteksi Kerusakan Sistem Mesin pada sepeda motor Honda Beat.
4. Untuk menguji aplikasi sistem pakar dalam mendeteksi Kerusakan Sistem Mesin pada sepeda motor Honda Beat dengan Algoritma *K-Nearest Neighbor (KNN)*.

1.5 Manfaat Penelitian

Manfaat penelitian merupakan dampak dari tercapainya tujuan dan terjawabnya rumusan masalah secara akurat. Adapun manfaat penelitian adalah sebagai berikut:

1. Dapat membantu penelitian di dalam memecahkan permasalahan yang terjadi selama ini berkenaan dengan mendeteksi Kerusakan Mesin pada sepeda motor Honda Beat dengan menggunakan *K-Nearest Neighbor (KNN)*.
2. Dapat membantu pengguna motor Honda Beat dalam mendeteksi kerusakan Mesin pada sepeda motor Honda Beat.
3. Dapat membantu pembaca sebagai referensi peneliti untuk mengembangkan ilmu pengetahuan khususnya dibidang teknik Mesin.

1.6 Metodologi Penelitian

Langkah langkah yang dilakukan dalam penelitian ini adalah sebagai berikut :

1. Studi Kepustakaan

Pada tahap ini dilakukan studi kepustakaan yaitu proses mengumpulkan informasi dengan melakukan pengumpulan, mempelajari,

dan membaca berbagai bahan referensi yang berkaitan dengan aplikasi sistem pakar serta algoritma *K-Nearest Neighbor (KNN)*. Adapun literatur yang digunakan meliputi buku, artikel, paper, jurnal, makalah, internet dan sumber lainnya.

2. Analisis dan Perancangan

Pada tahap ini dilakukan analisis spesifikasi aplikasi sistem pakar dan melakukan perancangan aplikasi sistem pakar, seperti perancangan proses dan antarmuka yang meliputi desain database sebagai *Back End*, dan sketsa tampilan sebagai *Front End*, dan lain sebagainya.

3. Pengkodean

Pada tahap ini dilakukan pengkodean aplikasi sistem pakar sesuai dengan analisis spesifikasi dan perancangan yang telah ditentukan.

4. Pengujian Aplikasi

Pada tahap ini dilakukan pengujian terhadap aplikasi sistem pakar yang telah dibangun, dan tingkat keakuratan dari sistem aplikasi yang telah dibuat.

5. Penyusunan Laporan

Pada tahap ini dilakukan penulisan dokumentasi dan laporan dari aplikasi sistem pakar yang dikembangkan.

1.7 Sistematika Penulisan

Sistematika penulisan Tugas Skripsi ini dibagi atas beberapa bab, di mana masing-masing bab dibagi atas beberapa sub agar mempermudah penjelasan mengenai penelitian yang dilakukan dan mempermudah pembaca dalam

memahami isi penelitian. Adapun sistematika penulisan Tugas Skripsi ini adalah sebagai berikut :

BAB I PENDAHULUAN

Pendahuluan berisi tentang Latar Belakang Masalah, Rumusan Masalah, Tujuan, Manfaat, Batasan Masalah, Metodologi Penelitian dan Sistematika Penulisan dalam pembuatan Tugas Skripsi.

BAB II LANDASAN TEORI

Bab ini berisi teori-teori pengetahuan dasar yang di peroleh dari studi kepustakaan atau literatur dan dokumentasi internet yang digunakan untuk memahami permasalahan yang dibahas pada penelitian ini. Teori – teori pengetahuan dasar yang disajikan antara lain tentang aplikasi sistem pakar serta algoritma *K-Nearest Neighbor (KNN)*.

BAB III ALGORITMA PENELITIAN

Bab ini menguraikan tahapan-tahapan sistematis yang digunakan untuk melakukan kajian penelitian. Tahapan-tahapan tersebut merupakan kerangka yang dijadikan pedoman penelitian untuk mencapai tujuan yang telah ditetapkan. Tahapan tersebut dimulai dari waktu dan tempat penelitian serta alat dan bahan yang digunakan dalam aplikasi sistem pakar dengan menggunakan algoritma *K-Nearest Neighbor (KNN)*.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi tentang hasil dan pembahasan dari aplikasi sistem pakar menggunakan algoritma *K-Nearest Neighbor (KNN)* yang telah dibuat.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari keseluruhan uraian bab bab penulisan skripsi dan saran yang diajukan untuk pengembangan lebih lanjut.

BAB II

TINJAUAN PUSTAKA

2.1 Sepeda Motor

Sepeda motor adalah kendaraan beroda dua yang digerakkan oleh sebuah mesin. Letak kedua roda sebaris lurus dan pada kecepatan tinggi sepeda motor tetap stabil disebabkan oleh gaya giroskopik. Sedangkan pada kecepatan rendah, kestabilan atau keseimbangan sepeda motor bergantung kepada pengaturan setang oleh pengendara. Penggunaan sepeda motor di Indonesia sangat populer karena harganya yang relatif murah, terjangkau untuk sebagian besar kalangan dan penggunaan bahan bakarnya serta biaya operasionalnya cukup hemat .

Sepeda motor merupakan pengembangan dari sepeda konvensional yang lebih dahulu ditemukan. Pada tahun 1868, Michaux ex Cie, suatu perusahaan pertama di dunia yang memproduksi sepeda dalam skala besar, mulai mengembangkan mesin uap sebagai tenaga penggerak sepeda. Namun usaha tersebut masih belum berhasil dan kemudian dilanjutkan oleh Edward Butler, seorang penemu asal Inggris. Butler membuat kendaraan roda tiga dengan suatu motor melalui pembakaran dalam. Sejak penemuan tersebut, semakin banyak dilakukan percobaan untuk membuat motor dan mobil (Yori Herwangi¹, Ibnu Syabri², & Iwan Kustiwan², 2015).

2.2 Sistem Pakar

Aplikasi berbasis komputer yang banyak dipergunakan dalam penyelesaian permasalahan yang berkaitan dengan pemikiran ataupun keahlian seorang pakar di

sebut dengan Sistem pakar, yang mencoba dalam memecahkan masalah yang tidak dapat diselesaikan awam dan hanya bisa diselesaikan oleh seorang pakar dibidangnya, sistem pakar dipandang berhasil ketika mampu menghasilkan sebuah keputusan yang sama seperti yang dilakukan oleh pakar aslinya baik dalam sisi proses pengambilan keputusannya maupun dari hasil keputusan yang diperoleh.

Sistem pakar hadir menjadi pembantu atau asisten yang akan menuntun seseorang menyelesaikan permasalahan dengan dukungan data kepakaran yang disimpan dalam komputer. Dengan bantuan kepakaran, informasi dirangkum dalam database sebagai sumber penanganan deteksi kerusakan sampai solusi yang akan dilakukan sebagai langkah penyelesaian permasalahan (Yuliyana & Sinaga, 2019).

Istilah yang ada pada sistem pakar bersumber dari istilah *knowledge-based expert system*. Penyebab istilah ini muncul adalah untuk memecahkan sebuah masalah yang jarang dapat diselesaikan oleh awam. (Ongko, 2013).

Pada sistem pakar ada dua bagian penting, yaitu *development environment* (lingkungan pengembangan) dan *consultation environment* (lingkungan konsultasi). Lingkungan pengembangan dipergunakan oleh pembuat dan perancang sistem pakar dalam membangun komponen-komponennya dan memperkenalkan pengetahuan kedalam sebuah *knowledge base* (basis pengetahuan).

Pengetahuan adalah informasi atau maklumat yang diketahui atau disadari oleh seseorang. Pengetahuan termasuk, tetapi tidak dibatasi pada deskripsi, hipotesis, konsep, teori, prinsip (Tamando, 2018).

Pengetahuan merupakan suatu saringan atau inti sari dari informasi. Pengetahuan diklasifikasikan menjadi :

1. Pengetahuan procedural (*procedural knowledge*), lebih menekankan bagaimana melakukan sesuatu.
2. Pengetahuan deklaratif (*declarative knowledge*), menjawab pertanyaan dengan jawaban yang bernilai salah atau benar.
3. Pengetahuan tacit (*tacit knowledge*), pengetahuan yang tidak bisa dijelaskan dengan bahasa.

Representasi Pengetahuan adalah sebuah cara yang dipergunakan dalam melakukan pengkodekan pengetahuan kedalam sebuah sistem pakar. Representasi pengetahuan dimaksudkan untuk menangkap seluruh sifat-sifat penting dalam permasalahan yang akan diselesaikan dan membuat informasi itu dapat digunakan oleh prosedur pemecahan masalah. Berikut ini adalah karakteristik dari representasi pengetahuan dalam sistem pakar:

1. Sistem harus dapat diprogram menggunakan bahasa pemrograman dan hasil dari proses tersebut disimpan ke dalam memori.
2. Dirancang dengan sedemikianrupa sehingga isinya bisa dipergunakan pada proses penalaran.

Model representasi pengetahuan merupakan suatu struktur data komputer yang bisa dimanipulasi oleh sebuah Mesin inferensi dan melakukan pencarian dalam aktivitas pencocokan pola- pola kasus yang ditangani sistem.

Mesin Inferensi merupakan sebuah otak dari aplikasi sistem pakar. Dimana dalam mesin inferensi inilah kemampuan pakar ini disisipkan. Apa yang dikerjakan oleh mesin inferensi, didasarkan pada pengetahuan- pengetahuan yang ada dalam basis pengetahuan yang telah diambil dari seorang pakar (Sihotang, Sistem Pakar

Mendiagnosa Penyakit Kolesterol Pada Remaja Dengan Metode Certainty Factor (Cf) Berbasis Web, 2014).

Inferensi adalah sebuah prosedur (program) yang mempunyai kemampuan dan melakukan penalaran. Inferensi ditampilkan kedalam suatu komponen yang disebut Mesin inferensi yang mencakup prosedur-prosedur mengenai pemecahan masalah. Semua pengetahuan yang dimiliki seorang pakar disimpan pada basis pengetahuan oleh sistem pakar. Tugas Mesin inferensi adalah mengambil kesimpulan berdasarkan basis pengetahuan yang dimilikinya. Ada dua algoritma yang penting dalam sistem pakar yaitu :

1. Runut maju (*forward chaining*) adalah teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian IF dari *rules IF-THEN*.
2. Runut balik (*backward chaining*) adalah algoritma inferensi yang bekerja mundur kearah kondisi awal. Proses diawali dari *Goal* (yang berada dibagian *THEN* dari *rule IF-THEN*), kemudian pencarian mulai dijalankan untuk mencocokkan apakah fakta-fakta yang ada cocok dengan premis-premis di bagian *IF*. (Ramadhan Nasution, 2017).

Komponen pada sistem pakar disusun dengan berdasarkan dua bagian utama, di antaranya:

1. Lingkungan konsultasi (*Consultation Environment*)
2. Lingkungan pengembangan (*development Environment*)

Lingkungan konsultasi dipakai oleh pengguna yang bukanlah pakar dengan maksud untuk memperoleh pengetahuan pakar, sedangkan lingkungan pengembangan sistem pakar dipakai untuk memasukkan pengetahuan pakar pada

lingkungan sistem pakar tersebut. Berikut adalah komponen-komponen yang terdapat pada arsitektur atau struktur sistem pakar.

1. Akuisisi pengetahuan

akuisisi pengetahuan adalah transfer, transformasi keahlian, dan akumulasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan pada program komputer. Dalam tahap ini juga Knowledge Engineer akan berusaha menyerap pengetahuan yang di mana selanjutnya akan di transfer dalam basis pengetahuan.

2. Mesin/Motor Inferensi

komponen ini adalah komponen yang mengandung sebuah mekanisme pola pikir serta penalaran yang dipakai oleh pakar untuk menyelesaikan suatu masalah. Mesin inferensi adalah sebuah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang telah ada dalam bentuk basis pengetahuan dan dalam workplace, serta untuk memformulasikan kesimpulan.

3. Perbaikan Pengetahuan

Pakar tentunya memiliki kemampuan untuk meningkatkan kinerjanya dan menganalisis serta sebuah kemampuan untuk belajar dari kinerjanya. Dan kemampuan tersebut lah yang terpenting untuk pembelajaran ter-komputerisasi.

4. Antarmuka Pengguna

Antarmuka pengguna merupakan sebuah mekanisme yang dipakai oleh seorang pengguna dan juga sistem pakar untuk berkomunikasi. Antarmuka akan menerima informasi dari pengguna lalu mengubahnya ke dalam bentuk yang bisa diterima oleh sistem.

5. Basis Pengetahuan

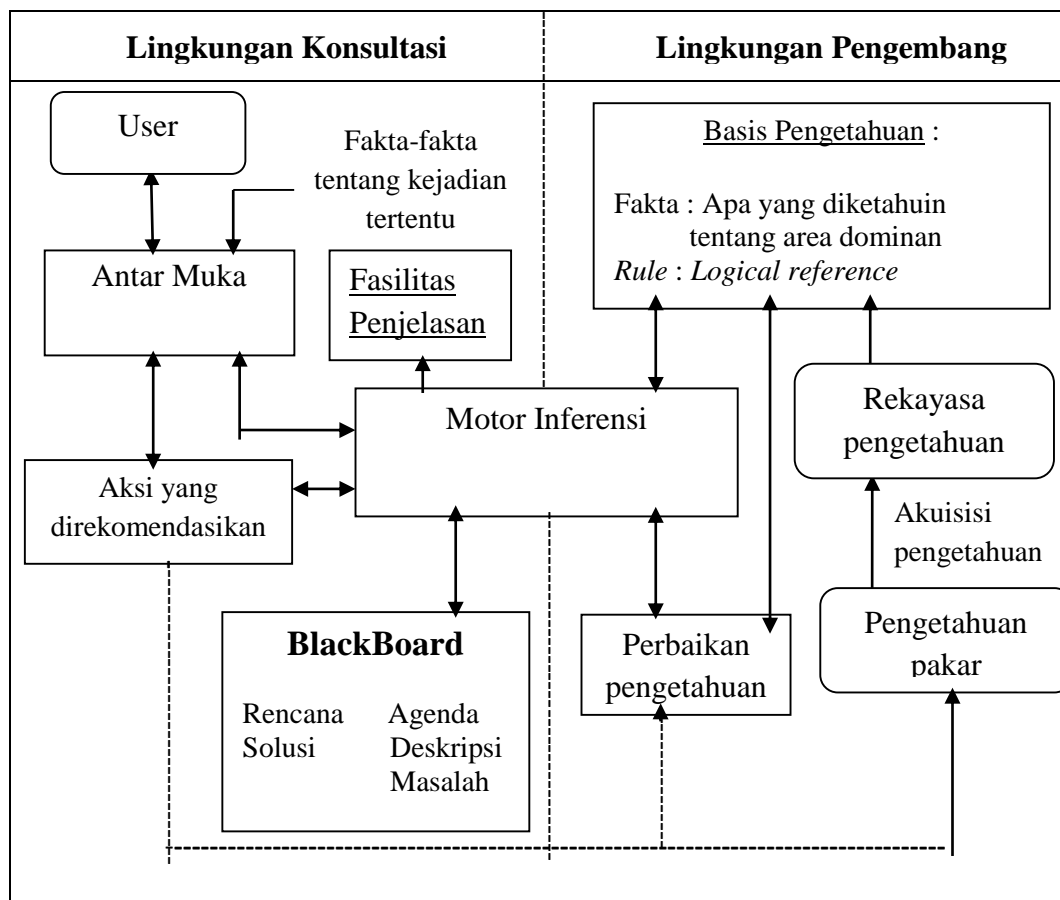
basis pengetahuan adalah komponen yang mengandung pengetahuan untuk formulasi, penyelesaian masalah, dan juga pemahaman.

6. Fasilitas Penjelasan

Fasilitas penjelasan adalah sebuah komponen tambahan yang di mana akan meningkatkan sebuah kemampuan khusus sistem pakar. Fasilitas penjelasan ini digunakan untuk memberikan penjelasan dan melacak respon tentang kelakuan dari sistem pakar secara interaktif.

7. Workplace/Blackboard

Workplace atau lembar kerja adalah area dari sekumpulan memori kerja, yang di mana di pakai untuk merekam kejadian yang sedang berlangsung.



Gambar 2.1 Komponen Sistem Pakar (Level Perdana, 2018)

2.3 *K-Nearest Neighbor (KNN)*

K-Nearest Neighbor (K-NN) adalah suatu algoritma yang menggunakan algoritma *supervised* dimana hasil dari sampel uji yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada K-NN. Tujuan dari algoritma ini adalah mengklasifikasi objek baru berdasarkan atribut dan sampel latih .

Pengklasifikasian tidak menggunakan model apapun untuk dicocokkan dan hanya berdasarkan pada memori. Diberikan titik uji, akan ditemukan sejumlah K objek (titik *training*) yang paling dekat dengan titik uji. Klasifikasi menggunakan *voting* terbanyak di antara klasifikasi dari K objek. Algoritma K-NN menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari sampel uji yang baru. Dekat atau jauhnya tetangga biasanya dihitung berdasarkan jarak *Euclidian* (Farokhah, 2020).

Langkah-langkah untuk menghitung algoritma *K-Nearest Neighbor* sebagai berikut :

1. Menentukan parameter K (jumlah tetangga paling dekat).
2. Menghitung kuadrat jarak Euclid (*query instance*) masing-masing objek terhadap data sampel yang diberikan dengan persamaan.

$$\text{Jarak} = \frac{(S1*A) + (S2*B) + (S3*C) + (S4*D) + (S5*E) + (S6*F) + \dots}{(A + B + C + D + E + F)} \dots \dots \dots (2.1)$$

S1 Sampai S6 = Nilai *Similarity*

A Sampai F = Bobot

Nilai dari *similarity* dikali dengan bobot dan hasil dari setiap perkalian di jumlahkan, lalu dibagi dari hasil penjumlahan nilai bobot.

3. Kemudian mengurutkan objek-objek tersebut ke dalam kelompok yang mempunyai jarak *Euclid* terkecil.
4. Mengumpulkan kategori Y (klasifikasi *nearest neighbor*).

Secara umum untuk mendefinisikan jarak antara dua objek x dan y , digunakan rumus jarak *Euclidean* adalah sebagai berikut:

$$ND = \sum \frac{(\text{Nilai_Kondisi} * \text{Nilai_Bobot})}{\text{Total_Nilai_Bobot}} \dots\dots\dots (2.2)$$

ND = Nilai *Distance* = Jarak Kedekatan

Pada *fase training*, *algoritma* ini hanya melakukan penyimpanan *vektor-vektor* fitur dan klasifikasi data *training sample*. Pada fase klasifikasi, fitur-fitur yang sama dihitung untuk *testing* data (yang klasifikasinya tidak diketahui). Jarak dari vektor baru yang ini terhadap seluruh vektor *training sample* dihitung dan sejumlah *k* buah yang paling dekat diambil (Shidiq, 2021).

2.4 Pemodelan UML (*Unified Modelling Language*)

Pada perkembangan teknologi perangkat lunak, diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak (Suendri, 2018). Seperti yang kita ketahui bahwa menyatukan banyak kepala untuk menceritakan sebuah ide dengan tujuan untuk memahami hal yang sama tidaklah mudah. Oleh karena itu, diperlukan sebuah bahasa pemodelan perangkat lunak yang dapat dimengerti oleh banyak orang (Aprianti, et al., 2016).



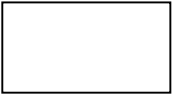
Unified Modelling Language (UML) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek. Abstraksi konsep dasar UML terdiri dari *structural classification*, *dynamic behavior*, dan *model management* dapat kita pahami main *concepts* sebagai *term* yang akan muncul pada saat membuat *diagram* dan *view* adalah kategori dari diagram tersebut. (Sutejo, 2016).



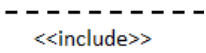
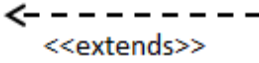
2.4.1 Use Case Diagram

Use Case atau diagram *use case* merupakan pemodelan yang digunakan untuk menggambarkan sebuah kasus interaksi antara aktor dan sistem meliputi apa yang dapat dilakukan seorang pengguna terhadap sistem yang dijalankan.

Suatu *Use Case Diagram* akan ditujukan untuk menyatakan visualisasi interaksi yang terjadi diantara pengguna (aktor) dengan sistem. Ada 2 elemen penting yang harus digambarkan, yaitu aktor dan *use case*. Berikut ini adalah simbol simbol *Use Case Diagram* (Hendini, 2016):

Tabel 2.1 Simbol *Use Case Diagram*








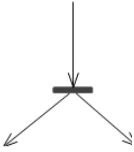

Gambar	Keterangan
	Use Case menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktif, yang dinyatakan dengan menggunakan kata kerja
	<i>Actor</i> atau Aktor adalah <i>Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi sikan aktif, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks
	<i>Boundary System</i> . Menggambarkan batasan antara sistem dengan aktor

	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau Syarat terpenuhi

2.4.2 Activity Diagram

Diagram ini menggambarkan berbagai aktifitas dalam sistem yang sedang dirancang, mulai dari titik awal, melalui kondisi (*decision*) yang mungkin terjadi, kemudian sampai pada titik akhir. *Diagram* ini tidak menggambarkan perilaku/proses *internal* sebuah sistem maupun interaksi antar subsistem, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas secara umum atau global (Hendini, 2016).

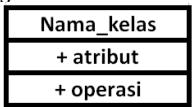

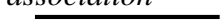
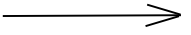
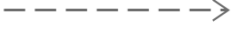
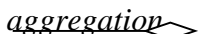
Tabel 2.2 Simbol *Activity Diagram*

Simbol	Dekripsi
Status awal 	Start Point, diletakkan pada pojok kiri atas dan merupakan awal aktivitas
Aktivitas 	<i>Activities</i> , menggambar kan suatu proses/kegiatan bisnis
Flow 	<i>Flow</i> , menggambarkan arah dari aktifitas
Percabangan / <i>decision</i> 	<i>Decision Points</i> , menggambar kan pilihan untuk pengambilan keputusan, true atau false
Penggabungan / <i>join</i> 	asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjaadi satu
Status akhir 	<i>End Point</i> , akhir aktivitas
<i>Swimlane</i>  Atau	memisahkan aktivitas yang bertanggung jawab terhadap aktivitas yang terjadi.
	<i>Fork</i> /percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi

2.4.3 Class Diagram

Diagram kelas atau *class diagram* adalah diagram yang menggambarkan struktur yang berjalan pada sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan algoritma atau operasi (Rosa, 2017).

Tabel 2.3 Simbol *Class Diagram*

Simbol	Dekripsi
Kelas 	kelas pada struktur <i>system</i>
Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi / <i>association</i> 	relasi antarkelas meliputi makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah / <i>directed association</i> 	relasi antarkelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
kebergantungan / <i>dependency</i> 	relasi antarkelas meliputi makna kebergantungan antarkelas
Agregasi / <i>aggregation</i> 	relasi antarkelas meliputi makna semua-bagian (<i>whole-part</i>)

Class Diagram secara khas meliputi : Kelas (*Class*), Relasi *Associations*, *Generalisation* dan *Aggregation*, atribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality* (Hendini, 2016).

Tabel 2.4 Simbol *Class Diagram*

<i>Multiplicity</i>	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimal 4

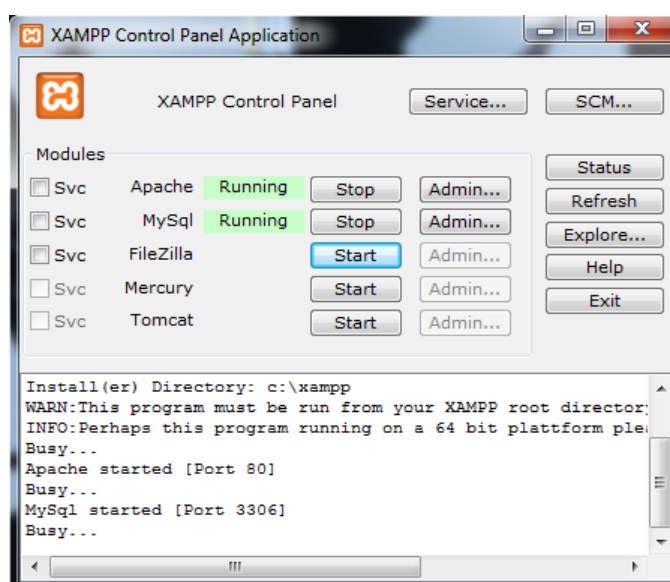
2.5 Aplikasi Pengembangan Sistem

Adapun aplikasi yang digunakan dalam membangun aplikasi sistem pakar ini adalah sebagai berikut:

2.5.1 XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. 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 (Palit, 2015). *XAMPP* adalah salah satu paket *software web server* yang didalamnya telah terdapat *Apache*, *MySQL*, *PHP*, dan *PHPMyAdmin*. Proses instalasi *XAMPP* sangat mudah, karena tidak perlu melakukan konfigurasi *Apache*, *PHP* dan *MySQL* secara manual, *XAMPP* melakukan instalasi dan konfigurasi secara otomatis. *XAMPP* adalah salah satu paket instalasi *Apache*, *PHP*, dan *MySQL* secara instant yang dapat digunakan untuk membantu proses instalasi ketiga produk tersebut”.



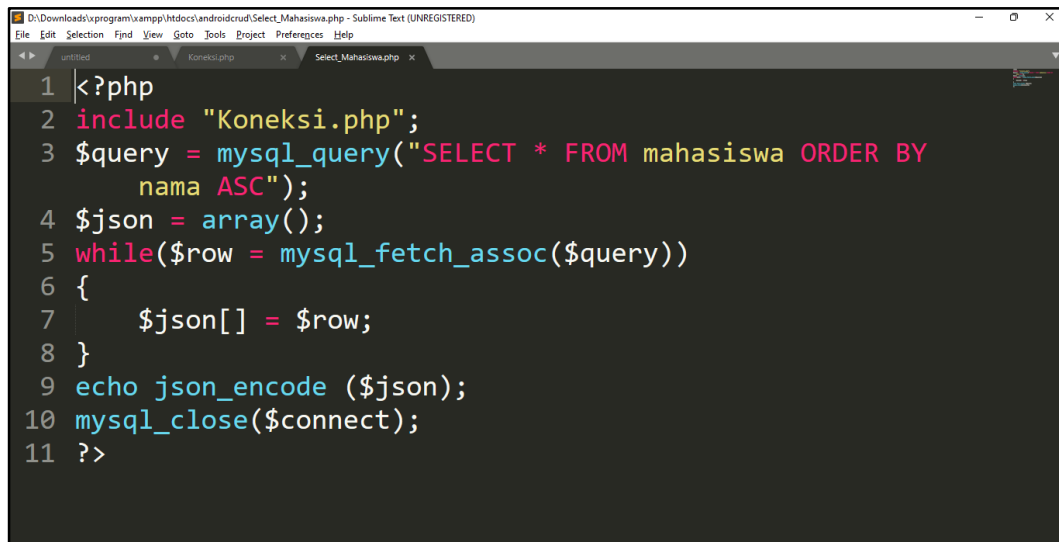
Gambar 2.2 Tampilan Untuk Mengaktifkan *XAMPP*

2.5.2 *Sublime Text*

Sublime text adalah *text editor* berbasis *Python*, sebuah *text editor* yang elegan, kaya fitur, *cross platform*, mudah dan sederhana yang cukup terkenal dikalangan *developer* (pengembang) dan desainer. (Pradiatiningtyas, 2017).

Sublime text adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai *platform operating system* dengan menggunakan teknologi *Phyton API*. Terciptanya aplikasi ini terinspirasi dari aplikasi *Vim*, Aplikasi ini sangatlah

flexibel dan *powerfull*. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan *sublime-packages*. Berikut ini adalah tampilan dari *Sublime text* :



```

1 | <?php
2 | include "Koneksi.php";
3 | $query = mysql_query("SELECT * FROM mahasiswa ORDER BY
   |     nama ASC");
4 | $json = array();
5 | while($row = mysql_fetch_assoc($query))
6 | {
7 |     $json[] = $row;
8 | }
9 | echo json_encode ($json);
10 | mysql_close($connect);
11 | ?>

```

Gambar 2.3 Tampilan *Sublime text*

2.5.3 Bahasa Pemrograman

PHP atau kependekan dari *Hypertext Preprocessor* adalah salah satu bahasa pemrograman *open source* yang sangat cocok atau dikhususkan untuk pengembangan web dan dapat ditanamkan pada sebuah skripsi HTML. Bahasa *PHP* dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti C, Java, dan *Perl* serta mudah untuk dipelajari (Firman, 2016). *PHP* adalah pemrograman *interpreter* yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti computer secara langsung pada saat baris kode dijalankan. *PHP* disebut sebagai pemrograman *Server Side Programming*, hal ini dikarenakan seluruh prosesnya dijalankan pada *server*. Kelebihan *PHP* dari Bahasa pemrograman lain :

1. Bahasa pemrograman *PHP* adalah sebuah Bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. *Web Server* yang mendukung *PHP* dapat ditemukan di mana-mana dari mulai IIS sampai dengan *Apache*, dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan developer yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, *PHP* adalah Bahasa scripting yang paling mudah karena referensi yang banyak.

PHP adalah Bahasa *open source* yang dapat digunakan diberbagai mesin (*linux, unix, windows*) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah sistem.

PHP mendukung berbagai jenis variabel yaitu :

1. *Integer* (bilangan bulat).
2. Bilangan *floating point* (presisi tunggal, ganda).
3. *Boolean*.
4. *Null* (untuk variabel yang belum diset).
5. *String* dan *Array*.
6. *Object*.
7. *Resource*

2.5.4 Database

SQL (*Structured Query Language*) adalah bahasa non procedural untuk mengakses data pada *database* relasional. SQL adalah bahasa *database* yang dipergunakan dalam menyelesaikan permasalahan dalam *database* serta mempunyai kelebihan dalam mengolah data. Standar SQL didefenisikan oleh ISO

(*International Standards Organization*) dan ANSI (*the American National Standards Institute*) yang dikenal dengan sebutan SQL86 (Eka Iswandy, 2015).

MySQL (My Structured Query Language) adalah Suatu sistem basis data *relation* atau *Relational Database managemnt System (RDBMS)* yang mampu bekerja secara cepat dan mudah digunakan *MySQL* juga merupakan program pengakses *database* yang bersifat jaringan, sehingga sapat digunakan untuk aplikasi multi user (banyak pengguna). *MySQL* didistribusikan gratis dibawah lisensi GPL (*General Public License*). Dimana setiap program bebas menggunakan *MySQL* namun tidak bisa dijadikan produk turunan yang dijadikan *closed source* atau komersia (Destiningrum & Adrian, 2017).



Gambar 2.4 Logo *MySQL*

MySQL atau dibaca “*My Sekuel*” dengan adalah suatu RDBMS (*relational Database Management System*) yaitu aplikasi sitem yang menjalankan fungsi pengolahan data. *MySQL* pertama dikembangkan oleh *MySQL AB* yang kemudian diakuisisi *Sun Microsystem* dan terakhir dikelola oleh *Oracle Corporation*.

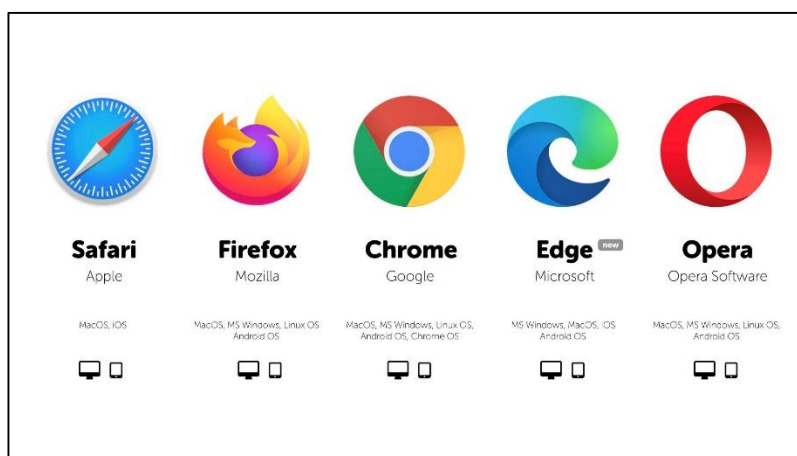
Data Definition Language (DDL) adalah suatu tata bahasa definisi data pada *MySQL*. DDL digunakan untuk mendefinisikan suatu *database, table, tablespace, logfile group, server, index*. DDL pada umumnya digunakan untuk mendefinisikan suatu wadah data atau *record*. DDL terdiri dari *create, alter, drop, rename*.

Data Manipulation Language (DML) merupakan sebuah tata bahasa manipulasi data pada *MySQL*, DML digunakan untuk memanipulasi data pada *TABLE Database*. *MySQL Utility Statement* adalah suatu tata bahasa tambahan data pada *MySQL*. *MySQL Utility Statement* digunakan untuk menampilkan informasi *database* dan *table*.

Stored Procedure merupakan sebuah fungsi yang tersimpan pada *database* yang berisi kumpulan- kumpulan perintah dalam melakukan pengolahan data yang dapat dijalankan dengan menggunakan program maupun sesama fungsi dalam *database*. *Stored Procedure* selain memiliki kemampuan pengolahan yang lebih cepat, juga berfungsi untuk memudahkan interaksi antara program dan *database*.

2.5.5 Web Browser

World Wide Web (WWW) adalah layanan internet yang paling populer. WWW berjalan dengan *hypertext Transfer Protocol* (HTTP). Halaman web merupakan file teks murni (*plain text*) yang berisi sintaks-sintaks HTML yang dapat dibuka, dilihat, diterjemahkan dengan internet *browser*.



Gambar 2.5 Web *Browser* Populer

Browser web adalah *software* yang digunakan untuk menampilkan informasi dari *server web*. *Software* ini kini telah dikembangkan dengan menggunakan *user interface grafis* sehingga pemakai dapat melakukan '*point and click*' untuk pindah antardokumen. Contohnya :

1. *Mozilla Firefox*
2. *Google Chrome*
3. *Opera*
4. *Internet Explorer*
5. *Safari*

Berdasarkan sifatnya, jenis-jenis website terbagi sebagai berikut :

1. Website Dinamis

Website yang menyediakan isi (*content*) yang selalu berubah-ubah setiap saat.

2. Website Statis

Website yang isinya sangat jarang diubah dan tidak menggunakan *database*.

Contoh website statis adalah web profil organisasi.

Berdasarkan tujuannya, jenis-jenis website terbagi sebagai berikut :

1. *Personal Website* : website yang berisi informasi pribadi seseorang.
2. *Corporate Website* : website yang dimiliki oleh sebuah perusahaan.
3. *Portal Website* : website yang mempunyai banyak layanan, mulai dari layanan berita, email, dan jasa-jasa lainnya.
4. *Forum Website* : sebuah web yang bertujuan sebagai media diskusi dan berinteraksi bersama pengguna, contohnya kaskus.
5. *E-Commerce Website* : web yang dibuat dengan tujuan bisnis penjualan melalui web tersebut, contohnya Toko bagus, e-buy, dan tokopedia.

2.6 Penelitian Terdahulu

Penelitian terdahulu adalah upaya peneliti untuk mencari perbandingan dan selanjutnya untuk menemukan inspirasi baru untuk peneltiain selanjutnya di samping itu kajian terdahulu membantu penelitian dapat memposisikan penelitian serta menunjukkan orsinalitas dari penelitian. Penelitian sejenis yang digunakan peneliti sebagai bahan referensi dalam melakukan penelitian, sebagai berikut:

Tabel 2.5 Penelitian Terdahulu

No.	Judul	Peneliti	Tahun	Hasil Penelitian
1	Implementasi Metode K-Nn Untuk Deteks Kerusakan Sepeda motor Honda Beat	Firman Dwi Wahyudi , Paulus Harsadi , Dwi Remawati	2018	Tercapainya tujuan untuk merancang, membangun, dan mengimplementasikan sistem pakar deteksi kerusakan sepeda motor Honda Beat yang sulit dideteksi kerusakannya dan penanganannya sehingga harus dibawa ke bengkel servis sepeda motor Honda Beat terlebih dahulu dan memakan waktu yang relatif lama. Dengan aplikasi sistem pakar ini deteksi kerusakan sepeda motor Honda Beat dapat ditangani secara tepat dan cepat dengan metode K-Nearest Neighbour
2	Sistem Pakar Kerusakan Liver Menggunakan K- Nearest Neighbors Algoritm Berbasis Website	Citra Yustitya Gobel	2018	Berdasarkan hasil penelitian dan dari hasil analisis, perancangan sistem, pembuatan program sampai tahap penyelesaian program, maka penulis dapat mengambil kesimpulan sebagai bahwa Aplikasi Sistem Pakar Kerusakan Liver Menggunakan Metode K-Nearest

				<p>Neighbors Algorithm dapat membantu dalam proses diagnosa kerusakan Liver berdasarkan gejala-gejala yang dirasakan oleh penderita dan terdapat dalam system sesuai dengan tujuan penelitian. Sedangkan proses Perhitungan Sistem Pakar diagnosa Kerusakan Liver ini Menggunakan Metode Klasifikasi K-Nearest Neighbors Algorithm dihitung berdasarkan dari proses basis aturan yang telah ditentukan sebelumnya sehingga pembuatan system dianggap cukup fleksibel dan hasil sesuai yang diharapkan.</p>
3	<p>Sistem Pakar Diagnosa Kerusakan VGA dengan Metode Certainty Factor dan Algoritma K-Nearest Neighbor (K-NN)</p>	<p>Rizal Maulana Yusuf Effendi</p>	2021	<p>Sistem diagnosa untuk mendeteksi kerusakan pada VGA menggunakan Algoritma K-Nearest Neighbour sebagai machine learning dan Metode Certainty Factor sebagai media perhitungan dalam menentukan jarak dari jenis kerusakan serta memiliki saran dalam tindakan selanjutnya untuk menangani dan mencegah kerusakan tersebut terjadi dan juga memiliki kemungkinan kerusakan lain yang serupa dengan kerusakan yang dialami dapat diakses dengan cepat dan mudah untuk dipahami, dalam pembuatan</p>

				penelitian dilakukan berurutan memudahkan pengerjaan.	ilmiah secara untuk proses
--	--	--	--	---	-------------------------------------