

BAB II

KAJIAN PUSTAKA

2.1 Kecerdasan Buatan

Kecerdasan buatan adalah salah satu bidang ilmu komputer yang memungkinkan komputer untuk berperilaku cerdas seperti manusia. Ilmu Komputer ini mengembangkan perangkat lunak dan perangkat keras untuk memungkinkan komputer tersebut dapat meniru perilaku manusia (Rahim, Nurhayati and Martono, 2015).

2.2 Sistem Pakar

Sistem pakar merupakan cabang dari kecerdasan buatan. Sistem ini adalah sistem Komputer yang dapat menandingi atau meniru Kemampuan ahli untuk membuat sistem ini bekerja Transfer pengetahuan manusia ke komputer Basis pengetahuan (Basis Pengetahuan) dengan sistem inferensi untuk menunjang Fungsi dari seorang pakar dalam pemecahan suatu masalah (Fanny, Hasibuan and Buulolo, 2017).

2.3 Tujuan Sistem Pakar

Sistem pakar (expert system) sendiri merupakan sistem perangkat lunak atau program komputer yang diprogram khusus sebagai media konsultasi secara instan dan media untuk membantu dalam memecahkan suatu masalah pada bidang-bidang spesialisasi tertentu seperti sains, perekayasaan, matematika, kedokteran, pendidikan dan lain-lain. Sistem pakar termasuk himpunan bagian dari Artificial Intelligence (Arhami, 2005).

Pada dasarnya sistem pakar diimplementasikan untuk mendukung kegiatan pemecahan masalah. Beberapa kegiatan pemecahan masalah yang disebutkan adalah: (Lestari, 2012).

1. *Interpretasi.*

Membuat kesimpulan atau deskripsi dari sekumpulan data mentah. pengolahan keputusan dari hasil pencarian, termasuk impresi suara, analisis gambar, bentuk sinya, dll.

2. *Prediksi.*

Memproyeksikan akibat-akibat yang dimungkinkan dari situasi-situasi tertentu. Contoh: prediksi demografi, prediksi ekonomi, dll.

3. *Diagnosis.*

Identifikasi penyebab malfungsi dalam situasi kompleks berdasarkan gejala yang diamati dari diagnosis medis, elektronik, dan mekanis, dll.

4. *Perancangan (desain).*

Menentukan konfigurasi komponen sistem untuk tujuan kinerja tertentu yang memenuhi batasan tertentu. Contoh: tata letak sirkuit, desain bangunan.

5. *Perencanaan.*

Melakukan rencana serangkaian tindakan yang dapat mencapai serangkaian tujuan, dengan kondisi awal yang spesifik. Contoh: perencanaan keuangan, militer, dll.

6. *Monitoring.*

Membandingkan hasil pengamatan dengan kondisi yang diharapkan. Contoh: computer aided monitoring system.

7. *Debugging.*

Menentukan dan menginterpretasikan cara-cara untuk mengatasi malfungsi. Contoh: memberikan resep obat terhadap kegagalan.

8. *Instruksi.*

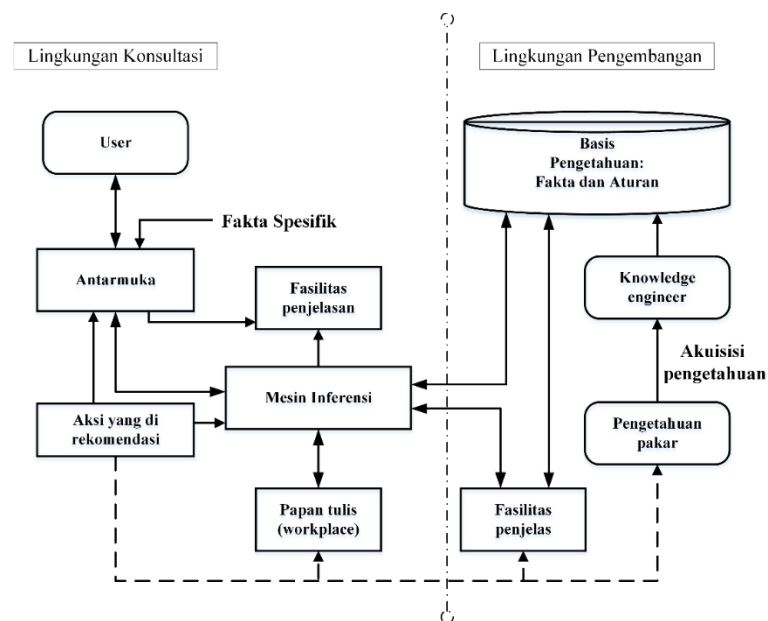
Mendeteksi dan mengoreksi defisiensi dalam pemahaman domain subjek. Contoh: menjalankan instruksi untuk analisis dan debugging.

9. Kontrol.

Mengatur tingkah laku suatu environment yang kompleks. Contoh: mengontrol interpretasi, prediksi, peningkatan, dan pemantauan perilaku sistem.

2.4 Struktur Sistem Pakar

Sistem pakar terdiri atas dua bagian utama, yakni: lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Pada lingkungan pengembangan berperan sebagai pembangun sistem pakar baik dari segi pembangunan komponen maupun basis pengetahuan. Lingkungan konsultasi dipergunakan oleh seseorang yang bukan ahli untuk berkonsultasi (Kusumadewi, 2003).

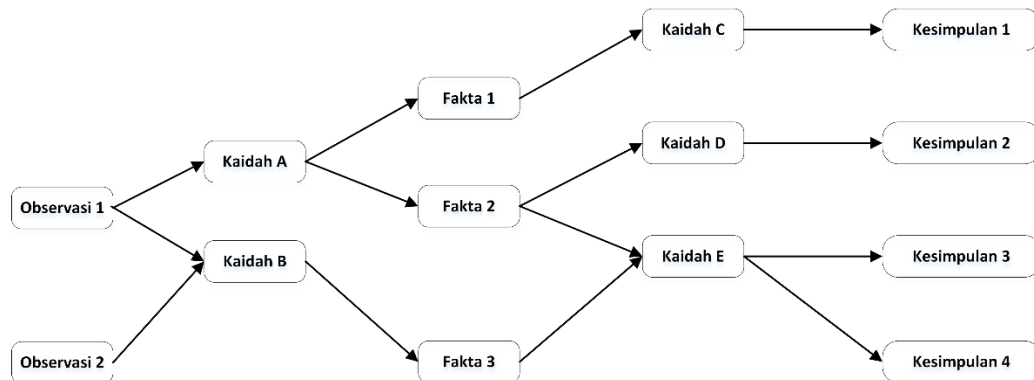


Gambar 2.1 Struktur Sistem Pakar

2.5 Metode Forward Chaining

Forward Chaining merupakan metode pencarian atau pelacakan prospektif yang dimulai dari isu yang terdapat dan menggabungkan *rule* dengan tujuan mencapai suatu konklusi atau tujuan (Pondaag *et al.*, 2015). Pelacakan maju ini sangat baik apabila bekerja dengan perkara yg dimulai dengan rekaman informasi awal dan ingin dicapai penyelesaian akhir, sebab semua proses akan dikerjakan secara berurutan

maju. Langkah inferensi yg dimulai dengan sekumpulan fakta pengetahuan, memperoleh fakta baru memakai rule dimana premis-premis sesuai menggunakan fakta-informasi pengetahuan, dan meneruskan prosesnya sampai sebuah tujuan yg ditetapkan sudah tercapai. Berikut merupakan diagram pelacakan kedepan untuk menghasilkan sebuah goal yang dapat dilihat pada Gambar 2.2 dibawah ini.



Gambar 2.2 Metode Forward Chaining

2.6 Model Riasec

Realistic, Investigative, Artistic, Social, Enterprising, dan Conventional (RIASEC) adalah sebuah model pemetaan minat bakat seseorang ke dalam 6 kriteria. Proses identifikasi pendeteksian minat dan bakat dilakukan dengan mengisi kuisisioner berisikan ciri-ciri atau karakter yang ada di model *RIASEC*. Output yang dihasilkan yaitu nilai tertinggi dari kriteria yang dipilih dan menjadi rekomendasi dari minat dan bakat (Hidayat & Wahyuni, 2019).

Berikut ciri-ciri dari kepribadian dari tiap tipe *RIASEC* :

1. *Realistic*

Individu dengan kepribadian tersebut lebih senang berinteraksi dengan dunianya melalui aktualisasi diri artistik, cenderung menghindari situasi yang konvensional serta interpersonal. Individu ini lebih baik berorientasi kepada karis contohnya seni dan menulis.

Berikut ciri-ciri dari kepribadian ini adalah:

- 1) Kemampuan mekanikal, psikomotor, dan atletik yang baik.

- 2) Jujur
- 3) Setia
- 4) Suka kegiatan-kegiatan diluar
- 5) Senang bekerja dengan mesin, alat-alat, tumbuhan, dan hewan.
- 6) Senang dengan kegiatan-kegiatan yang bersifat fisik.
- 7) Menyukai kegiatan kongkrit.
- 8) Cenderung kurang suka bersosialisai.
- 9) kurang menyukai hal-hal yang kompleks (lebih menyukai kesederhanaan)
- 10) Individu dengan kepribadian tersebut direkomendasikan memilih karir-karir yang bersifat praktis, seperti pegawai pabrik, bertani, supir , teknisi, dan konstruksi.

2. Investigative

Individu dengan kepribadian ini lebih berorientasi pada konsep dan teori. Mereka lebih berperan sebagai pemikir daripada sebagai pelaksana. Mereka sering menghindari adanya hubungan interpersonal dan lebih sesuai dengan karir-karir yang berkaitan dengan matematika dan pengetahuan.

Adapun ciri-cirinya adalah:

- 1) Kemampuan memecahkan masalah dan analitis yang baik.
- 2) Cenderung berfikir matematis.
- 3) Senang mengobservasi, mendalami, dan mengevaluasi.
- 4) Lebih suka bekerja sendiri.
- 5) Pemberi ide.
- 6) Kritis, hati-hati, dan selalu ingin tahu lebih
- 7) Suka kedisiplinan.
- 8) Berorientasi tugas.
- 9) Sistematis

3. *Artistic*

Tipe kepribadian ini menghargai orang lain yang ekspresif dan mandiri. Secara alami mereka mengagumi seni kreatif termasuk musik dan tulisan. Individu tipe ini melihat diri mereka ekspresif dan orisinal dan lebih suka menghindari aktifitas yang berulang atau rutin.

Adapun ciri-cirinya adalah:

- 1) Berpikir abstrak.
- 2) Menyukai estetika (keindahan).
- 3) Orang yang kreatif, suka hal-hal kompleks, emosional, intuitif, ideal.
- 4) Lebih menyukai bekerja secara mandiri.
- 5) Suka menyanyi, berakting, menulis, dan melukis.
- 6) Imaginatif.
- 7) Kurang menyukai hal-hal yang konvensional.
- 8) Tidak dapat diduga.
- 9) Tidak suka keteraturan.

4. *Social*

Individu ini memiliki kemampuan lisan serta korelasi interpersonal yang baik. mereka lebih cocok memasuki dunia karir yg bekerjasama dengan manusia seperti mengajar, pekerja sosial, konseling, serta semacamnya.

Adapun ciri-cirinya adalah:

- 1) Komunikatif.
- 2) Bersahabat, mudah bergaul.
- 3) Suka memberi dan membantu.
- 4) Baik, impulsive.
- 5) Bertanggung jawab.
- 6) Berjiwa kelompok.
- 7) Mempunyai toleransi yang cukup baik.
- 8) Dapat memahami.
- 9) Kemampuan verbal dan personal yang baik.

5. *Enterprising*

Individu ini menggunakan kemampuan verbalnya untuk memimpin orang lain, mendominasi individu, dan menjual produk atau hal yang lain. Individu ini lebih sesuai dalam karir seperti sales, bidang politik, dan manajemen.

- 1) Adapun ciri-ciri kepribadian ini adalah:
- 2) Percaya diri, bersikap asertif.
- 3) Mudah beradaptasi.
- 4) Ambisius.
- 5) Kemampuan berbicara dan memimpin yang baik.
- 6) Suka menggunakan pengaruh seseorang.
- 7) Kemampuan interpersonal yang cukup baik.
- 8) Penuh energi, ekstrovert, optimis, persuasif.
- 9) Suka mengambil resiko, spontan.
- 10) Suka mengontrol.

6. *Conventional*

Individu ini lebih menyukai kegiatan yang bersifat terstruktur. Mereka lebih suka pekerjaan yang berupa bawahan seperti pegawai bank, sekretaris, dan petugas arsip.

Adapun ciri-ciri kepribadian ini adalah:

- 1) Tergantung pada orang lain.
- 2) Tidak kreatif.
- 3) Suka kedisiplinan dan ketepatan.
- 4) Suka memperhatikan detail.
- 5) Efisien.
- 6) Melaksanakan tugas secara teratur.
- 7) Kemampuan klerikel dan numerical yang baik.
- 8) Terorganisir.
- 9) Bersifat tradisional

2.7 Penelitian Terdahulu

Penelitian terdahulu ini menjadi salah satu acuan penulis dalam melakukan penelitian sehingga penulis dapat memperkaya teori yang digunakan dalam mengkaji penelitian yang dilakukan. Berikut merupakan penelitian terdahulu berupa beberapa jurnal terkait dengan penelitian yang dilakukan penulis.

Table 2.1 Penelitian Terdahulu

Nama Peneliti	Judul Penelitian	Hasil Penelitian
(Hidayat & Wahyuni, 2019)	Pendeteksian Minat Dan Bakat Menggunakan Metode Riasec	Pembuatan sistem pakar pendeteksian minat dan bakat dengan metode <i>RIASEC</i> . Kemudian Teorema dan probabilitas di gunakan untuk validasi nilai perbandingan variable.kemudian di implementasikan kedalam sebuah sistem pakar berbasis <i>website</i> .
Perbedaan : penelitian yang dilakukan Hidayat, Wahyuni, metode yang di gunakan hanya menggunakan metode <i>RIASEC</i> dalam pendeteksian minat dan bakat sedangkan peneliti mengkombinasikan metode <i>Forward Chaining</i> dan <i>RIASEC</i> .		

Nama Peneliti	Judul Penelitian	Hasil Penelitian
(Rahim, 2015)	Sistem Cerdas Peminatan Program studi Di Universitas Diponegoro Bagi Siswa Menengah Atas	Pembuatan sistem cerdas Peminatan Program studi Di Universitas Diponegoro Bagi Siswa Menengah Atas. Metode <i>Forward Chaining</i> di gunakan untuk

		memprediksi dan mencari solusi berdasarkan aturan yang premisnya cocok.kemudian di implementasikan kedalam sebuah sistem cerdas berbasis desktop.
<p>Perbedaan : penelitian yang dilakukan Rahim, Nurhayati, Martono metode yang di gunakan hanya menggunakan metode <i>Forward Chaining</i> dan menentukan pertanyaan quesioner berdasarkan ide mereka dalam memecahkan masalah, sedangkan peneliti mengkombinasikan metode <i>Forward Chaining</i> yang di masukan kumpulan pertanyaan berdasarkan ciri kepribadian <i>RIASEC</i> sebagai variable sehingga memperkuat standar pertanyaa dan argumen sistem.</p>		

Nama Peneliti	Judul Penelitian	Hasil Penelitian
(Parastowo, 2021)	Implementasi Metode Simple Additive Weighting Dan Riasec Pada Rancang Bangun Aplikasi Pemilihan Bidang Ukm Terbaik	Hasil penelitian menunjukkan bahwa metode <i>RIASEC</i> dan SAW berhasil diterapkan pada aplikasi pemilihan bidang UKM terbaik.
<p>Perbedaan : penelitian yang dilakukan parastowo, dalam menentukan minat dan bakat menggunakan metode <i>RIASEC</i>. Kemudian dalam menentukan sebuah alternatif pilihan terbaik dalam kasus ini yaitu pilihan UKM menggunakan <i>Simple Additive Weighting (SAW)</i>, sedangkan peneliti mengkombinasikan antara metode <i>Forward Chaining</i> dengan <i>RIASEC</i> untuk menentukan rekomendasi program studi yang ada di Kampus IIB Darmajaya.</p>		

2.8 Sistem

Sistem adalah sekelompok komponen dan elemen yang saling bergabung bersama untuk meraih suatu tujuan tertentu. Kata sistem berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*systema*) Suatu entitas yang terdiri dari bagian-bagian atau elemen yang saling berhubungan dengan aliran informasi, material, atau sumber daya energi untuk mencapai suatu tujuan. Istilah ini sering digunakan untuk memetakan entitas yang berbeda. Seringkali dimungkinkan untuk berinteraksi dan membangun model matematika untuk itu. sistem Jaringan proses yang saling berhubungan, Melakukan suatu kegiatan atau tujuan tertentu (Bachry and Yuliawati, 2019).

2.9 Element Pembangun Website

CMS adalah suatu perangkat lunak yang berfungsi untuk mengelola content. maksud dari konten di sini adalah semua bentuk informasi digital, yang berupa file gambar, audio, video, teks dan format file komputer lainnya. CMS berupa aplikasi berbasis web yang diimplementasikan dalam bahasa pemrograman yang memiliki kemampuan pemrograman berbasis web (Siambaton and Fakhriza, 2016).

Menggunakan CMS ini bisa dibilang tidak perlu belajar bahasa pemrograman web, karena untuk mengatur menu tampilan dan lain-lain bisa menggunakan *tools* yang telah disediakan oleh masing-masing CMS. *Content Management System* yang ada saat ini adalah *WordPress*, *Joomla*, *Drupal*, *Magento*, *Shopify*, *Squarespace* dan masih banyak lagi. CMS yang paling populer dari beberapa CMS di atas adalah *WordPress* dan *Joomla*.

2.10 Side Scripting

Side Scripting adalah salah satu metode untuk mendesain *website*. Bahasa pemrograman *website* secara umum dibagi menjadi dua bagian berdasarkan sisi pemrosesan program ketika *website* diakses. Dua bagian tersebut adalah *Client Side* dan *Server Side*. Penjelasannya adalah seperti berikut ini (Adi, 2012).

2.10.1 Client Side Scripting

Client Side atau sisi klien adalah salah satu pemrograman *web* yang pengolahan datanya diproses oleh klien. Klien yang dimaksud adalah *web browser* yang akan menerjemahkan bahasa program yang dibuat. *Web browser* yang berperan sebagai klien memiliki library yang biasanya telah dimiliki oleh setiap *web browser* seperti *Google Chrome*, *Mozilla Firefox*, *Opera Mini* dan lain-lain. *Library* yang dimaksud adalah *web engine* yang memiliki peran untuk menerjemahkan semua perintah yang ada di dalam halaman sebuah *website* yang menggunakan *client side scripting*. Masalahnya adalah dari banyaknya *web-browser* yang ada saat ini memiliki *web-engine* yang berbeda-beda sehingga terkadang akan menyebabkan tampilan layout *website* yang telah dibuat akan memiliki tampilan yang berbeda-beda di setiap *browser* yang berbeda, sehingga sangat disarankan untuk menggunakan *web-browser* yang telah mendapat sertifikasi dari *World Wide Web Consortium (W3C)*. Bahas pemrograman yang menggunakan metode *client side scripting* ini antara lain adalah HTML, CSS, JavaScript, XML dan JQUERY. Bahasa program yang akan dibuat dalam *website* dalam penelitian ini adalah HTML dan CSS.

1. *Hypertext Markup Language (HTML)*

Hypertext Markup Language (HTML) sinkron namanya artinya bahasa program web yg berfungsi menjadi markup buat website. HTML berupa kode-kode acara yang disimpan pada bentuk file berekstensi “nama_file.html”. syarat supaya website HTML ini dapat diakses menggunakan internet yaitu file-file HTML yang sudah dibuat harus diunggah ke dalam server online agar siapapun yg terhubung ke internet dapat mengaksesnya.

2. *Cascading Style Sheet (CSS)*

Cascading Style Sheets (CSS) ialah termasuk bahasa pemrograman desain web (*style sheet language*) berperan mengontrol format tampilan sebuah *page web* yang ditulis dengan menggunakan bahasa penanda (*markup language*). umumnya CSS digunakan saat mendesain sebuah halaman HTML dan XHTML, tetapi kini bahasa pemrograman CSS mampu diaplikasikan buat

segala dokumen XML, termasuk SVG dan XUL. Penggunaan CSS bisa dalam satu file HTML maupun dibuat secara terpisah. Menggunakan CSS secara terpisah bisa mengatur banyak halaman HTML hanya dengan menggunakan satu buah file CSS, sehingga perubahan pada banyak halaman bisa dilakukan secara serentak hanya dengan cara memodifikasi satu file CSS yang memang dibuat untuk halaman-halaman HTML tersebut. Menggunakan CSS ini modifikasi halaman *website* menjadi lebih mudah dan efisien serta dengan CSS ini dapat memperkecil kemungkinan terjadinya kesalahan dalam *styling website* (Ginting, 2013).

2.10.2 Server Side Scripting

Server Side Scripting atau *script* pemrograman sisi *server* adalah pemrograman web yang pengolahan aksesnya dibebankan kepada *server*. *Server* yang dimaksud adalah *web-server* yang telah terintegrasi dengan *web browser* sehingga bisa diakses menggunakan *web-browser*. *Web-server* saat ini selain yang bersifat *online* juga ada *web-server* lokal komputer atau *local host* yang bisa digunakan secara *offline*. Akan tetapi *web-server* lokal (*localhost*) hanya berperan sebagai *server* untuk komputer itu sendiri, yang berarti apabila *website* diunggah ke *web-server* lokal tersebut hanya bisa diakses menggunakan komputer itu sendiri, dengan arti lain bahwa komputer tersebut bisa memiliki peran ganda yaitu sebagai *server* dan sebagai *client* sekaligus dalam satu komputer yang sama.

1. Cloud Sever

Web-server yang bersifat *cloud server* saat ini telah banyak sekali pilihan penyedia layanan untuk menyewa *server online* dengan berbagai macam fasilitas layanan dengan tawaran harga yang bervariasi sesuai dengan masing-masing spesifikasi *server* yang ditawarkan.

2. Server Local (local host)

Web-server lokal (*localhost*) berarti *server* yang digunakan berada di dalam satu komputer dengan pembuatan program *website*. *Web-server* lokal dapat dibangun dengan aplikasi yang mendukung untuk pembuatan *web-server*

lokal di dalam komputer. Semua sistem operasi saat ini telah memiliki aplikasi khusus yang dibuat untuk masing-masing sistem operasi, yang tentunya harus di-*install* sendiri yang artinya tidak disertakan langsung dalam sistem operasi ketika sistem operasinya di-*install*. Beberapa contohnya pada sistem operasi windows dapat diinstall software WAMPP (*Windows, Apache, MariaDB, PHP, Perl*). Seperti kepanjangan dari WAMPP itu sendiri, bahwa dalam satu aplikasi ini terdapat banyak aplikasi pendukung yang ada di dalamnya. Sementara itu untuk sistem operasi linux bisa menggunakan software LAMPP (*Linux, Apache, MariaDB, PHP, Perl*) dan untuk Mac OS bisa menggunakan MAMPP (*Mac OS, Apache, MariaDB, PHP, Perl*). Sementara itu ada satu buah aplikasi untuk membangun *web-server* yang mendukung semua platform yaitu XAMPP (*Cross Platform, Apache, MariaDB, PHP, Perl*).

2.11 Personal Home page (PHP)

Aplikasi untuk pemrograman *website* bisa menggunakan bahasa program PHP (*Personal HomePage*) atau disebut juga PHP *Hypertext Preprocessor*. PHP adalah bahasa pemrograman umum dan juga bahasa pemrograman *website* yang bersifat *server side scripting* yang bekerja pada sisi *server*. Awalnya PHP merupakan bahasa program yang dibuat untuk mengelola halaman pribadi yang dikembangkan dari bahasa C. Seorang yang menemukan bahasa program ini adalah Rasmus Lerdorf pada tahun 1995 yang pada waktu itu hanya digunakan untuk mengelola halaman *website* pribadinya, sehingga bahasa program ini kemudian disebut bahasa program PHP yaitu singkatan dari Personal Home Page. Saat ini bahasa program PHP dikembangkan oleh *The PHP Group*. PHP bersifat *server side scripting* sehingga bekerja atau diproses pada *server*, berbeda dengan bahasa yang bersifat *client side scripting* seperti HTML dan JavaScript yang bekerja pada *web browser client*.

PHP dapat digunakan dengan gratis karena bersifat *open source*, sehingga siapapun bisa menggunakannya secara gratis. Penggunaan bahasa program PHP sebenarnya tidak berdiri sendiri akan tetapi biasanya disisipkan di dalam HTML (*Hypertext Markup Language*) dan untuk mempercantik tampilannya ditambahkan (*CSS Cascading Style Sheet*). *Website* yang dibuat menggunakan HTML bersifat statis

atau tetap, sementara jika menggunakan tambahan PHP di dalamnya maka akan lebih bersifat dinamis, sehingga bisa menyesuaikan dengan situasi sesuai dengan perangkat yang digunakan. Menggunakan PHP juga dapat menyimpan data ke dalam *database* dan menampilkan ke dalam *website*.

2.12 Domain Name System (DNS)

Domain name system merupakan sebuah perangkat lunak service pada internet yang menerjemahkan sebuah domain name ke *IP address* dan merupakan jenis sistem yang menaungi permintaan pemetaan *IP address* ke *FQDN (Fany Qualified Domain Name)* dan dari *FQDN* ke *IP address*. *DNS* umumnya dipergunakan di perangkat lunak yg berafiliasi ke internet sererti *Web Browser* atau *e-mail*, Dimana *DNS* membantu memetakan *host name* sebuah komputer ke *IP Address*. Selain digunakan di *internet DNS juga bisa di implementasikan ke private network* atau internet (Tedyyana *et al.*, 2016).

2.13 My SQL

MySQL merupakan sebuah perangkat lunak sistem manajemen basis data SQL(bahasa Inggris: database management system) atau DBMS yg multithread,multi-user, dengan kurang lebih enam juta instalasi di penjuru dunia. MySQLAB menghasilkan MySQL disediakan sebagai *software* gratis dibawah lisensi GNU General Public License (GPL), namun mereka juga menjual dibawah lisensi komersial untuk kasus-kasus di mana penggunaannya kurang cocok dengan penggunaan GPL (Bangsawan, 2015).

2.14 Extreme Programming

Extreme Programming adalah suatu model yang termasuk dalam pendekatan agile yang diperkenalkan oleh Kent Back. Menurut penjelasannya, definisi XP adalah sebagai berikut: “*Extreme Programming (XP)* adalah metode pengembangan software yang cepat, efisien, beresiko rendah, fleksibel, terprediksi, scientific, dan menyenangkan.“. Model ini cenderung memakai pendekatan Object-Oriented. Proses tahapan yang dilalui yaitu : Planning, Design, Coding, dan Tesing. Sasaran *Extreme*

Programming adalah tim yang dibentuk berukuran antara kecil sampai medium saja, tidak perlu menggunakan sebuah tim yang besar. Hal ini bertujuan untuk mengatasi requirements yang tidak jelas ataupun adanya perubahan-perubahan requirements yang sangat cepat. *Extreme Programming* merupakan *agile methods* yang banyak digunakan saat ini dan menjadi sebuah pendekatan yang sangat terkenal.

Menurut Prabowo dalam (Supriyatna, 2018) *Extreme Programming (XP)* adalah sebuah proses rekayasa perangkat lunak yang cenderung memakai pendekatan berorientasi objek dan metode ini memiliki sasaran yaitu tim yang dibentuk dalam skala kecil sampai medium serta metode *extreme programming* sangat sesuai diterapkan apabila tim dihadapkan dengan requirement yang tidak jelas maupun terjadi perubahan-perubahan requirement yang sangat cepat. Sedangkan menurut Ferdiana dalam (Lubis, 2016) *Extreme Programming (XP)* dikenal dengan metode atau “technical how to” bagaimana suatu tim teknis mengembangkan perangkat lunak secara efisien melalui berbagai prinsip dan teknik praktis pengembangan perangkat lunak. XP menjadi dasar bagaimana tim bekerja sehari-hari (Carolina *et al.*, 2019).

Seperti yang telah dijelaskan di atas, terdapat empat tahapan yang harus dikerjakan pada metode *extreme programming (XP)* yaitu:

1. *Planning* (Perencanaan)

Tahapan ini merupakan langkah awal dalam pembangunan sistem dimana dalam tahapan ini dilakukan beberapa kegiatan perencanaan yaitu, identifikasi permasalahan, menganalisa kebutuhan sampai dengan penetapan jadwal pelaksanaan pembangunan sistem.

2. *Design* (Perancangan)

Tahapan design merupakan tahapan perancangan dimana pada tahapan ini dilakukan aktivitas pemodelan yang dimulai dari pemodelan sistem, pemodelan arsitektur sampai dengan pemodelan basis data.

3. *Coding* (Pengkodean)

Tahapan ini merupakan kegiatan penerapan pemodelan yang sudah dibuat kedalam bentuk user interface dengan menggunakan bahasa pemrograman.

4. *Tesing* (Pengujian)

Setelah tahapan pengkodean selesai, kemudian dilakukan tahapan pengujian sistem untuk mengetahui kesalahan apa saja yang timbul saat aplikasi sedang berjalan serta mengetahui apakah sistem yang dibangun sudah sesuai dengan kebutuhan pengguna.

2.15 Object Oriented Programming

Pemrograman berorientasi objek (OOP) adalah strategi pengembangan perangkat lunak yang mengatur perangkat lunak menjadi kumpulan objek. Ini termasuk data dan operasi yang dilakukan pada objek. Singkatan dari pemrograman berorientasi objek atau OOP, sebuah paradigma atau metode pemrograman berorientasi objek..(Rais, 2019)

Dari penjelasan diatas dapat disimpulkan bahwa *Object Oriented Programming* (OOP) adalah suatu metode atau cara terbaru untuk membangun program atau merancang sebuah sistem dengan mengamati objek.

Manfaat dari penggunaan metodologi berorientasi objek adalah sebagai berikut:

1. Peningkatan produktivitas

Sebab kelas dan objek yang ditemukan pada suatu masalah masih dapat digunakan kembali untuk masalah lainnya yang melibatkan objek tersebut (reusable).

2. Kecepatan pengembangan

Karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada pengkodean.

3. Kemudahan pemeliharaan

Karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering berubah-ubah.

4. Adanya konsistensi

Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.

5. Meningkatkan kualitas perangkat lunak

Sebab pendekatan pengembangan lebih dekat dengan dunia yang konkret dan adanya konsistensi pada saat pengembangannya, *software* yang dihasilkan akan mampu memenuhi kebutuhan pemakai dan mempunyai sedikit kesalahan.

Metodologi pengembangan sistem berorientasi objek memiliki beberapa konsep dasar yang perlu dipahami:

1. Kelas (*Class*)

Kelas adalah kumpulan objek dengan karakteristik yang sama. Kelas adalah definisi statis dan set objek yang sama dapat lahir atau dibuat oleh kelas itu. Kelas memiliki properti (atribut), perilaku (operasi / metode), hubungan (hubungan) dan makna. Kelas dapat diwariskan ke kelas lain, di mana atribut dan kelas asli dapat diwariskan ke kelas baru.

2. Objek (*Object*)

Objek merupakan gagasan dan hal-hal yang mewakili dunia nyata seperti objek, orang, unit organisasi, tempat, peristiwa, struktur, status, atau lain sebagainya yang abstrak. Suatu objek adalah entitas yang mampu menyimpan informasi (status) dan memiliki operasi (behavior) yang dapat diterapkan atau dapat diartikan sebagai objek statis. Objek memiliki siklus hidup yang diciptakan, dimanipulasi dan dihancurkan.

3. Metode (*Method*)

Operasi atau metode atau metode dalam kelas hampir sama dengan fungsi atau prosedur dalam metodologi struktural. Kelas mungkin memiliki lebih dari satu metode atau operasi. Metode atau operasi yang berfungsi untuk memanipulasi objek itu sendiri. Operasi atau metode adalah fungsi atau transformasi yang dapat dilakukan pada objek atau dilakukan oleh objek.

4. Atribut (*Attribute*)

Atribut kelas adalah variabel global yang termasuk dalam kelas. Atribut dapat berupa nilai atau elemen data yang dimiliki oleh objek di kelas objek. Atribut dimiliki secara individual oleh suatu objek, misalnya berat, jenis, nama, dan sebagainya. Atribut harus bersifat pribadi untuk mempertahankan enkapsulasi.

5. Abstraksi (*Abstraction*)

Prinsip menghadirkan dunia nyata yang kompleks ke dalam bentuk model sederhana dengan mengabaikan aspek lain yang tidak sesuai dengan masalah.

6. Enkapsulasi (*Encapsulation*)

Pembungkusan atribut data dan layanan (operasi) milik objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak tahu cara kerjanya.

7. Pewarisan (*Inheritance*)

Mekanisme yang memungkinkan suatu objek untuk mewarisi beberapa atau semua definisi dan objek lain sebagai bagian dari dirinya sendiri.

8. Antarmuka (*Interface*)

Antarmuka atau antarmuka sangat mirip dengan kelas, tetapi antara atribut kelas tanpa metode yang dinyatakan tanpa konten. Deklarasi metode pada antarmuka dapat diimplementasikan di kelas lain. Kelas dapat menerapkan lebih dari satu antarmuka di mana kelas ini menyatakan metode pada antarmuka yang diperlukan oleh kelas itu serta mendefinisikan isinya dalam kode program kelas itu. Metode pada antarmuka yang diimplementasikan dalam kelas harus persis cocok dengan yang ada di antarmuka. Antarmuka atau antarmuka biasanya digunakan sehingga kelas lain tidak secara langsung mengakses kelas, mengakses antarmuka.

9. Penggunaan Kembali (*Reusability*)

Penggunaan kembali objek yang telah didefinisikan untuk masalah dalam masalah lain yang melibatkan objek. Misalnya, aplikasi pinjaman buku diperlukan untuk kelas anggota, jadi ketika membuat aplikasi VCD, kelas anggota ini dapat digunakan kembali dengan perubahan kecil pada aplikasi sewa VCD tanpa harus membuatnya dari awal.

10. Generalisasi dan Spesialisasi

Menunjukkan hubungan antara kelas umum dan objek dan kelas dan objek tertentu. Misalnya, penerapan konsep pemrograman berorientasi objek untuk aplikasi penulisan surat. Kelas yang lebih umum (generalisasi) adalah kendaraan darat dan kelas khusus (spesialisasi) adalah mobil, sepeda motor, dan kereta api..

11. Komunikasi Antar Objek

Komunikasi antar objek dilakukan melalui pesan yang dikirim dari satu objek ke objek lainnya..

12. Polimorpisme (*Polymorphism*)

Kemampuan suatu objek yang akan digunakan untuk berbagai tujuan dengan nama yang sama sehingga menghemat jalur program.

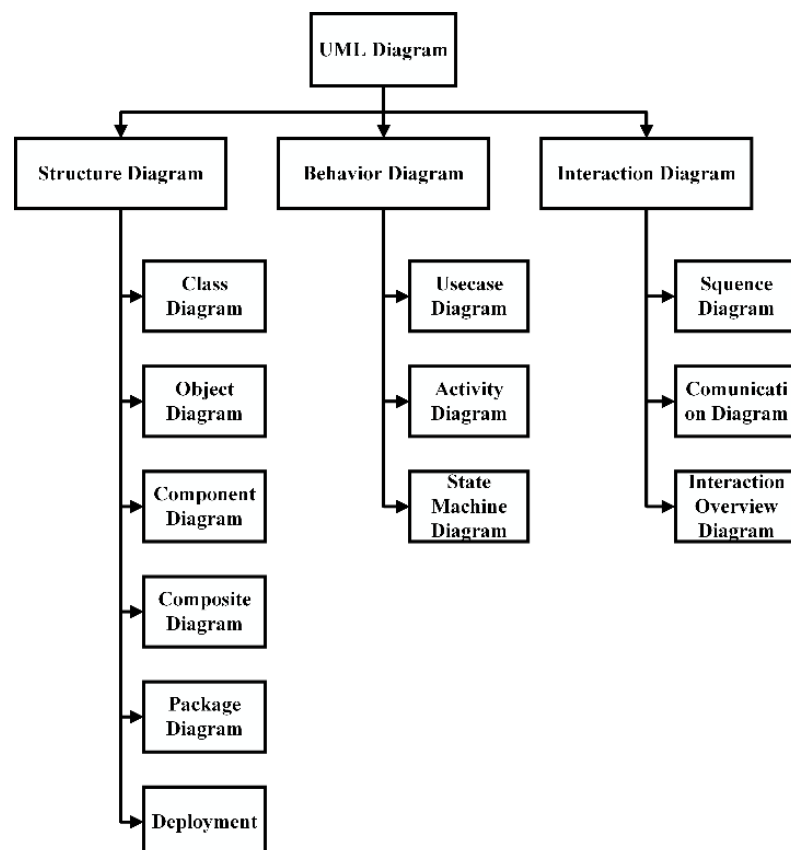
13. Package

Paket adalah wadah atau kemasan yang dapat digunakan untuk mengelompokkan kelas sehingga memungkinkan beberapa kelas dengan nama yang sama untuk disimpan dalam paket yang berbeda.

2.16 Unified Modeling Language (UML)

Metodologi analisis berorientasi objek adalah metode analisis yang memeriksa syarat-syarat dari sudut pandang kelas-kelas dan objek-objek yang ditemui pada ruang lingkup permasalahan dengan tujuan untuk memahami domain masalah dan meningkatkan ketelitian, konsistensi, kelengkapan analisis. Alat yang digunakan dalam metodologi ini salah satunya adalah UML (Yuliawati, Saleh and., 2018).

UML (*Unified Modeling Language*) adalah bahasa pemodelan untuk sistem atau *software* yang berparadigma berorientasi objek. Pemodelan (modeling) sebenarnya digunakan untuk penyederhanaan permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami (Nugroho, 2010). Menurut (Sukanto & Shalahuddin, 2014) UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar dibawah ini :



Gambar 2.3 Diagram Uml