**BAB II
TINJAUAN PUSTAKA**

**2.1 Android**

Sistem operasi Android adalah sebuah sistem operasi untuk perangkat mobile berbasis Linux yang mencakup sistem operasi, middleware, dan aplikasi[8]. Android menyediakan akses terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak.

Android memiliki OS yang sangat baik, cepat dan kuat serta memiliki antarmuka pengguna intuitif yang dikemas dengan pilihan dan fleksibilitas. Android SDK (Software Development Kit) menyediakan tools dan API yang diperlukan untuk mengembangkan aplikasi pada platform android dengan menggunakan bahasa pemrograman Java[9]. Application Programming Interface (API) yang disediakan menawarkan akses ke hardware, maupun data-data pinsel sekalipun, atau data sistem sendiri. Bahkan pengguna dapat menghapus aplikasi inti dan menggantikannya dengan aplikasi pihak ketiga.

Android menawarkan sebuah lingkungan yang berbeda untuk pengembangan. Setiap aplikasi memiliki tingkatan yang sama. Android tidak membedakan antara aplikasi inti dengan aplikasi pihak ketiga.

**2.1.1 Android Version**

Berikut adalah Tabel Error! No text of specified style in document.**.1** yang menggambarkan perkembangan android:

**Tabel** Error! No text of specified style in document.**.1 Android Version**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Code Name | API Level | Linux Kernel in AOSP |
| 1.5 | Cupcake | 3 | 2.6.27 |
| 1.6 | Donut | 4 | 2.6.29 |
| 2.0 | Eclair | 5, 6, 7 | 2.6.29 |
| 2.2 | Frozen Yogurt | 8 | 2.6.32 |
| 2.3 | Gingerbread | 9, 10 | 2.6.35 |
| 3.0 | Honeycomb | 11, 12, 13 | 2.6.36 |
| 4.0 | Ice Cream Sandwich | 14, 15 | 3.0.1 |
| 4.1 | Jelly Bean | 16, 17, 18 | 3.0.31 |
| 4.4 | Kitkat | 19, 20 | 3.10 |
| 5.0 | Lollipop | 21, 22 | 3.16 |
| 6.0 | Marshmallow | 23 | 3.18 |
| 7.0 | Nougat | 24, 25 | 4.4 |
| 8.0 | Oreo | 26, 27 | 4.10 |
| 9.0 | Pie | 28 | 4.14.42 |

**2.1.2 Arsitektur Android**

Secara garis besar arsitektur android dapat dijelaskan pada Gambar Error! No text of specified style in document.**.1** berikut:



**Gambar** Error! No text of specified style in document.**.1 Arsitektur Android**

**2.1.3 Application**

*Application layer* berada pada lapisan terluar dari Arsitektur Android. Pengguna awam Android pasti akan berinteraksi dengan lapisan ini untuk fungsi umum seperti menelepon, mengakses website, dll. Lapisan ini biasanya diakses oleh *developer*, *programmer*, atau sejenisnya. Pada layer terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain. Hampir semua aplikasi ditulis menggunakan bahasa pemrograman Java.

**2.1.3.1 Application Framework**

Applications frameworks Android adalah “Open Development Platform” di mana Android menawarkan kepada pengembang sebuah kebebasan untuk mengakses penuh API *framework* seperti yang dilakukan oleh aplikasi kategori inti. Arsitektur aplikasi dirancang supaya kita dengan mudah dapat menggunakan kembali komponen yang sudah digunakan.

*Application Framework layer* ini berinteraksi langsung dengan aplikasi yang dikembangkan. Program-program di atas memanajemen fungsi dasar dari perangkat seperti manajemen *resource*, manajemen panggilan, manajemen *window*, dan lain-lain.

Beberapa program penting pada Application Framework antara lain:

* 1. **Activity Manager** − Mengontrol semua aspek dari siklus hidup aplikasi dan *Activity Stack*.
	2. **Content Providers** − Mengizinkan aplikasi untuk mempublikasikan dan berbagi data dengan aplikasi lainnya.
	3. **Resource Manager** − Memberikan akses kepada resources yang bukan kode seperti *strings, setting* warna, dan *layout User Interface*.
	4. **Notifications Manager** − Membuat aplikasi dapat menampilkan pengingat dan notifikasi kepada pengguna.
	5. **View System** − Digunakan untuk membuat *User Interface* aplikasi.

**2.1.3.2 Libraries**

Libraries adalah *layer* dimana fitur-fitur Android berada. Berjalan di atas Kernel, *layer* ini meliputi berbagai library C/C++ inti seperti Libc SSL. Contoh dari *Library* yang termasuk dalam kategori ini adalah *Library* yang memfasilitasi pembangunan *User Interface*, Penggambaran Grafik dan akses *Database,* juga *library* yang terdapat pada *Application Framework*.

Rangkuman dari beberapa Library Android yang tersedia untuk Android Developer adalah sebagai berikut :

* 1. **android.app** − Memberikan akses ke model aplikasi dan fondasi dari semua aplikasi Android.
	2. **android.content** − Memfasilitasi akses konten, mempublikasikan dan mengirim pesan antar aplikasi dan komponen aplikasi.
	3. **android.database** − Digunakan untuk mengakses data yang dipublikasikan oleh content providers, juga termasuk kelas manajemen basis data SQLite.
	4. **android.opengl** − antarmuka Java ke Api penerjemah OpenGL ES 3D Graphic
	5. **android.os** − Menyediakan akses aplikasi ke OS system service stAndar seperti perpesanan, system service, dan IPC (Inter Process Communication ).
	6. **android.text** − Digunakan untuk menyajikan dan memanipulasi teks yang ditampilkan pada perangkat.
	7. **android.view** − Dasar dari pembangunan user interface aplikasi.
	8. **android.widget** − Koleksi dari beberapa user interface yang telah dibuat dan dapat langsung digunakan seperti tombol, label, tabel, view, layout manager, tombol pilihan, dll.
	9. **android.webkit** − Sekumpulan kelas untuk memungkinkan kemampuan web browsing yang akan dibangun ke dalam aplikasi.

**2.1.3.3 Android Runtime**

Layer yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan implementasi Linux. Di dalam Android *Runtime* dibagi menjadi dua bagian yaitu:

1. **Core Libraries**

Aplikasi Android dibangun dalam bahasa Java, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menterjemahkan bahasa Java yang ditangani oleh *Core Libraries*.

1. **Dalvik Virtual Machine**

Virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsifungsi secara efisien, dimana mampu membuat Linux Kernel untuk melakukan threading dan manajemen tingkat rendah.

**2.1.3.4 Linux Karnel**

Karena Android merupakan salah satu turunan dari sistem operasi Linux, maka Android juga memiliki kernel. Bagi yang belum pernah menyentuh sistem operasi Linux pasti akan bertanya — tanya apa itu *“KERNEL”. Kernel*merupakan inti dari sistem operasi berbasis UNIX. Kernel merupakan lapisan terdalam yang berada pada sistem operasi baik itu Linux maupun Android. Lapisan ini tidak benar — benar berinteraksi dengan pengguna maupun developer, tapi lapisan ini merupakan jantung dari seluruh sistem di Android karena lapisan inilah yang memberikan fungsi-fungsi berikut pada sistem Android:

1. Abstraksi Hardware.
2. Program Manajemen Memory.
3. Pengaturan Sekuritas.
4. Manajemen Energi Software ( Baterai ).
5. Driver (Driver adalah program yang mengontrol hardware).
6. Network Stack.

**2.2 Java**

Java adalah bahasa pemprograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin arus bawah yang minimal.

Java memiliki beberapa kelebihan dibandingkan dengan bahasa pemograman lain, diantaranya:

1. Multi-platform
2. Object Oriented Programming
3. Library Class yang lengkap
4. Mewarisi kekayaan C/C++
5. Pengumpulan Sampah Otomatis

Ada dua macam pemprograman yang dapat dibuat menggunakan bahasa pemprograman Java, yakni sebagai berikut[10]:

1. Applet Java yaitu program yang disisipkan dalam halaman web.
2. Aplikasi Java yaitu program *stand alone* yang pada umumnya dibagi menjadi dua macam:
3. Aplikasi Konsol yang hanya mendukung keluaran berupa karakter pada layar.
4. Aplikasi window yang dapat membuat dan mengatur banyak window seperti penggunaan menu, toolbar, dialog, dan lain-lain.

Versi awal Java ditahun 1996 sudah merupakan versi rilis sehingga dinamakan Java Versi 1.0. Java versi ini menyertakan banyak paket standar awal yang terus dikembangkan pada versi selanjutnya:

1. java.lang: Peruntukan kelas elemen-elemen dasar.
2. java.io: Peruntukan kelas input dan output, termasuk penggunaan berkas.
3. java.util: Peruntukan kelas pelengkap seperti kelas struktur data dan kelas kelas penanggalan.
4. java.net: Peruntukan kelas TCP/IP, yang memungkinkan berkomunikasi dengan komputer lain menggunakan jaringan TCP/IP.
5. java.awt: Kelas dasar untuk aplikasi antarmuka dengan pengguna (GUI).
6. java.applet: Kelas dasar aplikasi antar muka untuk diterapkan pada penjelajah web.

**2.3 Web Service**

W3C mendefinisikan web service sebagai sebuah software aplikasi yang dapat teridentifikasi oleh URL dan memiliki interface yang didefinisikan, dideskripsikan, dan dimengerti oleh XML atau JSON dan juga mendukung interaksi langsung dengan software aplikasi yang lain dengan menggunakan message berbasis XML atau JSON melalui protokol internet.

Web service adalah sebuah software aplikasi yang tidak terpengaruh oleh platform, menyediakan method-method yang dapat diakses oleh network. Web Service juga akan menggunakan XML untuk pertukaran data, khususnya pada dua entitas bisnis yang berbeda. Beberapa karakteristik dari web service adalah:

1. Message-based
2. Standards-based
3. Programming language independent
4. Platform-neutral

**2.4 JSON**

JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data. JSON terbuat dari dua struktur, yaitu:

1. Kumpulan pasangan nama atau nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (object), rekaman (record), daftar berkunci (keyed list), atau associative array.
2. Daftar nilai terurutkan (an ordered list of values). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (array), vektor (vector), daftar (list), atau urutan (sequence).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemrograman modern mendukung sruktur data ini dalam bentuk yang sama maupun berlainan. JSON menggunakan bentuk sebagai berikut:

1. Objek yang merupakan sepasang nama atau nilai yang tidak terurutkan.
2. Larik yang merupakan nilai yang sudah terurut.
3. Nilai atau *value* yang dapat berupa sebuah *string*, angka, *boolean*, atau sebuah objek atau larik.
4. String yang merupakan kumpulan dari nol atau lebih karakter *unicode,* yang dibungkus dengan tanda kutip ganda.
5. Angka yaitu dengan format desimal, bukan oktal maupun heksadesimal.

**2.5 Application Programming Interface**

Application Programming Interface atau API adalah sekumpulan perintah, fungsi, dan protokol yang dapat digunakan saat membangun perangkat lunak untuk sistem operasi tertentu. API memungkinkan programmer untuk menggunakan fungsi standar untuk berinteraksi dengan system operasi. API juga merupakan suatu dokumentasi yang terdiri dari antar muka, fungsi, kelas, struktur untuk membangun sebuah perangkat lunak.

Dengan adanya API, maka memudahkan seorang programmer untuk membongkar suatu software untuk kemudian dapat dikembangkan atau diintegrasikan dengan perangkat lunak yang lain. API dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya. Suatu rutin standar yang memungkinkan developer menggunakan system function. Proses ini dikelola melalui operating system. Keunggulan dari API ini adalah memungkinkan suatu aplikasi dengan aplikasi lainnya untuk saling berinteraksi.

Penggunaan API dapat bervariasi tergantung pada jenis bahasa pemrograman yang terlibat. API untuk bahasa prosedural seperti Lua dapat terdiri terutama dari rutinitas dasar untuk mengeksekusi kode, memanipulasi data, atau menangani kesalahan; sementara API untuk bahasa berorientasi objek, seperti Java, akan memberikan spesifikasi kelas dan metode kelasnya[9].

API menggunakan bahasa yang lebih terstruktur dan mudah dimengerti daripada bahasa system call. Hal ini sangat penting dalam hal editing dan pengembangan. System call interface ini berfungsi sebagai penghubung antara API dan system call yang dimengerti oleh sistem operasi. System call interface ini akan menerjemahkan perintah dalam API dan kemudian akan memanggil system calls yang diperlukan. Untuk membuka suatu file tersebut user menggunakan program yang telah dibuat dengan menggunakan bantuan API, maka perintah dari user tersebut diterjemahkan dulu oleh program menjadi perintah open().

Perintah open() ini merupakan perintah dari API dan bukan perintah yang langsung dimengerti oleh kernel sistem operasi. Oleh karena itu, agar keinginan pengguna dapat dimengerti oleh sistem operasi, maka perintah open() tadi diterjemahkan ke dalam bentuk system call oleh system call interface. Implementasi perintah open() tadi bisa bermacam-macam tergantung dari sistem operasi yang digunakan. Cara menggunakan API :

1. Dilakukan dengan mengimpor package/kelas
2. Ada beberapa kelas bernama sama dipackage yang berbeda, yaitu :
	1. Import salah satu dan gunakan nama lengkap untuk yang lain
	2. Gunakan nama lengkap semua kelas.

Kebanyakan sistem operasi seperti Windows, menyediakan fasilitas API sehingga programmer dapat melakukan aktivitas programming dengan lebih konsisten.

**2.6 OO Analisis Design**

Analisis dan desain berorientasi objek adalah cara baru dalam memikirkan satu masalah dengan menggunakan model yang dibuat menurut konsep sekitar dunia nyata. Tujuan dari analisis berorientasi objek adalah untuk mengembangkan model yang menggambarkan perangkat lunak komputer karena bekerja untuk memenuhi seperangkat persyaratan yang ditentukan user. Tools yang dapat digunakan pada pendekatan analisis pengembangan sistem secara objek dapat menggunakan UML[11].

Dalam membangun block UML ada 3 hal yang harus diperhatikan, yaitu object (memodelkan konsep), relationship (mengkoneksikan object), dan diagram (grouping yang saling mengkoneksikan antara object dan relationship.

**2.6.1 Behavior Diagram**

*Behavior Diagram* pada *Unified Modelling Language* digunakan untuk mendeskripsikan interaksi antara aktor dan sebuah use case, yaitu bagaimana seorang aktor menggunakan sistem. *Behavior diagram* terdiri dari:

**Use Case Diagrams** yang merupakan gambaran fungsionalitas yang diharapkan dari sebuah sistem.

**Activity Diagrams** yang menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing aliran berawal, keputusan yang terjadi, dan bagaimana mereka berakhir.

**Sequence Diagrams** adalah penggambaran tentang interaksi antar objek di dalam dan di sekitar sistem berupa pesan yang digambarkan berdasarkan dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

**2.6.2 Struktural Diagram**

*Structural Diagram* pada *Unified Modelling Language* digunakan untuk mendeskripsikan relasi antarkelas. Structural diagram terdiri dari:

1. Class Diagrams menggambarkan pandangan statis suatu sistem dalam hal kelas dan hubungan antar kelas.
2. Component Diagrams adalah sesuatu yang diperlukan untuk menjalankan fungsi stereotip. Contoh stereotip dalam komponen termasuk file yang dapat dieksekusi, dokumen, tabel database, file, dan file perpustakaan.
3. Deployment Diagrams memodelkan penyebaran fisik artefak pada node. Untuk menggambarkan situs web, misalnya, diagram penempatan akan menunjukkan komponen perangkat keras yang ada misalnya, server web, server aplikasi, dan server database.

**2.7 Clarifai**

Clarifai adalah perusahaan kecerdasan buatan (AI) yang berspesialisasi dalam visi komputer dan menggunakan pembelajaran mesin dan jaringan saraf yang dalam untuk mengidentifikasi dan menganalisis gambar dan video. Perusahaan menawarkan solusinya melalui API, SDK seluler, dan solusi di tempat.

API Clarifai menawarkan pengenalan gambar dan video sebagai layanan. Apakah Anda memiliki satu gambar atau miliaran, Anda hanya beberapa langkah lagi dari menggunakan kecerdasan buatan untuk mengenali konten visual Anda.

 Berikut adalah jenis-jenis layanan pada Clarifai API:

**2.7.1 Prediksi**

API dibangun berdasarkan ide sederhana. Pengguna mengirim input (gambar atau video) ke layanan dan layanan akan mengembalikan prediksi. Jenis prediksi didasarkan pada model apa Anda menjalankan input. Misalnya, jika Anda menjalankan input Anda melalui model 'food', prediksi yang dikembalikan akan berisi konsep-konsep yang diketahui oleh model 'food' atau makanan. Berikut adalah Gambar Error! No text of specified style in document.**.2** yang mengilustrasikan proses tersebut



**Gambar** Error! No text of specified style in document.**.2 Proses Prediksi Clarifai**

Sebelum bisa menggunakan Clarifai, pengembang aplikasi harus terlebih dahulu membuat akun dan menerima API Key secara gratis. Semua akses API melewati HTTPS dan diakses melalui domain https://api.clarifai.com. Lebih disarankan menggunakan *Client Library* untuk mengakses API yang sudah tersedia dalam berbagai bahasa pemprogramman seperti JavaScript, Phyton, Java, C#, dan PHP.

Selain itu, terdapat juga *Authentication* ke API yang ditangani melalui API Key. Pengembang dapat membatasi ruang lingkup sebuah API Key, yang memungkinkan Aplikasi yang menggunakan Key tersebut untuk melakukan operasi yang sangat spesifik.

**2.7.2 Pencarian**

Kasus umum untuk menggunakan Clarifai adalah untuk mendapatkan konsep-konsep yang diprediksi dalam sebuah gambar dan kemudian menggunakan konsep-konsep itu untuk mencari data.

API Pencarian memungkinkan Pengembang untuk mengirim gambar (berupa url) ke layanan dan membuat mereka diindeks oleh konsep model 'umum' dan representasi visual mereka. Setelah diindeks, Pengguna dapat mencari gambar berdasarkan konsep dari hasil prediksi. Berikut adalah Gambar Error! No text of specified style in document.**.3** yang mengilustrasikan proses tersebut:



**Gambar** Error! No text of specified style in document.**.3 Proses Pencarian Clarifai**

Pengembang juga dapat mengatur agar Clarifai mencari gambar berdasarkan gambar lainnya. Dalam hal ini, Pengguna menginputkan gambar (url) dan hasilnya layanan akan mengembalikan semua gambar dalam indeks pencarian yang secara visual mirip dengan gambar yang dimasukkan.

**2.8 Instagram**

Instagram adalah sebuah aplikasi berbagi foto dan video yang memungkinkan pengguna mengambil foto, mengambil video, menerapkan filter digital, dan membagikannya ke berbagai layanan jejaring sosial, termasuk milik Instagram sendiri. Satu fitur yang unik di Instagram adalah memotong foto menjadi bentuk persegi, sehingga terlihat seperti hasil kamera Kodak Instamatic dan polaroid. Hal ini berbeda dengan rasio aspek 4:3 atau 16:9 yang umum digunakan oleh kamera pada peranti bergerak.

Instagram API memerlukan otentikasi dengan permintaan khusus yang dibuat atas nama pengguna. Permintaan otentikasi memerlukan sebuah *access token*.

Instagram Api memiliki fitur yang bernama *Search for hastag.* Fitur ini memungkinkan user menemukan foto dan video publik yang telah ditandai dengan tagar khusus.

**2.9 Pinterest**

Pinterest adalah virtual pinboard di mana pengguna bisa mengunggah foto atau gambar yang bisa dimasukkan ke dalam kategori-kategori yang fungsinya semacam album atau folder yang bisa di customize namanya. Semisal kategori 'food and drink' di mana berisi foto dan gambar makanan dan minuman atau juga ‘fashion outfit’ yang nantinya akan berisikan foto tentang cara berbusana. User bisa saling mengikuti akun lain dan re-pin pin milik akun lain (seperti retweet pada Twitter) tanpa harus mengikuti pemilik pinboard.

Untuk menggunakan Pinterest API, user harus membuat aplikasi. Setelah membuat aplikasi, user akan dapat mengujinya dan menambahkan sejumlah kolaborator untuk membantu pengujian aplikasi. Ada dua jenis kolaborator: pengembang dan penguji.

Pemanfaatan Pinterest API juga bisa untuk mengambil info profil, *boards*, *pins, suggest boards* dan *suggest pins*, serta *search board* dan *search pin* untuk pengguna yang diautentikasi.

* + - 1. **Board** adalah koleksi *pin* yang didedikasikan untuk sebuah tema tertentu seperti halnya puisi, melancong, atau pernikahan.
			2. **Pin** adalah gambar yang telah ditautkan dari situs web maupun diunggah langsung ke Pinterest. *Pin* yang disimpan dari *board* seorang *user* dapat disimpan ke *board* milik *user* lainnya.

**2.10 OpenWeather Map**

Open Weather Map adalah layanan online yang menyediakan API untuk data cuaca, termasuk data saat cuaca, ramalan, dan data historis untuk para pengembang layanan web dan aplikasi mobile. Untuk sumber data, dengan menggunakan layanan meteorologi siaran, data mentah dari stasiun cuaca bandara, data mentah dari stasiun radar, dan data mentah dari stasiun cuaca resmi lainnya. Semua data yang diproses oleh OpenWeatherMap sehingga ia mencoba untuk memberikan secara online cuaca data dan cuaca peta yang akurat, seperti untuk awan atau hujan.

Open Weather Map API memiliki beberapa fitur, diantaranya adalah:

Current Weather Data yang memberikan akses untuk data cuaca pada setiap lokasi di lebih dari 200.000 kota.

5 Day / 3 Hour Forecast adalah fitur ramalan cuaca lima hari ke depan dengan rentang cuaca tiap 3 jam.

16 day / Daily Forecast adalah fitur ramalan cuaca 16 hari yang meliput data cuaca setiap hari.

Historical Data yang menyediakan riwayat data cuaca lebih dari 37.000 kota dengan rentang data per jam.

Weather Map Layers yaitu peta cuaca yang memberikan info curah hujan, awan, tekanan cuaca, suhu, angin, dan lain-lain.

**2.11 Android Location**

Berisi Framework berupa kelas-kelas API yang mendefinisikan layanan berbasis lokasi Android dan terkait. Beberapa kelas diantaranya adalah

* 1. Classes Address: Kelas yang mewakili Alamat, yaitu, seperangkat String yang menggambarkan lokasi.
	2. Classes Geocoder: Kelas untuk menangani geocoding dan membalikkan geocoding.
	3. Classes Location: Kelas data yang mewakili lokasi geografis.
	4. Classes Manager: Kelas ini menyediakan akses ke layanan lokasi sistem.

**2.12 Sistem Pakar**

Sistem pakar adalah program komputer yang mensimulasikan penilaian dan perilaku manusia yang memiliki pengetauan dan pengalaman ahli dalam bidang tertentu. Biasanya, sistem seperti ini berisi basis pengetahuan yang beisi akumulasi pengalaman dari satu set aturan untuk menerapkan pengetahuan dasar untuk setiap situasi tertentu[12].

Sistem pakar yang mencoba memecahkan masalah yang biasanya hanya bisa dipecahkan oleh seorang pakar, dipandang berhasil ketika mampu mengambil keputusan seperti yang dilakukan oleh pakar aslinya baik dari sisi proses pengambilan keputusannya maupun hasil keputusan yang diperoleh.

Sebuah sistem pakar memiliki 2 koomponen utama yaitu basis pengetahuan dan mesin inferensi.

* 1. Basis pengetahuan merupakan tempat menyimpan pengetahuan dalam memori komputer, di mana pengetahuan ini diambil dari pengetahuan pakar.
	2. Mesin inferensi merupakan otak dari aplikasi sistem pakar, bagian inilah yang menuntun user untuk memasukkan fakta sehingga diperoleh suatu kesimpulan. Apa yang dilakukan mesin inferensi ini didasarkan pada pengetahuan yang ada dalam basis pengetahuan.

Ada banyak cara untuk merepresentasikan pengetahuan di antaranya adalah logika, jaringan semantik, object-attribut value (OAV), bingkai, dan kaidah produksi[13].

**2.12.1 Knowledge**

Sistem pakar awalnya merepresentasikan fakta tentang dunia sebagai pernyataan sederhana dalam basis data yang datar dan menggunakan aturan sebagai pemikiran (*reasoning*) dan sebagai hasil (*result*). Mewakili pengetahuan secara eksplisit melalui aturan memiliki beberapa keunggulan yaitu[14]:

1. Akuisisi dan pemeliharaan. Para ahli dapat mendefinisikan dan memelihara aturan itu sendiri daripada melalui seorang programmer.
2. *Explaination* atau Penjelasan. Mengikuti rantai kesimpulan yang mengarah pada diagnosis dan menggunakan fakta-fakta ini untuk menjelaskan diagnosis.

**2.12.2 Rule-based**

Aturan digunakan sebagai cara untuk menyimpan dan memanipulasi pengetahuan untuk menafsirkan informasi dengan cara yang bermanfaat. Rule-Based Reasoning sering digunakan dalam aplikasi dan penelitian kecerdasan buatan.

Aturan adalah konstruksi komputasi yang efisien untuk representasi yang ringkas, modular, dan penalaran langsung dan efisien. Aturan mampu menggambarkan dengan seksama pengetahuan yang relevan[15].

Sistem berbasis aturan biasanya memiliki tiga komponen dasar yaitu daftar aturan, mesin inferensi, dan antarmuka pengguna. Berikut adalah penjelasannya:

* + - 1. **Daftar aturan** merupakan jenis basis pengetahuan tertentu.
			2. **Mesin inferensi** atau pemikir semantik, yang menyimpulkan informasi atau mengambil tindakan berdasarkan interaksi input dan basis aturan.
			3. **Antarmuka pengguna** yang menghubungkan antara pengguna dengan sistem.

**2.12.3 Case-based**

Penalaran berbasis kasus dalah proses penyelesaian masalah baru berdasarkan solusi dari masalah masa lalu yang serupa. Seorang mekanik mobil yang memperbaiki mesin dengan menarik kembali mobil lain yang menunjukkan gejala serupa menggunakan penalaran berbasis kasus.

Telah dikemukakan bahwa penalaran berbasis kasus tidak hanya metode yang kuat untuk penalaran komputer, tetapi juga pemecahan masalah sehari-hari; bahwa semua alasan didasarkan pada kasus-kasus masa lalu yang dialami secara pribadi. Pandangan ini terkait dengan teori prototipe, yang paling dieksplorasi dalam ilmu kognitif.

Penalaran berbasis kasus telah diformalkan untuk keperluan penalaran komputer sebagai proses empat langkah[16]:

1. **Retrieve**: Diberikan masalah target, ambil dari case memori yang relevan untuk menyelesaikannya.
2. **Reuse**: Memetakan solusi dari kasus sebelumnya ke masalah target.
3. **Revise**: Meninjau kembali solusi yang diberikan. Setelah memetakan solusi sebelumnya ke situasi target, uji solusi baru dengan simulasi, dan jika perlu lakukan revisi.
4. **Retain**: Setelah solusi berhasil disesuaikan dengan masalah target, simpan pengalaman yang dihasilkan sebagai kasus baru dalam memori.

**2.13 Pengertian Fesyen**

Fesyen adalah gaya berpakaian. Pakaian adalah busana atau baju atau kostum yang berfungsi tidak hanya sebagai pelindung tubuh, namun juga sebagai ciri khas yang dapat dilihat dari kutipan “*Fashion and some of its many facets; it sources in culture and social structure, and processes by which it diffuses within and among societies, the purposes it serves in social differentiation and social integration, the psychologcal needs it is said to satisfy, and not least of all, its implications for modern economic life*.”[17].

**2.14 Pengertian OOTD**

OOTD adalah singkatan dari *Outfit of The Day*, sebagai padanan kata untuk menunjukan pakaian apa yang Anda pakai di hari itu. Konsep OOTD sangat sederhana dan apa adanya untuk beraktivitas di hari tersebut.

Berkaitan dengan fenomena OOTD yang muncul dan populer di berbagai jejaring sosial yang dalam perkembangannya saat ini tidak hanya digunakan untuk sekedar berbagi foto lewat jejaring sosial berbasis gambar seperti Instagram ataupun Pinterest, namun juga menjadi sarana promosi, memasarkan, dan berkomunikasi yang sangat mumpuni untuk para pelaku industri di bidang fesyen.

**2.15 Pengertian Dress Code**

Aturan berbusana atau *dress code* adalah peraturan tertulis ataupun tidak tertulis yang mengatur mengenai cara berpakaian. Dalam waktu sehari, seseorang mungkin perlu mengatur antara dua atau lebih aturan berbusana, setidaknya cara mereka berpakaian di tempat kerja, di sekolah, dan di rumah. Perbedaan budaya dalam dalam masyarakat menyebabkan norma-norma yang mengatur cara berpakaian juga berbeda-beda.

Aturan tertulis misalkan seperti seragam sekolah SMA saat senin harus dengan celana biru dan baju putih, bertopi, dan berdasi. *Dress codes are considered a permissible regulation of student expression because the classroom is considered a non public, rather than a public*[18]. Sedangkan yang tidak tertulis adalah ketika pergi ke kondangan, wanita cenderung menggunakan kebaya, sedangkan pria mengenakan kemeja batik.