BAB 2
LANDASAN TEORI

2.1 Pakaian

Pakaian adalah kebutuhan pokok manusia selain makanan dan tempat berteduh/tempat tinggal (rumah). Manusia membutuhkan pakaian untuk melindungi dan menutup dirinya. Namun seiring dengan perkembangan kehidupan manusia, pakaian juga digunakan sebagai simbol status, jabatan, ataupun kedudukan seseorang yang memakainya. Perkembangan dan jenis-jenis pakaian tergantung pada adat-istiadat, kebiasaan, dan budaya yang memiliki ciri khas masing-masing[4].

Berikut ini beberapa pakaian yang sampai saat ini masih digunakan dalam kehidupan sehari-hari:

a. Pakaian Jas

   Jas adalah pakaian resmi model Eropa, berlengan panjang, dan dipakai diluar kemeja. Setelan jas sedikitnya terdiri dari sebuah jas dan sebuah celana panjang yang dibuat dari kain yang sama.

b. Pakain Kemeja

   Pakaian Kemeja berasal dari Bahasa Portugis yaitu camisa yang merupakan sebuah baju atau pakaian atas. Pakaian ini menutupi tangan, bahu, dan dada sampai keperut. Pada umumnya kemeja berkerah dan berkancing depan. Kemeja ada yang lengan panjang dan lengan yang pendek.

c. Celana Panjang

   Celana panjang sudah ada sejak 570 SM memalui relief di Persepolis, Iran. Celana panjang ini digunakan kaum nomaden Iran sebagai busana berkuda yang akan melindungi paha dan kaki dari gesekan dengan tubuh kuda. Selanjutnya, celana panjang menjadi seragam kaum pemburu Iran juga Persia yang biasanya dibuat dari kulit binatang[4].

Berikut ini beberapa tipe karakter fasion yang digemari masyarakat:

1. Kasual, merupakan pakaian santai yang biasanya digunakan ketika melakukan kegiatan rekreasi bersama keluarga. Pakaian kasual biasanya mengenakan celana jeans, kaos T-shirt tak berkerah, baju pantai, dan sandal.
2. Formal, merupakan pakaian resmi yang biasanya digunakan buat kerja pada suatu instansi atau perusahaan untuk mengikuti aturan-aturannya. Pakaian formal biasanya mengenakan baju kemeja, dasi, jas, celana kain untuk laki-laki, rok untuk wanita, dan jas.
3. Sporty, merupakan pakaian olahraga yang digunakan saat berolahraga atau saat menonton acara yang berhubungan dengan olahraga. Pakaian sporty biasanya mengenakan baju kaos jersey, celana pendek, dan sepatu olahraga.
4. Trendy, merupakan pakai yang sedang populer dengan gaya pakaian kekinian biasanya dikenakan oleh remaja yang bersifat dinamis, aktif dan percaya diri.

Hubugan pakaian pada penelitian ini karena pada penelitian ini bertujuan membuat suatu aplikasi yang bisa merekomendasikan pakaian sesuai warna baju yang dimiliki serta warna kulit masyarakat tersebut yang akan menggunakan aplikasi ini.

2.2 Android

Android adalah sistem operasi yang dimodifikasi untuk *mobile device* berbasis linux yang terdiri dari sistem operasi, software penghubung antar sebuah aplikasi dengan aplikasi lain, dan aplikasi-aplikasi utama. Android dikembangkan oleh Android Inc. perusahaan ini kemudian dibeli oleh Google pada tahun 2005 dan dirilis secara resmi pada tahun 2007[2].

Android telah mengalami beberapa pembaruan sejak pertama kali rilis, berikut ini jenis-jenis versi dan *Codename* android bisa dilihat pada table 2.1[7].

Tabel 2.1 Daftar Versi Android

Versi	Peluncuran	Kode
1.5	30 April 2009	Cupcake
1.6	15 September 2009	Donut
2.0/2.1	26 Oktober 2009	Éclair
2.2	20 Mei 2010	Frozen Yoghurt (Froyo)
2.3	6 Desember 2010	Gingerbread
3.0	22 Februari 2011	Honeycomb
4.0	19 Oktober 2011	Ice Cream Sandwich
4.1/4.2/4.3	27 Juni 2012	Jelly Bean
4.4	3 September 2013	Kitkat
5.0 – 5.1	12 November 2014	Lollipop
6.0	5 Oktober 2015	Marshmallow
7.0/7.1	Agustus / September 2016	Nougat
8.0/8.1	Agustus 2017	Oreo

Secara umum, arsitektur Android terdiri dari beberapa lapisan software, yaitu *Applications and Widgets*, *Applications Framework*, *Libraries*, *Android Run-time* sebagai lapisan perantara (*middleware*), serta *Linux Kernel* yang membungkus di bawahnya[5]. Untuk penjelasan setiap lapisan tersebut adalah sebagai berikut :

1. *Applications and Widget*, Lapisan dimana pengguna hanya berhubungan dengan aplikasi saja.
2. *Applications Framework*, Lapisan dimana para pengembang melakukan pembuatan aplikasi yang akan dijalankan di sistem operasi Android dengan komponen-komponennya meliputi view, sumberdaya, pemberitahuan (notifikasi), dan lain-lain.

3. *Libraries*, Android menyediakan sejumlah *Libraries* seperti grafis 2D dan 3D, *multimedia playback*, *browser engine*, pencetakan font, database, dan lain-lain.
4. *Android Run-time*, Lapisan yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan implementasi Linux yang terbagi menjadi dua bagian yaitu Core Libraries dan Dalvik virtual Machine.
5. *Linux Kernel*, Menyediakan layanan dasar seperti keamanan, pengelolaan proses, pengelolaan file, pengelolaan sumber daya memori, dan hard ware.

Alasan digunakannya Android pada penelitian ini karena Android menjadi sebuah sistem operasi perangkat bergerak (*mobile device*) yang paling banyak digunakan di dunia. Dari sample 50 orang yang sudah dilakukan didapat data 100% orang memiliki perangkat android. Teknologi yang terdapat pada perangkat android bisa dimanfaatkan untuk membantu permasalahan yang sedang diteliti saat ini.

2.3 Java

Java diciptakan oleh Patrick Naughton dan james Gosling bersama timnya dalam suatu proyek dari Sun Microsystem dengan tujuan untuk menghasilkan bahasa pemrograman yang sederhana. Awalnya nama bahasa pemrograman ini bukanlah Java, tetapi OAK terinspirasi dari sebuah pohon yang berada pada sebrang kantornya. Namun Sun menyadari OAK merupakan bahasa pemrograman komputer yang sudah ada. Maka Sun mengubahnya menjadi Java yang terinspirasi saat mereka sedang menimati secangkir kopi di sebuah kedai kopi. Java termasuk bahasa pemrograman tingkat tinggi dan bahasa pemrograman berorientasi objek untuk membuat dan menjalankan perangkat lunak pada komputer[8].

Berikut ini beberapa fitur java yang ditawarkan java API antara lain[9] :

1. Applet

   Program Java yang dapat berjalan di atas browser, yang dapat membuat halaman HTML lebih dinamis dan menarik.
2. Java Networking

Sekumpulan API (*Application Programming Interface*) yang menyediakan fungsi-fungsi untuk aplikasi-aplikasi jaringan, seperti penyediaan akses untuk TCP, UDP, IP address dan URL. Namun pada Java Networking tidak menyediakan ICMP karena *security* dan hanya administrator (root) yang bias memanfaatkan protocol ICMP.

3. Java Database Connectivity (JDBC)

   JDBC menyediakan sekumpulan API yang dapat digunakan untuk mengakses database seperti Oracle, MySQL, PostgreSQL, Microsoft SQL Server

4. Java Security

   Java Security menyediakan sekumpulan API untuk mengatur security dari aplikasi Java baik secara *high level* atau *low level*, seperti *public/private key management* dan *certificates.*

5. Java Swing

   Java Swing menyediakan sekumpulan API untuk membangun aplikasi – aplikasi GUI (*Graphical User Interface*) dan model GUI yang diinginkan bisa bermacam-macam, bisa model Java, model Motif/CDE atau model yang dependent terhadap platform yang digunakan

6. Java RMI

   Java RMI menyediakan sekumpulan API untuk membangun aplikasi – aplikasi Java yang mirip dengan model RPC (Remote Procedure Call) jadi objek-objek Java bisa di panggil secara remote pada jaringan komputer.

7. Java 2D/3D

   Java 2D/3D menyediakan sekumpulan API untuk membangun grafik-grafik 2D/3D yang menarik dan juga akses ke printer.

8. Java Server Pages

   Berkembang dari Java Servlet yang digunakan untuk menggantikan aplikasi-aplikasi CGI, JSP (Java Server Pages) yang mirip ASP dan PHP merupakan alternatif terbaik untuk solusi aplikasi Internet.

9. Java Native Interface (JNI)

JNI menyediakan sekumpulan API yang digunakan untuk mengakses fungsi-fungsi pada library (*.dll atau *.so) yang dibuat dengan bahasa pemrograman yang lain seperti C,C++, dan Basic.

10. Java Sound

    Java Sound menyediakan sekumpulan API untuk manipulasi suara.

11. Java IDL + CORBA

    Java IDL (*Interface Definition Language*) menyediakan dukungan Java untuk implementasi CORBA (*Common Object Request Broker*) yang merupakan model distributed-Object untuk solusi aplikasi besar di dunia *networking*

12. Java Card

    Java Card utamanya digunakan untuk aplikasi-aplikasi pada smart card, yang sederhana wujudnya seperti SIM Card pada handphone.

13. Java Telephony API (JTAPI)

    Java Telephony API menyediakan sekumpulan API untuk memanfaatkan devices-devices telephony, sehingga akan cocok untuk aplikasi-aplikasi CTI (*Computer Telephony Integration*) yang dibutuhkan seperti ACD (*Automatic Call Distribution*), PC-PBX dan lainnya.

Adapun hubungan Java pada penelitian ini karena aplikasi yang dibuat menggunakan tools Android studio dengan bahasa pemrograman yang digunakan yaitu bahasa pemrograman Java. Android Studio dipilih karena memiliki banyak fitur yang memudahkan saat pembangunan aplikasi android.

2.4 OOP (*Object Oriented Programming*)

Object oriented programming atau pemrograman berorientasi objek adalah suatu pendekatan pemrograman untuk membuat kode program yang lebih terkelompok berdasarkan objek-objek yang terlibat. Untuk membuat aplikasi yang besar kita harus membagi aplikasi tersebut menjadi bagian-bagian kecil, pada kasus seperti ini programmer menerapkan konsep pemrograman berorientasi

objek dengan membagi-bagi kode program aplikasi menjadi kumpulan bungkusan objek[8].

Sebuah bahasa pemrograman dinyatakan bisa menggunakan OOP jika sudah memenuhi syarat-syarat berikut, yaitu :

1. *Encapsulation* atau pembungkusan yaitu memaketkan atribut bersama dengan metode-metodenya dalam sebuah *Class* dan memberi akses langsung terhadap atribut dan metode-metode yang ingin dilindungi di dalam *Class* tersebut..
2. *Inheritence* atau pewarisan yaitu mewariskan sesuatu dari suatu *Class* untuk dipakai *Class-class* yang lain yang memiliki beberapa sifat atau perilaku dari *Class* induknya. Misalnya seorang anak yang mendapatkan beberapa sifat atau prilaku dari ibu/bapaknya. Pada pemrograman berbasis objek ini inherent merupakan suatu hubungan dari dua buah class atau lebih dimana dari class yang berhubungan tersebut memiliki atribut dan metode yang sama.
3. *Polymorphism* berarti banyak bentuk yaitu pembuatan metode-metode dengan nama yang sama tetapi berbeda *Class* atau parameter masukkan.

Hubungan OOP (*Object Oriented Programming*) pada penelitian ini adalah pada penelitian ini menggunakan konsep OOP yaitu membagi-bagi kode program menjadi kelas-kelas yang di buat berdasarkan objek-objek yang terlibat saat pembangunan program.

2.5 UML *(Unified Modeling Language)*

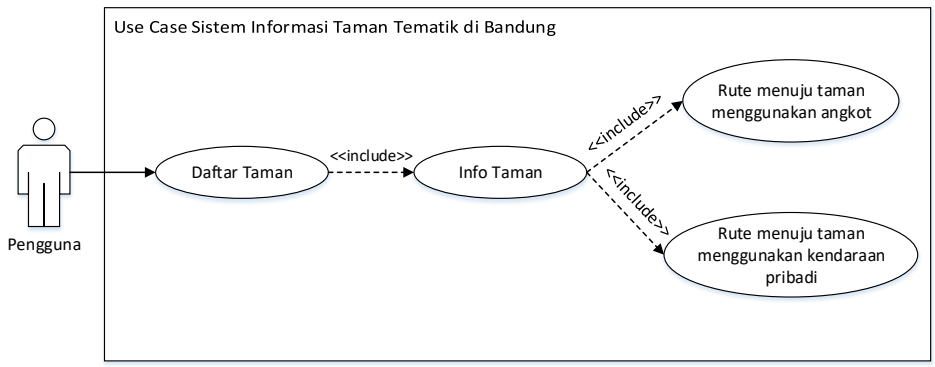
UML *(Unified Modeling Language)* adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML mulai diperkenalkan oleh Object Management Group, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat (tool) desain berorientasi objek dari IBM[11].

Berikut ini diagram-diagram UML yang paling sering digunakan dalam memodelkan sebuah sistem berorientasi objek diantaranya :

1. *Use Case* Diagram

   Use case diagram adalah form tertinggi tentang suatu sistem yang merupakan gambaran pada pihak manajemen, customer, atau orang lain yang terlibat langsung pada sistem tersebut tetapi tidak secara mendetail[11].

   Sebuah use case dapat meng-include fungsionalitas use case lain sebagai bagian dari proses dalam dirinya. Selain Include, Sebuah use case juga dapat meng-extend use case lain dengan behaviour-nya sendiri. Berikut ini adalah contoh dari *Use Case* Diagram



Gambar 2.1 Contoh Use Case Diagram

2. *Class* Diagram

   Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain[11].

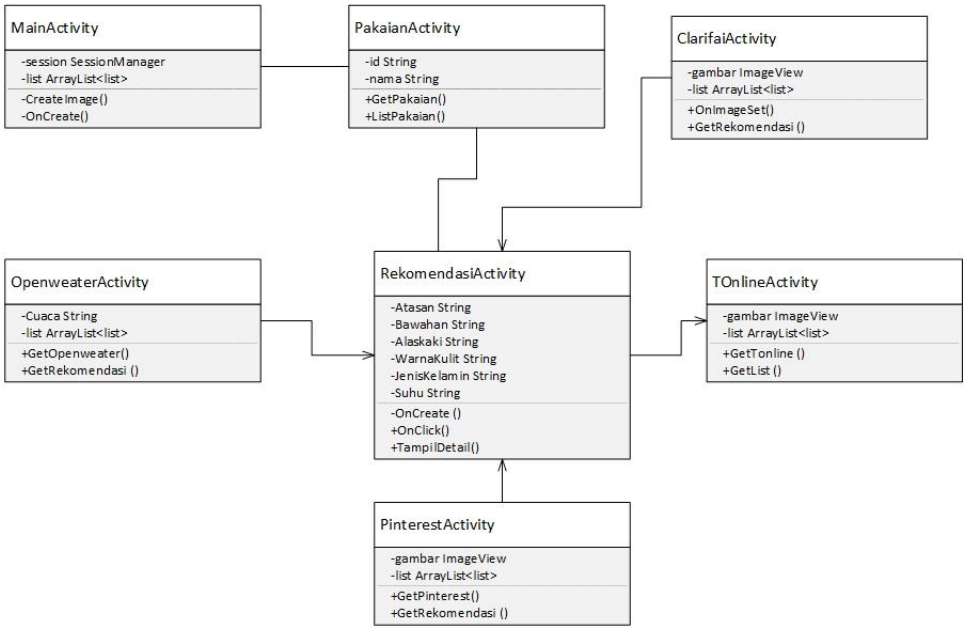
   Class memiliki tiga area pokok:

   1. Nama memiliki fungsi merepresentasikan identitas dari sebuah class.
   2. Atribut memiliki fungsi merepresentasikan atribut – atribut yang ada di dalam suatu class.
   3. Metode memiliki fungsi merepresentasikan metode-metode yang ada pada suatu class.

   Pada atribut dan metode bisa memiliki sifat sifat berikut :

1. *Private*, hanya bisa dipanggil oleh *Class* yang bersangkutan, tidak bisa dipanggil oleh *Class* lain yang tidak bersangkutan.
2. *Public*, bisa di panggil oleh seluruh *Class*.
3. Protected, hampir sama sifatnya dengan *Private* dan Class turunannya.

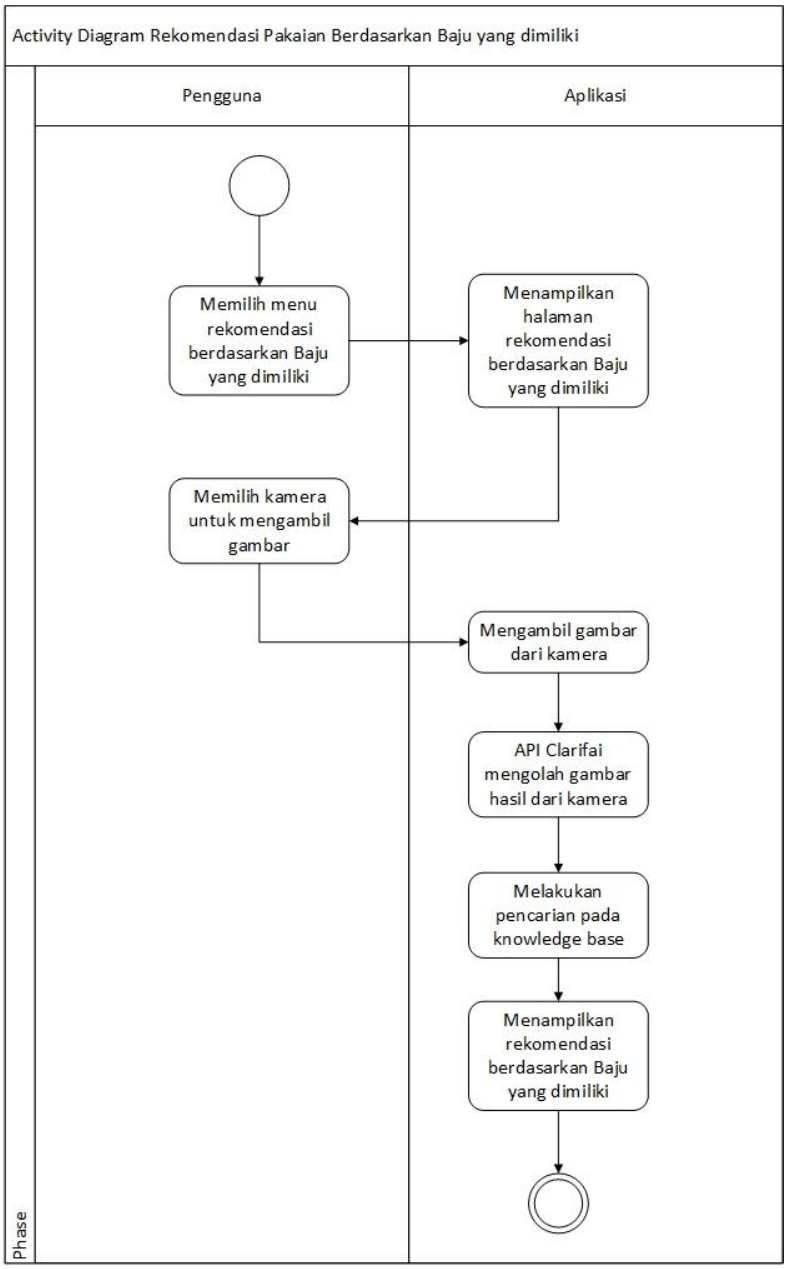
Pada gambar 2.2 adalah contoh dari *Class* diagram:



Gambar 2.2 Contoh Class Diagram

3. *Activity* Diagram

   Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin. Pada Gambar 2.3 adalah contoh dari *Activity* Diagram

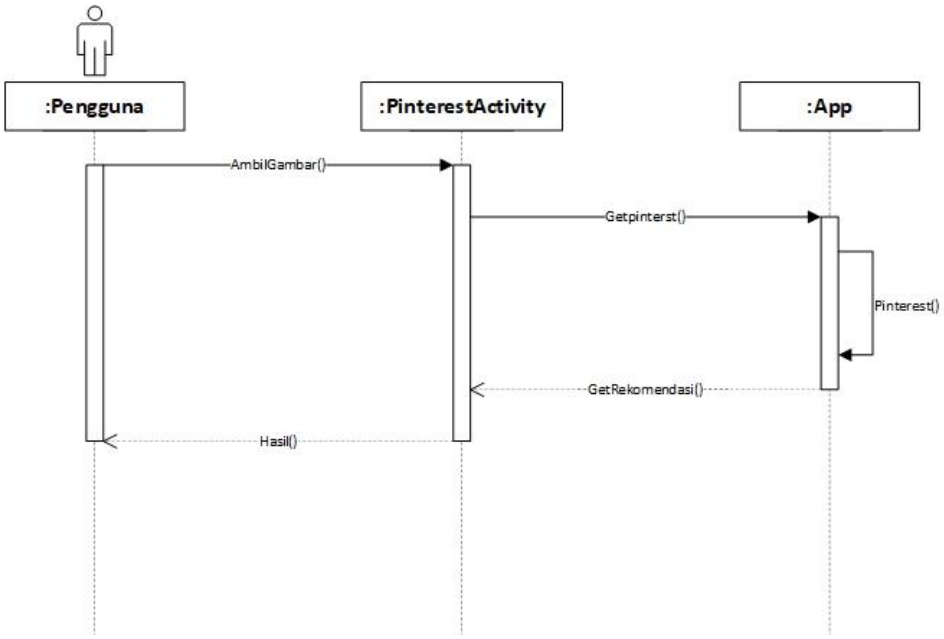


Gambar 2.3 Contoh Activity Diagram

4. *Sequence* Diagram

   Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. *Sequence* diagram terdiri antar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan[12]. Pada Gambar 2.4 adalah contoh dari *Sequence* Diagram



Gambar 2.4 Contoh Squence Diagram

Adapun hubungan UML pada penelitian ini adalah pada penelitian ini untuk memodelkan suatu aplikasi yang akan dibuat menggunakan konsep berorientasi objek. Sehingga bahasa pemodelan yang digunakan dalam perancangan aplikasi adalah UML.

2.6 API Clarifai

Teknologi klarifai bergantung pada penggunaan Convolutional Neural Networks (CNN) untuk memproses gambar, dan kemudian menghasilkan daftar tag yang mendeskripsikan gambar tersebut. Pendekatan Clarifai pertama kali diusulkan untuk tantangan klasifikasi ImageNet pada tahun 2013[13].

Clarifai bekerja melalui analisis gambar untuk menghasilkan daftar label deskriptif yang mewakili gambar yang diberikan. Untuk setiap tag dalam daftar

ini, sistem juga memberikan nilai probabilitas. Probabilitas ini mewakili kemungkinan
menggambarkan gambar menggunakan tag spesifik. API Clarifai dapat diakses sebagai layanan web jarak jauh[13].

API Clarifai dapat diakses menggunakan opsi uji coba atau digunakan dalam mode pembayaran. Dalam hal ini, mode uji coba telah digunakan untuk mendapatkan tag gambar. Mode ini memungkinkan untuk mendapatkan tag untuk jumlah gambar yang dibatasi. Menggunakan mode uji coba, kita hanya dapat memperoleh 20 tag per gambar dengan probabilitas terkaitnya. Mengenai waktu pemrosesan, Clarifai memiliki waktu latensi hampir 1 menit[13].

Hubungan API Clarifai pada penelitian ini adalah pada penelitian ini memanfaat API Clarifai untuk mendeteksi warna kulit dari hasil gambar yang diinputkan pengguna. Hasil dari API Clarifai ini akan digunakan sebagai data masukkan untuk aplikasi yang akan dibuat.

2.7 GPS (*Global Positioning System*)

Fitur yang semakin terjangkau dan populer pada piranti bergerak adalah kemampuan GPS yang memungkinkan piranti memberitahu lokasi setiap saat. Fungsi utama GPS adalah pemetaan, pencarian lokasi, dan arah. Teknologi GPS adalah teknologi sistem navigasi yang awalnya dikembangkan untuk menunjang kagiatan militer, namun kemudian dibuka untuk umum.

Teknologi GPS terdiri dari dari konsentelasi 27 satelit yang mengelilingi bumi (24 satelit aktif ditambah 3 satelit cadangan). Tiap satelit mengelilingi bumi dua kali dalam satu hari. Orbit tiap satelit sudah di atur sehingga pada tiap titik di seluruh permukaan bumi, orang akan menemukan paling tidak 4 setelit di langit tiap saat. Bicara piranti GPS artinya mengacu pada piranti penerima yang tugas utamanya adalah menemukan lokasi empat satelit atau lebih, mendapat jarak piranti penerima kemasing-masing satelit, dan menentukan lokasi piranti penerima berdasarkan jarak ke masing-masing satelit[2].

Selain memberikan keuntungan, tentu ada beberapa tantangan yang harus dihadapi ketika menggunakan layanan berbasis lokasi, yaitu :

1. Konsumsi Energi

   Tingkat konsumsi energi penyedia layanan berbeda-beda. Ketika aktif, piranti GPS paling banyak menguras energi disbanding Cell-ID atau Wi-Fi. Oleh karena itu, penting untuk mematikan piranti ini bila tidak diperlukan.

2. Akurasi

   Tingkat akurasi penyedia layanan lokasi juga berbeda-beda, LocationManager.NETWORK_PROVIDER memiliki akurasi terendah. Sedangkan LocationManager.GPS_PROVIDER memiliki akurasi lebih baik.

3. Ketersediaan Layanan

   Layanan berbasis lokasi dengan GPS, Cell-ID, atau Wi-Fi hanya dapat dinikmati bila piranti Anda mendukungnya. Namun, meskipun sudah mendukung, tetap tidak ada jaminan layanan akan selalu tersedia. Sinyal GPS akan bagus apabila Anda berada diruang terbuka dan posisi pancaran sinyal dari satelit tidak terhalang. Bila berada dibawah tanah atau ruang berdinding tebal, penerimaan sinyal GPS mungkin buruk karena tidak cukup kuat untuk menembusnya.

4. Waktu Untuk Memperoleh Data Lokasi

   Penentuan lokasi piranti penerimaan GPS membutuhkan waktu yang lama di bandingkan Cell-ID. Lama waktu tergantung piranti penerima GPS[2].

Hubungan GPS pada penelitian ini adalah pada penelitian ini membutuhkan GPS untuk menemukan lokasi pengguna untuk mengambilan data cuaca lokasi pengguna saat itu.

2.8 Android Studio

Android Studio adalah IDE (*Integrated Developmenat Environment*) resmi Android yang dirancang untuk membantu Anda membangun aplikasi berkualitas tinggi untuk setiap perangkat Android. Android Studio merupakan pembaharuan IDE dari Eclipse yaitu versi IDE yang sebelumnya. Saat ini, Android Studio sudah dilengkapi dengan berbagai kemudahan dan kenyamanan[14].

Kemampuan yang ditawarkan dari Android Studio antara lain :

1. Mendukung pengembangan berbasis Gradle.
2. Dapat menspesifikasikan pemfaktoran android dan melakukan perbaikan dengan cepat.
3. memiliki lint tools untuk menangkap kinerja, kegunaan kompatibilitas versi dan masalah lainnya.
4. memiliki kemampuan penandaan aplikasi dan ProGUard.
5. memiliki kemampuan wizard berdasarkan template untuk membuat desain dan komponen android yang umum.
6. sebuah layout editor yang kaya dan memungkinkan anda untuk melakukan darg-drop UI, pratinjau layout pada beberapa konfigurasi layar, dan banyak lagi.
7. di bangun untuk mendukung Google Cloud Platform, sehingga mudah untuk mengintegrasikan Google Cloud Messaging dan App Engine sebagai komponen server-side[15].

Alasan digunakan Android Studio pada penelitian karena tools yang digunakan dalam pembangunan aplikasi android yaitu menggunakan IDE Android Studio. Android studio menyediakan alat pengembang android yang terintegrasi dalam pengembangan dan debugging program.

2.9 JSON

JSON adalah sebuah format data-interchange yang ringan. JSON merupakan format teks yang sepenuhnya independen tetapi menggunakan konvensi yang familiar dengan bahasa pemrograman dari keluarga-C, termasuk C, C++, C#, Java, JavaScript, Perl, dan Phyton[19].

JSON dibangun dalam dua struktur, yaitu :

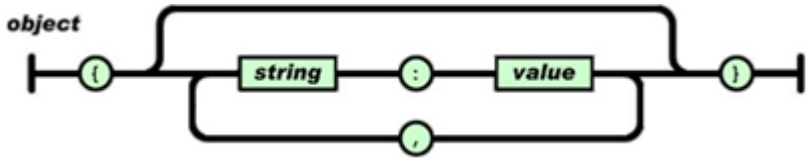
1. Beberapa pasangan dari nama / nilai. Dalam beberapa bahasa pemrograman biasa disebut dengan istilah object, record, struct, tabel hash, key list atau associative array.
2. Nilai – nilai tersusun secara ordered list. Biasa disebut dengan array, vector, list atau daftar dalam bahasa pemrograman.

JSON adalah struktur data yang universal, bisa digunakan dalam berbagai bahasa pemrograman. Hampir semua bahasa pemrograman mendukung penuh JSON dalam berbagai format. Hal ini memungkinkan format data yang dapat dipertukarkan menggunakan bahasa pemrograman juga menggunakan dasar dari struktur JSON.

JSON menggunakan bentuk sebagai berikut:

A. Object

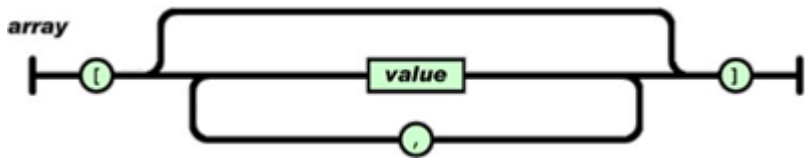
Objek adalah sepasang nama / nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma). Objek biasanya digunakan untuk menyimpan data tunggal dalam bentuk JSON.



Gambar 2.5 Object JSON

B. Array

Array adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung kotak buka) dan diakhiri dengan] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma). Larik dalam JSON dapat digunakan sebagai value dari JSON object hal ini dapat berguna jika JSON menyimpan data bertingkat.



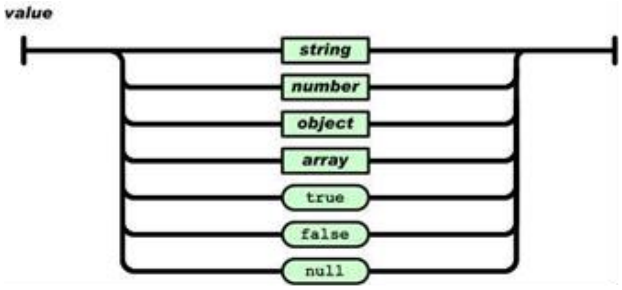
Gambar 2.6 Array JSON

Bentuk data JSON objek dan larik dapat saling dikombinasikan untuk mendukung struktur data yang lebih kompleks. JSON mendukung

beberapa tipe data untuk menjadi *value* seperti Angka, *String*, *Boolean* dan nilai NULL.

C. Nilai (*Value*)

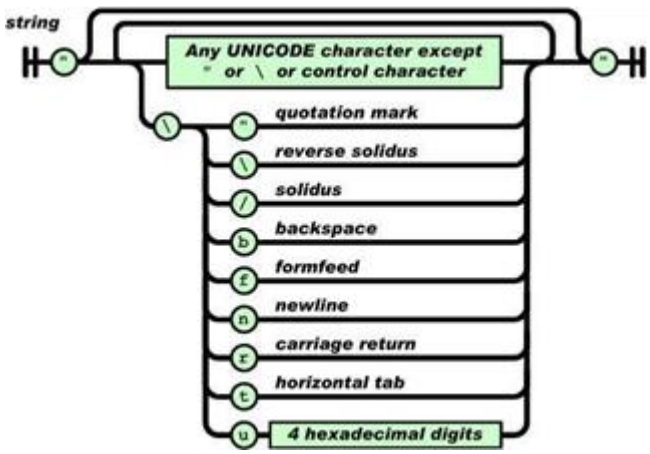
Nilai (*value*) dapat berupa sebuah *string* dalam tanda kutip ganda, atau angka, atau *true* atau *false* atau *null* atau sebuah objek atau sebuah larik. Struktur-struktur tersebut dapat disusun bertingkat.



Gambar 2.7 Nilai (*Value*) JSON

D. String

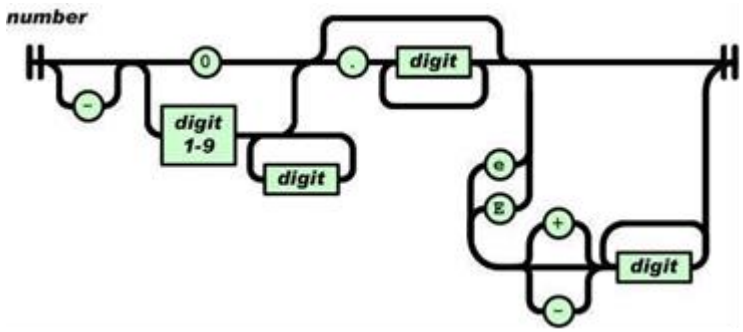
String adalah kumpulan dari nol atau lebih karakter *Unicode*, yang dibungkus dengan tanda kutip ganda. Pada *string* dapat digunakan *backslash escapes* "\" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada *string*. *String* sangat mirip dengan *string* C atau Java .



Gambar 2.8 String JSON

E. Angka

Angka adalah sangat mirip dengan angka di C atau Java, kecuali format oktal dan heksadesimal tidak digunakan.



Gambar 2.9 Angka JSON

2.10 Case-Based Reasoning (CBR)

Case Based Reasoning (CBR) didefenisikan sebagai sebuah metodologi untuk penyelesaian masalah dengan memanfatkan pengalaman sebelumya. CBR merupakan sebuah paradigma utama dalam penalaran otomatis (automated reasoning) dan mesin pembelajaran (machine learning). Di dalam CBR, seseorang yang melakukan penalaran dapat menyelesaikan masalah baru dengan memperhatikan kesamaannya dengan satu atau beberapa penyelesaian dari permasalahan sebelumnya[18].

Tahapan roses Case Based Reasoning adalah sebagai berikut :

1. Retrieve

   Tahap Retrieve adalah mengambil kembali permaslaahan yang sama. Pada langkah ini dilakukan proses pencarian atau kalkulasi dari kasus-kasus yang memiliki kesamaan.

2. Reuse

   Pada tahap Reuse menggunakan kembali informasi dan pengetahuan dalam kasus tersebut untuk mengatasi masalah baru. Pada langkah ini dicari solusi dari kasus serupa pada kondisi sebelumnya untuk permasalahan baru

3. Revise

   Lalu pada tahap Revise meninjau kembali solusi yang diberikan. Pada langkah ini dicari solusi dari kasus serupa pada kondisi sebelumnya untuk permasalahan yang terjadi kemudian.

4. Retain

   Dan terakhir pada tahap Retain mendalami bagian dari pengalaman sebelumnya untuk digunakan dalam pemecahan masalah berikutnya

Hubungan Case Based Reasoning (CBR) pada penelitian ini adalah metode CBR digunakan sebagai data knowledge untuk menentukan rekomendasi pakaian yang cocok berdasarkan warna kulit ataupun baju yang sudah dimiliki.

2.11 Pengujian Kuisioner

Kuesioner merupakan teknik pengumpulan data yang dilakukan dengan cara memberi beberapa pertanyaan atau pernyataan tertulis kepada responden untuk mendapatkan jawaban dari pertanyaan atau pernyataan tersebut. Metode yang digunakan dalam kuisionare ini menggunakan skala Likert. Skala likert diciptakan oleh Rensis Likert untuk pengukuran sikap, pendapat, dan presepsi seseorang atau kelompok orang terhadap sesuatu, misalnya tentang fenomena sosial. Untuk setiap satu pernyataan responden memilih salah satu tanggapan berupa SS = Sangat Setuju, S = Setuju, R = Ragu, TS = Tidak Setuju, STS = Sangat Tidak Setuju. Tanggapan responden dikodekan kedalam bilangan 1 sampai dengan 5. Pemberian bilangan ini bergantung kepada bentuk pernyataan yang dijawab oleh responden[17].

Ada dua macam bentuk pernyataan dalam skala likert, pertama adalah pernyataan yang diharapkan untuk disetujui oleh responden, disebut bentuk positif. Yang kedua adalah pernyataan yang diharapkan untuk tidak disetujui oleh responden, bentuk negatif. Biasanya kuesioner berisikan pernyataan positif dan negatif agar responden berpikir dulu sebelum memberi jawaban.

Tabel 2.2 Skala Likert

Jenis Pernyataan	Pernyataan	
	Positif	Negatif
Sangat Setuju	5	1
Setuju	4	2
Ragu	3	3
Tidak Setuju	2	4
Sangat Tidak Setuju	1	5

Hubungan pengujian kuisionar dengan penelitian ini adalah pada panelitian ini menggunakan kuisionare untuk mendapatkan jawaban dari responden berupa pernyataan positif dan negatif terhadap penelitian yang dilakukan. Dengan mengukur sikap, pendapat, dan persepsi seseorang orang mengenai suatu fenomena saat memilih pakaian.

2.12 Pengujian *Black Box*

Pengujian *Black Box* berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program. Black Box Testing cenderung untuk menemukan hal-hal berikut:

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (interface errors).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (performance errors).
5. Kesalahan inisialisasi dan terminasi[16].

Saat ini terdapat banyak metoda atau teknik untuk melaksanakan Black Box Testing, antara lain:

1. Equivalence Partitioning

2. Boundary Value Analysis/Limit Testing
3. Comparison Testing
4. Sample Testing
5. Robustness Testing
6. Behavior Testing
7. Requirement Testing
8. Performance Testing
9. Uji Ketahanan (Endurance Testing)
10. Uji Sebab-Akibat (Cause-Effect Relationship Testing)

Alasan menggunakan Pengujian *Black Box* pada penelitian ini karena pengujian pada penelitian ini berfokus pada spesifikasi fungsional dari perangkat lunak. Pada pengujian Black Box dilakukan dengan mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak yang dibangun.