

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1 Sistem Hidroponik**

Sistem hidroponik adalah sistem budidaya menggunakan air yang mengandung nutrisi dan mineral tanpa tanah [12]. Hidroponik berasal dari bahasa latin, *hydro* dan *phonos*. *Hydro* artinya air dan *phonos* artinya kerja. Maka dalam konteks ini, hidroponik didefinisikan sebagai air kerja. Namun dalam bidang pertanian, hidroponik merupakan kegiatan pertanian yang menggunakan air sebagai media utama untuk menggantikan tanah. Maka para pelaku kegiatan hidroponik dapat dikatakan sebagai petani hidroponik [13]. Hidroponik juga bisa diartikan sebagai sistem tanam tanpa menggunakan media tanah. Yang membedakan metode bercocok tanam hidroponik dengan bercocok tanam konvensional adalah pada media bercocok tanamnya. Pada hidroponik digunakan media air sebagai media bercocok tanam, sedangkan metode konvensional menggunakan tanah [14].

Sistem hidroponik memiliki beberapa keunggulan dibandingkan dengan sistem budidaya konvensional, yaitu mengurangi risiko atau masalah budidaya yang berhubungan dengan tanah seperti gangguan serangga, jamur dan bakteri yang hidup di tanah. Sistem hidroponik juga lebih mudah dalam pemeliharaan seperti tidak melibatkan proses penyiangan dan pengolahan tanah dalam budidaya tanamannya [12].

#### **2.2 Tanaman Selada (*Lactuca sativa* L.)**

Selada (*Lactuca sativa* L.) termasuk dalam famili *compositae*. Bentuk selada bermacam-macam, ada yang membentuk krop dan ada yang membentuk “Rosette”. Tanaman selada berwarna hijau hingga putih kekuningan. Tanaman selada merupakan tanaman hortikultura yang bernilai ergonomis tinggi. Tanaman selada digemari oleh masyarakat karena memiliki kandungan gizi yang tinggi. Umumnya tanaman ini dipasarkan di supermarket atau di pasar tradisional.

Tanaman selada dapat dimakan secara mentah atau sebagai campuran berbagai olahan makanan seperti hamburger, hot dog, beer steak, atau masakan rumahan lainnya biasanya sebagai campuran salad. Selada memiliki ciri khas yaitu bunganya mengumpul dalam tandan membentuk rangkaian. Selada terdapat bermacam-macam varietas yaitu selada keriting hijau (green lollo), selada keriting ujung merah (lollo rosa), dan selada cos (romaine lettuce) [15].

Selada keriting hijau merupakan tanaman sayuran yang banyak ditanam di daerah timor. Sayuran ini paling sering ditanam dan dibudidayakan. Selada keriting memiliki ciri khas daunnya keriting mulai dari ujung sampai tepi daun, serta berwarna hijau.



**Gambar 2.1 Selada Keriting Hijau**

Selada keriting ujung merah merupakan jenis sayuran selada yang banyak dikonsumsi oleh masyarakat. Selada ujung merah memiliki ciri khas daun berwarna merah segar keunguan. Seorang peneliti di University of Glasgow menemukan kandungan dari lettuce lollo rossa mempunyai kandungan antioksidan 100 kali lebih banyak dibandingkan dengan selada yang pada umumnya.



**Gambar 2.2 Selada Keriting Ujung Merah**

Selada cos salah satu jenis memiliki ciri membentuk krop seperti selada kepala namun berbentuk lonjong dengan pertumbuhan meninggi, daunnya lebih tegak, dan kropnya berukuran besar dan padat. Terdapat kandungan gizi dalam selada cos diantaranya energi 72 KJ, karbohidrat 3,3 g, diet serat 2,1 g, lemak 0,3 g, protein 1,2 g, air 95 g, kalsium 33 mg, besi 0,97 mg, fosfor 30 mg, kalium 247 mg.

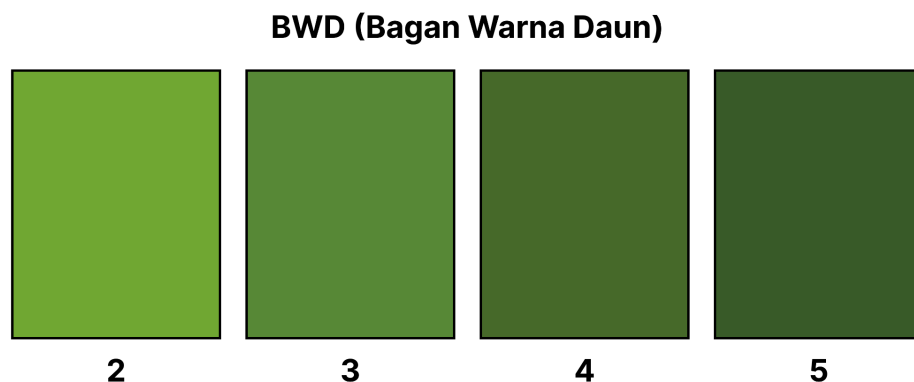


**Gambar 2.3 Selada Cos**

Selada keriting hijau dan selada keriting ujung merah termasuk dalam kultivar selada daun (*L. Sativa Crispa*). Kedua jenis selada ini memiliki tepian daun bergerigi dan berwarna hijau atau merah. Helaian pada kedua daun selada ini lepas dan memiliki ciri khas yaitu membentuk krop dengan umur genjah dan memiliki toleransi terhadap temperatur rendah. Kedua selada ini dapat dibedakan berdasarkan batang daun masing-masing. Selada keriting hijau memiliki daun dan batang berwarna hijau sedangkan selada keriting ujung merah memiliki daun dan batang berwarna merah. Pertumbuhan kedua selada ini membentuk krop yang lonjong. Pertumbuhan kedua selada ini dapat mencapai tinggi 25 cm dengan daun lebih tegak dan tidak menjuntai ke bawah seperti selada varietas lainnya.

### 2.3 Bagan Warna Daun

Untuk mengukur apakah tanaman kekurangan nitrogen atau tidak, petani umumnya menerka apakah warna hijaunya sudah pas atau belum. Kalau dirasa belum cukup, maka AB Mix akan terus ditambahkan. Kondisi demikian sering menyebabkan tanaman kelebihan nitrogen (warna daunnya terlalu hijau). Untuk menghindari hal demikian, terdapat bagan warna daun (BWD), yaitu bagan skala warna yang tersusun dari hijau kekuningan sampai hijau tua, sesuai dengan warna-warna daun di lapangan. Bagan warna daun dapat digunakan untuk mengukur warna daun dengan cara mencocokkannya. Bila suatu nilai warna daun lebih rendah dari batas kritis (warna daun sehat), maka tanaman memerlukan AB Mix tambahan [16].



**Gambar 2.4 Bagan Warna Daun**

Bagan warna daun ini merupakan alat yang cocok untuk mengoptimalkan penggunaan N, dengan berbagai sumber pupuk N; AB Mix. Bagan warna daun terdiri dari empat warna hijau, dari hijau kekuningan (No. 2 pada kartu) sampai hijau tua (No. 5 pada kartu). BWD tak dapat menunjukkan perbedaan warna hijau daun yang terlalu kecil sebagaimana pada klorofil meter (SPAD). Namun Bagan warna daun bisa dibandingkan dengan SPAD untuk menentukan ketepatan relatifnya dalam menentukan status N tanaman [17].

#### **2.4 Smartphone**

Menurut Williams dan Sawyer *smartphone* adalah telepon seluler dengan mikroprosesor, memori, layar dan modem bawaan. *Smartphone* merupakan ponsel multimedia yang menggabungkan fungsionalitas Personal Computer (PC) dan handset sehingga menghasilkan gadget yang mewah dimana terdapat pesan teks, kamera, pemutar musik, video, game, akses email, tv digital, search engine, pengelola informasi pribadi, fitur GPS, jasa telepon internet, dan bahkan terdapat telepon yang juga berfungsi sebagai kartu kredit [18]. Di Dalam penelitian ini *smartphone* digunakan untuk membantu aplikasi dalam mengakses perangkat keras yang terdapat pada sebuah sistem operasi Android untuk kemudian dapat memberikan data gambar ke server.

#### **2.5 Android**

Android merupakan sistem operasi berbasis Linux yang dikembangkan untuk perangkat mobile yang mencakup sistem operasi, middleware, aplikasi. Menurut Akhmad Dharma Kasman, “Android adalah sebuah sistem operasi telepon seluler dan komputer tablet layar sentuh (touchscreen) yang berbasis linux” [19].

Android merupakan sebuah sistem operasi yang bersifat Open Source artinya memberikan kebebasan bagi para pembuat ataupun pengembang untuk mengembangkan, memodifikasi, memperbaiki dan mendistribusikan perangkat lunaknya [20].

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*, *resource*, pemberitahuan (notifikasi), dan lain-lain.
3. *Libraries*, Android menyediakan sejumlah *Libraries* seperti grafis 2D dan 3D, *multimedia playback*, mesin pencari, pencetakan font, basis data, 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 juga perangkat keras.

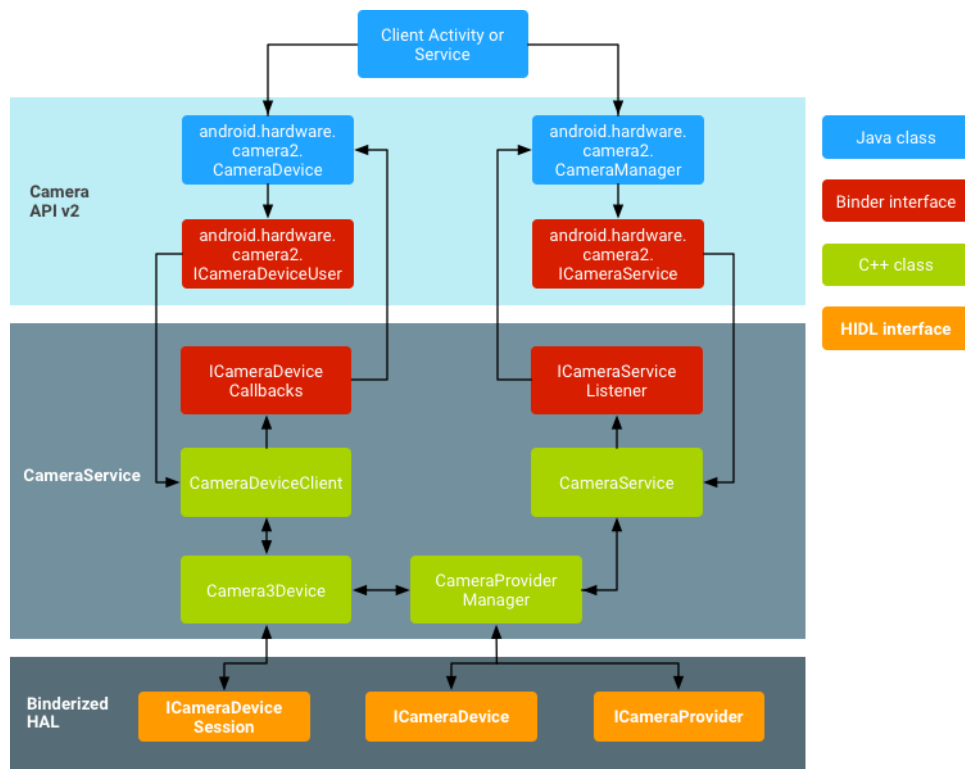
Alasan digunakannya Android pada penelitian ini karena Android menjadi sebuah sistem operasi perangkat bergerak (*mobile device*) yang paling banyak digunakan di dunia. Teknologi yang terdapat pada perangkat android juga bisa dimanfaatkan untuk membantu permasalahan yang sedang diteliti saat ini.

## 2.6 Kamera

*Hardware Abstraction Layer* (HAL) kamera Android menghubungkan API *framework* kamera tingkat yang lebih tinggi di Camera 2 ke driver kamera dan perangkat keras yang mendasarinya [21]. Android 8.0 memperkenalkan *Treble*, mengalihkan *CameraHal API* ke antarmuka stabil yang ditentukan oleh *HAL*

*Interface Description Language* (HIDL). Jika sebelumnya pengguna telah mengembangkan modul dan driver HAL kamera untuk Android 7.0 dan yang lebih rendah, perhatikan perubahan signifikan pada pipeline kamera.

Gambar 2.10 berikut menjelaskan komponen HAL dari arsitektur kamera.



**Gambar 2.5 Arsitektur Kamera**

### 1. app framework

Pada tingkat *app framework* adalah kode aplikasi, yang menggunakan API Kamera 2 untuk berinteraksi dengan perangkat keras kamera. Secara internal, kode ini memanggil antarmuka *Binder* yang sesuai untuk mengakses kode asli yang berinteraksi dengan kamera.

### 2. AIDL

Antarmuka binder yang terkait dengan *CameraService* dapat ditemukan di *frameworks/av/camera/aidl/android/hardware*. Kode yang dihasilkan memanggil kode asli tingkat yang lebih rendah untuk mendapatkan akses ke kamera fisik dan mengembalikan data yang digunakan untuk membuat objek *CameraDevice* dan akhirnya *CameraCaptureSession* di tingkat kerangka kerja.

### 3. native framework

*Framework* ini berada di *frameworks/av/* menyediakan *native* yang setara dengan kelas *CameraDevice* dan *CameraCaptureSession*. Lihat juga referensi *NDK camera2*.

### 4. binder IPC interface

Antarmuka *binder IPC* memfasilitasi komunikasi melalui batas-batas proses. Ada beberapa kelas *binder* kamera yang terletak di direktori *frameworks/av/camera/camera/aidl/android/hardware* yang memanggil *camera service*. *ICameraService* adalah antarmuka ke *camera service*; *ICameraDeviceUser* adalah antarmuka ke perangkat kamera tertentu yang dibuka; dan *ICameraServiceListener* dan *ICameraDeviceCallbacks* adalah masing-masing callback *CameraService* dan *CameraDevice* ke framework aplikasi.

### 5. camera service

*camera service*, terletak di *frameworks/av/services/camera/libcameraservice/CameraService.cpp*, adalah kode aktual yang berinteraksi dengan HAL.

### 6. HAL

*Hardware Abstraction Layer* mendefinisikan antarmuka standar yang dipanggil oleh *camera service* dan yang harus pengembang terapkan agar perangkat keras kamera pengguna berfungsi dengan benar.

## 2.7 Aplikasi

Menurut *Cambridge Dictionary*, Aplikasi merupakan sebuah program komputer yang dirancang untuk tujuan tertentu. Menurut Ali Zaki dan Smitdev Community, Aplikasi merupakan komponen yang bermanfaat sebagai media untuk menjalankan pengolahan data ataupun berbagai kegiatan lainnya seperti pembuatan ataupun pengolahan dokumen dan file. Menurut Yuhefizar, Aplikasi adalah program yang sengaja dibuat dan dikembangkan sebagai pemenuh kebutuhan penggunaannya dalam menjalankan suatu pekerjaan tertentu [22].

Dari pengertian diatas peneliti dapat menyimpulkan bahwa aplikasi adalah sebuah program yang dibuat untuk suatu tujuan tertentu agar dapat menyelesaikan



permasalahan baik umum maupun khusus guna memenuhi kebutuhan pengguna dalam melakukan kegiatannya. Hal ini didasarkan pada permasalahan yang bisa saja ditemui bersifat unik sehingga tidak ada program yang sesuai untuk digunakan. Oleh karena itu dengan terpaksa diperlukan pengembangan program aplikasi itu sendiri.

## 2.8 Android Studio

Android Studio merupakan IDE (Integrated Development Environment) resmi untuk melakukan pengembangan aplikasi android yang bersifat open source. Android Studio pertama kali diluncurkan oleh Google pada 16 Mei 2013 saat event Google I/O untuk tahun 2013. Sejak saat itu, Android Studio mulai digunakan dan menggantikan IDE resmi sebelumnya yaitu Eclipse untuk mengembangkan aplikasi android [18].

Android Studio dikembangkan berdasarkan IntelliJ IDEA yang hampir sama dengan Eclipse tetapi disertai dengan plugin ADT (Android Development Tools). Diantaranya Android Studio memiliki fitur:

1. Proyek yang berbasis pada *Gradle Build*.
2. *Refractory* dan pengecekan bug yang cepat.
3. Memiliki tools baru yang bernama *Lint* dan diklaim dapat memonitor kecepatan, guna, serta kompatibilitas aplikasi dengan cepat.
4. Mendukung *Proguard* dan *App-Signing* yang berguna untuk keamanan.
5. Memiliki GUI (*Graphical User Interface*) untuk aplikasi android yang lebih mudah.
6. Didukung oleh *Google Cloud Platform* untuk setiap aplikasi yang dikembangkan.

## 2.9 Flutter

*Flutter* merupakan suatu teknologi untuk membangun mobile apps yang dibuat oleh *Google*. *Flutter* merupakan *framework* yang menggunakan bahasa pemrograman *Dart* yang juga dibuat oleh *Google* yang juga merupakan bahasa dengan *general purpose*, artinya *Dart* mampu berjalan di banyak platform atau

biasa disebut *multi-platform*. Begitupun dengan *Flutter* sendiri, *Flutter* merupakan *cross-platform framework*, yang berarti aplikasi dapat digunakan untuk lebih dari satu platform [23].

*Flutter* ditujukan untuk mempermudah dan mempercepat dalam proses pengembangan aplikasi *mobile* yang dapat berjalan di Android dan iOS sekaligus. Dengan menggunakan *Flutter*, para pengembang dapat membangun aplikasi Android dan iOS dengan menggunakan bahasa yang sama yaitu *Dart*, sehingga para pengembang pun hanya perlu mempelajari satu bahasa pemrograman saja [24].

## 2.10 Dart

*Dart* adalah bahasa pemrograman yang sudah dioptimalkan bagi klien untuk mengembangkan aplikasi cepat di platform apa pun. Tujuan dari *Dart* adalah untuk menawarkan bahasa pemrograman yang paling produktif untuk pengembangan aplikasi *multi-platform*, dipasangkan dengan platform *runtime* eksekusi yang fleksibel untuk *framework* aplikasi [25].

*Dart* dirancang untuk ruang lingkup teknis yang sangat cocok untuk pengembangan di sisi klien, memprioritaskan kualitas pengembangan yang cepat (*sub-second stateful hot reload*) dan pengalaman hasil produksi berkualitas tinggi di berbagai target kompilasi (web, mobile, dan desktop) [26].

## 2.11 API

API (Application Programming Interface) merupakan *interface* yang bisa menghubungkan satu aplikasi dengan aplikasi lainnya. Peran API adalah sebagai perantara dari berbagai aplikasi yang berbeda, baik dalam satu platform yang sama ataupun lintas platform [5]. API menggunakan protokol HTTP untuk kerjasama antara program yang berbeda yaitu layanan web (REST atau SOAP). Pada dasarnya, interaksi API adalah tipe permintaan dan respons antara klien dan server.

API adalah mekanisme yang memungkinkan dua komponen perangkat lunak untuk berkomunikasi satu sama lain menggunakan seperangkat definisi dan

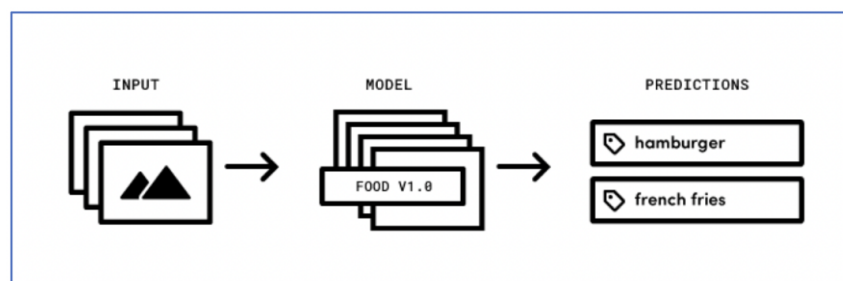
protokol. Misalnya, sistem perangkat lunak biro cuaca berisi data cuaca harian. Aplikasi cuaca di ponsel "berbicara" dengan sistem ini melalui API dan menunjukkan pembaruan cuaca harian di ponsel.

## 2.12 Clarifai

Clarifai adalah alat yang dapat mengidentifikasi atau mengenali gambar atau video yang dimasukkan sebagai input sehingga dapat memberikan hasil berupa prediksi tentang apa yang ada di dalam gambar atau video tersebut berupa nilai besaran probabilitas kemungkinannya [27].

Clarifai menawarkan platform terdepan untuk membangun solusi perangkat lunak bertenaga AI. Clarifai mendukung siklus hidup pengembangan AI secara penuh; termasuk persiapan kumpulan data, pelatihan model, dan penerapan. Clarifai mengkhususkan diri dalam model *deep learning* yang digunakan untuk memahami data gambar, video, teks, dan audio yang tidak terstruktur. Clarifai membantu pelanggan sektor publik dan perusahaan memecahkan kasus penggunaan yang kompleks melalui klasifikasi objek, deteksi, pelacakan, geolokasi, pencarian visual, dan pemrosesan bahasa alami [28].

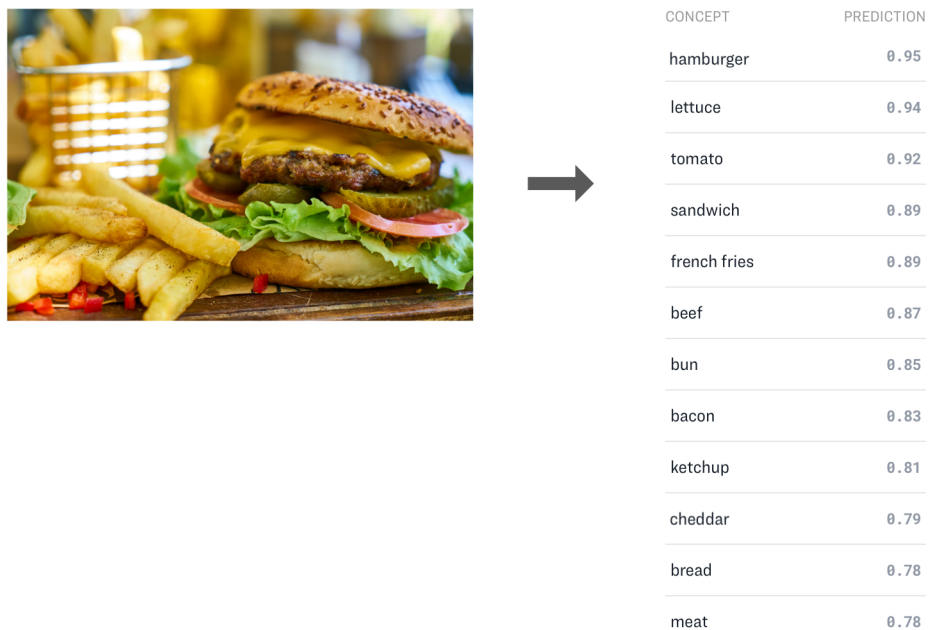
Inti dari teknologi Clarifai didasarkan pada convolutional neural networks, yaitu proses yang memungkinkan komputer untuk belajar dari contoh data dan menarik kesimpulannya sendiri, memberikan kemampuan pada aplikasi untuk memprediksi tag atau nama yang benar pada gambar atau video yang akan dideteksi.



**Gambar 2.6 Clarifai API's workflow**

Teknologi clarifai bergantung pada pemakaian Convolutional Neural Networks ( CNN) guna memproses foto, setelah itu menghasilkan daftar tag yang

mendeskripsikan foto tersebut. Pendekatan Clarifai pertama kali diusulkan untuk tantangan klasifikasi ImageNet pada tahun 2013, Clarifai bekerja lewat analisis foto guna menciptakan daftar label deskriptif yang mewakili foto yang diberikan. Untuk tiap tag dalam catatan ini, sistem pula membagikan nilai probabilitas. Probabilitas ini mewakili nilai kemungkinan gambar menggunakan tag spesifik. API Clarifai dapat diakses sebagai layanan web jarak jauh [29].



**Gambar 2.7 Contoh Prediksi Clarifai**

Penggunaan API Clarifai pada penelitian ini adalah pada penelitian ini memanfaatkan API Clarifai untuk mendeteksi tanaman dari hasil foto yang dimasukkan oleh pengguna. Hasil dari API Clarifai ini akan digunakan sebagai data masukan untuk aplikasi yang akan dibuat.

### 2.13 JSON

JSON adalah format pertukaran data yang ringan. JSON adalah format teks yang sepenuhnya independen tetapi menggunakan konvensi yang familiar dengan bahasa pemrograman dari keluarga C, termasuk C, C++, C#, Java, JavaScript, Perl, dan Python [29].

JSON dibangun dalam dua struktur, yaitu :

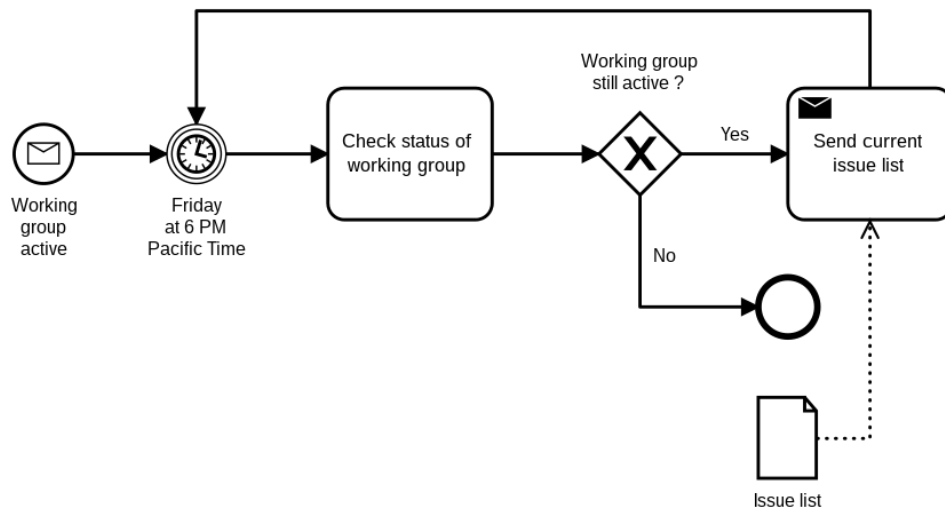
1. Beberapa pasangan nama/nilai. Dalam beberapa bahasa pemrograman, biasanya disebut sebagai objek, record, struct, tabel hash, key list atau array asosiatif.
2. Nilai-nilai tersebut disusun secara berurutan. Biasa disebut array, vektor, list atau daftar dalam bahasa pemrograman.

JSON adalah struktur data universal, dapat 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 struktur JSON dasar.

## **2.14 BPMN**

*Business Process Model and Notation* (BPMN) adalah notasi grafis yang menggambarkan langkah-langkah dalam proses bisnis. BPMN menggambarkan aliran end-to-end dari proses bisnis. Notasi telah dirancang secara khusus untuk mengkoordinasikan urutan proses dan pesan yang mengalir di antara peserta proses yang berbeda dalam serangkaian aktivitas yang terkait [30]. BPMN akan memberikan kemampuan kepada bisnis untuk memahami prosedur bisnis internal mereka dalam notasi grafis dan akan memberi organisasi kemampuan untuk mengomunikasikan prosedur ini dengan cara yang standar.

BPMN ditargetkan pada tingkat tinggi untuk pengguna bisnis dan pada tingkat yang lebih rendah untuk pelaksana proses. Pengguna bisnis harus dapat dengan mudah membaca dan memahami diagram proses bisnis BPMN. Proses pelaksana harus dapat menghiasi diagram proses bisnis dengan lebih detail untuk mewakili proses dalam implementasi fisik. BPMN ditujukan untuk pengguna, vendor, dan penyedia layanan yang perlu mengkomunikasikan proses bisnis secara standar.


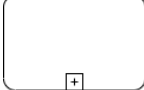










**Gambar 2.8 Contoh BPMN Untuk Proses Dengan Aliran Normal**

Dunia proses bisnis telah berubah secara dramatis selama beberapa tahun terakhir. Proses dapat dikoordinasikan dari belakang, di dalam dan di atas batas-batas alami organisasi. Sebuah proses bisnis sekarang mencakup banyak peserta dan koordinasi bisa menjadi kompleks. Hingga BPMN, belum ada teknik pemodelan standar yang dikembangkan untuk mengatasi masalah ini. BPMN telah dikembangkan untuk memberikan notasi bebas royalti kepada pengguna. Ini akan menguntungkan pengguna dengan cara yang sama di mana UML menstandarisasi dunia rekayasa perangkat lunak. Akan ada kursus pelatihan, buku, dan kumpulan pengetahuan yang dapat diakses pengguna untuk mengimplementasikan proses bisnis dengan lebih baik.

**Tabel 2.1 Simbol Diagram BPMN**

Simbol	Nama	Deskripsi
	Start	Bertindak sebagai pemicu proses.
	Intermediate	Mewakili sesuatu yang terjadi antara awal dan akhir peristiwa.
	End	Mewakili hasil dari suatu proses.

	Task	Sebuah tugas mewakili satu unit kerja yang tidak atau tidak dapat dipecah ke tingkat lebih lanjut dari detail proses bisnis.
	Sub Process	Digunakan untuk menyembunyikan atau mengungkapkan tingkat detail proses bisnis tambahan.
	Transaction	Suatu bentuk sub-proses di mana semua aktivitas yang terkandung harus diperlakukan secara keseluruhan.
	Call Activity	Titik dalam proses di mana proses global atau Tugas global digunakan kembali.
	Exclusive	Digunakan untuk membuat aliran alternatif dalam suatu proses.
	Inclusive	Digunakan untuk membuat aliran alternatif di mana semua jalur dievaluasi.
	Parallel	Dua proses paralel dimulai berdasarkan suatu peristiwa, tetapi tidak ada evaluasi terhadap peristiwa tersebut.
	Sequence Flow	Sequence Flow diwakili dengan garis padat dan panah, dan menunjukkan urutan aktivitas yang dilakukan.
	Message Flow	Alur Pesan diwakili dengan garis putus-putus, lingkaran terbuka di awal, dan panah terbuka di akhir.
	Association	Sebuah Asosiasi diwakili dengan garis putus-putus.

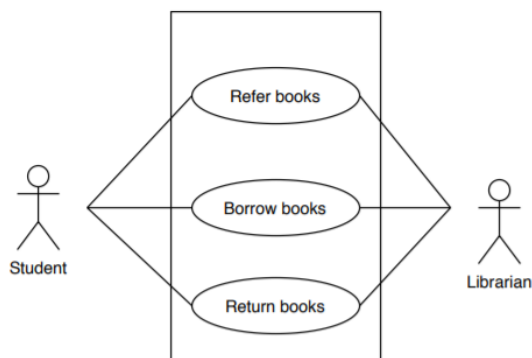
## 2.15 UML (*Unified Modeling Language*)

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 [23].

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. Sebuah use case dapat memasukkan 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 2.9 Contoh Use Case Diagram**

### 2. *Class Diagram*

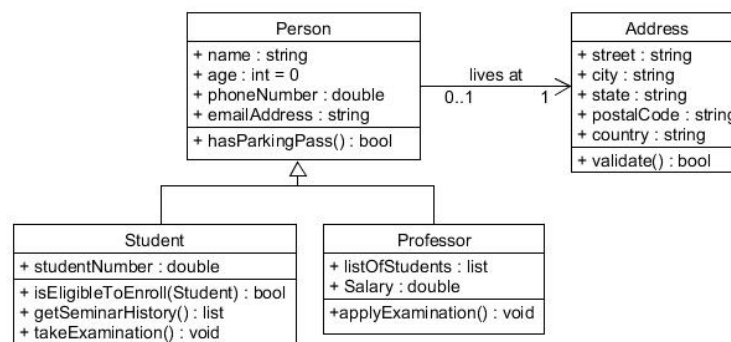
Class diagram menggambarkan struktur dan deskripsi class, package dan objek serta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. Class memiliki tiga area pokok:



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 method bisa memiliki sifat sifat berikut:

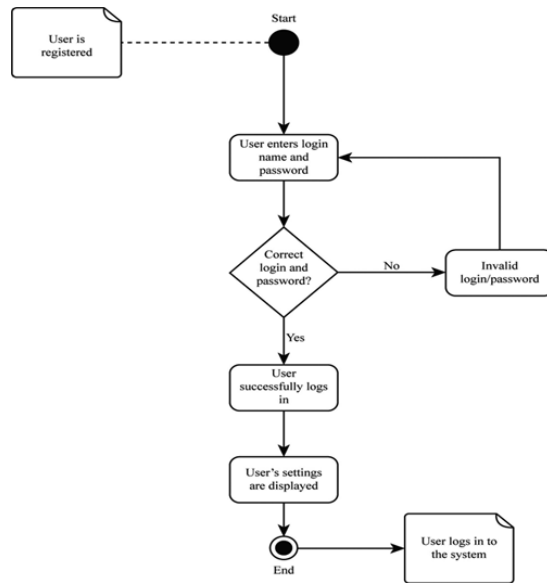
1. *Private*, hanya bisa dipanggil oleh Class yang bersangkutan, tidak bisa dipanggil oleh Class lain yang tidak bersangkutan.
2. *Public*, bisa di panggil oleh seluruh Class.
3. *Protected*, hampir sama sifatnya dengan *private* dan hanya bisa diakses oleh Class turunannya.



**Gambar 2.10 Contoh Class Diagram**

### 3. Activity Diagram

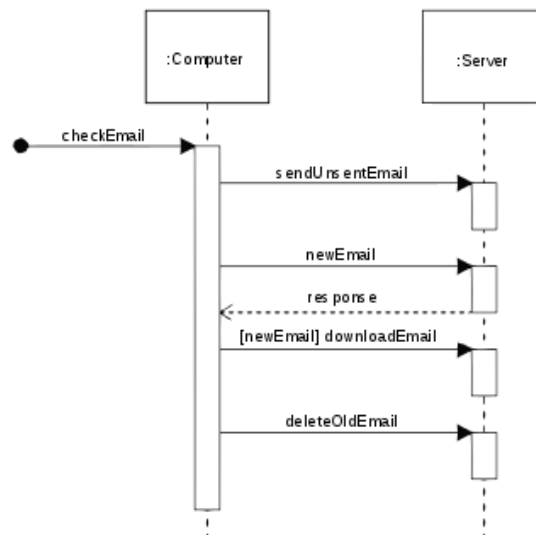
*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, pemilihan keputusan yang mungkin terjadi. Pada Gambar 2.15 adalah contoh dari *Activity Diagram*.



**Gambar 2.11 Contoh Activity Diagram**

#### 4. Sequence Diagram

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



**Gambar 2.12 Contoh Sequence Diagram**