

## **BAB 2 TINJAUAN PUSTAKA**

Dalam BAB ini berisikan berbagai konsep dasar dan teori-teori yang berkaitan dengan pembangunan APLIKASI CHATBOT DIAGNOSA PENYAKIT TANAMAN PADI DAN REKOMENDASI OBAT MENGGUNAKAN TENSORFLOW.

### **2.1 Tanaman Padi**

Tanaman Padi merupakan tanaman semusim yang berumur pendek kurang dari 1 tahun[8]. Akarnya yang serabut , tinggi batang yang beragam, berbatang bulat dan berongga yang disebut jerami. Padi memiliki bunga yang mana terdiri dari tangkai, kelopak, palea, putik,kepala putik, tangkai sari, kepala sari, dan bulu. Ciri khas daun tanaman padi yaitu adanya lidah dan telinga daun , hal ini menyebabkan daun tanaman padi dapat dibedakan dari jenis rumput lain. Padi sendiri memiliki beberapa bagian yakni Helaiian daun, Upih / pelepah daun dan Lidah daun[9]. Padi sendiri mempunyai fase – fase pertumbuhan yakni ada tiga fase yaitu Vegetatif ( Awal sampai pembentukan malai), dua Reproduksi ( pembentukan mulai sampai pembungaan dan yang terakhir Pematangan (Pembungaan sampai gabah matang) (Arafah, 2009). Ketika gabah matang dan padi mulai di panen kemudian jadilah beras yang menjadi bahan pokok makanan masyarakat Indonesia.

### **2.1 Aplikasi**

Aplikasi dapat diartikan sebagai suatu program dimana digunakan untuk melaksanakan suatu fungsi atau tugas bagi pengguna nya. Menurut kamus Komputer ekeskutif, aplikasi memiliki suatu pengertian yakni pemecah masalah yang menggunakan suatu tehnik pemrosesan data aplikasi yang mengacu pada sebuah komputasi yang diharapkan maupun pemrosesan nya. Pengertian Aplikasi sendiri menurut Kamus Besar Bahasa Indonesia yakni penerapan dari sebuah rancangan sistem

yang digunakan untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu[10].

## 2.2 Diagnosa

Menurut kamus besar bahasa Indonesia, Diagnosa merupakan suatu penentuan jenis penyakit dengan cara meneliti dengan melihat gejala – gejalanya[11]. Adapun didalam Wikipedia, diagnosis merupakan suatu penentuan kondisi kesehatan seseorang yang sedang dialami sebagai dasar pengambilan keputusan medis untuk prognosis dan pengobatan[12].

## 2.3 Chatbot

Chatbot merupakan suatu aplikasi teknologi secara virtual yang dibuat layaknya dua orang yang sedang bercakap – cakap namun yang satu adalah sebuah robot chat yang memiliki suatu tujuan tertentu dengan contoh sebagai hiburan. Chatbot sendiri memiliki banyak fungsi nya bisa sebagai teman chat biasa atau ditujukan dengan pengetahuan. Yang mana chat bot ini dilengkapi oleh suatu sistem Artificial Intelegence, Machine Learning, maupun bahasa yang digunakan agar terlihat mirip dengan percakapan aslinya yaitu menggunakan Natural language processing. [13] Chatbot pada dasarnya memiliki 2 komponen utama yaitu Chat yang dapat diartikan sebagai pembicaraan dan Bot merupakan sebuah program yang mengandung sejumlah data, jika diberikan masukan maka akan memberikan jawaban. Chatbot dapat menjawab pertanyaan dengan membaca tulisan yang diketikkan oleh pengguna melalui keyboard [14]

## 2.4 Penyakit Tanaman Padi

Didalam sebuah situs / berita online *liputan6.com* menyebutkan terdapat 8 hama dan penyakit yang sering menyerang pada tanaman padi yaitu diantaranya :

- a. Tikus dan Tungro

Tikus merupakan suatu hama yang memiliki mobilitas yang tinggi. Tikus biasanya menyerang tanaman padi dan memakan bulir – bulir padi sehingga tikus dapat merusak tanaman padi tersebut. Tungro merupakan suatu penyakit yang disebabkan oleh virus diantaranya virus Rice Tungro Bacilliform Virus dan Rice Tungro Spherical Virus. Kedua virus tersebut menginfeksi tanaman secara bersamaan karena tidak memiliki kekerabatan serologi.

b. Ulat dan Penyakit HDB

Ulat merupakan suatu hama yang sama halnya seperti tikus namun objek yang dimakan nya berbeda. Ulat tersebut memakan daun – daun padi yang dimana terjadi pada fase setelah tanam namun belum terbentuknya bulir padi. Adapun HDB didefinisikan yaitu Hawar Daur Bakteri atau dengan kata lain penyakit ini disebabkan oleh bakteri yang menyerang daun pada tanaman padi. Bakteri tersebut diantaranya *X.oryzae* pv. *oryzae* atau *X. Compestris* pv.*oryzae*. Penyakit ini bisa timbul pada musim kemarau atau pun musim hujan.

c. Walang sangit dan wereng

Walang sangit merupakan suatu hama yang cukup meresahkan dan punya mobilitas yang tinggi pula dalam merusak tanaman padi dikarenakan hama ini bisa terbang atau pun loncat dari batang satu ke batang tanaman padi yang lain. Adapun wereng merupakan suatu hama yang menyerang daun ataupun batang sehingga berakibat tanaman padi tersebut kemungkinan mati.

d. Burung dan busuk batang

Burung merupakan suatu hama yang cukup meresahkan bagi petani , karena burung juga merupakan hama yang memiliki mobilitas yang tinggi juga karena dating nya berkelompok dan dapat terbang dari batang satu ke batang yang lainnya. Burung

tersebut memakan bulir padi yang datang pada masa panen yang semakin dekat. Adapun busuk batang merupakan Busuk batang merupakan dan menyebabkan tanaman menjadi mudah rebah. Gejala awal berupa bercak berwarna kehitaman serta bentuknya tidak teratur pada sisi luar pelapah daun dan secara bertahap membesar. [15]

## 2.5 Tensorflow Api

TensorFlow Object Detection API adalah open source framework yang dapat digunakan untuk mengembangkan, melatih, dan menggunakan model deteksi objek. Sistem ini sudah banyak diterapkan pada berbagai produk Google antara lain pencarian image, deteksi wajah dan plat nomor kendaraan pada Google Streetview, Google Assistant, Waymo atau Self Driving Car, dan lain-lain (“Tensorflow Object Detection API,” n.d.) API ini juga menyediakan lima model berbeda yang masing-masing mempunyai trade-off antara kecepatan dan akurasi dalam mendeteksi objek. Model SSD dengan menggunakan MobileNet dapat berjalan dengan komputasi ringan, sehingga dapat dijalankan secara real time di perangkat seluler (Liu et al., 2016). Sedangkan model Faster-RCNN lebih berat secara komputasi, tetapi menghasilkan pendeteksian yang jauh lebih akurat (Huang et al., 2017). Tensorflow detection model terdiri dari yang pertama yakni Model deteksi yang bisa dilatih, Frozen weight, Jupyter Notebook, dan yang terakhir Local Training Steps.[16]

## 2.6 Bahasa Pemrograman Phyton

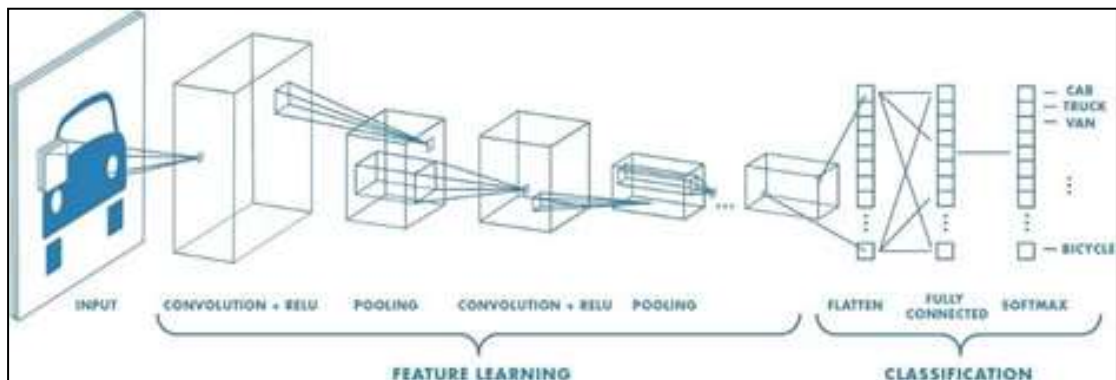
Phyton merupakan bahasa pemrograman yang kemampuan, menggabungkan kapabilitas, dan sintaksis kode yang sangat jelas, dan juga dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Walaupun Python tergolong bahasa pemrograman dengan level tinggi, nyatanya Python dirancang sedemikian rupa agar mudah dipelajari dan dipahami. Python sendiri dikembangkan oleh Guido Van Rossum, yaitu seorang programmer yang berasal dari Belanda. Pembuatannya berlangsung di kota Amsterdam, Belanda pada tahun 1990. Pada tahun 1995 Python dikembangkan lagi agar lebih kompatibel oleh Guido Van Rossum. Selanjutnya pada awal tahun 2000, terdapat pembaharuan versi Python hingga mencapai Versi 3 sampai saat ini. Pemilihan nama Python sendiri diambil dari sebuah acara televisi yang lumayan terkenal yang bernama Mothy Python Flying Circus yang merupakan acara sirkus favorit dari Guido van Rossum.[17]

## 2.7 Convolutional Neural Network (CNN)

Convolutional Neural Network adalah salah satu metode machine learning dari pengembangan Multi Layer Perceptron (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis Deep Neural Network karena dalamnya tingkat jaringan dan banyak diimplementasikan dalam data citra. CNN memiliki dua metode yakni klasifikasi menggunakan feedforward dan tahap pembelajaran menggunakan backpropagation. Cara kerja CNN memiliki kesamaan pada MLP, namun dalam CNN setiap neuron dipresentasikan dalam bentuk dua dimensi, tidak seperti MLP yang setiap neuron hanya berukuran satu dimensi.

Convolutional Neural Network (CNN) merupakan salah satu metode Deep learning (DL) yang dapat digunakan untuk mendeteksi dan mengenali sebuah objek pada sebuah citra digital.

©©



**Gambar 2.0.1 Proses CNN**

CNN adalah jaringan saraf yang berisi beberapa layer di dalamnya diantaranya adalah convolutional layer, pooling layer, activation layer. Convolutional layer bertugas untuk memeriksa pola dari suatu gambar dengan cara memberi filter lalu menghasilkan bobot berupa vector dan bobot tersebut akan bernilai tinggi jika dianggap melihat pola sebelumnya. Kombinasi bobot tinggi dari berbagai filter memungkinkan jaringan memprediksi konten suatu gambar. Lalu setelah memasuki tahap convolutional layer jaringan akan memasuki pooling layer yang berguna untuk mengurangi spatial dimension tetapi tidak untuk kedalamannya. Cara kerja pooling layer adalah dengan cara membagi pooling menjadi max-pooling dengan output maksimum nilai dari sub-region dan average-pooling dengan output rata-rata nilai dari sub-region[18]

## 2.8 Keras

Keras adalah Deep Learning API tingkat tinggi yang membuatnya sangat mudah untuk melatih dan menjalankan jaringan saraf. Keras dapat berjalan di atas TensorFlow, Theano, atau Microsoft Cognitive Toolkit (sebelumnya dikenal sebagai CNTK). Keras adalah API Deep Learning tingkat tinggi yang memungkinkan untuk membangun, melatih, mengevaluasi, dan mengeksekusi semua jenis jaringan saraf dengan mudah.[19]



**Gambar 2.0.2 Logo Keras**

## 2.9 Telegram Aps

*Telegram Messenger* adalah aplikasi *messaging* yang berfokus pada kecepatan, keamanan, sederhana, dan dapat diunduh secara gratis. Telegram dapat digunakan di semua perangkat dalam satu akun dalam waktu yang sama. Telegram juga dapat mengirim pesan, foto, video, dan beberapa jenis *file* (doc, zip, mp3, dll), serta dapat membuat group hingga 5000 orang atau *channel* untuk *broadcasting* untuk khalayak terbatas. Beberapa fitur lain yang terdapat dalam *Telegram Messenger* :

- a. Enkripsi pesan, personal dan *bussiness secret*.
- b. Menghapus pesan secara otomatis dengan *timer*.
- c. Menyimpan media di dalam *cloud*.
- d. Membangun sistem/*tools* sendiri dengan menggunakan API telegram.



**Gambar 2.0.3 Logo Telegram**

Salah satu fitur telegram yang berbeda dengan aplikasi *messenger* lainnya adalah Telegram menyediakan API (*Application Programming Interface*) yang terbuka 100% untuk publik yang ingin mengembangkan aplikasi menggunakan API Telegram.[20]

#### 2.10 Telegram API

Telegram mempunyai dua jenis APIs untuk *developer*, yaitu:

a. Bot API

Bot API memungkinkan *developer* untuk menghubungkan Bot dengan sistem Telegram. Telegram Bots adalah sebuah akun khusus yang tidak memerlukan nomor telepon tambahan dalam pengaturannya. Akun ini berfungsi sebagai antarmuka untuk tempat berjalannya kode di pada suatu *server*.





**Gambar 2.0.4 Logo Bot API (Bot Father)**

Botfather Telegram adalah salah satu *tools* resmi dari telegram yang digunakan untuk membuat atau menciptakan *bot* telegram dimana nantinya Botfather tersebut akan mengeluarkan token API yang digunakan sebagai kunci mengelola *bot* yang akan dipakai pada kode program.[21]

b. Telegram API

Telegram API memungkinkan *developer* untuk membangun sendiri Telegram *clients* yang diinginkan. Telegram API terbuka 100% untuk semua *developer* yang ingin membuat aplikasi dengan *platform* Telegram.

2.11 Docker

Docker adalah platform perangkat lunak yang memungkinkan Anda membuat, menguji, dan menerapkan aplikasi dengan cepat. Docker mengemas perangkat lunak ke dalam unit standar yang disebut kontainer yang memiliki semua yang diperlukan perangkat lunak agar dapat berfungsi termasuk pustaka, alat sistem, kode, dan waktu proses. Dengan menggunakan Docker maka dengan cepat menerapkan dan menskalakan aplikasi ke lingkungan apa pun dan yakin bahwa kode akan berjalan.



**Gambar 2.0.5 Logo Docker**

\

Docker tersedia di berbagai sistem operasi seperti Windows, Macintosh dan Linux. Untuk versi linux sendiri docker mensupport Ubuntu, Debian, Fedora, CentOS, dan Raspbian. Pada Ubuntu minimal versi sistem operasi yang dibutuhkan adalah versi 16.04 (LTS).

#### 1. Docker Compose

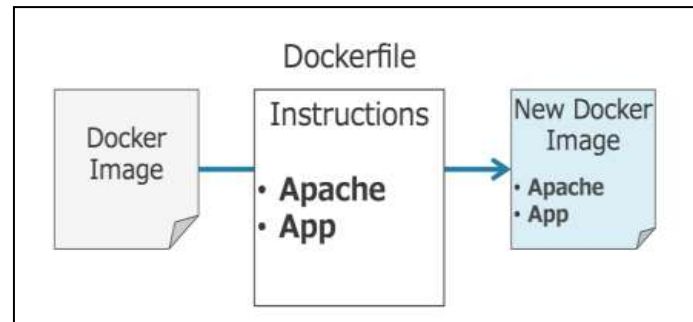
Docker-compose adalah sebuah alat dari docker yang digunakan untuk mendefinisikan dan menjalankan aplikasi multi kontainer. Dengan docker compose kita bisa menjalankan kontainer 1 dengan yang lainnya dengan 1 perintah. Docker-compose juga menggunakan yaml file untuk menyimpan konfigurasi dari service yang dibuat.

#### 2. Docker File

Dockerfile merupakan kumpulan script atau intruksi perintah berbasis *command line* yang digunakan untuk mengunduh dan menjalankan aplikasi secara otomatis pada *docker container*. Pembentukan Dockerfile merupakan tahap awal untuk membangun sebuah komponen docker image yang nantinya berguna untuk menjalankan aplikasi di environment *development* atau *production*. Setiap dockerfile yang dibuat nantinya akan dikemas kedalam sebuah file teks berekstensi Dockerfile untuk dijadikan docker image, dalam hal ini proses continuous delivery berguna untuk mengotomatisasi pembuatan docker image tanpa adanya campur tangan *software engineer*.

Hasil dari pengemasan dockerfile yaitu adalah sebuah *docker image* yang nantinya akan dipakai sesuai dengan kebutuhan. Tahapan pengemasan dockerfile berawal dari pemilihan docker image dasar yang digunakan hal ini sangat penting untuk dilakukan dikarenakan kita akan membuat landasan paling awal dockerfile, landasan tersebut bisa

berupa sistem operasi, bahasa pemrograman dan lain sebagainya. Setelah kita melakukan pemilihan docker image dasar kita akan diwajibkan dengan memasukan instruksi – instruksi yang dibutuhkan pada pembangunan *docker image* menggunakan dockerfile, instruksi – instruksi berupa instalasi software maupun konfigurasi yang dibuthkan. Hasil akhir dari pengemasan dockerfile adalah sebuah *docker image*.



**Gambar 2.0.6 Proses Pengemasan Dockerfile**

### 3. Docker Container

*Docker Container* dapat berjalan diatas *docker images* dan juga dapat melakukan sath atau lebih proses yang berjalan pada *docker images*. *Docker container* sangat berbeda dengan *docker image*, karena *docker container* berbentuk sebuah *docker image* yang dapat dikemas, dibaca dan ditulis (*writable*). *Layer* baru diatas dasar *docker image* akan terbentuk disaat ada perubahan yang disimpan dalam *docker container*.

Di dunia nyata proses pengiriman barang melalui kontainer atau *shipping container* merupakan ide awal yang mencetus adanya *docker container* saat ini. Bila dianalogikan dalam istilah *shipping container*, setiap kontainer berisi aplikasi perangkat lunak yang berarti seperti halnya kargo, kemudian kontainer tersebut dapat diberikan intruksi (*command*) untuk melakukan *created*, *started*, *stopped*, *restarted* dan *destroyed*.



**Gambar 2.0.7 Skema Docker Container**

#### 4. Docker Image

*Docker Image* adalah template yang bersifat *read only* (hanya bisa membaca saja) dan bertujuan untuk mendefinisikan docker container. *Docker image* yang berisi sistem operasi, aplikasi perangkat lunak maupun kode yang akan pada *docker container* anda. Docker image sendiri bisa digunakan menjadi *source code* untuk *docker container* yang mana sangat portabel.

Kode yang terdapat pada *docker image* merupakan format berlapis menggunakan *union file system* yang dibangun langkah demi langkah menggunakan beberapa instruksi, seperti:

##### 1. Add a file

Add a file adalah salah satu fitur yang terdapat di dalam docker image yang berguna untuk melakukan copy file dari folder aplikasi kedalam *docker image*.

##### 2. Run a command

Didalam *docker image* anda dapat melakukan proses menjalankan aplikasi dengan menggunakan attribute CMD atau RUN pada instruksi *docker image*.

##### 3. Open a v port

Didalam *docker image* anda dapat memunculkan port tertentu (*expose port*) sesuai dengan aplikasi yang dijalankan didalam *docker image*. [22]

## 2.13 Object Oriented Programming

Pemrograman berorientasi objek adalah paradigma pemrograman yang didasarkan pada konsep "objek", yang berisi data, dalam bentuk *field* atau dikenal juga sebagai atribut; serta kode, dalam bentuk fungsi atau prosedur atau dikenal juga sebagai *method*. Semua data dan fungsi pada paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya,[14]

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik peranti lunak skala besar. Lebih jauh lagi, pendukung *OOP* mengklaim bahwa *OOP* lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan *OOP* lebih mudah dikembangkan dan dirawat.

### 2.13.1 Objek

Objek adalah pembungkus data dan berfungsi bersama menjadi satu kesatuan suatu unit dalam sebuah program komputer; objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.

### 2.13.2 Kelas

Kelas merupakan kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh '*class of dog*' adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah *class* adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi object. Sebuah *class* secara tipikal sebaiknya dapat dikenali oleh seorang *non-programmer* sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah *class* sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak

menggunakan *OOP*). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.

#### **2.14 UML**

*Unified Modeling Language* (UML) merupakan kumpulan struktur ataupun teknik yang digunakan untuk memodelkan desain program berorientasi objek (*OOP*). UML merupakan metode yang digunakan untuk mengembangkan sistem program berorientasi objek (*OOP*) dan sekelompok perangkat tool untuk pengembangan sistem tersebut. UML 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*.

#### **2.15 Obat Tanaman Padi**

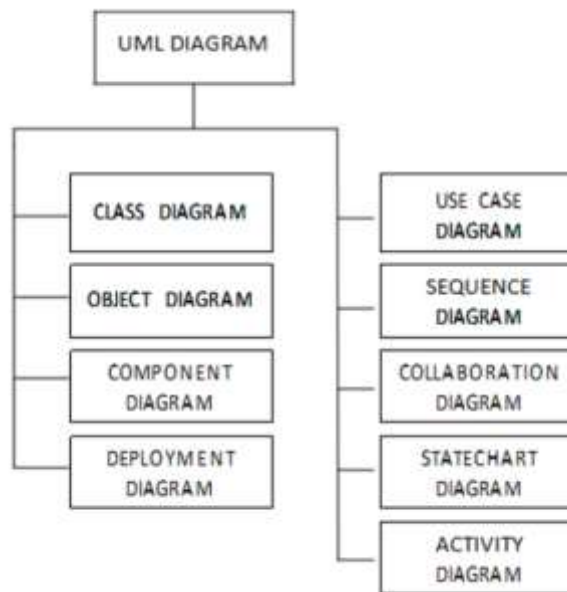
Obat Tanaman Padi merupakan Obat yang dianjurkan untuk diberikan ke tanaman padi yang berpenyakit atau menimbulkan gejala – gejala atau tanda – tanda tanaman padi itu terserang penyakit agar cepat diatasi atau tidak menimbulkan efek yang menyebar ke tanaman padi lainnya.

#### **2.16 Pupuk Tanaman Padi**

Pupuk Tanaman Padi merupakan vitamin untuk tanaman padi agar tanaman padi subur, dan menghasilkan bulir – bulir padi yang banyak, sehingga terciptanya panen yang hasil.

### 2.13.1 UML Diagram

Diagram-diagram yang terdapat pada UML sangat banyak, berikut ini beberapa diagram yang sering digunakan dalam pengembangan sistem dapat dilihat pada gambar dibawah .



**Gambar 2.0.8 UML IAGRAM**

### 2.13.2 Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi) [15]. Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan

lain-lain. Class memiliki tiga area pokok yaitu Nama (dan stereotype), Atribut dan Metoda. Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. Private(-), tidak dapat dipanggil dari luar class yang bersangkutan.
2. Protected(#), hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya.
3. Public(+), dapat dipanggil oleh siapa saja.

Hubungan Antar Class :

1. Asosiasi, yaitu hubungan statis antar class. Umumnya menggambarkan class yang memiliki atribut berupa class lain, atau class yang harus mengetahui eksistensi class lain. Panah navigability menunjukkan arah query antar class. Lambang :



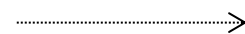
2. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”) atau biasa disebut relasi mempunyai sebuah. Lambang :



3. Composition, yaitu sebuah kelas tidak bisa berdiri sendiri dan harus merupakan bagian dari class yang lain, maka class tersebut memiliki relasi composition. Lambang:

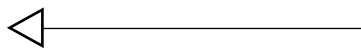


4. Dependency, yaitu hubungan yang menunjukkan operasi pada suatu kelas yang menggunakan class yang lain. Lambang :

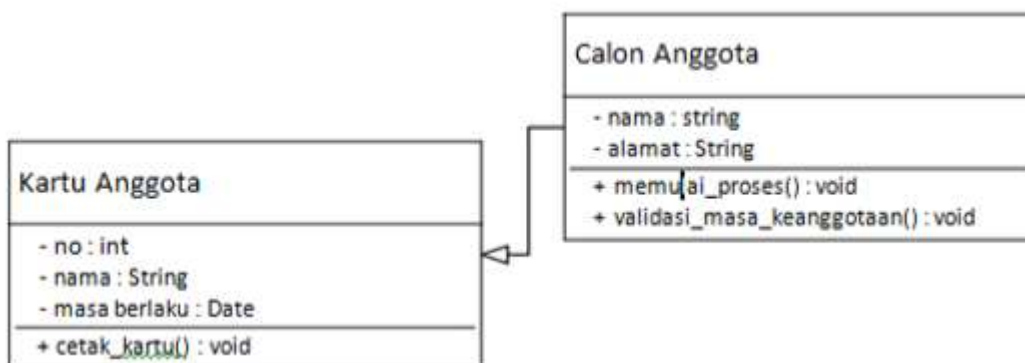




5. Generalisasi / Pewarisan, yaitu hubungan hirarkis antar class. Class dapat diturunkan dari class lain dan mewarisi semua atribut dan metoda class asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari class yang diwaris.



Berikut Contoh Class Diagram dapat dilihat pada Gambar dibawah.

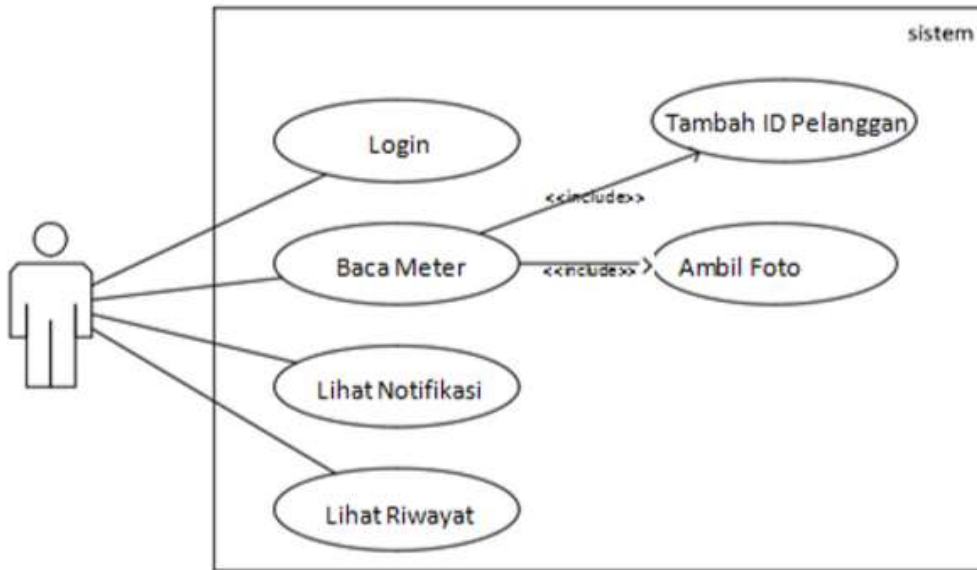


**Gambar 2.0.9 Class diagram**

### 2.13.3 Use Case Diagram

Use Case Diagram yaitu diagram yang digunakan untuk menggambarkan hubungan antara sistem dengan aktor. Diagram ini hanya menggambarkan secara global. Karena use case diagram hanya menggambarkan sistem secara global, maka elemen-elemen yang digunakan pun sangat sedikit, berikut ini elemen-elemen yang digunakan pada use case diagram :

1. Sistem, merupakan batasan-batasan proses yang sudah kita deskripsikan dalam sebuah sistem.
2. Aktor, elemen yang menjadi pemicu sistem. Aktor bisa berupa orang, mesin ataupun sistem lain Yang berinteraksi dengan use case.
3. Use Case, potongan proses Yang merupakan bagian dari sistem.
4. Association, menggambarkan interaksi antara use case dan aktor.
5. Dependency, menggambarkan relasi (relationship) antara dua use case. Ada 2 (dua) tipe dari dependency yaitu, include dan extends. Include merupakan tipe dari dependency yang menghubungkan dua use case dimana, satu use case membutuhkan use case yang satunya sedangkan extends adalah tipe dari dependency yang menghubungkan dua use case dimana satu use case terkadang akan memanggil use case yang satunya, tergantung pada kondisi.
6. Generalization, menggambarkan pewarisan antara dua aktor atau use case dimana salah satu aktor atau use case mewarisi properties ke aktor atau use case yang satunya. Berikut Contoh Use Case Diagram dapat dilihat pada Gambar dibawah.



**Gambar 2.0.10 Gambar Use Case Diagram**

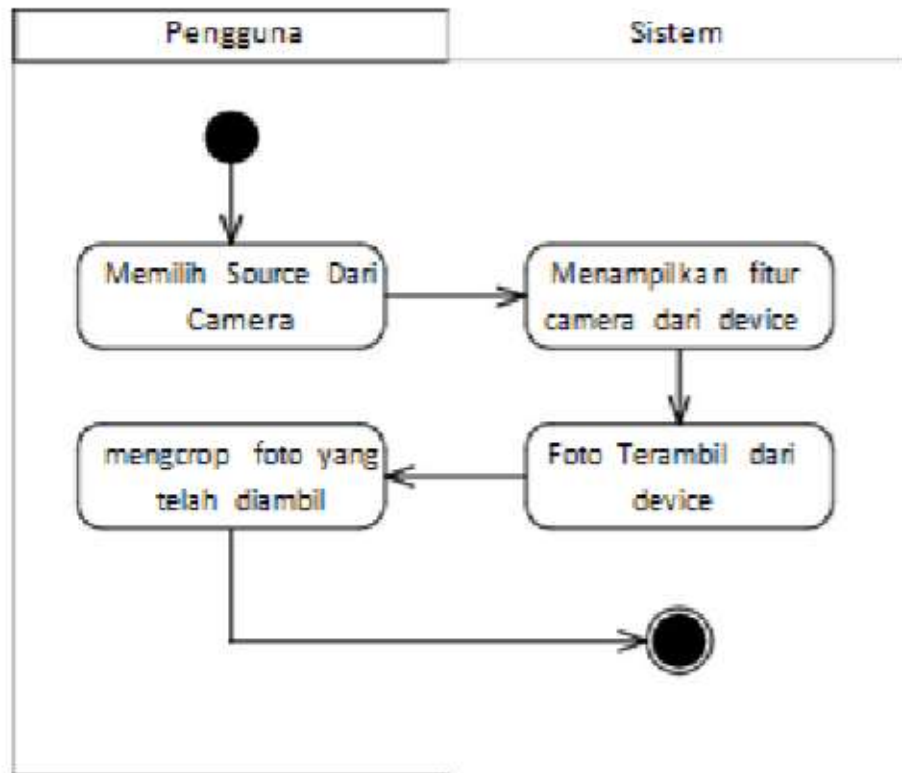
#### 2.13.4 Activity Diagram

Activity Diagram yaitu diagram yang digunakan untuk menggambarkan alur kerja (aktivitas) pada use case (proses), logika, proses bisnis dan hubungan antara aktor dengan alur-alur kerja use case. Jika sudah terbiasa dengan flowchart, maka tidak akan merasa kesulitan dalam mempelajari activity diagram, karena activity diagram identik dengan flowchart, hanya saja ada beberapa notasi tambahan yang digunakan untuk kasus- kasus tertentu. Berikut ini di jelaskan elemen- elemen dari activity diagram.

1. Activity, yaitu elemen yang digunakan untuk menggambarkan aktivitas.

2. Transitions, yaitu elemen yang digunakan untuk menggambarkan transisi dari elemen yang satu ke elemen yang lainnya
3. Decisions, yaitu elemen yang digunakan untuk percabangan logika. Elemen ini sering kita jumpai pada flowchart terutama flowchart yang digunakan untuk menggambarkan sebuah algoritma.
4. Merge Point, yaitu elemen yang digunakan untuk menggabungkan percabangan proses. Elemen ini merupakan kebalikan dari elemen decisions, dimana jika decisions digunakan untuk percabangan, sedangkan merge point digunakan sebagai penggabungan dari percabangan.
5. Start Point, yaitu elemen yang digunakan untuk memulai activity diagram.
6. End Point, yaitu elemen yang digunakan untuk mengakhiri activity diagram.
7. Swimlines, yaitu elemen yang digunakan untuk memisahkan antara aktor dan sistem ataupun antara aktor yang satu dengan aktor yang lain atau antara sistem yang satu dengan sistem yang lain.

Berikut Contoh Activity Diagram dapat dilihat pada



**Gambar 2.0.11 Gambar Acitivity Diagram**

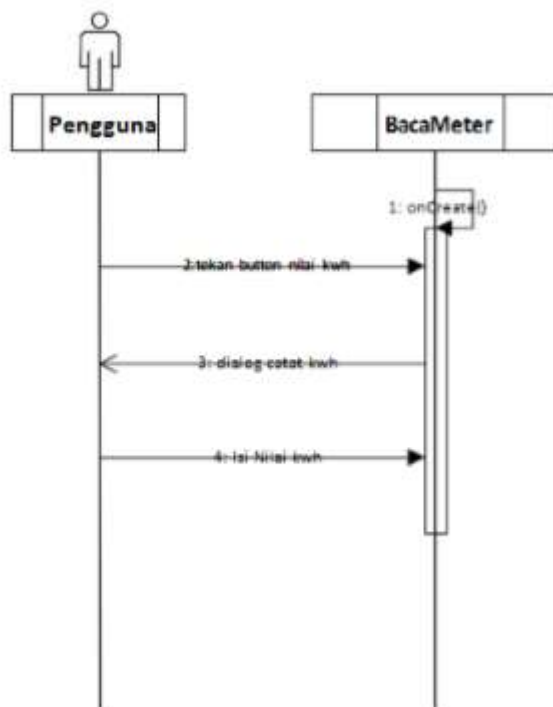
### 2.13.5 Sequence Diagram

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 atas 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. Masing-masing objek, termasuk aktor, memiliki lifeline vertikal.

Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari class. Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message. Berikut Contoh sequence diagram:



**Gambar 2.0.12 Gambar Activity diagram**

UML merupakan dasar bagi perangkat (tool) desain berorientasi objek dari IBM.[23]