

BAB 2

LANDASAN TEORI

2.1 Unified Modeling Language (UML)

UML Adalah sekumpulan diagram yang diguakan untuk melakukan abstraksi sebuah system berbasis objek oriented (OOP). UML dapat digunakan untuk mempermudah pengembangan aplikasi berkelanjutan. UML juga bias dikatakan sebagai acra standar untuk mevisualisasikan desain sebuah system. Terdiri dari berbagai elemen-elemen grafis yang disatukan pada bentuk diagram, fungsinya utuk menyajikan berbagai sedut pandang dai sebuah system [8].

2.2 Pemrograman Beorientasi Objek

object-oriented programming atau OOP adalah suatu metode pemrograman yang berorientasi pada objek. Program-program yang telah ada merupakan gabungan dari beberapa komponen-komponen kecil yang sudah ada sebelumnya. Hal itu dapat mempermudah pekerjaan seorang programmer dalam melakukan pengembangan program. Objek-objek yang saling berkaitan dan disusun kedalam satu kelompok ini disebut dengan class. Nantinya, objek-objek tersebut akan saling berinteraksi untuk menyelesaikan masalah program yang rumit. Dalam object oriented programming, dikenal empat prinsip yang menjadi dasar penggunaannya. keempat prinsip OOP tersebut adalah sebagai berikut:

1. Encapsulation

Encapsulation atau pengkapsulan adalah konsep tentang pengikatan data atau metode berbeda yang disatukan atau “dikapsulkan” menjadi satu unit data. Maksudnya, berbagai objek yang berada dalam class tersebut dapat berdiri sendiri tanpa terpengaruh oleh yang lainnya. Encapsulation dapat mempermudah pembacaan kode. Hal tersebut terjadi karena informasi yang disajikan tidak perlu dibaca secara rinci dan sudah merupakan satu kesatuan. Proses enkapsulasi mempermudah untuk menggunakan sebuah objek dari suatu kelas karena kita tidak perlu mengetahui segala hal secara rinci.

2. Abstraction

Prinsip selanjutnya yaitu abstraction. Prinsip ini sendiri berarti memungkinkam seorang developer memerintahkan suatu fungsi, tanpa harus mengetahui bagaimana fungsi tersebut bekerja. Lebih lanjut, abstraction berarti menyembunyikan detail

latar belakang dan hanya mewakili informasi yang diperlukan untuk dunia luar. Ini adalah proses penyederhanaan konsep dunia nyata menjadi komponen yang mutlak diperlukan. Seperti kala menggunakan handphone, kamu cukup memberikan suatu perintah, tanpa tahu bagaimana proses terlaksananya perintah tersebut.

3. Inheritance

Inheritance dalam konsep OOP adalah kemampuan untuk membentuk class baru yang memiliki fungsi turunan atau mirip dengan fungsi yang ada sebelumnya. Konsep ini menggunakan sistem hierarki atau bertingkat. Maksudnya, semakin jauh turunan atau subclass-nya, maka semakin sedikit kemiripan fungsinya.

4. Polymorphism

Prinsip terakhir dalam OOP adalah polymorphism. Pada dasarnya polymorphism adalah kemampuan suatu pesan atau data untuk diproses lebih dari satu bentuk. Salah satu ciri utama dari OOP adalah adanya polymorphism. Tanpa hal ini, suatu pemrograman tidak bisa dikatakan sebagai OOP. Polymorphism sendiri adalah konsep di mana suatu objek yang berbeda-beda dapat diakses melalui interface yang sama [9].

2.3 Sequence Diagram

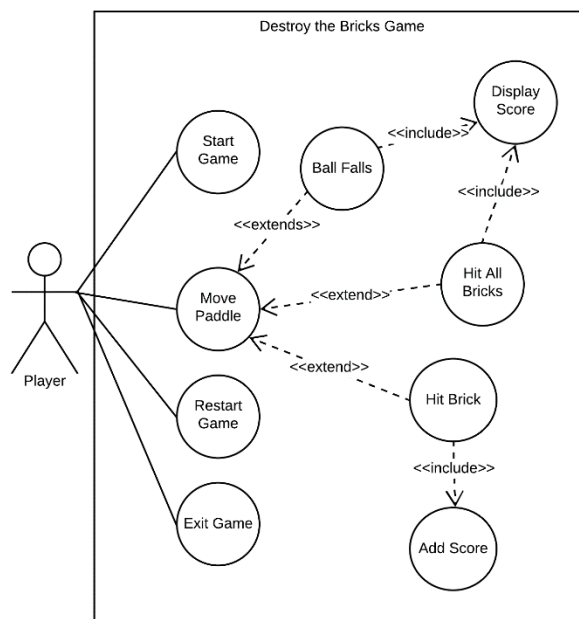
Sequence Diagram adalah salah satu dari diagram - diagram yang ada pada UML, sequence diagram berfungsi untuk menggambarkan kolaborasi secara dinamis antara sejumlah objek. Selain itu, sequence diagram berfungsi untuk menunjukkan hubungan dari pesan yang dikirim antara objek – objek yang saling berinteraksi [10].

1. Object - adalah komponen berbentuk kotak yang mewakili sebuah class atau object. Mereka mendemonstrasikan bagaimana sebuah object berperilaku pada sebuah system.
2. Activation boxes - adalah komponen yang berbentuk persegi panjang yang menggambarkan waktu yang diperlukan sebuah object untuk menyelesaikan tugas. Lebih lama waktu yang diperlukan, maka activation boxes akan lebih panjang.
3. Actors - adalah komponen yang berbentuk stick figure. Komponen yang mewakili seorang pengguna yang berinteraksi dengan system.

4. Lifeline - adalah komponen yang berbentuk garis putus - putus. Lifeline biasanya memuat kotak yang berisi nama dari sebuah object. Berfungsi menggambarkan aktifitas dari object.

2.4 Usecase Diagram

Usecase Diagram merupakan suatu diagram yang mendeskripsikan hubungan antara aktor dengan sistem. Usecase diagram dapat mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang dibuat, usecase diagram juga dapat digunakan untuk mengetahui fungsi apa saja yang terdapat dalam sistem, hal tersebut kemudian dapat menjelaskan komunikasi antara aktor dan sistem. Manfaat dari usecase sendiri adalah memberikan pemahama yang pas tentang requirement yang dibutuhkan oleh sistem [8].



Gambar 2. 1 Contoh Usecase Diagram

Usecase Diagram terdiri dari 3 komponen yaitu:

1. Sistem

Menyatakan Batasan aktor yang menggunakan sistem tersebut dan mebatasi fitur-fitur yang harus disediakan dalam sistem.

2. Aktor

Aktor adalah segala hal diluar sistem atau yang akan mennggunakan sistem tersebut. Biasanya merupakan manusia,sistem atau device




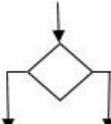
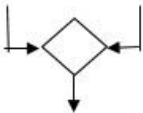

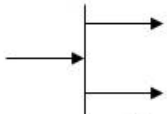
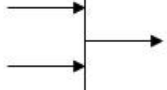
yang memiliki peranan dalam keberhasilan suatu sistem.

4. Use Case

Gambaran fungsional dari sebuah sistem, dengan demikian antara konsumen dan pengguna akan paham mengenai fungsi suatu sistem yang sedang di bangun.

2.5 Activity Diagram

Activity diagram adalah rancangan aliran aktivitas pada sebuah sistem yang akan dijalankan. Activity diagram juga dapat digunakan untuk mendefinisikan alur tampilan sistem. Activity diagram memiliki komponen atau symbol-simbol tertentu yang dihubungkan menggunakan tanda panah dari awal hingga akhir [11].

- Initial State

 Initial State adalah awal dimulainya suatu aliran kerja pada activity diagram dan pada sebuah activity diagram hanya terdapat satu initial state.
- Final State

 Final State adalah bagian akhir dari suatu aliran kerja pada sebuah activity diagram dan pada sebuah activity diagram bisa terdapat lebih dari satu final state.
- Activity

 Aktivitas adalah aktivitas atau pekerjaan yang dilakukan dalam aliran kerja.
- Decision

 Decision berfungsi untuk menggambarkan pilihan kondisi dimana ada kemungkinan perbedaan transisi, untuk memastikan bahwa aliran kerja dapat mengalir ke lebih dari satu jalur.
- Merge

 Merge berfungsi untuk menggabungkann kembali aliran kerja yang sebelumnya telah dipecah oleh Decission.
- Transition / Association

 Transition untuk menghubungkan aktivitas selanjutnya setelah aktivitas sebelumnya
- Synchronization
 - Synchronization Fork

 Synchronization Fork digunakan untuk memecah behavior menjadi aktivitas yang paralel (Contoh : User dapat memilih menu yang dapat dilakukan secara paralel).
 - Synchronization Join

 Synchronization Join digunakan untuk menggabungkan kembali aktivitas yang paralel.

Gambar 2. 2 Simbol-simbol Activity diagram

2.6 Class Diagram

Class diagram merupakan salah satu diagram utama dari Unified Modelling Language (UML). Class diagram bertujuan untuk menggambarkan blueprint object pada sebuah sistem. Class diagram menggambarkan bagaimana hubungan atau interaksi yang terjadi antar class pada konstruksi sebuah *software* seperti hubungan inheritance, agregasi, asosiasi, dan komposisi [12].

2.7 Algoritma Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan algoritma deep learning yang berkerja dengan cara menerima input berupa gambar, memutuskan sebuah aspek dari objek yang berupa gambar atau foto yang tujuan akhirnya adalah mengenali gambar tersebut dan dapat membedakannya dari gambar yang lain. Convolutional Neural Network (CNN) memiliki arsitektur yang mirip dengan pola koneksi sel safaf atau neuron dalam otak manusia. Algoritma ini terinspirasi dari *visual contex*, merupakan bagian dari otak yang memiliki fungsi untuk memproses informasi – informasi yang berbentuk visual. Dengan arsitektur yang meniru cara kerja pada manusia ini, Convolutional Neural Network dapat dilatih untuk mempelajari, memahami sebuah gambar dengan detail. Dengan demikian, Convolutional Neural Network dapat memahami dependensi temporal dan spasial dalam sebuah gambar setelah diberikan filter yang relevan [13].

2.8 Artificial Intelligence (AI)

Artificial Intelligence atau kecerdasan buatan adalah sebuah simulasi dari kecerdasan seorang manusia yang ditanam dalam mesin dan telah diprogram sedemikian rupa untuk berpikir, meniru tindakan layaknya manusia seperti pada umumnya. Artificial Intelligence juga merupakan bagian dari ilmu komputer untuk mengetahui mesin komputer apakah AI ini dapat melakukan pekerjaan seperti manusia atau mungkin bisa lebih baik [14].

2.9 Machine Learning

Machine Learning atau pembelajaran mesin merupakan salah satu ilmu dari kecerdasan buatan (Artificial Intelligence) dikembangkan dengan tujuan agar komputer bisa belajar secara mandiri tanpa adanya arahan dari pengguna. Istilah *machine learning* pada dasarnya memiliki arti yaitu proses computer untuk belajar sesuatu berdasarkan pada data atau bisa disebut juga dengan *learn from data*. Tanpa adanya data, komputer tidak bisa belajar apa-apa. Oleh karena itu semua pengetahuan *machine learning* pastinya akan melibatkan data, karena data adalah dasar dari *machine learning*. Banyak hal yang dipelajari, akan tetapi pada dasarnya ada 4 hal pokok yang dipelajari dalam machine learning [15].

1. Pembelajaran Terarah (Supervised Learning)
2. Pembelajaran Tak Terarah (Unsupervised Learning)

3. Pembelajaran Semi Terarah (Semi-supervised Learning)
4. Reinforcement Learning.

2.10 Image Processing

Image processing adalah suatu metode pengolahan gambar yang ditransformasikan menjadi gambar lain sebagai keluarannya dengan Teknik tertentu. Awalnya fungsi dari image processing adalah untuk memperbaiki kualitas dari suatu gambar, namun seiring berkembangnya teknologi fungsi image processing juga berkembang menjadi memungkinkan untuk mengambil informasi, manipulasi, dan Analisa pada suatu gambar [16].

Operasi image processing dapat dikelompokkan berdasarkan dari tujuan transformasinya, yaitu:

1. Image Enhancement (peningkatan kualitas gambar)
2. Image Restoration (pemulihan gambar)
3. Image Compression (kompresi gambar)
4. Image Refresention & Modelling (representasi dan permodelan gambar).

2.11 Post-training quantization

Post-training quantization adalah teknik konversi yang memiliki fungsi untuk mengurangi ukuran sebuah model sekaligus meningkatkan kinerja dari CPU dan mengurangi latensi akselerator pada perangkat keras, dengan sedikit menurunkan akurasi dari model yang dibuat [17].

2.12 Java

Java adalah bahasa pemrograman yang dibuat oleh James Gosling. Sintaks Java mirip seperti C/C++, karena Java banyak terinspirasi dari keuda bahasa ini. Java dirancang untuk tujuan umum (general-purpose) dan sepenuhnya menganut paradigma OOP (Object Oriented Programming). OOP adalah paradigma pemrograman yang berbasis pada objek. Setiap membuat program, maka wajib membuat objek terlebih dahulu [18].

2.13 Android

Android adalah sistem operasi yang dirancang oleh Google dengan basis kernel Linux untuk mendukung kinerja perangkat elektronik layar sentuh, seperti tablet atau smartphone. Jadi, android digunakan dengan sentuhan, gesekan ataupun ketukan pada layar gadget anda.

Android bersifat open source atau bebas digunakan, dimodifikasi, diperbaiki dan didistribusikan oleh para pembuat ataupun pengembang perangkat lunak. Dengan sifat open source perusahaan teknologi bebas menggunakan OS ini diperangkatnya tanpa lisensi. Awal pembuatannya, Android ditargetkan bagi penggunaan perangkat kamera digital. Akan tetapi, para pencipta Android, yaitu Andy Rubin, Chris White, dan Nick Sears berpendapat bahwa pasar untuk kamera digital tidak terlalu besar. Maka dari itu, sistem operasi ini kemudian dialihkan penggunaannya pada ponsel pintar. Pada tahun 2004, Android mulai dipasarkan dan berhadapan dengan saingan smartphone berbasis sistem operasi Symbian dan Windows Mobile [19].

2.14 Android Studio

Android Studio adalah sebuah tools IDE (Interpreter Development Environment) dikembangkan oleh tim JetBrains yang suport penuh oleh Google sebagai alat pengembang Android yang sangat powerful, dalam IDE tersebut bisa melakukan pengembangan android menggunakan bahasa Java, Kotlin, C# [20].

2.15 TensorFlow

Tensorflow adalah sebuah *framework machine learning* yang dapat membantu membuat sebuah neural network (jaringan artifisial) dalam skala besar. Tensorflow biasa digunakan pada sebuah penelitian yang menyangkut *artificial intelligence* (AI) atau *deep learning* [21].

2.16 Keras

Keras adalah deep learning API yang dikembangkan menggunakan bahasa python, API tingkat tinggi keras berjalan di platform machine learning tensorflow. Keras dikembangkan dengan maksud pengguna dapat menikmati fast experimentation dan mendapatkan hasil secepat mungkin pada tensorflow. Keras dibuat dengan tujuan agar mudah digunakan, mudah diperluas, modular, dan dapat bekerja dengan Bahasa pemrograman Python. API "dirancang untuk manusia, bukan mesin, " dan "mengikuti praktik terbaik untuk mengurangi beban kognitif". Alasan terbesar untuk menggunakan Keras adalah karena dokumentasinya yang lengkap, dan terutama tentang mudahnya keras untuk digunakan oleh pengguna. Di luar kemudahan yang ditawarkan dalam membuat dan membangun sebuah model, keras juga menawarkan kelebihan lain yaitu luasnya dukungan untuk berbagai

pilihan penyebaran produksi, penggunaan yang luas. Untuk luasnya penyebaran produksi tensorflow sudah terintegrasi dengan lima mesin *back-end* yaitu MXNet, CNTK, Tensorflow, Theano, dan PlaidML. Keras juga sudah *support* dengan berbagai jenis GPU guna memberikan pengalaman yang lebih baik untuk pengguna [22] [23].

2.17 MobileNets

MobileNets, merupakan salah satu arsitektur convolutional neural network (CNN) yang dapat digunakan untuk mengatasi kebutuhan akan computing resource berlebih. Seperti namanya, Mobile, para peneliti dari Google membuat arsitektur CNN yang dapat digunakan untuk ponsel. Perbedaan mendasar antara arsitektur MobileNet dan arsitektur CNN pada umumnya adalah penggunaan lapisan atau layer konvolusi dengan ketebalan filter yang sesuai dengan ketebalan dari input image. MobileNet membagi konvolusi menjadi *depthwise convolution* dan *pointwise convolution* [24].

2.18 Google Cloud Storage

Google Cloud Storage adalah layanan yang dikembangkan oleh Google yang memungkinkan untuk menyimpan dan mengelola data secara langsung di infrastruktur Google. Dengan Google Cloud Storage, pengguna dapat mengunduh atau mengunggah file, menghapus file, mengambil daftar file, atau mengubah ukurannya [25].

2.19 Google Colabs

Google Collaboratory atau Google Colab merupakan tools yang berbasis cloud dan free untuk tujuan penelitian. Google colab dibuat dengan environment jupyter dan mendukung hampir semua library yang dibutuhkan dalam lingkungan pengembangan Artificial Intelligence (AI).

Pada dasarnya google colab sama dengan Jupyter Notebook dan bisa dikatakan google colab adalah jupyter notebook yang dijalankan secara online dan gratis. Berikut adalah beberapa kelebihan dalam menggunakan google colab [26].

2.20 Internet of Things (IoT)

Internet of Things atau IoT pertama kali dikeluarkan oleh Kevin Ashton pada tahun 1999 di salah satu presentasinya. Kini banyak perusahaan besar mulai

mendalami Internet of Things sebut saja Intel, Microsoft, Oracle, dan banyak lagi lainnya.

Banyak yang memprediksikan bahwa IoT adalah “The Next Big Thing” di dunia teknologi informasi, hal ini karena IoT menawarkan banyak potensi yang bisa dikembangkan kembali. Contohnya adalah implementasi dari internet of Things (IoT) misalnya adalah kulkas yang dapat memberitahukan kepada pemiliknya via SMS atau email tentang makanan dan minuman apa saja yang sudah habis dan harus distok lagi.

Casagras (Coordination and support action for global RFID-related activities and standardisation) mendefinisikan, bahwa Internet Of Things (IoT) adalah sebuah infrastruktur jaringan global, yang dapat menghubungkan perangkat keras dan virtual melalui eksploitasi data capture serta kemampuan komunikasi. Dalam Infrastruktur terdiri dari jaringan yang sudah ada dan internet beserta pengembangan jaringannya. Sehingga, IoT ini menawarkan objek, sensor dan kemampuan koneksi agar dapat menyediakan layanan dan aplikasi ko-operatif yang independen [27].

2.21 Pink Eyes

Pink Eye merupakan sebuah penyakit mata yang menular pada hewan ternak seperti domba, sapi, dan kambing, penyakit ini biasanya bersifat epizootik dan memiliki gejala dengan keruhnya mata dan memerahnya conjunctiva. Pink eyes tidak menyebabkan kematian pada hewan ternak, tetapi akan menyebabkan penurunan berat badan, kebutaan, serta biaya pengobatan yang mahal. Penyakit pink eyes ini disebabkan oleh sebuah virus, bakteri, chlamydia maupun rikketsia, namun penyebab pink eyes yang paling sering ditemukan adalah akibat dari bakteri *Maraxella bovis* [28].

Penularan:

1. Kontak antara ternak peka dengan ternak penderita
2. Serangga yang bisa memindahkan mikroorganisme
3. Iritasi debu
4. Sumber-sumber lain yang dapat menyebabkan goresan atau luka mata.

Gejala Klinis

1. Mata berair, kemerahan pada bagian mata yang putih dan kelopaknya
2. Bengkak pada kelopak mata
3. Menjulingkan mata untuk menghindari sinar matahari.
4. Selaput bening mata/kornea menjadi keruh
5. pembuluh darah tampak menyilangnya.
6. Terjadi borok atau lubang pada selaput bening mata. Borok dapat pecah dan mengakibatkan kebutaan.
7. Sembuh dalam waktu 1 – 4 minggu, tergantung kepada penyebabnya dan keganasan penyakitnya.

2.22 Thelaziasis

Thelaziasis merupakan suatu penyakit cacing mata yang disebabkan oleh *Thelazia* sp. dan dapat menyerang berbagai jenis ternak yaitu sapi, kerbau, kuda, kambing, anjing, kucing, kelinci dan domba. *Thelazia* sp. yang banyak menyerang ternak sapi adalah *T. rhodisii*, *T. glukosa* dan *T. Skrijabini* (Tjahajati dan Husniyati, 2012).

Infeksi cacing mata *Thelazia* sp. dapat terjadi pada salah satu mata atau kedua mata. Pada hari ke-3 atau ke-4 setelah infeksi oleh larva cacing dapat terjadi konjungtivitis ringandiserelai lacrimasi. Pada perkembangan selanjutnya dapat mengakibatkan kongesti konjungtiva dan photobia. Bila keadaan ini dibiarkan dapat terjadi kekeruhan kornea mata, konjungtiva membengkak karena adanya penyumbatan duktus lakrimalis oleh nanah [29].

Gejala :

1. Hewan penderita nampak tidak tenang karena ada rasa sakit atau iritasi pada mata.
2. Nafsu makan semakin menurun akibatnya hewan menjadi kurus dan nampak lemas.
3. Adanya keluar mata yang berlebihan pada hewan penderita atau biasa disebut lacrimasi.
4. Area konjungtiva mata pada hewan yang terserang *thelazia*, mengalami kemerahan.

5. Pada stadium lanjut dapat terjadi keratitis ulserasi pada kornea dan akhirnya terjadi kebutaan.
6. Perubahan patologis anatomi pada ternak yang terinfeksi thelaziasis, tidak ada yang khas kecuali adanya cacing mata dibawah membran niktitan atau ductus lacrimalis atau di dalam kantong konjungtiva, tergantung spesies cacing.
7. Siklus hidup cacing mata /thelaziasi ini berawal dari stadium larva (mikrofilaria) yang terdapat dalam air mata sapi. Larva ini kemudian dihisap oleh vektor (lalat Musca sp), dalam tubuh vektor lalat tersebut larva thelazia mengalami perkembangan menjadi larva stadium kedua dan stadium ketiga (larva infeksi). Larva infeksi yang ada di tubuh lalat musca sp tersebut kemudian dipindahkan dan ditularkan ke mata sapi atau hewan lainnya, untuk selanjutnya berkembang menjadi larva stadium keempat dan kelima (dewasa). Cacing betina dewasa selanjutnya menghasilkan larva stadium pertama atau yang biasa disebut mikrofilaria.
8. Untuk mendiagnosa penyakit Thelazia pada hewan ternak dilakukan dengan cara melihat gejala klinis yang timbul yaitu adanya cacing pada mata. Diagnosa juga dapat diteguhkan dengan pemeriksaan laboratorium untuk mengidentifikasi cacing mata tersebut.