

BAB II

LANDASAN TEORI

2.1 Pengolahan Citra

Pengolahan Citra merupakan bidang yang bersifat multidisiplin. Terdiri dari banyak aspek antara lain fisika, elektronika, matematika, seni, fotografi, dan teknologi komputer [1].

Secara harfiah, citra (Image) adalah gambar pada bidang dua dimensi. Citra merupakan fungsi lanjutan dari intensitas cahaya pada bidang dua dimensi. Sumber cahaya menerangi objek, objek lalu memantulkan kembali sebagai dari bekas cahaya tersebut [1]. Secara umum pengolahan citra menunjuk pada pemrosesan gambar 2 dimensi dengan menggunakan komputer. Pengolahan citra mengacu pada pemrosesan 2 dimensi [10].

2.1.1 OCR (Optical Character Recognition)

Optical Character Recognition (OCR) merupakan salah satu bidang citra yang mengkonversi gambar huruf menjadi karakter ASCII yang dikenal oleh komputer. Gambar huruf yang dihasilkan dapat berupa hasil scan dokumen, halaman website, hasil foto, dan lain sebagainya [2].

OCR merupakan aplikasi yang menerjemahkan gambar karakter menjadi bentuk teks dengan cara menyesuaikan pola karakter per baris dengan pola yang telah tersimpan dalam database aplikasi. Hasil dari proses OCR adalah berupa teks sesuai dengan gambar output scanner dimana tingkat keakuratan penerjemah karakter tergantung dari tingkat kejelasan gambar dan metode yang digunakan [3].

2.2 Smartphone

Smartphone merupakan perkembangan dari handphone yang ditambahi fitur – fitur seperti email, personal organizer, dan juga konektivitas tambahan seperti wifi dan bluetooth yang dapat di instal di device. Dari segi arsitektur device sudah

dilengkapi dengan inputan seperti QWERTY miniatur keyboard dan touchscreen. Aplikasi pada ponsel pintar dapat dikembangkan oleh operator dari device itu sendiri ataupun pihak ketiga yang ikut mengembangkan untuk kepentingan komersial [4].

2.3 Android

Android adalah sistem operasi untuk telepon seluler yang berbasis di linux (Safaat Nazruddin, 2011). Android menyediakan platform yang bersifat open source bagi para pengembang untuk menciptakan sebuah aplikasi. Pada awalnya, Google Inc. mengakuisisi Android Inc. yang mengembangkan software untuk ponsel yang terletak di Palo Alto, California Amerika Serikat. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, yaitu konsorsium dari 34 perusahaan hardware, software, dan telekomunikasi. Pada tanggal 22 Oktober 2008 HTC Dream merupakan telepon pertama yang memakai sistem operasi Android [5]. Saat ini android tidak hanya menjadi sistem operasi pada smartphone saja tetapi, Android juga menjadi pesaing bagi Apple pada sistem operasi PC dan Tablet. Selain itu Android adalah platform yang lengkap, baik sistem operasi, aplikasi dan tool pengembangan, market aplikasi android, serta dukungan yang sangat tinggi dari komunitas opensource [5]. Arsitektur Android dapat dijelaskan dan digambarkan sebagai berikut:

- Applications dan Widgets Applications dan Widgets ini adalah layer dimana berhubungan dengan aplikasi saja, di mana biasanya download aplikasi dijalankan kemudian dilakukan instalasi dan jalankan aplikasi tersebut [5].
- Applications Frameworks ini adalah layer di mana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi android, karena pada layer inilah aplikasi dapat dirancang dan dibuat, seperti contentproviders yang berupa sms dan panggilan telepon [5].
- Libraries ini adalah layer di mana fitur-fitur android berada, biasanya para pembuat aplikasi mengakses libraries untuk menjalankan aplikasinya. Berjalan di atas kernel, Layer ini meliputi berbagai library C/C++ inti seperti Libc dan SSL [5].

- Android Run Time Layer yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan Implementasi Linux [5].
- Linux Kernel adalah layer di mana inti dari operating system dari android itu berada. Berisi file - file system yang mengatur sistem processing, memory, resource, drivers, dan sistem - sistem operasi android lainnya. Linux kernel yang digunakan android adalah linux kernel release 2.6 [5].

2.4 Machine Learning

Machine Learning merupakan metode dari salah satu percabangan kecerdasan buatan (artificial intelligence) untuk mengoptimalkan performa dari sistem dengan mempelajari data sampel atau data histori [11]. *Machine learning* dapat mempelajari bagaimana sebuah mesin atau komputer dapat belajar dari pengalaman atau bagaimana cara memprogram mesin untuk dapat belajar [6]. *Machine learning* berfokus pada pengembangan program – program komputer yang dapat mengerjakan dirinya sendiri untuk tumbuh dan berubah bila diberikan data baru [6].

Inti dari machine learning adalah untuk membuat model yang merefleksikan pola – pola data. machine learning memungkinkan komputer atau suatu program dapat menemukan pengetahuan tanpa perlu diprogram secara eksplisit [12].

2.4.1 Computer Vision

Computer Vision adalah cabang dari ilmu komputer yang mengembangkan metode komputasi untuk mengekstrak informasi melalui gambar visual. Aplikasi computer vision yang familiar termasuk pengenalan wajah pada Facebook, pencarian gambar Google dan otonomi navigasi kendaraan [13].

Computer Vision juga merupakan metode yang menggunakan komputer untuk mendapatkan pemahaman yang komprehensif, Pemahaman tentang konten atau makna gambar atau video dari digital yaitu visual. Informasi yang diperoleh dapat digunakan seperti untuk klasifikasi citra atau pengenalan wajah [14].

2.5 Firebase Machine Learning Kit (MLKit)

Firebase Machine Learning kit (MLKit) merupakan salah satu produk dari Google yang menghadirkan keahlian dari Machine Learning untuk diterapkan pada platform Android atau IOS. MLKit mudah diaplikasikan dan untuk memulai sebuah projek tidak perlu pengetahuan yang dalam mengenai ilmu Artificial Neural Network (jaringan syaraf tiruan) [6].

Di dalam MLKit terdapat banyak proses yang bida dikembnagkan seperti *Text Recognition* dan *Text To Speech*. *Text Recognition* memiliki fokus utama pada pengenalan karakter pada dokumen sedangkan *Text To Speech* adalah proses mengkonversi atau mengubah teks dalam format suatu bahasa menjadi ucapan dalam bentuk suara sesuai dengan pembacaan teks sesuai dengan bahasa yang dipilih [6].

2.5.1 Google MLKit

Google MLKit merupakan sebuah SDK Seluler yang menghadirkan keahlian Machine Learning Google untuk aplikasi Android dan IOS dalam paket andal dan mudah digunakan [9].

Cara kerja dari Google ML Kit yaitu mempermudah penerapan teknik ML di aplikasi dengan menghadirkan teknologi ML dari Google, seperti Google Cloud Vision API, TensorFlow Lite, dan Android Neural Networks API secara terpadu dalam satu SDK. Hanya dengan menerapkan beberapa baris kode, ML Kit dapat digunakan untuk memperoleh berbagai hal, seperti pemrosesan berbasis cloud, kemampuan real-time pada model di perangkat yang dioptimalkan untuk seluler, atau fleksibilitas model TensorFlow Lite kustom [9].

Alur Implementasi dari Google MLKit yaitu :

1. Mengintegrasikan SDK ke dalam program menggunakan Gradle.
2. Mempersiapkan data input. Jika menggunakan fitur *Vision*, proses mengambil data gambar dari kamera dan membuat metadata yang diperlukan seperti Rotasi Gambar, atau meminta pengguna untuk memilih foto dari galeri.

3. Menerapkan model machine learning ke data aplikasi [9].

2.6 UML (Unified Modeling Language)

Unified Modelling Language (UML) adalah sebuah pemodelan yang menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem [7].

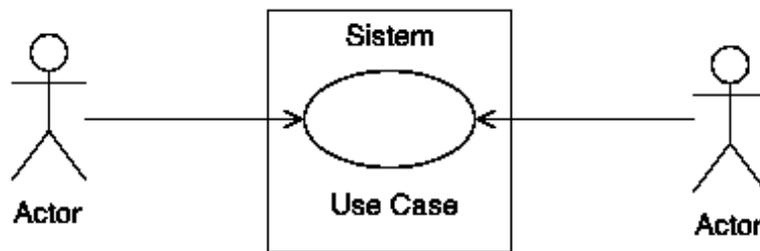
UML merupakan sebuah bahasa untuk menentukan visualisasi, konstruksi dan mendokumentasikan artifact dari sistem perangkat lunak. Dengan menggunakan UML dapat membuat suatu model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut adakan dapat berjalan pada perangkat keras. Sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman [10].

Setiap sistem yang kompleks seharusnya bisa dipandang dari sudut yang berbedabeda sehingga kita dapat mendapatkan pemahaman secara menyeluruh. Untuk upaya tersebut UML menyediakan berbagai diagram yang biasa dikelompokan berdasarkan sifatnya statis atau dinamis, dengan demikian adapun diagram yang digunakan dalam penelitian ini adalah sebagai berikut:

2.6.1 Use Case Diagram

Use Case Diagram bersifat statis. Diagram ini memperlihatkan himpunan use case dan actor-aktor. Diagram ini sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna. Use case diagram merupakan salah satu diagram untuk memodelkan aspek perilaku sistem. Masing-masing diagram use case menunjukkan sekumpulan use case, actor, dan hubungannya. use case diagram adalah hal yang penting untuk memvisualisasikan, menspesifikasikan, dan mendokumentasikan kebutuhan perilaku sistem, subsistem, dan kelas. use case adalah interaksi antara actor eksternal dan sistem, hasil yang dapat diamati oleh actor, berorientasi pada tujuan, dideskripsikan di diagram use case dan teks [8].

Berikut ini merupakan contoh dari Use Case Diagram, dapat dilihat dari Gambar II-1. Use Case Diagram.



Gambar II-1 Contoh Use Case Diagram

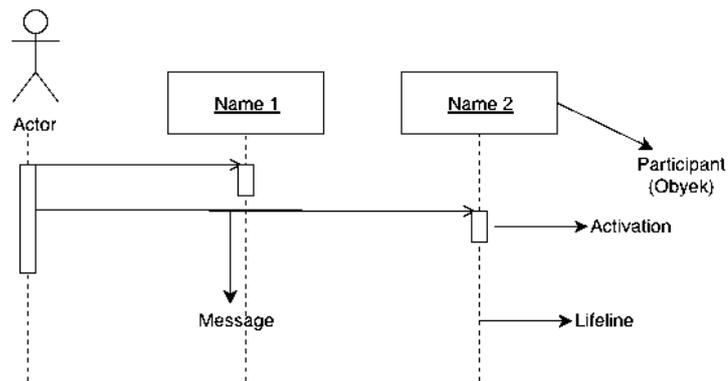
2.6.2 Activity Diagram

Activity Diagram bersifat dinamis. Diagram ini adalah tipe khusus dari diagram state yang memperlihatkan aliran dari suatu aktifitas ke aktivitas lainnya dalam suatu sistem. Diagram ini penting dalam pemodelan fungsifungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antarobjek. Pada dasarnya diagram aktivitas adalah diagram flowchart yang diperluas yang menunjukkan aliran kendali satu aktivitas ke aktivitas lain. Diagram ini digunakan untuk memodelkan aspek dinamis sistem [8].

2.6.3. Sequence Diagram

Sequence Diagram bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu. Sequence Diagram digunakan untuk memodelkan scenario penggunaan. Scenario penggunaan adalah barisan kejadian yang terjadi selama satu eksekusi sistem. Sequence Diagram menunjukkan objek sebagai garis vertical dan tiap kejadian sebagai panah horizontal dari objek pengirim ke objek penerima [8].

Berikut ini merupakan contoh dari Sequence Diagram, dapat dilihat dari Gambar II-2. Sequence Diagram.



Gambar II-2 Contoh Sequence Diagram

2.6.4 Class Diagram

Class Diagram bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas membuat kelas-kelas aktif. Class Diagram merupakan diagram paling umum dipakai di semua pemodelan berorientasi objek. Pemodelan kelas merupakan pemodelan paling utama di pendekatan berorientasi objek. Pemodelan kelas menunjukkan kelaskelas yang ada di sistem dan hubungan antar kelas-kelas itu, atribut-atribut dan operasi di kelas-kelas [8].