BAB II
TEORI PENUNJANG

2.1 OCR (*Optical Character Recognition)*

OCR merupakan software aplikasi yang digunakan untuk mendeteksi sebuah teks, angka, maupun sebuah pola karakter dari sebuah gambar. Dalam pengaplikasiannya di dunia nyata, OCR dapat membantu melakukan berbagai macam hal, contohnya seperti mendeteksi plat kendaraan untuk kebutuhan area parkir, scan barcode untuk penjualan di pasar swalayan atau minimarket, untuk mendeteksi teks bahasa asing, dan lainnya [2].

Gambar 2.1-1 Cara Kerja OCR

2.2 Tesseract OCR

Tesseract OCR adalah suatu engine *Optical Character Recognition. Tesseract* OCR merupakan sebuah mesin optical character recognition yang sekarang dikembangkan oleh google. Pada sebuah software aplikasi, tesseract OCR biasanya digunakan sebagai tambahan library baru dari sebuah project dalam pembuatan software maupun aplikasi yang nantinya berfungsi sebagai pendeteksian. Salah satu contohnya dalam pembuatan aplikasi android dengan menggunakan software android studio dalam pengembangannya, android studio tidak memiliki atribut OCR di dalamnya sehingga saat pembuat aplikasi ingin membuat sebuah aplikasi dengan mengandalkan OCR sebagai point utama dari

aplikasi tersebut, maka pembuat aplikasi harus menambahkan library baru untuk OCR salah satunya Tesseract OCR engine.[3]

2.3 *Text to Speech*

Sistem konversi *text-to-speech* (TTS) merupakan suatu sistem yang mampu memproduksi sinyal ucapan secara otomatis melalui transkripsi grafem-ke-fonem untuk kalimat yang diucapkan. Perbedaan sistem TTS dengan *talking machine* biasa adalah keotomatisannya dalam mengucapkan kata-kata baru, oleh karena itu TTS memungkinkan untuk diimplementasikan pada bidang aplikasi yang beragam seperti aplikasi sms bicara, buku digital dan pembaca email otomatis. Luasnya aplikasi yang ditawarkan oleh sistem TTS ini, dan berkembangnya beberapa perangkat/platform, seperti ponsel dan PDA, telah mendorong diimplementasikannya sistem TTS pada berbagai platform untuk berbagai keperluan. Dukungan hardware dan software yang memadai memungkinkan sistem TTS untuk diimplementasikan pada perangkat tersebut[5].

2.4 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah Bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan artifacts (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, artifact tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak[5]. UML dibuat oleh salah satu tokoh dari *Grady Booch, James Rumbaugh* dan *Ivar Jacobson*.

2.4.1 Bagian – bagian dari UML

Bagian-bagian utama dari UML, antara lain :

1. View

View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. Ada beberapa jenis view dalam UML antara lain sebagai berikut:

a. Use Case View

   Ialah bentuk fungsionalitas dari sistem yang diinginkan oleh user (dalam hal ini aktor). View ini digambarkan dalam user case diagram dan seringkali juga dibuat dalam class diagram. View digunakan dalam pelanggan, perancang, pengembang dan penguji sistem.

b. Logical View

   Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (class, object dan ralationship) dan kolaborasi dinamis yang terjadi ketika object mengirim pesan ke object lain dalam suatu fungsi tertentu. View digunakan untuk perancang dan pengembang.

c. Component View

   Mendeskripsikan implementasi dan ketergantungan modul. Komponen ini ialah tipe lainnya dari code module diperlihatkan dengan struktur dan ketergantungannya juga alokasi sumber daya komponen dan informasi administrasi lainnya. View digunakan untuk pengembang.

d. Deployment View

   Mendeskripsikan fisik dari sistem seperti komputer dan perangkat dan bagaimana hubungannya dengan lainnya. View digunakan untuk pengembang, pengintegrasi dan penguji.

2. Diagram

Diagram ialah presentasi grafis dari sekumpulan elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu view tertentu. Adapun jenis diagram antara lain:

a. Use Case Diagram

Menggambarkan sejumlah external actors dan hubungannya ke use case yang diberikan oleh sistem. Use case ialah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari use case symbol namun dapat juga dilakukan dalam activity diagrams. Use case digambarkan hanya dilihat dari luar actor dan bukan fungsi yang ada di dalam sistem.

b. Class Diagram

Menggambarkan struktur statis class di dalam sistem. Class merepresentasikan sesuatu yang ditangani oleh sistem. Class dapat berhubungan dengan yang lain melalui berbagai cara, yaitu: associated, dependent, specialed atau package.

c. Statechart Diagram

Menggambarkan semua state yang dimiliki oleh suatu object dari suatu class dan keadaan yang menyebabkan state berubah. Kejadian dapat berupa object lain yang mengirim pesan. State class tidak digambarkan untuk semua class, hanya yang mempunyai sejumlah state yang terdefinisi dengan baik.

d. Sequence Diagram

Menggambarkan semua state yang dimiliki oleh suatu object dari suatu class dan keadaan yang menyebabkan state berubah. Kejadian dapat berupa object lain yang mengirim pesan. State class tidak digambarkan untuk semua class, hanya yang mempunyai sejumlah state yang terdefinisi dengan baik

e. Collaboration Diagram

Menggambarkan kolaborasi dinamis seperti sequnce diagram. Dalam menunjukkan pertukaran pesan, collaboration diagram menggambarkan object dan hubungannya.

f. Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti use case atau interaksi.

g. Component Diagram

Menggambarkan alokasi semua kelas dan object kedalam komponen-komponen dalam desain fisik sistem software. Diagram ini memperlihatka pengaturan dan ketergantungan antara komponen-komponen software seperti source code, binary code dan komponen yang tereksekusi.

h. Deployment Diagram

Menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak sistem, menunjukkan hubungan komputer dengan perangkat satu sama lain dan jenis hubungannya.

2.5 *Smartphone*

Sebuah perangkat yang memungkinkan untuk melakukan komunikasi (seperti telp atau sms) juga di dalamnya terdapat fungsi PDA (*Personal Digital Assistant*) dan berkemampuan seperti layaknya komputer. Kata "*smartphone*" didefinisikan sebagai "ponsel yang menggabungkan (*Personal Digital Assistant*) PDA" oleh Amerika Kamus Oxford. Dalam pengertian singkat *smartphone* adalah sebuah perangkat yang dapat digunakan untuk melakukan komunikasi seperti menelpon atau mengirim pesan singkat, juga di dalamnya terdapat fungsi *personal digital assistant* dan berkemampuan seperti layaknya komputer dan kemampuan mengolah pesan pada *smartphone*[6].

Gambar 2.5-1 Smartphone

2.6 Android

Android adalah *operating sistem* (OS) berlisensi platform terbuka berbasis linux yang diperuntukan untuk mobile device seperti *smartphone* maupun tablet PC. *Android Open Source Project* (AOSP) yang dibuat oleh Google. Beberapa produsen mobile menanamkannya untuk menjadikan android sebagai sistem operasi. Berbagai keunggulan sistem operasi android mampu memenuhi kebutuhan informasi manusia secara cepat dan menarik. Android disematkan berbagai fitur sebagai fungsi utama[7].

1. Kerangka aplikasi, memungkinkan penggunaan dan penghapusan komponen yang tersedia.
2. Dalvik mesin virtual, mesin virtual dioptimalkan untuk perangkat telepon seluler.
3. Grafik: grafikdi 2DD dan grafis 3D berdasarkan pustaka OpenGL.
4. SQLite: sebagai penyimpanan data.
5. Dukungan pembacaan terhadap banyak format media.
6. GSM, *Bluetooth*, EDGE, 3G, 4G dan WiFi (tergantung piranti keras).
7. Kamera, *Global Positioning System* (GPS), kompas, NFC dan *accelerometer*.

Gambar 2.6-1 Sistem Operasi Android

2.7 Java

Bahasa java merupakan bahasa pemrograman yang *simple, object oriented, distributed, interpreted, robust, secure, architecture neutral, portable, high performance, multithreaded dan dynamic. Simple* karena java dikembangkan dari bahasa C++ dengan menghilangkan fungsi yang jarang digunakan dan kerumitan yang ada didalamnya serta memiliki sintaks yang hampir sama yang akan memberikan kemudahan dalam penggunaannya. *Object Oriented* karena menggunakan mekanisme obyek dalam desain dan pemrogramannya. Java memiliki dukungan luas terhadap pemrograman jaringan dan internet sehingga sangat banyak digunakan dalam aplikasi sistem terdistribusi dan jaringan internet[8].

Gambar 2.7-1 Java

2.8 Android Studio

Android studio adalah IDE (*Integrated Development Environment*) resmi untuk pengembangan aplikasi Android dan bersifat open source atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 mei 2013 pada event Google I/O Conference untuk tahun 2013. Sejak saat itu, Android Studio mengantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android. Android studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan *Eclipse* disertai dengan ADT *plugin (Android Development Tools*). Android studio memiliki fitur[9]:

1. Projek berbasis pada *Gradle Build* .
2. *Refactory* dan pembenahan bug yang cepat .
3. *Tools* baru yang bernama "Lint" dikalim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.
4. Mendukung *Proguard And App-signing* untuk keamanan.
5. Memiliki GUI aplikasi android lebih mudah .
6. Didukung oleh *Google Cloud Platfrom* untuk setiap aplikasi yang dikembangkan.

Gambar 2.8-1 Android Studio