

BAB 2

TINJAUAN PUSTAKA

2.1 Remaja Tunagrahita

2.1.1 Pengertian Remaja Tunagrahita

Definisi remaja adalah masa perpindahan dari anak-anak menjadi dewasa dan telah mencapai kematangan mulai dari kematangan mental, sosial, fisik, dan emosional [11]. Usia remaja diklasifikasi menjadi 3 kelompok, yaitu masa remaja awal yang memiliki rentang umur dari usia 12 tahun hingga 15 tahun, masa remaja pertengahan yang memiliki rentang umur dari usia 15 hingga 18 tahun, dan remaja akhir dengan rentang umur yaitu dari 18 tahun hingga 21 tahun. Sedangkan definisi tunagrahita menurut American Association on Mental Deficiency (AAMD) adalah kelainan yang dimiliki oleh individu yang memiliki fungsi intelektual di bawah rata-rata, yaitu 84 ke bawah berdasarkan tes yang dilakukan saat pemeriksaan jenis kebutuhan khusus di lembaga khusus [12].

Setiap individu penyandang tunagrahita sama halnya dengan remaja pada umumnya akan mengalami masa remaja juga. Maka, remaja tunagrahita dapat didefinisikan sebagai individu yang memiliki IQ di bawah rata-rata yang sedang berada di masa remaja yang secara fisik tidak memiliki perbedaan dengan remaja pada umumnya dalam perubahan fisik yang berhubungan dengan seksualnya, tetapi tingkah laku remaja tunagrahita masih seperti anak-anak karena kemampuan maksimalnya sama seperti anak normal usia 12 tahun sehingga perilakunya nampak tidak seimbang.

2.1.2 Keterampilan Melindungi Diri pada Remaja Tunagrahita

Remaja tunagrahita perlu diberikan keterampilan melindungi diri agar remaja tunagrahita dapat mengidentifikasi dan mengatasi diri dari situasi yang mengancam dirinya. Keterampilan melindungi diri adalah sebuah kemampuan diri dalam meningkatkan kewaspadaan, mendeteksi, dan merespons situasi yang membahayakan untuk melindungi dirinya [2]. Keterampilan melindungi diri menjadi upaya mencegah terjadinya kekerasan seksual. Terdapat program *Three Rs*

of *Child Self Protection* dari The Catholic Church dan Boy Scouts of America yang terdiri dari *recognize* (mengenali), *resist* (melawan), dan *report* (melaporkan).

1) *Recognize* (mengenali)

Pada bagian ini terdiri dari mengenali sentuhan baik (sentuhan pantas), sentuhan buruk (sentuhan tidak pantas), dan ciri-ciri orang jahat. Sentuhan baik (sentuhan yang pantas) berupa pelukan dan ciuman dari orang tua, guru menyentuh kepala dengan mengatakan “hebat usahamu”, pelukan hangat saudara karena keberhasilan, ciuman singkat di dahi atau pipi, bersalaman atau high five, dan sentuhan yang tidak membuat takut atau merasa tidak nyaman. Sedangkan sentuhan buruk (sentuhan yang tidak pantas) berupa sentuhan yang membuat takut atau malu atau senewen, sentuhan yang memaksa dan menyakitkan, sentuhan yang diminta untuk merahasiakan, memegang atau meraba-raba bagian terlarang (bagian tubuh pribadi), mencium biri, dan memukul pantat.

2) *Resist* (melawan)

Resist adalah kemampuan mempertahankan diri dengan cara melawan. Cara yang bisa dilakukan, yaitu berteriak minta tolong, mengabaikan rayuan atau bujukan, menolak secara lantang ketika ada yang mencoba melakukan tindak kekerasan seksual, memukul, menggigit, menendang, dan melarikan diri dari pelaku.

3) *Report* (melaporkan)

Report, yaitu kemampuan untuk melaporkan perilaku kurang menyenangkan secara seksual yang diterimanya dari orang dewasa, bersikap terbuka kepada orang tua agar orang tua nya dapat memantau kondisi anaknya. Pada komponen report, remaja dengan hambatan kecerdasan ringan diajari agar mampu bersikap terbuka atas tindakan kekerasan seksual yang diterimanya, dan mampu melaporkan pelaku pada orang dewasa atau lembaga lain yang berkepentingan dan dipercaya oleh anak untuk membantunya.

2.1.3 Kekerasan Seksual Pada Remaja Tunagrahita

Kekerasan seksual adalah tindakan seksual yang dilakukan dengan paksaan, tanpa persetujuan, dan bahkan ancaman. World Health Organization menjelaskan bahwa kekerasan seksual adalah setiap tindakan seksual, upaya untuk mendapatkan tindakan seksual, atau tindakan lain yang ditunjukkan terhadap seksualitas seseorang dengan paksaan, oleh siapapun tanpa memandang hubungannya dengan korban [13]. Kekerasan seksual pada tunagrahita akan berdampak pada psikologis, fisik, dan sosial [14]. Kekerasan seksual yang terjadi pada remaja tunagrahita diakibatkan oleh pelaku yang memanfaatkan keterbatasan yang dimiliki remaja tunagrahita. Selain itu, kurangnya pendidikan seksual pada remaja tunagrahita menjadi salah satu faktor yang berperan terhadap meningkatnya risiko kekerasan pada individu dengan hambatan kecerdasan. Pendidikan seksual yang rendah pada remaja tunagrahita disebabkan oleh beberapa faktor, salah satunya kesulitan memahami materi pendidikan seksual akibat sulitnya konsentrasi pada tunagrahita saat belajar pendidikan seksual.

2.1.4 Dampak Ketunagrahitaan terhadap Fokus Belajar

Rendahnya fungsi intelektual yang dimiliki oleh tunagrahita membuat remaja tunagrahita memiliki masalah dalam belajar. Tunagrahita mengalami gangguan pada susunan saraf pusat dan mempunyai kemampuan berfikir yang rendah, perhatian serta daya ingat yang lemah, konsentrasi yang mudah beralih, menyebabkan anak mengalami kesulitan menerima pelajaran atau mengikuti pembelajaran yang diikuti [15]. Gangguan atau keterlambatan kecerdasan menyebabkan seseorang dengan penyandang tunagrahita mengalami kesulitan untuk menerapkan materi dan fokus terhadap materi yang dipelajari dari rata-rata pada umumnya. Siswa tunagrahita cenderung sulit fokus dan harus ada sesuatu yang menarik bagi mereka belajar agar tertarik mengikuti pembelajaran hingga tuntas. Padahal fokus dibutuhkan pada saat pembelajaran berlangsung agar siswa remaja tunagrahita mampu memahami materi yang disampaikan guru.

2.2 Fokus

Fokus memiliki artian pemusatan perhatian atau tetap terjaga saat konsentrasi pada sesuatu hal. Fokus juga bisa diartikan dengan kemampuan konsentrasi pada sesuatu atau kegiatan yang sedang dilakukan. Fokus dapat dilihat dari tiga aspek, yaitu fokus perilaku dan kognitif [16]. Fokus Perilaku dan alokasi tatapan saling berkaitan sehingga orang cenderung melihat lebih lama pada item yang akan mereka pilih dibandingkan pada item yang akan mereka tolak, sehingga menimbulkan efek bias tatapan [18]. Maka, siswa dikatakan fokus belajar apabila penglihatannya berada pada area objek pembelajaran dan siswa tidak fokus belajar apabila penglihatannya tidak berada pada area objek pembelajaran. Fokus kognitif adalah tingkat investasi siswa dalam pembelajaran. Waktu yang dihabiskan siswa untuk tugas membaca berhubungan secara signifikan dengan tingkat fokus kognitif mereka. Selain itu, tingkat fokus kognitif dapat diprediksi berdasarkan durasi tugas membaca [16].

Berdasarkan penjelasan di atas, maka seseorang fokus dapat dideteksi oleh API *library* yang akan digunakan peneliti ketika sudut wajah secara absolut menghadap 20 derajat maka wajah dinyatakan wajah menghadap ke samping. Kemudian jika probabilitas mata terbuka di atas 0,9 maka mata dinyatakan terbuka, sebaliknya jika probabilitas mata terbuka di bawah 0,9 maka mata dinyatakan tertutup atau mengantuk. Lalu untuk pupil mata dapat dideteksi tidak fokus jika rasio pupil mata menunjukkan arah pandang horizontal kurang dari sama dengan 0,35 maka pupil mata dideteksi melihat ke arah kanan, lalu jika rasio pupil mata menunjukkan arah pandang horizontal lebih dari sama dengan 0,65 maka pupil mata dideteksi melihat ke arah kiri, dan jika rasio pupil mata menunjukkan arah pandang vertikal 0,40 menunjukkan pupil mata melihat ke arah atas dan arah pandang vertikal 0,65 menunjukkan pupil mata melihat ke arah bawah. Sehingga dapat disimpulkan remaja tunagrahita tidak fokus. Tetapi jika pupil mata menunjukkan arah pandang horizontal dan vertikal 0,5 pupil mata melihat ke arah depan atau *center*, sehingga dapat disimpulkan remaja tunagrahita fokus.

2.3 Aplikasi

Aplikasi adalah perangkat lunak yang dibuat memiliki tujuan untuk mengolah dokumen, membuat hiburan seperti permainan, melakukan aktivitas seperti mengirim pesan, dan mengatur sistem-sistem di Windows maupun perangkat mobile seperti android atau IOS [19]. Menurut sejarah juga, aplikasi komputer dibuat pertama kali yaitu untuk memecahkan masalah, perhitungan, atau pengolahan data [20]. Ketut Darmayuda mengategorikan aplikasi ke dalam dua kategori, yaitu aplikasi terkoneksi dan aplikasi terputus [21]. Aplikasi terkoneksi adalah aplikasi di mana pemakai aplikasi tersebut secara terus menerus melakukan koneksi ke suatu database sepanjang aplikasi itu dijalankan, sedangkan aplikasi terputus adalah sebuah aplikasi oleh pemakai aplikasi tidak secara konstan melakukan koneksi ke suatu database, melainkan koneksi hanya dibuka pada saat mengambil atau menyimpan suatu data saja ke database.

2.4 Aplikasi Interaktif

Interaktif memiliki arti yaitu seperangkat teknologi untuk perangkat keras (*hardware*) yang biasanya berada pada perangkat yang memiliki layar yang menyediakan interaksi antara perangkat dan pengguna. Dengan artian lain Media interaktif yaitu melakukan sesuatu pada perangkat (komputer, tablet, *smartphone*) dan perangkat yang bersangkutan melakukan sesuatu kembali kepada usernya.

Aplikasi interaktif dapat berupa website, aplikasi *kiosk*, aplikasi yang berjalan di perangkat seluler, permainan video, dan perangkat yang memberikan pengalaman user yang bisa berinteraksi langsung dengan komputer/sensor yang biasa ada di ruang publik. Semua dari Aplikasi interaktif ini dikembangkan dengan bahasa pemrograman yang berbeda-beda, dijalankan pada jenis perangkat keras (*hardware*) yang berbeda, dan memiliki tujuan yang berbeda-beda juga. Namun, aplikasi media interaktif memiliki kesamaan dalam pembuatannya, yaitu untuk menyediakan hubungan antara user dengan sistem yang sedang digunakan [17].

2.5 Smartphone

Smartphone adalah telepon seluler yang memiliki kemampuan yang sangat banyak, mulai dari perangkat kerasnya yang sangat modern, fitur yang sangat banyak dan beragam, hingga memiliki sistem operasi yang bisa menjalankan berbagai sistem dan aplikasi yang tersedia. Banyaknya fitur yang ada di smartphone juga semakin membuat mudah pengguna dalam mengakses apapun yang pengguna inginkan. Seperti mengakses permainan yang sekarang bisa dilakukan dimana saja, mencari informasi, mengirim atau menerima pesan, mendengarkan musik, menonton video, berselancar di internet, mencari ilmu, bahkan hingga berfoto [20]. Smartphone sebagai perangkat teknologi yang paling praktis dan bisa dibawa kemana saja membuat smartphone banyak diminati oleh penggunanya. Williams mengemukakan bahwa smartphone merupakan ponsel multimedia yang menggabungkan fungsionalitas PC dan handset sehingga menghasilkan gadget yang mewah, di mana terdapat pesan teks, kamera, pemutar musik, video, permainan, akses email, tv digital, pengelola informasi pribadi, GPS, dan internet [22]. Hal tersebut menyebabkan saat ini di kalangan masyarakat banyak yang minat terhadap smartphone berbasis sistem android.

2.6 Android

Android didirikan pertama kali oleh Rich Miner, Chris White, Nick Sears, dan Andy Rubin pada tahun 2003 sebelum diakuisisi oleh Google pada tahun 2005. Google mengakuisisi Android karena Google yakin akan membuat suatu perangkat yang akan menarik banyak konsumen dalam Android ini [23]. Android adalah sistem operasi yang dikhususkan untuk perangkat mobile yang berbasis linux sebagai dasar operasi sistemnya. Hal ini sejalan dengan Safaat yang menyatakan juga bahwa Android merupakan sebuah sistem operasi pada smartphone yang sifatnya terbuka dan berbasis pada sistem operasi Linux [24].

Pada tahun 2007 dibuatlah suatu usaha pengembangan untuk android yang dinamakan Open Handset Alliance (OHA), yaitu sebuah kerjasama untuk mengembangkan Android dari 34 perusahaan di bidang perangkat lunak, perangkat

keras, dan telekomunikasi yang diantaranya adalah HTC, Google, Motorola, Intel, Qualcomm, T-Mobile, dan Nvidia [25].



Gambar 2. 1 Logo Android

Perkembangan dari Android sangat pesat. Hal ini ditandai dengan banyaknya versi Android. Android 1.0/1.1 pertama kali muncul pada 22 Oktober 2008, kemudian disusul dengan Android 1.5 (Cupcake), Android 1.6 (Donut), Android 2.1 (Eclair), Android 2.2 (Froyo), Android 2.3 (GingerBread). Android 3.0 (Honey Comb), Android 4.0 (Ice Cream Sandwich), Android 4.1 (Jelly Bean), Android 4.2 (Jelly Bean), Android 4.3 (Jelly Bean), Android 4.4 (Kitkat), Android 5.0 (Lollipop), Android 6.0 (Marshmallow), Android 7.0 (Nougat), Android 8.0 (Oreo), Android 9.0 (Pie), Android 10.0 (Q), dan yang terbaru adalah Android 11.0 (R) [24].

Sistem Android merupakan satu sistem yang terbuka bagi para pengembang aplikasi untuk menciptakan aplikasi mereka sendiri menggunakan smartphone berbasis Android sehingga kemudahan mengembangkan aplikasi ini menyebabkan banyaknya pilihan aplikasi yang bermanfaat di toko aplikasi smartphone Android, yaitu Google Play [22]

2.7 Android Studio

Android Studio adalah sebuah IDE (*Integrated Development Environment*) untuk pengembangan atau pembuatan aplikasi untuk sistem operasi *android*. Aplikasi ini dibuat dan dikembangkan oleh Google pada tanggal 16 Mei 2013 dan bisa digunakan secara gratis oleh siapapun. Sebelum adanya Android Studio ini, untuk pengembangan dan membuat aplikasi *android* ini menggunakan *Eclipse* yang dibuat pada tahun 29 November 2001 [23].



Gambar 2. 2 Logo Android Studio

Android Studio menawarkan banyak fitur untuk meningkatkan produktivitas pengembangan aplikasi Android, yaitu sistem versi berbasis gradle yang fleksibel, emulator yang cepat dan kaya fitur, lingkungan yang menyatu untuk pengembangan bagi semua perangkat android, instant run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru kode templat dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh, serta memiliki alat pengujian dan kerangka kerja yang ekstensif [26].

2.7.1 Java

Java dalam ilmu komputer adalah bahasa yang digunakan dalam dunia pemrograman yang diperkenalkan kepada publik pada tahun 1995 oleh Sun Microsystem Inc., yang saat itu Java diciptakan dalam pimpinan James Gosling. Sebelum menggunakan nama Java, bahasa pemrograman ini diciptakan dengan nama Oak. Awalnya Java diciptakan dikarenakan Microsystem Inc. ingin membuat sebuah bahasa pemrograman yang dapat berjalan di semua perangkat tanpa harus ada *platform* yang digunakan oleh perangkat tersebut [23].



Gambar 2. 3 Logo Java

Java merupakan bahasa pemrograman berorientasi objek. Java menerapkan konsep Write Once, Run Anywhere. Kode Java yang telah dikompilasi dalam bentuk *bytecode* dapat dijalankan pada berbagai platform [27].

2.7.2 Integrated Development Environment (IDE)

Integrated Development Environment adalah sebuah aplikasi untuk pembuatan aplikasi perangkat lunak dengan fungsi yang sudah terintegrasi seperti kode, pengeditan, *debugger*, *compiler*, dan masih banyak lagi [23].

2.7.3 Android SDK

Android SDK (Software Development Kit) adalah alat yang digunakan untuk kepentingan pembuatan atau pengembangan pada sistem operasi *Android* yang bahasa pemrogramannya memakai bahasa java. Dalam SDK ini memiliki fitur yang diantaranya adalah : *debugger*, *emulator*, *compiler*, *emulator*, kode pembantu, pengumpulan dan pengolahan dokumen, *libraries* dan *tutorial*. Semua fitur tersebut sangat berguna dan membantu dalam pengembangan atau pembuatan aplikasi *Android* [23].

2.7.4 JDK (Java Development Kit)

JDK adalah sebuah paket API yang berisi kumpulan perintah-perintah dalam baris untuk membuat aplikasi dalam java yang berfungsi untuk melakukan penggabungan dari program java yang sudah dibuat oleh pengembang [23].

2.8 API (Application Programming Interface)

API (Application Programming Interface) adalah fasilitas yang saling terhubung sebagai media sarana untuk bertukar data informasi yang menggunakan 2 atau lebih dalam mengimplementasikannya. API ini bisa menggunakan berbagai bahasa pemrograman yang berbeda-beda. Kegunaan API ini mempermudah pengguna dalam mengakses sebuah informasi dari aplikasi pengguna dengan aplikasi yang lainnya [23].

2.8.1 ML Kit

ML Kit adalah SDK *smartphone* yang menyediakan *library machine learning* yang disediakan oleh Google untuk sistem operasi Android dan iOS disatukan dalam paket yang bagus dan mudah untuk digunakan oleh pemula ataupun yang sudah berpengalaman dalam menggunakan dan mengimplementasikan *machine learning*. ML Kit juga menyediakan API yang mudah digunakan dan bisa sangat membantu pemula dalam menggunakan *machine learning* pendeteksi wajah di *smartphone* berbasis Android [28]. Untuk pendeteksian wajah menggunakan ML Kit ini pertama pastikan versi API Android yang akan digunakan ialah menggunakan versi Android API level 21 atau lebih tinggi agar ML Kit bisa dijalankan, kemudian dalam file *build.gradle* memasukkan repositori Maven Google di bagian *buildscript* dan *allprojects*, untuk memaketkan model ML Kit dapat dilihat pada gambar 2.4 dibawah ini

```
dependencies {  
    // ...  
    // Use this dependency to bundle the model with your app  
    implementation 'com.google.mlkit:face-detection:16.1.7'  
}
```

Gambar 2. 4 Cara memaketkan model MI Kit

Setelah melakukan pemasangan *library* Android ML Kit pada gradle, langkah selanjutnya yaitu mengonfigurasi detektor wajah yang ingin diterapkan, sebagai contoh mengonfigurasi wajah dapat dilihat pada gambar 2.5 dibawah ini

```
// High-accuracy landmark detection and face classification
FaceDetectorOptions highAccuracyOpts =
    new FaceDetectorOptions.Builder()
        .setPerformanceMode(FaceDetectorOptions.PERFORMANCE_MODE_ACCURATE)
        .setLandmarkMode(FaceDetectorOptions.LANDMARK_MODE_ALL)
        .setClassificationMode(FaceDetectorOptions.CLASSIFICATION_MODE_ALL)
        .build();

// Real-time contour detection
FaceDetectorOptions realTimeOpts =
    new FaceDetectorOptions.Builder()
        .setContourMode(FaceDetectorOptions.CONTOUR_MODE_ALL)
        .build();
```

Gambar 2. 5 Mengonfigurasi Pendeteksi Wajah

Pengonfigurasi deteksi wajah bisa berbeda-beda dalam pengimplementasiannya, bagaimana kebutuhan developer yang akan membangun suatu aplikasinya.

2.8.2 CameraX

CameraX adalah *library* yang dibuat untuk memudahkan pengguna yang ingin membuat atau mengembangkan aplikasi menggunakan kamera dalam aplikasi yang dibuat. CameraX ini memiliki *nterface* API yang mudah digunakan dan bisa digunakan di versi Android 5.0 ke atas [28]. Untuk menambahkan *library* ini didalam *code* yang akan dibuat, dapat menggunakan langkah-langkah sebagai berikut:

1. Tambahkan PreviewView ke dalam *codingan* pada aplikasi yang akan dibuat. Kode dapat dilihat pada gambar 2.6 dibawah ini

```
<FrameLayout
    android:id="@+id/container">
    <androidx.camera.view.PreviewView
        android:id="@+id/previewView" />
</FrameLayout>
```

Gambar 2. 6 PreviewView CameraX

2. Setelah menambahkan PreviewView pada *codingan* yang sedang dikerjakan, lalu tambahkan kode untuk meminta CameraProvider kedalam *codingan* sistem yang akan dibuat. Kode untuk meminta CameraProvider dapat dilihat pada gambar 2.7 dibawah ini

```
import androidx.camera.lifecycle.ProcessCameraProvider
import com.google.common.util.concurrent.ListenableFuture

public class MainActivity extends AppCompatActivity {
    private ListenableFuture<ProcessCameraProvider> cameraProviderFuture;

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        cameraProviderFuture = ProcessCameraProvider.getInstance(this);
    }
}
```

Gambar 2. 7 Request CameraProvider CameraX

3. Lalu setelah meminta CameraProvider kedalam kode yang sedang dibangun, verifikasi bahwa instalasi berhasil ketika *view* dibuat. Dibawah ini merupakan kode untuk melakukan cara melakukan cek apakah instalasi CameraX berhasil dilakukan dapat dilihat pada gambar 2.8 dibawah ini

```

cameraProviderFuture.addListener(() -> {
    try {
        ProcessCameraProvider cameraProvider = cameraProviderFuture.get();
        bindPreview(cameraProvider);
    } catch (ExecutionException | InterruptedException e) {
        // No errors need to be handled for this Future.
        // This should never be reached.
    }
}, ContextCompat.getMainExecutor(this));

```

Gambar 2. 8 Check CameraProvider Availability.

2.8.3 Gaze Tracking

Gaze Tracking merupakan *library machine learning* yang mampu melakukan deteksi pupil mata dan pelacakan mata dengan memanfaatkan teknologi kamera untuk melakukan deteksinya. Gaze Tracking juga bisa menjalankan pelacakan tatapan mata dengan cara mendeteksi bagian iris yang merupakan salah satu bagian mata atau bisa juga disebut dengan pupil [31]. Untuk pendeteksian pupil menggunakan API Gaze Tracking ini dapat dilihat pada gambar 2.9 berikut

```

gaze.refresh(frame)

frame = gaze.annotated_frame()
text = ""

if gaze.is_blinking():
    text = "Blinking"
elif gaze.is_right():
    text = "Looking right"
elif gaze.is_left():
    text = "Looking left"
elif gaze.is_center():
    text = "Looking center"

# cv2.putText(frame, text, (90, 60), cv2.FONT_HERSHEY_TRIPLEX, 1.6, (147, 58, 31), 2)

left_pupil = gaze.pupil_left_coords()
right_pupil = gaze.pupil_right_coords()
# cv2.putText(frame, "Left pupil: " + str(left_pupil), (90, 130), cv2.FONT_HERSHEY_TRIPLEX, 2, (147, 58, 31), 2)
# cv2.putText(frame, "Right pupil: " + str(right_pupil), (90, 200), cv2.FONT_HERSHEY_TRIPLEX, 2, (147, 58, 31), 2)

font_path = "Fonts/Times New Roman/times new roman bold.ttf"
font = ImageFont.truetype(font_path, 48)
img_pil = Image.fromarray(frame)
draw = ImageDraw.Draw(img_pil)
draw.text((90, 60), text, font=font, fill=(255, 255, 255))
draw.text((90, 130), "Left pupil: " + str(left_pupil), font=font, fill=(255, 255, 255))
draw.text((90, 200), "Right pupil: " + str(right_pupil), font=font, fill=(255, 255, 255))
frame = np.array(img_pil)

cv2.imwrite("test.jpg", frame)

```

Gambar 2. 9 Pendeteksian Pupil Gaze Tracking

2.9 RoomDB

RoomDB adalah sebuah pengembangan dari turunan database SQLite dengan mengadakan lapisan abstraksi di atas SQLite agar bisa memungkinkan akses basis data yang lancar. Pengembang sistem aplikasi yang memiliki jumlah data yang banyak bisa sangat dimudahkan dengan menyimpan basis data yang digunakan secara lokal [29].

2.10 Unified Modeling Language

Unified Modeling Language atau bisa disingkat menjadi UML adalah salah satu model untuk merancang pembuatan perangkat lunak yang menghasilkan sintandarisasi diagram dan berbagai ketentuan spesifikasi yang bisa memperbaiki interaksi dan komunikasi dari banyaknya *stakeholder*. *Stakeholder* ini memiliki keikutsertaan dalam memperbaiki kualitas yang dimiliki oleh perangkat lunak yang akan dibuat, meminimalisir kesalahan, dan membuat model yang dipakai oleh UML ini mudah untuk di masukan kedalam perangkat lunak pemodelan yang dapat membuat sekumpulan user, analisis, desainer, dan tester bisa bekerja sama [30]. Berikut ini adalah beberapa fungsi yang terdapat pada UML bisa dilihat sebagai berikut :

1. Visualisasi yaitu memberikan gambar dan visual yang dapat merepresentasikan kebutuhan, proses, solusi desain, dan arsitektur dengan menggunakan macam-macam bahasa pemrograman dan pembangunan rekayasa secara umum.
2. Penentuan dalam pemodelan yang dapat membantu dalam memperbaiki dari suatu solusi agar mendapatkan kualitas yang tidak membuat kesalah pahaman antara pengembang aplikasi dan pengguna aplikasi.
3. Memberikan sebuah informasi seperti gambaran model dengan menggunakan bahasa pemodelan visual yang benar dalam pengembangan sistem.
4. Bisa memberikan pemodelan visual pada sistem yang berorientasikan objek.
5. Mempermudah pengguna dalam membaca suatu sistem yang dibuat.
6. Bisa berguna menjadi sebuah gambaran rencana awal yang bisa menjelaskan sebuah informasi yang rinci pada perancangan yang berbentuk pengkodean dalam suatu program.

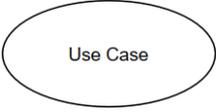
2.10.1 Use Case Diagram

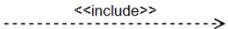
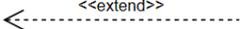
Use case diagram adalah ringkasan yang berisi dari fungsi sebuah sistem atau bisnis proses dari sudut pandang pengguna. Dalam hal ini pengguna yang menggunakan sistem menjadi langkah awal untuk pembuatan *use case diagram* [30]. Berikut ini adalah 3 komponen dari *use case diagram* :

1. Pembatasan Sistem yang berguna untuk menjadi pembeda antara aktor dan *use case*.
2. Aktor yang biasa di lambangkan dengan gambar *stick man* ini berguna untuk menunjukan seseorang atau aplikasi pihak ketiga yang berada di luar sistem yang dibuat.
3. *Use Case* ini berguna untuk menunjukan fungsi dari suatu sistem yang dibuat berdasarkan persyaratan dari pengguna.

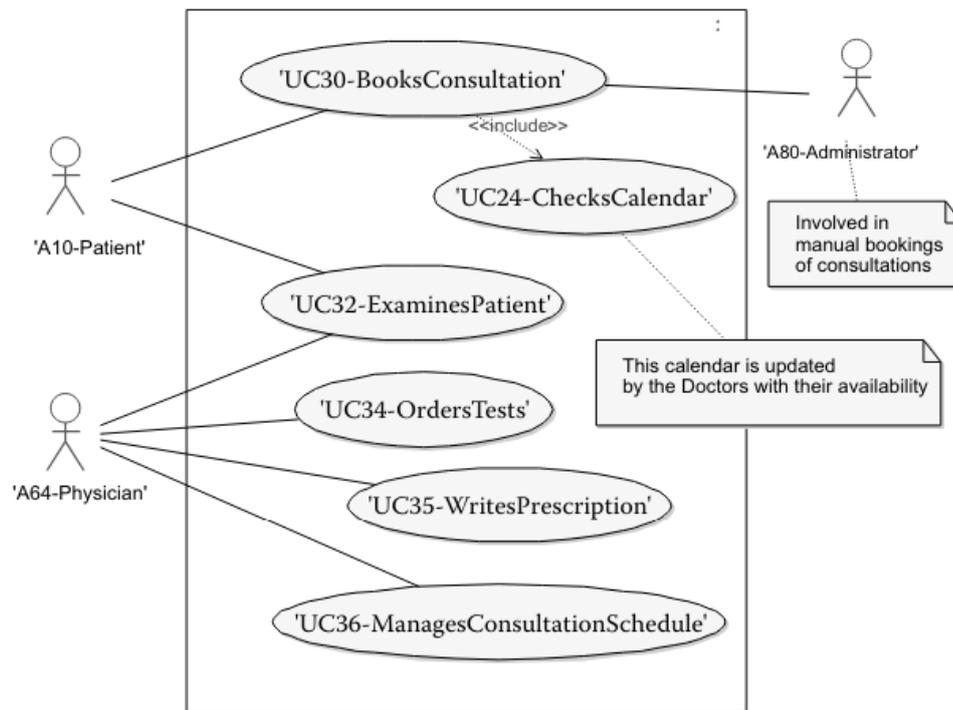
Berikut simbol use case diagram dapat dilihat pada tabel 2.1 berikut

Tabel 2. 1 Simbol Use Case Diagram

No.	Simbol	Nama	Keterangan
1.	 Actor	Actor	Menggambarkan pengguna, atau sistem eksternal, yang berinteraksi dengan Use Case
2.	 Use Case	Use Case	Abstraksi antara sistem dan Actor
3.		Association	Menggambarkan Penghubung antara Actor dengan Use Case

4.		Generalization	Menggambarkan bahwa spesialisasi aktor untuk dapat berpartisipasi dengan Use Case
5.		Include	Menggambarkan bahwa suatu Use Case adalah bagian dari Use Case lain
6.		Exclude	Menggambarkan bahwa suatu Use Case adalah fungsi tambahan dari Use

Dibawah ini adalah gambar contoh penggunaan *use case diagram* dapat dilihat pada gambar 2.10

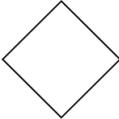
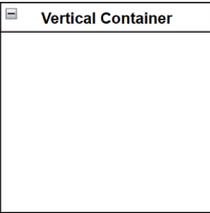


Gambar 2. 10 Contoh Use Case Diagram

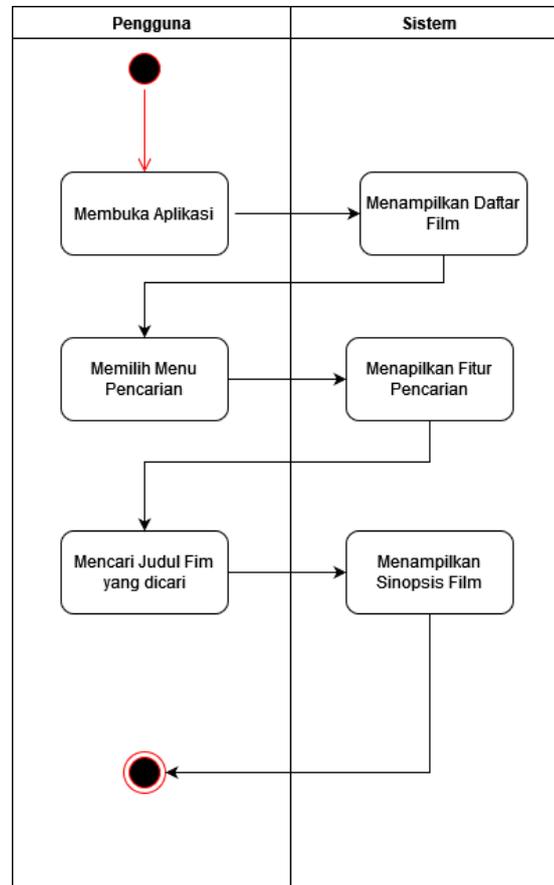
2.10.2 Activity Diagram

Activity diagram adalah sebuah diagram yang menggambarkan konsep dari aliran data, aksi yang tertata serta dirancang dengan baik dalam suatu system yang sedang dibangun [30]. *Activity diagram* juga memiliki alur khusus dalam *use case* yang bisa mendepresikan pengguna dalam melakukan interaksi utama dan interaksi alternatif, dan juga interaksi pengecualian pada system yang digambarkan dengan baik oleh *activity diagram* ini [30]. Berikut adalah contoh simbol yang pada *activity diagram* dapat dilihat pada tabel 2.2

Tabel 2. 2 Simbol Activity Diagram

No.	Simbol	Nama	Keterangan
1.		Start	Posisi Awal Activity
2.		Activity	Activity yang dilakukan sistem
3.		Decision	Percabangan Activity berdasarkan suatu pilihan
4.		Join	Penggabungan Activity dua Activity menjadi satu Activity
5.		End	Status Akhir Activity
6.		Swimlane	Pemisah dari organisasi yang bertanggung jawab pada suatu Activity

Dan berikut bisa dilihat contoh Activity diagram dapat dilihat pada gambar 2.11.

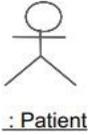


Gambar 2. 11 Contoh Penggunaan Activity Diagram

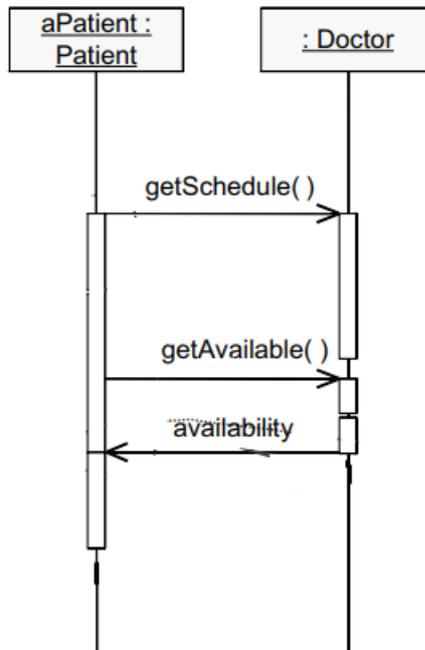
2.10.3 Sequence Diagram

Sequence diagram adalah model diagram yang memodelkan interaksi antar objek secara detail disebutkan maupun dapat berupa bobotnya. Objek yang dipakai tersebut dapat secara detail di sebutkan maupun dapat berupa objek tanpa nama termasuk kedalam suatu *class*. Urutan dari pengekseskusion dari setiap pesan antara objek saat sistem sedang berjalan dimodelkan menggunakan *sequence diagram* [30]. Berikut merupakan contoh simbol yang terdapat pada *sequence diagram* dapat dilihat pada tabel 2.3 berikut

Tabel 2. 3 Simbol Sequence Diagram

No.	Simbol	Nama	Keterangan
1.		Actor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem
2.		Object	Komponen yang menerima atau mengirim pesan secara berurutan
3.		Timeline	Menandakan kehidupan objek
4.		Focus of Control	Menandakan ketika suatu objek mengirim atau menerima pesan
5.		Message	Mengirim pesan
6.		Self message	Mengirim pesan pada diri sendiri
7.		Asynchronous Message	Pesan asynchronous
8.		Return Message	Pesan tanggapan
9.		Destroying Object	Menghancurkan object saat tidak digunakan

Berikut ini adalah contoh *sequence diagram* dapat dilihat pada gambar 2.12.

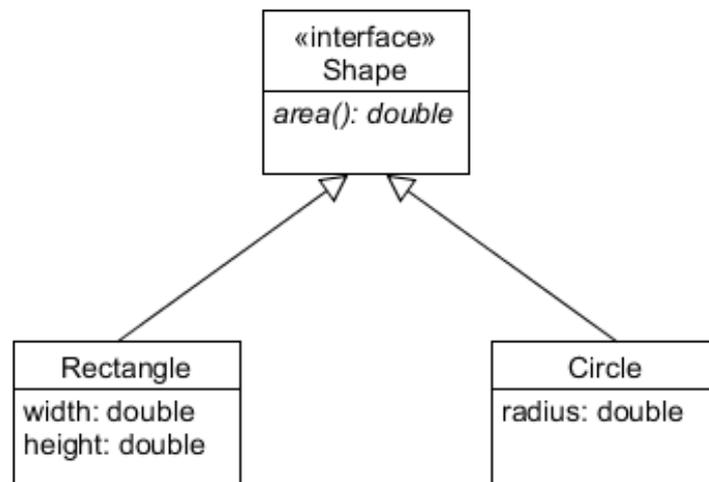


Gambar 2. 12 Contoh Sequence Diagram

2.10.4 Class Diagram

Class diagram adalah jenis diagram yang menunjukkan *class* yang ada dari suatu sistem dan memiliki hubungan logisnya. Maka dari itu, diagram ini dapat memberikan deskripsi tentang sistem dan ikatan yang berada di dalam sistem yang dikembangkan. Dengan demikian *class diagram* ini yaitu menggambarkan struktur statis yang berada dari suatu sistem [30].

Berikut ini dapat dilihat contoh dari penggunaan *class diagram* pada gambar 2.13.



Gambar 2. 13 Contoh Penggunaan Class Diagram