

BAB II

TINJAUAN PUSTAKA

2.1 Perancangan Aplikasi

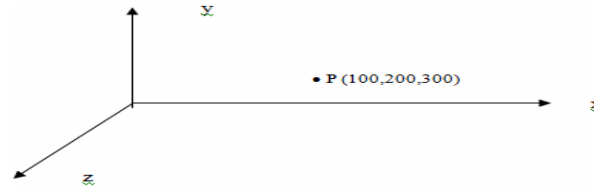
Perancangan aplikasi merupakan aplikasi yang akan dibuat. Untuk dapat merancang konsep dalam membuat aplikasi multimedia dibutuhkan kreatifitas. Kreatifitas adalah kemampuan untuk menyajikan gagasan atau ide baru. Sedangkan inovasi merupakan aplikasi dari gagasan atau ide baru tersebut. Untuk menciptakan ide yang orisinil tidaklah mudah, maka dapat digunakan beberapa teknik untuk menciptakan ide, yaitu penyesuaian (adaptasi).

2.2 Objek 3 Dimensi

Yang dimaksud realitas grafis 3D adalah bagaimana membentuk suatu objek 3D agar sesuai dengan kenyataan suatu objek yang dilihat sehari-hari. Sebagian besar objek nyata bersifat solid atau tidak tembus pandang. Objek 3D banyak yang dibuat masih dalam bentuk rangka (*wireframe*). Dalam implementasinya model rangka ini masih banyak dipakai untuk mengetahui bagian dalam suatu objek. Realitas grafis 3D dapat dilakukan dengan : mengarsir poligon, *rendering*, dan *shadowing*. Dan komponen dasar yang terdapat dalam Grafik 3D adalah

1. Objek yang akan ditampillkan, harus memiliki koordinat 3D (*world coordinate*-koordinat dunia) pada setiap titik pokoknya.
2. Posisi pengamat atau pada metode kamera sintetis disebut dengan posisi 'kamera', ditentukan dari koordinat dunia (x,y,z)
3. Titik fokus 'kamera', dalam koordinat dunia (x,y,z,) yang merupakan titik pusat pandang.

2.3 Sistem Koordinat 3D atau sistem Koordinat Dunia



Gambar 2.1 : Sistem Koordinat 3D

Setiap titik pada koordinat dunia memiliki 3 sumbu x,y, dan z. Misal titik P(100,200,300) berarti titik P tersebut berada pada sumbu

- x sejauh 100 satuan
- y sejauh 200 satuan
- z sejauh 300 satuan

koordinat dunia dalam grafika 3D dapat menggunakan satuan yang riil. Misal bila lingkup objek berupa sebuah kota, maka satuannya dapat berupa kilometer, bila lingkup suatu gedung satuannya dapat berupa meter, bila hanya suatu ruang dapat berupa centimeter, dan lain sebagainya.

2.4 Transformasi 3D

Pada objek tiga dimensi dapat dilakukan 3 transformasi pokok yaitu penggeseran (*translation*), perputaran (rotasi) , pembesaran (*scaling*). Adapun jenis transformasi lain adalah pembebanan (*shearing*), dan pencerminan (*mirroring*). Didalam program, transformasi ini juga dapat dilakukan dengan mengubah persamaan menjadi operasi matriks 4x4.

Matriks transformasi adalah matriks yang membuat sebuah objek mengalami perubahan, baik berupa perubahan posisi maupun perubahan ukuran. Matriks 3D dinyatakan dalam ukuran 4x4, di mana kolom ke-4 digunakan untuk menyediakan tempat untuk proses translasi. Transformasi 3D dapat dilakukan dengan menggunakan perkalian antara matriks transformasi dan vektor posisi dari setiap titik pada objek.

Transformasi 3D pada dasarnya hampir sama dengan transformasi 2D, hanya

pada 3D kita menghitung sumbu Z. Sama seperti pada 2D, ada tiga transformasi dasar yang dapat dilakukan terhadap verteks, yaitu:

1. Tranlansi
2. Pensekalaan
3. Rotasi

2.5 Image Processing

Image Processing adalah sebuah metode yang dipakai untuk memproses atau memanipulasi *image* dalam bentuk dua dimensi. Segala operasi untuk memperbaiki, menganalisa, atau mengubah suatu gambar disebut *image processing* (A. Usman, 2005, Pengolahan Citra Digital dan Teknik pemogramannya). Dalam sejarahnya, *image processing* telah diaplikasikan dalam berbagai bentuk, dengan tingkat kesuksesan yang cukup besar. Seperti berbagai cabang ilmu lainnya, *image processing* pula gabungan berbagai cabang-cabang ilmu, diantaranya adalah optik, elektronik, matematika, fotografi, dan teknologi komputer. Beberapa faktor yang menyebabkan sistem *image processing* menjadi berkembang pesat pada saat ini, salah satu yang utama adalah dibutuhkannya suatu teknologi yang dapat bekerja secara mandiri, dalam artian teknologi tersebut dapat mengambil keputusan sendiri dari hasil pengolahan data sebelumnya. Pada umumnya, objektifitas dari *image processing* adalah mentransformasikan atau menganalisis suatu gambar sehingga informasi baru tentang gambar dapat dibuat dengan lebih jelas.

2.6 Metode Morphing

Morphing adalah efek khusus dalam film dan animasi yang berubah (atau *morph*) satu gambar ke gambar lain melalui transisi mulus. Paling sering digunakan untuk menggambarkan seseorang berubah menjadi lain melalui sarana teknologi atau sebagai bagian dari fantasi atau urutan surealis. Secara tradisional penggambaran tersebut akan dicapai melalui teknik *cross*-memudar

pada film. Sejak awal 1990-an, ini telah digantikan oleh perangkat lunak komputer untuk membuat transisi lebih realistis.

Algoritma morphing terus maju dan program secara otomatis dapat *morph* gambar yang sesuai cukup erat dengan relatif sedikit instruksi dari pengguna. Hal ini telah menyebabkan penggunaan teknik *morphing* untuk menciptakan meyakinkan efek gerak lambat yang tidak pernah ada dalam film atau video rekaman asli dengan *morphing* antara setiap frame individu menggunakan optik aliran *technology*. *Morphing* juga muncul sebagai transisi teknik antara satu adegan dan satu lagi di acara televisi, bahkan jika isi dari dua gambar yang sama sekali tidak terkait. Algoritma dalam hal ini mencoba untuk menemukan titik yang sesuai antara gambar dan mendistorsi satu ke yang lain karena mereka *crossfade*.

2.7 Android

Arfida, Amnah dan Wibowo (2018 : 52) menguraikan bahwa Android merupakan sebuah sistem operasi yang berbasis Linux untuk perangkat portable seperti *smartphone* dan *computer tablet*. Android menyediakan *platform* terbuka bagi programmer untuk mengembangkan Aplikasi sendiri pada berbagai perangkat dengan sistem operasi Android.

2.8 Perangkat Lunak Yang Digunakan

2.8.1 Android Studio

Harison, Busran, Yogi Putra (2016 : 3) menguraikan bahwa Android studio adalah sebuah IDE (Integrated Development Environment) yang bisa digunakan untuk pengembangan aplikasi android, dan dikembangkan oleh Google. Android studio merupakan pengembangan dari eclipse IDE dan dibuat berdasarkan IDE java populer, yaitu intellij IDEA.

Juansyah (2015 : 17) menguraikan bahwa Android studio adalah IDE (Integrated Development Environment) resmi untuk pengembangan aplikasi Android dan

bersifat open source atau gratis. Android Studio menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android.

Yudhanto dan Wijayanto (2017 : 17) Android studio ini bersifat free dibawah Apache Lincense 2.0. Android Studio awalnya bermula dengan versi 0.1 pada 16 Mei 2013, kemudian dibuat versi beta 0.8 yang dirilis 2014. Berbasiskan JetBrains' IntelliJ IDEA, Studio di desain khusus untuk Android Development. Android studio memiliki fitur :

- a. Projek berbasis pada Gradle Build
- b. Refactory dan pembenahan bug yang cepat
- c. Tools baru yang bernama "Lint" dikalim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.
- d. Mendukung Proguard And App-signing untuk keamanan.
- e. Memiliki GUI aplikasi android lebih mudahf. Didukung oleh Google Cloud Platfrom untuk setiap aplikasi yang dikembangkan.

2.8.2 Java

Harison, Busran (2016 : 2) Java adalah bahasa pemrograman serbaguna. dapat digunakan membuat suatu program, mendukung sumber daya internet yang saat ini populer, yaitu World Wide Web atau sering disebut Web saja. Mendukung aplikasi klien/ server, baik dalam jaringan lokal (LAN) maupun WAN.

Wahana Komputer (2015 : 2) menguraikan bahwa Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer maupun telepon genggam. Bahasa pemrograman ini dibuat oleh James Gosling saat masih bergabung di Sun Microsystems, di mana saat ini merupakan bagian dari Oracle yang dirilis pada tahun 1995. Bahasa ini banyak mengadopsi sintaks yang terdapat pada C dan C++, tetapi dengan sintaksis mode objek yang lebih sederhana. Java merupakan bahasa pemrograman yang bersifat umum/nonspesifik dan secara khusus didesain untuk memanfaatkan implementasi semaksimal mungkin. Fungsi Java memungkinkan aplikasi Java mampu berjalan di beberapa platform sistem operasi yang berbeda. Java dikenal pula dengan slogannya "Tulis sekali, jalankan

di mana pun”. Saat ini Java secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web.

2.9 Metode Perangkat Lunak Yang Digunakan

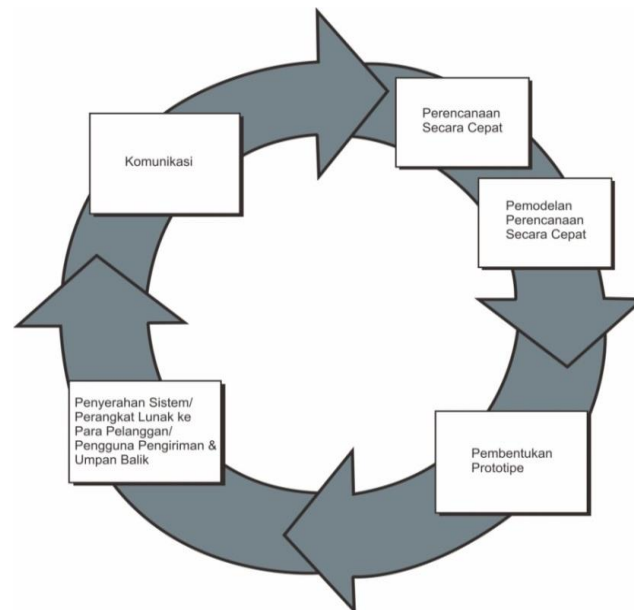
2.9.1 Metode Prototype

Metode pengembang perangkat lunak yang digunakan pada aplikasi ini adalah metode *prototype*. Dalam melakukan perancangan system yang akan dikembangkan dapat menggunakan metode *prototype*. Metode ini cocok digunakan untuk mengembangkan sebuah perangkat yang akan dikembangkan kembali. Kemudian membuat sebuah rancangan kilat yang selanjutnya akan dievaluasi kembali sebelum diproduksi secara benar.

(Pressman 2012) *Prototype* bukanlah merupakan sesuatu yang lengkap, tetapi sesuatu yang harus dievaluasi dan dimodifikasi kembali. Segala perubahan dapat terjadi pada saat *prototype* dibuat untuk memenuhi kebutuhan pengguna dan pada saat yang sama memungkinkan pengembang untuk lebih memahami kebutuhan pengguna secara lebih baik.

Berikut adalah tahapan dalam metode *prototype*:

- a) Komunikasi dan pengumpulan data awal, yaitu wawancara terhadap pihak yang terkait dalam penelitian dan analisis terhadap kebutuhan pengguna (wawancara terhadap owner atau karyawan salon)
- b) Perencanaan Secara Cepat yaitu pembuatan desain secara umum untuk selanjutnya dikembangkan kembali.
- c) Pembentukan *Prototype* yaitu pembuatan perangkat *prototype* termasuk pengujian dan penyempurnaan.
- d) Evaluasi terhadap *Prototype*, yaitu mengevaluasi *prototype* dan memperhalus analisis terhadap kebutuhan pengguna.
- e) Perbaikan *Prototype*, yaitu pembuatan tipe yang sebenarnya berdasarkan hasil dari evaluasi *prototype*.
- f) Produksi akhir, yaitu memproduksi perangkat secara benar sehingga dapat digunakan oleh pengguna.



Gambar 2.2 Metode *Prototype* (Pressman, 2012)

2.10 Pengujian *Black-box Testing*

Metode pengujian pada aplikasi ini menggunakan *Black Box Testing* yaitu yang berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program.

Black Box Testing bukanlah solusi alternatif dari *White Box Testing* tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh *White Box Testing* (Pressman 2012). *Black Box Testing* cenderung untuk menemukan hal-hal berikut:

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.

2.11 Sistem Pemodelan

2.11.1 *Unified Modeling Language* (UML)

Unified Modeling Language (UML) adalah sebuah bahasa yang berdasarkan gambar untuk memvisualisasikan, menspesifikasikan, membangun dan

pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis Objek. *Unified Modeling Language* (UML) bukanlah merupakan bahasa pemrograman tetapi model-model yang tercipta berhubungan langsung dengan berbagai macam bahasa pemrograman berorientasi obyek, seperti Java (Syafitri 2016). UML tersusun atas sejumlah elemen grafis membentuk diagram-diagram. Dalam penelitian ini melakukan desain hanya 2 diagram yaitu *Use Case Diagram* dan *Activity Diagram*.


2.11.2 Use Case Diagram



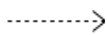


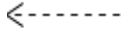
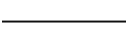

Munawar (2018 : 89) menguraikan bahwa *Use Case* adalah deskripsi fungsi sebuah system dari perspektif pengguna. *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antara pengguna sebuah system dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah system dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan system disebut sebagai *Scenario*. Setiap *Scenario* menggambarkan urutan kejadian. Setiap urutan di inialisasi oleh orang, system yang lain, perangkat keras atau urutan waktu. Dengan demikian, secara singkat bias dikatakan *Use Case* adalah serangkaian *Scenario* yang digabungkan bersaa-sama oleh tujuan umum pengguna.

Use Case dibuat berdasarkan kebutuhan Aktor. *Use Case* harus merupakan 'apa' yang dikerjakan software aplikasi, bukan 'bagaimana' software aplikasi mengerjakannya.

Tabel 2.1 pada halaman berikut ini adalah Simbol-simbol yang digunakan dalam *Use Case Diagram* :

Tabel 2.1 Simbol Use Case Diagram.

GAMBAR	NAMA	KETERANGAN
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .

	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

Tabel 2.1 (lanjutan).

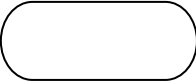


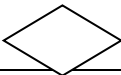

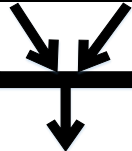
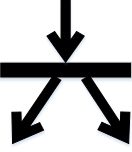
2.11.3 Activity Diagram

Munawar (2018 : 127) menguraikan bahwa *Activity Diagram* adalah bagian penting dari UML yang menggambarkan aspek dinamis dari Sistem. Logika Prosedural,

proses bisnis dan aliran kerja suatu bisnis bisa dengan mudah di deskripsikan dalam *Activity Diagram*.

Activity Diagram mempunyai peran seperti halnya *Flowchart*, akan tetapi perbedaannya dengan *Flowchart* adalah `; *Activity Diagram* bisa mendukung perilaku paralel sedangkan *Flowchart* tidak bisa. Simbol-simbol yang digunakan dalam *Activity Diagram* dapat dilihat pada tabel 2.2 dibawah ini :

Tabel 2.2 Simbol *Activity Diagram*.

Simbol	Keterangan
	<i>Activity</i> : Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Initial Node</i> : Bagaimana objek dibentuk atau diawali
	<i>Activity Final Node</i> : Bagaimana objek dibentuk dan diakhiri.
	<i>Decision</i> : Asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu.
	<i>Swimlane</i> : Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi.
	<i>Join</i> : Digunakan untuk menunjukkan kegiatan yang digabungkan.
	<i>Fork</i> : Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel

2.12 Penelitian Terkait

Dalam penyusunan skripsi ini, penulis terinspirasi dan mereferensi dari penelitian-penelitian sebelumnya yang berkaitan dengan skripsi ini. Daftar penelitian terkait sebagai berikut :

Tabel 2.3 Penelitian Terkait

No.	Nama	Judul	Tahun	Uraian
1.	Nurhuda Budi Pamungkas ¹ , Dedi Darwis ² , Ditha Nurjayanti ³ , Agung Tri Prastowo ⁴	Perbandingan Algoritma Pixel Value Differencing Dan Modulus Function Pada Steganografi Untuk Mengukur Kualitas Citra Dan Kapasitas Penyimpanan	2020	Masalah keamanan dalam transfer data menjadi perhatian yang paling utama karena pentingnya nilai informasi pada setiap aspek dapat memungkinkan adanya usaha pemindah alihan atau pencurian informasi ataupun data oleh pihak yang tidak berwenang[1,2]. Media penyimpanan dan penyebaran data atau informasi yang digunakan menjadi salah satu alasan rentannya data atau informasi mudah diambil oleh pihak yang tidak bertanggung jawab. H
2.	Ira Herawati Sada, Iwan Iwut Tritoasmoro, Gelar Budiman	Pemodelan Wajah 3d Melalui Pendeteksian Fitur Wajah 2d Menggunakan Teknik Morphing	2018	Pengenalan Aksara Lontara Tulis Tangan Menggunakan Metode Convolutional Neural Networks Berbasis Android

		3d Models Of Facial Features With The Detection Of 2d Faces Using Morphing Techniques		
3.	Abrar Hiswara ¹ , Aida Fitriyani ² , Reza Adi Nugraha	Sistem Steganography Dengan Metode Least Signifcant Bit (Lsb) & Metode Caesar Cipher Berbasis Android	2020	Perkembangan teknologi Internet memunculkan kejahatan yang disebut dengan cybercrime atau kejahatan melalui jaringan Internet. Munculnya beberapa kasus cybercrime di Indonesia, merupakan fenomena, seperti pencurian kartu kredit, hacking terhadap berbagai situs, penyadapan transmisi data orang lain, (misalnya email), dan manipulasi data dengan cara menyiapkan perintah yang tidak dikehendaki ke dalam programer komputer.
4	Taufiqurrahman, Apriansyah Putra, Ali Bardadi	Sistem Informasi Kemacetan Lalu Lintas Berbasis Teknologi Mobile Dengan Teknik	2017	Kemacetan merupakan masalah serius yang sering dihadapi oleh kota-kota besar di seluruh dunia. Secara umum, kemacetan

		Geofencing Menggunakan Metode Crowdsourcing		merupakan suatu kondisi menumpuknya volume kendaraan di jalan dalam suatu periode tertentu. Terdapat 26 faktor penyebab kemacetan[
5	Abdul Jahir1 , Kuart Indartono2	DETEKSI KHIMAR WANITA PADA CITRA WAJAH MENGUNAKAN METODE GAUSSIAN MIXTURE MODEL	2019	Dalam penelitian ini, untuk mendeteksi aurat pada sebuah gambar/foto digunakan tahapan pra pengolahan deteksi wajah dan kulit. Algoritme deteksi wajah yang dikembangkan oleh Viola-Jones merupakan metode yang biasa digunakan sebagai pengenalan wajah dan memiliki akurasi dan kecepatan yang sangat baik