

BAB 2

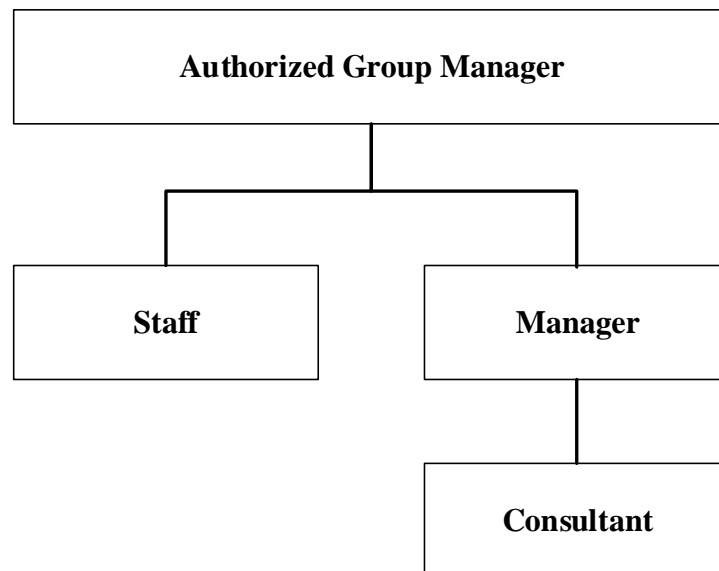
TINJAUAN PUSTAKA

2.1 Profil Tupperware AGM Cianjur

Authorized Group Manager (AGM) adalah Group Manager yang dipilih dan diangkat oleh Tupperware Indonesia sebagai perwakilan dalam mengembangkan promosi, pemasaran dan penjualan produk Tupperware di suatu wilayah kerja yang telah disetujui oleh Tupperware Indonesia serta untuk memudahkan akses dalam mengikuti kegiatan-kegiatan di wilayah tersebut. Tupperware AGM Cianjur beralamatkan di Jl. Pangeran Hidayatullah No. 10, Kel. Sawah Gede, Kec. Cianjur, Kab.Cianjur, Jawa Barat 43212.

2.2 Struktur Organisasi

Berikut merupakan struktur organisasi dan deskripsi jabatan di Tupperware AGM Cianjur.



Gambar 2.1 Struktur Organisasi

Adapun uraian tugas pada struktur organisasi di Tupperware AGM Cianjur sebagai berikut.

1. *Auhorized General Manager*

Mememlihara, membina, mengembangkan serta mengadakan pelatihan bagi manager baik secara sendiri maupun bersama-sama dengan Tupperware Indonesia.

2. *Manager*

Membina dan mengembangkan consultan dalam unitnya secara terus menerus dan berkelanjutan melalui monitoring atas rekapitulasi penjualan mingguan.

3. *Consultant*

Melakukan aktivitas Success Forumula sebagai *consultant* yang ditetapkan oleh Tupperware Indonesia. Hadir pada saat *assembly*, melakukan Tupperware *Party* , hingga menghadiri pertemuan unit dan *training*.

4. *Staff*

Melayani anggota Tupperware Cianjur dalam proses pesan hingga distribusi produk Tupperware.

2.3 Landasan Teori

Landasan teori merupakan penjelasan hal-hal apa saja yang terkait dalam penelitian Pembangunan Aplikasi *Augmented Reality* Sebagai Media Informasi Produk Berbasis Android Di Tupperware Cianjur. Bagian teori dalam penelitian ini meliputi penjelasan mengenai Aplikasi, Android, UML dan perangkat lunak pendukung seperti Unity, Belnder, Umlet dan Vuforia *Augmented Reality*.

2.3.1 Aplikasi

Aplikasi merupakan sebuah penerapan, penambahan atau pemakaian [11]. Menurut pengertian tadi, dapat disimpulkan bahwa aplikasi merupakan sebuah perangkat lunak yang berfungsi dalam melakukan berbagai macam pekerjaan maupun tugas tertentu, semacam penerapan, penggunaan dan penambahan pada data.

Aplikasi merupakan penggunaan dalam sebuah perangkat komputer, instruksi atau penyaraan yang telah disusun sehingga computer dapat melakukan dari proses masukan hingga keluaran [12].

Aplikasi merupakan sebuah unit perangkat lunak yang telah dibuat dengan tujuan untuk melayani kebutuhan dalam banyak aktivitas. Contohnya termasuk

perangkat lunak yang digunakan oleh sebuah perusahaan, perangkat lunak untuk akuntansi, perkantoran hingga grafis perangkat lunak serta pemutar media [13].

Berdasarkan beberapa pernyataan diatas, dapat disimpulkan bahwa aplikasi merupakan sebuah perangkat lunak dengan kegunaan untuk melakukan berbagai macam tugas semacam penerapan, pemakaian dan penambahan data. Program aplikasi merupakan sebuah program yang siap dipakai oleh pengguna. Aplikasi akan menggunakan sebuah sistem operasi computer dan beberapa aplikasi pendukung lainnya.

2.3.2 Android

Android merupakan sistem operasi untuk sebuah perangkat *mobile* dengan basis linux hingga mencakup pada sistem operasi, *middle ware* dan aplikasi. Android sendiri menyediakan *platform* yang bersifat terbuka untuk memudahkan para pengembang dalam menciptakan aplikasi sendiri. Pada awalnya, Android Inc. dibeli oleh Google Inc. , Android pada saat itu merupakan pendatang baru dalam proses pembuatan perangkat lunak yang diperuntukan untuk ponsel atau *smartphone*. Kemudian dibentuk *Open Handset Alliance* dalam mengembangkan Android, dengan pembiayaan bersama dari 34 perusahaan perangkat keras, lunak serta telekomunikasi yang termasuk Goole, Intel, HTC, Qualcom serta NVIDIA.

2.3.3 Pengertian Pemrograman Berbasis Objek

OOP (*Object Oriented Programming*) merupakan sebuah pendekatan dalam pengembangan sebuah perangkat lunak, dimana struktur dari perangkat lunak tersebut berdasar pada interaksi objek dalam proses penyelesaian sebuah proses maupun tugas [14].

Pemrograman berorientasi objek merupakan sebuah model pemrograman yang berorientasi pada objek. Seluruh data dan fungsi yang terdapat pada model ini disimpan dalam kelas ataupun objek. Jika dibanding dengan logika dari pemrograman terstruktur, dimana setiap objek dapat memproses data, menerima dan mengirim pesan pada objek lainnya. Dalam Pemrograman berbasis objek terdapat 2 bagian penting yaitu *event* dan *properties*, dimana *properties* memiliki nama serta nilai, sedangkan *event* hanya memiliki nama yang biasanya berhubungan antara *method*, *function* dan *action*.

Di dalam proses pemecahan masalah, pemrograman berbasis objek tidak melihat solusi dari pemecahan suatu masalah secara terstruktur, melainkan bagaimana menyelesaikan sebuah masalah dengan cara menentukan objek apa saja yang dapat dipecahkan dari masalah yang dihadapi.

Dengan model data yang berorientasi objek, dikatakan bahwa model ini dapat memberikan kemudahan yang lebih, seperti pada kemudahan dalam mengubah program, dan dapat digunakan secara luas dalam pembangunan perangkat lunak dalam skala yang besar. Pemrograman berbasis objek dapat dengan mudah dipelajari bagi para pemula jika dibandingkan dengan sebelumnya dan juga mudah dalam dikembangkan.

2.3.4 UML (*Unified Modeling Language*)

Unified Model Language (UML) merupakan sebuah Bahasa pemodelan untuk sistem ataupun perangkat lunak dengan model yang berorientasi objek. Gambaran konsep dasar dari sebuah UML terdiri dari *structural classification*, *dynamic behavior*, dan *model management* yang dapat kita pahami konsep utaa sebagai *term* yang akan muncul pada saat membuat diagram dan *view* adalah kategori dari diagram tersebut [15].

UML mendefinisikan beberapa diagram antara lain: *Class Diagram*, *Package Diagram*, *Use Case Diagram*, *Sequence Diagram*, *Communication Diagram*, *Statechart Diagram*, *Activity Diagram*, *Component Diagram* dan *Deployment Diagram* [16].

Dalam pengembangan aplikasi ini, hanya menggunakan 4 diagram UML yaitu Use Case Diagram, Activity Diagram, dan Sequence Diagram karena dapat membantu penelitian selanjutnya dalam mengembangkan sistem ini Perangkat Pemodelan Berbasis Objek.

2.3.4.1 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, membuat sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor

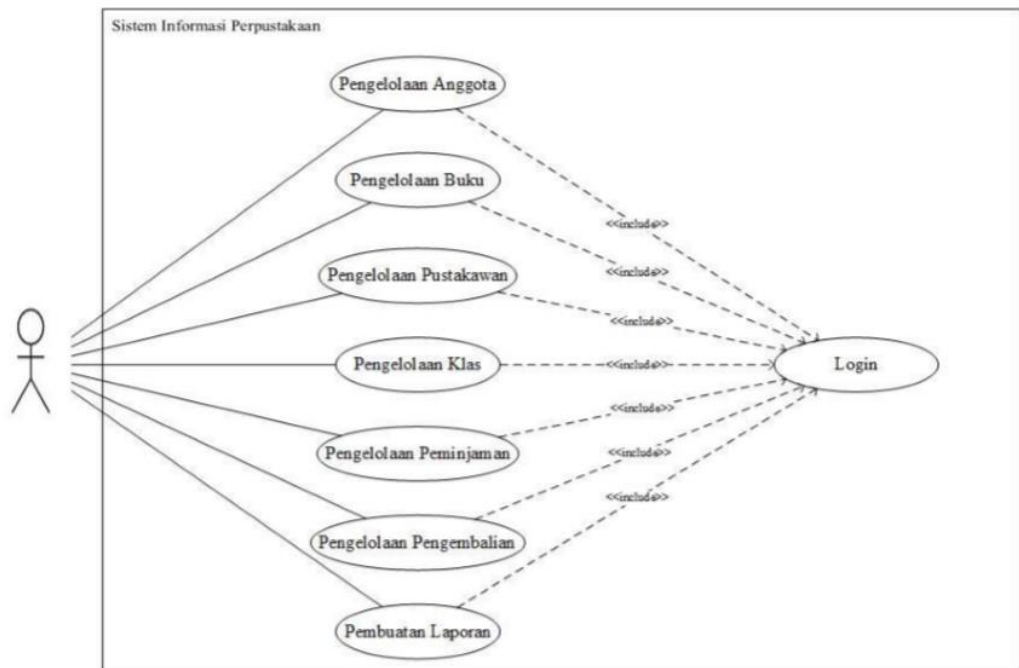
adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Adapun komponen-komponen dalam use case diagram diantaranya:

1. Aktor

Aktor merupakan suatu entitas yang berkaitan dengan sistem tapi bukan dari bagian dalam sistem itu sendiri. Aktor berbeda diluar sistem namun berkaitan erat dengan fungsionalitas didalamnya. Aktor dapat memiliki hubungan secara langsung terhadap fungsi utama baik terhadap salah satu atau semua fungsionalitas utama. Aktor juga dapat dibagi terhadap berbagai jenis atau tingkatan dengan cara digeneralisasi atau dispesifikasi tergantung kebutuhan sistemnya. Aktor biasanya dapat berupa pengguna atau database yang secara pandang berada dalam suatu ruang lingkup sistem tersebut.

2. *Use Case*

Use case merupakan gambaran umum dari fungsi proses utama yang menggambarkan tentang salah satu perilaku sistem. Perilaku sistem ini terdefinisi dari proses bisnis sistem yang akan dimodelkan. Tidak semua proses bisnis digambarkan secara fungsional pada use case, tetapi yang digambarkan hanya fungsionalitas utama yang berkaitan dengan sistem. Use case menitikberatkan bagaimana suatu sistem dapat berinteraksi baik antar sistem maupun di luar sistem.

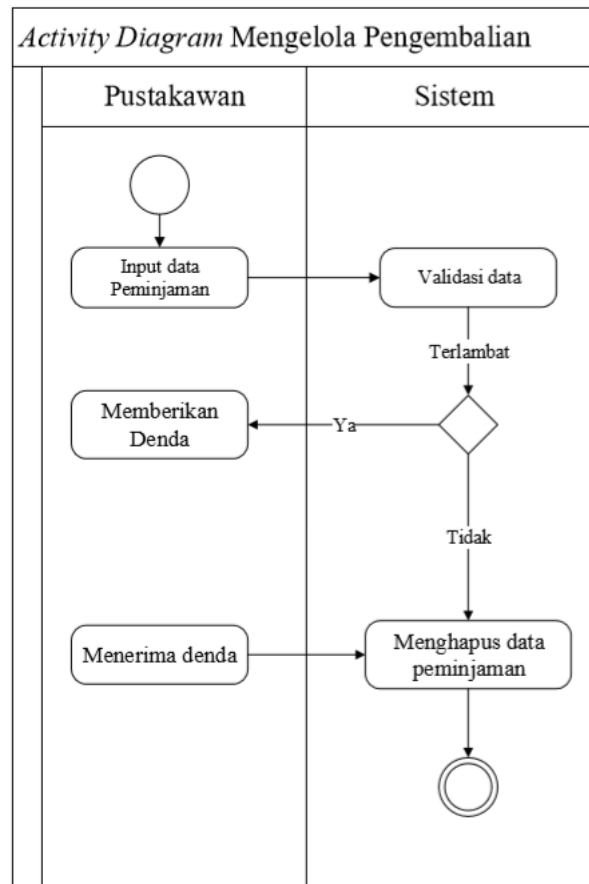


Gambar 2.2 Use Case Diagram

2.3.4.2 Activity Diagram

Activity diagram menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, keputusan yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

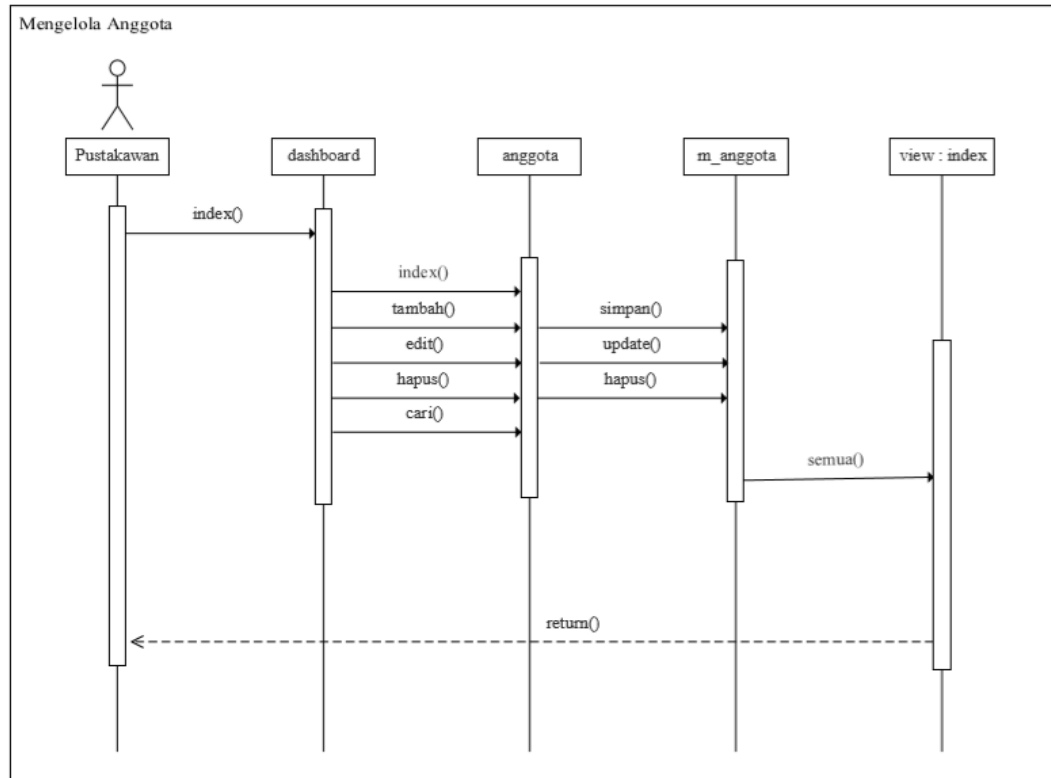
Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah aksi dan sebagian besar transisi di-*trigger* oleh selesainya state sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour* internal sebuah sistem (dan interaksi antar sub sistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.



Gambar 2.3 Activity Diagram

2.3.4.3 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, tampilan, dan sebagainya) berupa message yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horisontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.



Gambar 2.4 Sequence Diagram

2.3.5 Augmented Reality

Augmented Reality (AR) merupakan istilah untuk lingkungan yang menggabungkan dunia virtual yang dibuat oleh komputer dengan dunia nyata dimana batas antara keduanya akan menjadi tidak terasa. Menurut Ronald Azuma, *Augmented Reality* didefinisikan sebagai *system* yang memiliki karakteristik sebagai berikut:

1. Menggabungkan lingkungan nyata dengan virtual
2. Berjalan secara interaktif dengan *real time*
3. Integrasi dalam wujud tiga dimensi (3D)

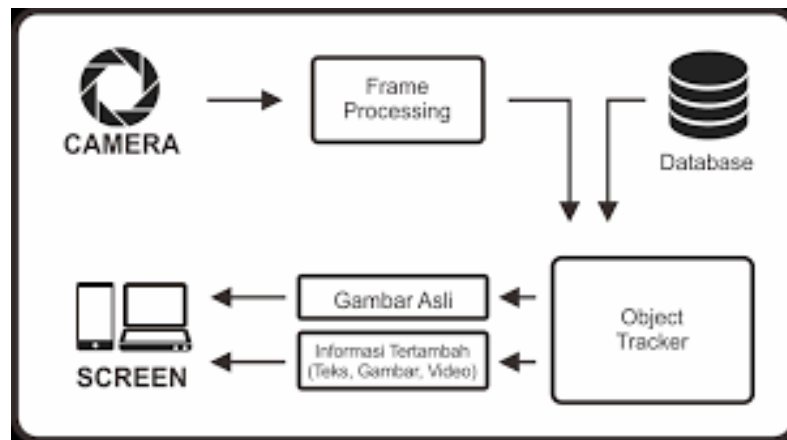
Sederhananya, *Augmented Reality* dapat diartikan sebagai lingkungan nyata dengan tambahan objek virtual. *Augmented Reality* merupakan sebuah ragam dari *Virtual Environments* (VE) atau dikenal dengan istilah *Virtual Reality* (VR). Teknologi VE akan menggabungkan pengguna ke dalam sebuah lingkungan virtual secara keseluruhan. Ketika pengguna tergabung dan berada di lingkungan virtual, pengguna tidak dapat melihat lingkungan yang nyata di sekitarnya. Namun

sebaliknya dengan *Augmented Reality* yang memungkinkan pengguna untuk melihat lingkungan nyata yang ada di sekitarnya, dengan tambahan objek virtual yang tergabung dengan lingkungan nyata. Tak seperti *Virtual Reality* yang total menggantikan lingkungan nyata, *Augmented Reality* hanya sekedar menambahkan atau melengkapi lingkungan di dunia nyata tanpa menggantikannya.

Tujuan yang utama dari teknologi *Augmented Reality* ini adalah untuk menciptakan sebuah lingkungan yang baru dengan menggabungkan antara dunia nyata dengan objek virtual, sehingga pengguna dapat merasakan lingkungan yang diciptakan adalah nyata. Dengan kata lain, bertujuan agar pengguna tidak merasakan adanya perbedaan antara *Augmented Reality* dengan apa yang dirasakan pada lingkungan yang nyata. Dengan adanya bantuan dari teknologi *Augmented Reality* (seperti pengenalan objek) lingkungan nyata di sekitar pengguna akan dapat berinteraksi dalam bentuk digital (virtual). Informasi mengenai objek dan lingkungan di sekitar dapat ditambahkan pada sistem *Augmented Reality* dan ditampilkan pada *layer* dunia nyata secara *real time* yang menjadikan informasi tersebut seolah terasa nyata. Informasi yang ditampilkan oleh objek virtual dapat membantu pengguna dalam melakukan kegiatan di dunia nyata. *Augmented Reality* banyak digunakan pada banyak bidang seperti pada bidang militer, kesehatan, industri manufaktur dan juga telah diaplikasikan dalam perangkat-perangkat yang orang banyak gunakan seperti pada telpon genggam. Tujuan dari penggunaan teknologi informasi adalah menjadikan proses bisnis dapat berjalan secara optimal dan menghasilkan informasi yang berkualitas [17].

2.3.5.1 Alur Kerja *Augmented Reality*

Alur kerja *Augmented Reality* secara umum dimulai dari pengambilan gambar marker dengan kamera atau webcam. *Marker* tersebut dikenali berdasarkan feature yang dimiliki, kemudian masuk ke dalam object tracker (objek yang dilacak) yang disediakan oleh *Software Development Kit* (SDK). Di sisi lain, marker tersebut telah didaftarkan dan disimpan kedalam *database*. *Object tracker* selanjutnya akan melacak dan mencocokkan *marker* tersebut agar dapat menampilkan informasi yang sesuai. Alur kerja *Augmented Reality* dapat dilihat pada gambar berikut.



Gambar 2.5 Alur Kerja *Augmented Reality*

- 1) *Camera*
Digunakan untuk menscan *marker/markerless*.
- 2) *Frame Processing*
Proses perhitungan pixel yang digunakan untuk pendeteksian gambar atau objek.
- 3) *Database*
Digunakan untuk menyimpan data *marker*.
- 4) *Object Tracker*
Objek yang dilacak yang akan di tampilkan informasi nya jika sesuai dengan data yang ada didalam database.
- 5) *Informasi*
Menampilkan informasi yang sesuai dengan object tracker yang sudah di inisiliasi di *database*.
- 6) *Screen*
Untuk menampilkan keluaran dari aplikasi yang berupa Teks, Gambar, Video, dan Objek 3D.

2.3.5.2 *Markerless User Defined Target*

Markerless merupakan sebuah metode dari pelacakan *augmented reality* dengan memanfaatkan objek di lingkungan nyata atau tidak menggunakan *marker* khusus sebagai target. Metode *Augmented Reality* tanpa penanda ini tidak

menggunakan pengenalan penanda (*facial marker*) melainkan menggunakan teknik pelacakan secara alami (*natural feature*) [18].

User Defined Target adalah Target Gambar yang dibuat oleh pengguna pada saat *runtime* dari fram kamera. Dengan kata lain, pada metode *userd defined target, marker* akan terbentuk pada saat pengguna melakukan pemindaian target [19].

Atribut rekomendasi untuk pengguna *User Defined Target*, aplikasi bertanggung jawab untuk tugas-tugas berikut.

1. Memulai proses pemindaian target
2. Memicu proses untuk membangun target baru
3. Menambahkan target yang baru diperoleh ke dalam *database*

Target ideal yang ditetapkan pengguna mencakup beberapa atribut sebagai berikut.

Tabel 2.1 Tabel Atribut Ideal Untuk *User Defined Target*

Atribut	Contoh
Kaya Akan Detail	Pemandangan Jalanan, Sekelompok Orang, Kolase dan Campuran Item
Kontras yang Baik	Termasuk daerah terang dan gelap, dan cukup terang
Tidak Ada Pola Berulang	Lapangan berumput, fasad rumah modern dengan jendela identic, dan kotak-kotak
Kemudahan Dalam Ketersediaan	Kartu Nama, Majalah dan memo.

2.3.6 Tools Yang Digunakan

Tools atau alat-alat yang diperlukan dalam merancang dan membuat aplikasi perangkat lunak yaitu sekumpulan alat yang wajib dan hanya dapat dibuat oleh fungsi masing-masing *tools*. Terdapat beberapa tools yang digunakan dalam pembuatan aplikasi perangkat lunak yaitu.

2.3.6.1 Unity

Unity adalah game developing software, dengan built-in IDE yang dikembangkan oleh Unity Technologies. Hal ini digunakan untuk mengembangkan video game untuk plugin web, platform desktop, konsol dan perangkat mobile, dan digunakan oleh lebih dari satu juta pengembang. Unity tumbuh dari OS X didukung

permainan alat pengembangan pada tahun 2005 untuk *game developing software game multi platform*. Mesin grafis menggunakan Direct3D (Windows , Xbox 360) , OpenGL (Mac , Windows , Linux , PS3) , OpenGL ES (Android , iOS) , dan kepemilikan API (Wii) . Ada dukungan untuk pemetaan mesh , pemetaan refleksi , pemetaan paralaks, bayangan dinamis menggunakan peta bayangan , merender ke tekstur dan efek post-processing layar penuh. Unity mendukung aset seni dan format file dari 3ds Max , Maya , Softimage , Blender , modo , ZBrush , Cinema 4D , Cheetah3D , Adobe Photoshop , Adobe Fireworks dan Substansi Allegorithmic . Aset ini dapat ditambahkan ke proyek game, dan dikelola melalui antarmuka pengguna grafis Unity. Bahasa ShaderLab digunakan untuk shader , mendukung kedua deklaratif " pemrograman " dari program tetap fungsi pipa dan shader ditulis dalam GLSL atau Cg . Shader A dapat mencakup beberapa varian dan spesifikasi fallback deklaratif, memungkinkan Unity untuk mendeteksi varian yang terbaik untuk kartu video saat ini, dan jika tidak ada yang kompatibel , jatuh kembali ke shader alternatif yang mungkin mengorbankan fitur untuk kinerja.

Unity selain gratis juga memungkinkan individual maupun team untuk membuat game secara mudah dan cepat. Unity telah diatur untuk membuat game berjenis *First Person Shooting (FPS)*, namun dapat juga membuat game berjenis *Role Playing Game (RPG)*, dan *real time strategy (RTS)* [20].

2.3.6.2 Blender

Blender merupakan sebuah perangkat lunak multimedia dengan ukuran instalasi relative kecil dan dapat di implementasikan pada semua platform komputer. Meskipun sering kali distribusinya tidak diiringi dengan dokumentasi yang jelas, perangkat lunak ini memiliki beberapa fitur yang menyerupai dengan perangkat lunak *modelling* termuthakhir lainnya. Beberapa kemampuan dari Blender adalah sebagai berikut:

1. Mendukung banyak macam dari bentuk geometri primitive seperti polygon yang tak beraturan, *metaballs* dan lainnya
2. Didukung dengan *keyframed animation tools* dengan berupa animasi nonlinier, pemberian bobot pada vertex, *particle based hair* dan partikel sistem dengan *collution detection*.

3. Didukung oleh Bahasa pemrograman Python dalam menciptakan *tools* yang baru serta *export* dan *import file* dari format berbeda seperti .FBX, .DFX, .OBJ.
4. Memiliki kemampuan dalam penyuntingan video ataupun audio.
5. Memiliki kemampuan untuk *editing* video atau audio yang nonlinier dan masih banyak lagi feature yang lain yang merupakan teknologi *high-end* [21].

2.3.6.3 Android SDK (Software Development Kit)

Android SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang di-release oleh Google. Saat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Sebagai platform aplikasi netral, Android memberi kesempatan untuk membuat aplikasi yang dibutuhkan yang bukan merupakan aplikasi bawaan *handphone/smartphone*.

2.3.6.4 Vuforia

Vuforia merupakan *software untuk augmented reality* yang dikembangkan oleh Qualcomm, yang menggunakan sumber yang konsisten mengenai *computer vision* yang fokus pada *image recognition*. Vuforia mempunyai banyak fitur-fitur dan kemampuan, yang dapat membantu pengembang untuk mewujudkan pemikiran mereka tanpa adanya batasan secara teknis. Dengan *support* untuk iOS, Android, dan Unity3D, platform Vuforia mendukung para pengembang untuk membuat aplikasi yang dapat digunakan di hampir seluruh jenis smartphone dan tablet. Pengembang juga diberikan kebebasan untuk mendesain dan membuat aplikasi yang mempunyai kemampuan antara lain:

1. Teknologi *computer vision* tingkat tinggi yang mengijinkan *developer* untuk membuat efek khusus pada *device*.
2. Terus-menerus mengenali *multiple image*.
3. *Tracking* dan *detection* tingkat lanjut.
4. Solusi pengaturan *database* gambar yang fleksibel.

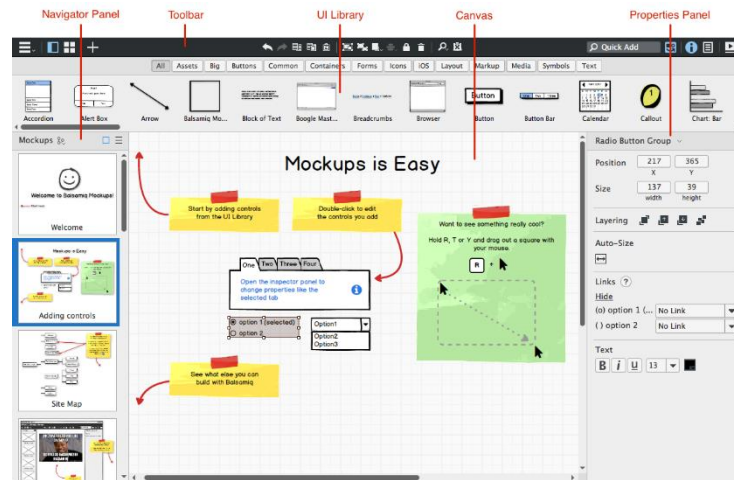
Target pada vuforia merupakan obyek pada dunia nyata yang dapat dideteksi oleh kamera, untuk menampilkan obyek virtual. Beberapa jenis target pada vuforia adalah :

1. *Image targets*, contoh : foto, papan permainan, halaman majalah, sampul buku, kemasan produk, poster, kartu ucapan. Jenis target ini menampilkan gambar sederhana dari *Augmented*.
2. *Frame markers*, tipe frame gambar 2D dengan pattern khusus yang dapat digunakan sebagai potongan permainan di permainan pada papan.
3. *Multi-target*, contohnya kemasan produk atau produk yang berbentuk kotak ataupun persegi. Jenis ini dapat menampilkan gambar sederhana *Augmented* 3D.
4. *Virtual buttons*, yang dapat membuat tombol sebagai daerah kotak sebagai sasaran gambar.

Pada Vuforia, ada 2 jenis *workflow* dengan dasar *database* yang dapat dipilih oleh developer, yaitu *Cloud Database* dan *Device Database*.

2.3.6.5 Balsamic Mockup

Balsamiq Mockup merupakan alat desain untuk membuat antarmuka pengguna secara digital dari ide produk yang telah direncanakan sebelumnya. Setiap file dari software ini dapat mewakili satu proyek dan dapat berisikan banyak tampilan antarmuka (mockup) dan gambar, yang disimpan bersama dalam satu file dengan jenis format keluaran yaitu .bmpr. Antarmuka pada software Balsamiq Mockups terdiri dari lima area utama yaitu tools, UI Library, kanvas, panel navigator, dan panel properti. Secara Umum Tampilan *Software* Balsamiq Mockup dapat dilihat pada gambar 2.6 [22].



Gambar 2.6 Tampilan Antarmuka *Software Balsamiq Mockup*

Adapun penjelasan dari Masing-masing area dijelaskan di bawah ini.

1. *Toolbar*

Pada bagian *Toolbar* berisi serangkaian ikon untuk melakukan tindakan pada area antarmuka pengguna yang terdiri dari tombol navigasi yang terdiri dari assets, symbols, dan trash, Tampilkan / sembunyikan panel kiri, Lihat mockup sebagai kotak thumbnail, Buat *mockup* baru, *Mockups* action seperti mengganti nama dll, menuju ke *mockup* selanjutnya ataupun sebelumnya. Bagian pertama pada toolbar dapat dilihat pada gambar 2.7 [22].



Gambar 2.7 Menu *Toolbar1*

Pada bagian kedua yaitu di tengah bilah adalah untuk fungsi kavas yaitu tindakan yang mungkin digunakan dari editor teks atau alat menggambar lain seperti salin, tempel, kelompokkan, perbesar. Untuk ikon terakhir adalah untuk mengaktifkan markap. Bagian ini diperjelas pada gambar 2.8 [22].



Gambar 2.8 Menu *Toolbar2*

Bagian terakhir dari menu toolbar di ujung kanan yaitu terdiri dari *Quick add*, ikon beralih untuk *UI Library*, *Property Inspector* dan *Project Info Panel*. Bagian ini dapat dilihat pada gambar 2.9 [22].



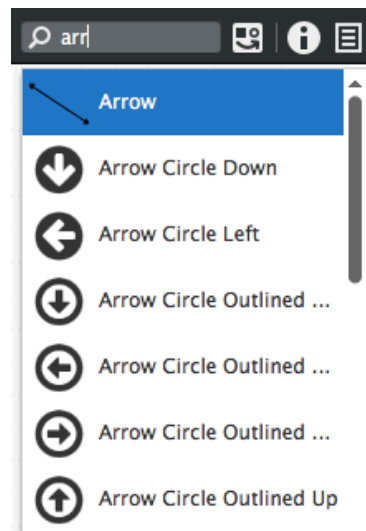
Gambar 2.9 Menu Toolbar3

2. UI Library

UI Library merupakan kumpulan tools untuk membuat perancangan yang sudah disediakan langsung oleh *Balsamiq Mockups*. Bagian *UI Library* dapat dilihat pada gambar 2.10 dan gambar 2.11 [22].



Gambar 2.10 UI Library1

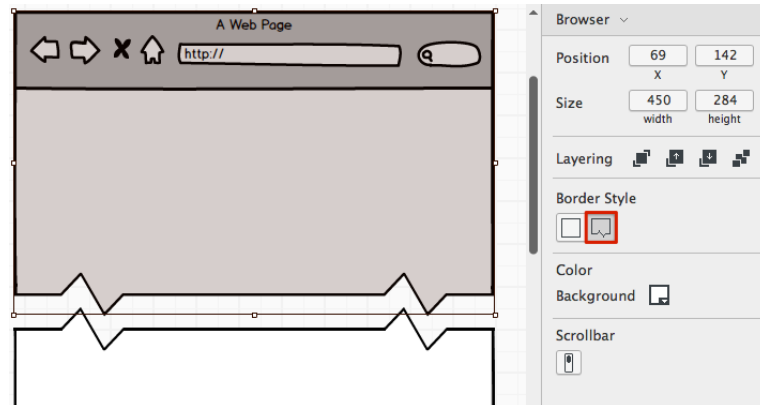


Gambar 2.11 UI Library2

3. Kanvas

Pada bagian ini merupakan area kerja utama sebagai tempat mockup UI anda hidup. Setelah menambahkan kontrol UI, pengguna dapat memindahkan,

mengubah ukuran dan menyesuaikan dengan kebutuhan. Area kerja utama atau kanvas dapat dilihat pada gambar 2.12 [22].



Gambar 2.12 Area Kerja

4. Panel Navigator

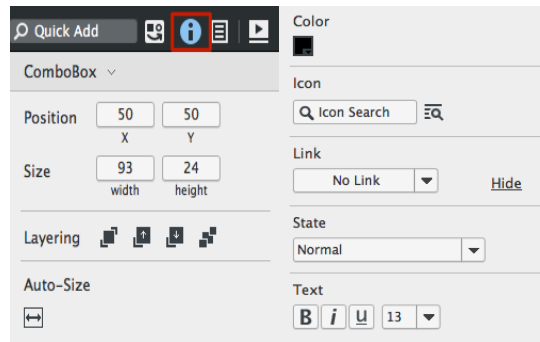
Panel navigator di sebelah kiri menunjukkan daftar *mockup*, aset, dan simbol dalam proyek pengguna. Pengguna dapat memilih beberapa objek di panel navigator. Berikut merupakan tampilan panel navigator pada gambar 2.13 [22].



Gambar 2.13 Panel Navigator

5. Panel Properti

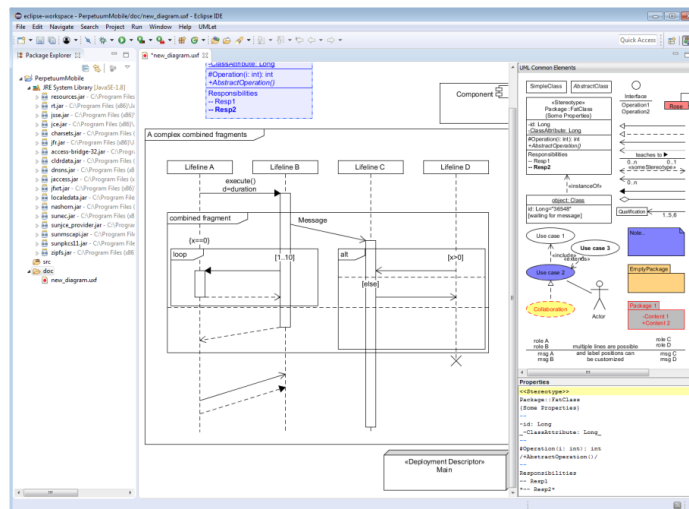
Panel dengan posisi kanan pada aplikasi menampilkan properti untuk objek atau kontrol yang dipilih. Menu ini bergantung pada ikon mana yang dipilih. Panel ini dapat dimunculkan ataupun disembunyikan. Panel Properti dapat dilihat pada gambar 2.14 [22].



Gambar 2.14 Panel Properti

2.3.6.6 Umlet

UMlet adalah *software open-source* gratis dengan antarmuka pengguna yang sederhana. Software ini dapat digunakan untuk menggambar diagram UML dengan cepat, menyusun diagram dan dapat mengekspor diagram ke *eps*, *pdf*, *jpg*, *svg*, dan *clipboard*. Fitur lain yang dimiliki oleh software ini yaitu dapat berbagi diagram menggunakan *Eclipse*. *UMlet* menjalankan secara *stand-alone* atau sebagai *plugin* *Eclipse* pada *Windows*, *OSX* dan *linux*. Tampilan *UMlet* dapat dilihat pada gambar 2.15 [23].



Gambar 2.15 Antarmuka Software Umlet

2.3.7 Pengujian Black Box

Pengujian *black box* berfokus pada keperluan penelusuran kesalahan fungsional dari *software*. Proses pengujian *black box* ini berfokus dalam menentukan kesalahan dalam beberapa kategori, diantaranya:

1. Fungsi-fungsi yang salah atau hilang
2. Kesalahan Antarmuka
3. Kesalahan dalam struktur data atau akses *database* eksternal
4. Kelasahan performa
5. Kesalahan inisialisasi dan terminasi.

Teknik pengujian *black box* terdiri dari 10 jeni, diantaranya *Equivalence Partitioning*, *Boundary Value Analyst/Limit Testing*, *Comparison Testing*, *Sample Testing*, *Robustness Testing*, *Cause-Effect Relationship Testing*. Salah satunya yang akan dibahas adalah *Equivalence Partitioning*.

Equivalence Partitioning merupakan sebuah metode uji coba *black box* yang membagi domain *input* program menjadi beberapa kelas data dari kasus ujicoba yang dihasilkan. Kasus uji penanganan *single* yang ideal dalam menemukan sejumlah kesalahan semisal kesalahan dalam pemrosesan dari seluruh data karakter yang merupakan syarat lain dari suatu kasus yang dieksekusi sebelum kesalahan umum diamati.

2.3.8 Pengujian Kuisisioner

Kuisisioner merupakan sebuah daftar pernyataan yang harus diisi oleh orang yang akan dievaluasi (responden). Dalam penelitian ini akan digunakan metode skala Likert dalam menghitung nilai kuisisioner penelitian. Dalam skala likert, responden diminta untuk membaca dengan seksama setiap pertanyaan yang telah disajikan, kemudian responden diminta untuk menilai pernyataan-pernyataan tersebut.

Derajat penilaian responden terhadap suatu pernyataan terbagi dalam 5 kategori yang disusun secara bertingkat, mulai dari Sangat Tidak Setuju (STS), Tidak Setuju (TS), Ragu-Ragu (R), Setuju (S) dan Sangat Setuju (SS) ataupun bisa sebaliknya. Pernyataan pada tiap kuisisioner dibuat berdasarkan aspek-aspek yang diteliti. Bobot pemberian skor yang digunakan dapat dilihat pada Tabel 2.2.

Tabel 2.2 Panduan Pemberia Skor

Jenis Pertanyaan	Bobot Pendapat				
	SS	S	R	TS	STS
Positif	5	4	3	2	1
Negatif	1	2	3	4	5

Skor yang telah dihitung dari setiap pernyataan kemudian akan dikalikan dengan masing-masing bobot tersebut sesuai dengan skenario kuisisioner yang telah dibuat. Kemudian seluruh bobot ditotalkan dan dibagi dengan total responden yang kelak akan menjadi nilai rata-rata. Nilai rata-rata inilah yang akan diambil sebagai acuan sikap dimana jika nilai rata-rata kurang dari 3, maka dapat diartikan responden bersikap negatif dan jika nilai rata-rata lebih dari sama dengan 3, maka dapat diartikan bahwa responden bersikap positif terhadap tujuan yang ingin dicapai.