

**APLIKASI MOBILE MUSEUM SATWA VIRTUAL BERBASIS
ANDROID**

SKRIPSI

INSTITUT TEKNOLOGI NASIONAL



Disusun Oleh:

GUNARDI AFRIDITA

NIM. 08.12.517

MALANG

**MILIK
PERPUSTAKAAN
ITN MALANG**

**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL
2015**

ORGANISASI LAINNYA DAN PERUSAHAAN LAINNYA

DIKEMUKAKAN

DIKEMUKAKAN

DIKEMUKAKAN
DIKEMUKAKAN
DIKEMUKAKAN

DIKEMUKAKAN
DIKEMUKAKAN
DIKEMUKAKAN
DIKEMUKAKAN
DIKEMUKAKAN

LEMBAR PERSETUJUAN

APLIKASI MOBILE MUSEUM SATWA VIRTUAL BERBASIS ANDROID

*Disusun dan Diajukan Untuk Melengkapi dan Memenuhi Syarat
Guna Mencapai Gelar Sarjana Teknik*

Disusun Oleh :

GUANRDI AFRIDITA

NIM : 08.12.517

Mengetahui,

Ketua Jurusan Teknik Elektro S-1



M. Ibrahim Ashari, ST, MT

NIP.P. 1030100358

Diperiksa dan Disetujui

Pembimbing I

Pembimbing II

Dr. Eng. Aryanto Soetedjo, ST, MT
NIP. Y.1030800417

Ir. Ni Putu Agustini, MT
NIP. Y.1030100371

**PROGRAM STUDI TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2015**

SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini :

Nama : GUNARDI AFRIDITA
NIM : 08.12.517
Program Studi : TEKNIK ELEKTRO S-1
Konsentrasi : TEKNIK KOMPUTER

Dengan ini menyatakan bahwa Skripsi yang saya buat adalah hasil karya sendiri, tidak merupakan plagiasi dari karya orang lain. Dalam Skripsi ini tidak memuat karya orang lain, kecuali dicantumkan sumbernya sesuai dengan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat, dan apabila di kemudian hari ada pelanggaran atas surat pernyataan ini, saya bersedia menerima sanksinya.

Malang, September 2015

Yang membuat Pernyataan,



Gunardi Afridita
NIM : 08.12.517

APLIKASI MOBILE MUSEUM SATWA VIRTUAL BERBASIS ANDROID

Gunardi Afridita
08.12.517

Jurusan Teknik Elektro S-1, Konsentrasi Teknik Komputer
Fakultas Teknologi Industri, Institut Teknologi Nasional Malang
Jl. Raya Karanglo Km 2 Malang
Email : otiddito@gmail.com

**Dosen Pembimbing : Dr. Eng. Aryuanto Soetedjo, ST, MT dan
Ir. Ni Putu Agustini, MT**

Abstrak

Saat ini banyak objek wisata yang bermunculan di berbagai tempat, namun kebanyakan hanya berupa tempat rekreasi hiburan keluarga yang kurang memberikan nilai pendidikan bagi para pengunjungnya. Sementara objek wisata seperti museum satwa yang mempunyai nilai sejarah justru tidak banyak kita jumpai, selain itu pengunjung objek-objek wisata museum satwa semakin menurun. Untuk mengatasi keadaan yang demikian, maka perlu diadakan usaha-usaha penyampaian informasi dan dokumentasi yang dikemas dengan menarik mengenai informasi didalam museum satwa berupa aplikasi virtual. Aplikasi ini akan mempermudah masyarakat untuk lebih mengenal museum satwa dan informasi objek-objek di dalamnya dengan berbasis 3D. Sehingga rasa keingintahuan masyarakat tentang museum, khususnya museum satwa akan bertambah.

Pada penelitian ini, desain marker digunakan untuk menyimpan data hewan beserta keterangan informasi dalam bentuk 3D. Smartphone Android sebagai media perangkat keras untuk menjalankan aplikasi dan Augmented Reality sebagai metode pembuatan aplikasi. Eclipse adalah software yang dipakai untuk membuat Aplikasi Mobile Museum Satwa Virtual Berbasis Android.

Pengujian aplikasi telah berhasil menangkap desain marker sehingga dapat menampilkan objek hewan beserta keterangan hewan tersebut. Dengan dibangunnya Aplikasi Mobile Museum Satwa Berbasis Android ini dapat di manfaatkan sebagai media pembelajaran.

Kata kunci : Augmented Reality, Android Smartphone, Marker

KATA PENGANTAR

Dengan mengucapkan syukur kehadiran Tuhan Yang Maha Esa yang dengan segala Kasih dan Anugerah-Nya, telah memberikan kekuatan, kesabaran, bimbingan dan perlindungan sehingga penulis dapat menyelesaikan laporan skripsi dengan judul:

” APLIKASI MOBILE MUSEUM SATWA VIRTUAL BERBASIS ANDROID”

Pembuatan skripsi ini disusun guna memenuhi syarat akhir kelulusan pendidikan jenjang Strata I di Institut Teknologi Nasional Malang. Dalam penyusunan skripsi ini penulis banyak mendapat bantuan baik moril maupun materiil, saran dan dorongan semangat dari berbagai pihak, untuk itu penulis mengucapkan terima kasih kepada :

1. Bapak Dr. Ir. Lalu Mulyadi, MTA. Selaku rektor ITN Malang.
2. Bapak Ir. Anang Subardi, MT. Selaku Dekan Fakultas Teknologi Industri.
3. Bapak M. Ibrahim Ashari, ST, MT. Selaku Ketua Jurusan Teknik Elektro S-1 ITN Malang.
4. Bapak Dr. Eng. I Komang somawirata, ST, MT. Selaku Sekretaris Jurusan Teknik Elektro S-1 ITN Malang.
5. Bapak Dr. Eng. Aryuanto Soetedjo, ST, MT. Selaku Dosen Pembimbing I.
6. Ibu Ir. Ni Putu Agustini, MT. Selaku Dosen Pembimbing II.
7. Bapak Dr. F. Yudi Limpraptono, ST, MT. Selaku Dosen Penguji I.
8. Ibu Yuli Wahyuni, ST, MT. Selaku Dosen Penguji II.
9. Dan semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini yang tidak bisa penulis sebutkan satu persatu.

Penulis menyadari bahwa laporan ini masih banyak yang perlu disempurnakan. Oleh sebab itu kritik dan saran yang membangun sangat diharapkan. Akhir kata, penulis mohon maaf kepada semua pihak bilamana selama penyusunan skripsi ini penyusun membuat kesalahan secara tidak sengaja dan semoga skripsi ini dapat bermanfaat bagi kita semua.

Malang, September 2015

Penulis

DAFTAR ISI

Lembar Persetujuan.....	i
Surat Pernyataan Orisinalitas	ii
Abstrak	iii
Kata Pengantar	iv
Daftar isi.....	v
Daftar Gambar.....	vii
Daftar Tabel	viii

BAB I PENDAHULUAN

1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Metode Penelitian.....	2
1.6 Sistematika Penulisan.....	3

BAB II LANDASAN TEORI

2.1 Augmented Reality.....	5
2.2 Sejarah Augmented Reality.....	6
2.3 Markeless Augmented Reality	6
2.4 Vuforia SDK	7
2.5 ARTooKit.....	8
2.6 Marker	9
2.7 Android.....	10
2.7.1 Versi Android.....	10
2.8 Sistem Operasi Android	13
2.8.1 Fitur Antar Muka Android	15
2.8.2 Fitur Aplikasi Android	16
2.8.3 Fitur Pengelolaan Memori Android	17
2.9 Arsitektur Android	18
2.10 Android SDK.....	21
2.11 ADT.....	21
2.12 Eclips IDE	22

BAB III ANALISA DAN PERANCANGAN SISTEM

3.1	Diskripsi Sistem.....	24
3.2	Analisa Kebutuhan Perangkat Keras	25
3.3	Metode Pengumpulan Data	27
3.4	Proses Alur Kerja Aplikasi Mobile Museum Satwa Virtual Berbasis Android	27
3.5	Analisi Resiko	28
3.6	Perancangan.....	29
	3.6.1 Flowchart Diagram.....	29
	3.6.2 Antar Muka Aplikasi	30
3.7	Kontruksi Sistem	32
	3.7.1 Penginstalan ARToolKit	32
	3.7.2 Perancangan Objek 3D.....	32
	3.7.3 perancangan Marker	32

BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1	Implementasi Perangkat Lunak	34
4.2	Interface Perangkat Lunak	34
4.3	Menu Capture	35
4.4	Menu About	36
4.5	Menu Keluar	37
4.6	Uji Sumber Marker	37
4.7	Pengujian Sistem dan Hasil	38

BAB V PENUTUP

5.1	Kesimpulan	42
5.2	Saran	42

Daftar Pustaka	43
----------------------	----

Lampiran

DAFTAR GAMBAR

Gambar 2.1 Penggunaan ARToolKit	9
Gambar 2.2 Sistem Operasi Android	13
Gambar 2.3 Arsitektur Platform Android	18
Gambar 3.1 Konsep Aplikasi Mobile Museum Satwa Virtual Berbasis Android	27
Gambar 3.2 Flowchart Aplikasi	29
Gambar 3.3 Tampilan <i>Men</i>	30
Gambar 3.4 Antar Muka Kamera.....	31
Gambar 3.5 Antar Muka About.....	31
Gambar 3.6 Marker Objek 3D.....	33
Gambar 4.1 Menu Utama Aplikasi	35
Gambar 4.2 Menu Capture Marker	36
Gambar 4.3 Menu About.....	36
Gambar 4.4 Menu Keluar.....	37
Gambar 4.5 Uji Sumber Marker.....	38

DAFTAR TABEL

Tabel 4.1 Hasil Pengujian Akurasi Sistem.....	39
--	-----------

BAB I PENDAHULUAN

1.1. Latar Belakang

Pada era globalisasi kini telah berkembang berbagai jenis smartphone yang mengusung banyak sistem operasi, salah satunya adalah sistem operasi android. Android merupakan sistem operasi yang di kembangkan oleh perusahaan bernama android Inc yang kemudian di kembangkan oleh google, karena google melihat banyaknya user yang online dengan perangkat mobile, maka google mengira bahwa perangkat mobile ini mempunyai masa depan yang cerah, sehingga android Inc diakuisisi oleh google di tahun 2005.

Augmented Reality adalah teknologi yang menggabungkan benda maya dua dimensi dan ataupun tiga dimensi ke dalam sebuah lingkungan nyata tiga dimensi lalu memproyeksikan benda-benda maya tersebut dalam waktu nyata (real time). Tidak seperti realitas maya yang sepenuhnya menggantikan kenyataan, namun *Augmented Reality* hanya menambahkan atau melengkapi kenyataan.

Museum satwa adalah sebuah tempat rekreasi yang menyajikan satwa-satwa yang di awetkan dan fosil-fosil purba. Pendiri Museum satwa didasari keinginan untuk memberikan pengetahuan dan pembelajaran satwa kepada pengunjung terutama usia sekolah sebagai pendukung pembelajaran satwa yang disajikan. Selain sebagai tempat rekreasi, museum satwa berfungsi sebagai tempat pendidikan, dan riset. Di sini museum satwa menampilkan segala hal yang berhubungan dengan seluk-beluk dunia satwa.

Dengan adanya *Augmented Reality* kini telah banyak dikembangkan yang memanfaatkan *Augmented Reality*, seperti pembuatan game, film. Dari pembahasan diatas, peneliti tertarik untuk memanfaatkan teknologi *Augmented Reality* dengan sebuah ide “*APLIKASI MOBILE MUSEUM SATWA VIRTUAL BERBASIS ANDROID*” dengan harapan dapat menjadi aplikasi pembelajaran.

1.2. Rumusan Masalah

1. Bagaimana Aplikasi Mobile Museum Satwa Virtual Berbasis Android dapat diterapkan di Smartphone menggunakan augmented reality.
2. Bagaimana membuat desain marker untuk menampilkan hewan 3D.

1.3. Batasan Masalah

Agar permasalahan mengarah sesuai dengan tujuan, maka pembahasan di batasi pada hal-hal sebagai berikut :

1. Pembuatan aplikasi mobile museum satwa virtual menggunakan *Augmented Reality*.
2. Menggunakan sistem operasi android.
3. Menggunakan Eclipse sebagai aplikasi open source yang berfungsi untuk membangun aplikasi mobile museum satwa virtual.

1.4. Tujuan

Dengan di buatnya Aplikasi Mobile Museum Satwa Virtual Berbasis Android dapat memberikan kemudahan dalam mengetahui informasi dan mengenal macam-macam satwa.

1.5. Metode Penelitian

Metodologi yang digunakan pada penelitian skripsi ini adalah sebagai berikut :

1. Studi Literatur

Pengumpulan data yang dilakukan dengan mencari bahan-bahan kepustakaan dan referensi dari berbagai sumber sebagai landasan teori yang ada hubungannya dengan permasalahan yang dijadikan objek penelitian.

2. Analisa Kebutuhan Aplikasi

Data dan informasi yang telah diperoleh akan dianalisa agar didapatkan kerangka global yang bertujuan untuk mendefinisikan kebutuhan sistem baik hardware maupun software, dimana nantinya akan digunakan sebagai acuan perancangan aplikasi.

3. Perancangan Aplikasi

Berdasarkan data dan informasi yang telah diperoleh serta analisa kebutuhan untuk membangun aplikasi ini, akan dibuat rancangan kerangka global yang menggambarkan mekanisme dari aplikasi yang akan dibuat.

4. Coding

Tahap ini dilakukan implementasi hasil rancangan kedalam baris-baris kode program yang dapat dimengerti oleh komputer.

5. Uji coba aplikasi

Pada tahap ini, aplikasi yang telah selesai dibuat dan akan diuji coba, yaitu pengujian berdasarkan struktur data tampilan program dan akan dilakukan koreksi dan penyempurnaan program jika diperlukan.

1.6. Sistematika Penulisan

Sistematika penulisan yang diuraikan dalam penyusunan skripsi ini adalah sebagai berikut :

BAB I : PENDAHULUAN

Bab ini berisi tentang Latar Belakang, Rumusan Masalah, Tujuan, Batasan Masalah, Metodologi Penelitian dan Sistematikan Penulisan.

BAB II : LANDASAN TEORI

Berisi tentang landasan teori mengenai permasalahan yang berhubungan dengan penelitian yang dilakukan.

BAB III : PERANCANGAN DAN DESAIN APLIKASI

Dalam bab ini berisi mengenai analisa kebutuhan sistem baik software maupun hardware yang diperlukan untuk membuat kerangka global yang menggambarkan mekanisme dari sistem yang akan dibuat.

BAB IV : UJI COBA APLIKASI

Bab ini berisi tentang pengujian aplikasi pada hardware yang sudah dibuat.

BAB V : PENUTUP

Pada bab ini berisikan kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk pengembangan penulisan selanjutnya.

BAB II LANDASAN TEORI

2.1. **Augmented Reality**

Augmented Reality adalah teknologi yang menggabungkan benda maya dua dimensi dan ataupun tiga dimensi ke dalam sebuah lingkungan nyata tiga dimensi lalu memproyeksikan benda-benda maya tersebut dalam waktu nyata (real time). Tidak seperti realitas maya yang sepenuhnya menggantikan kenyataan, namun *Augmented Reality* hanya menambahkan atau melengkapi kenyataan.

Menurut Ronald Azuma pada tahun 1997, *Augmented Reality* adalah menggabungkan dunia maya dan virtual, bersifat interaktif secara *real time*, dan bentuknya merupakan animasi 3D. Yang dimaksud interaktif disini adalah adanya interaksi dari *user* ke AR tersebut. Sehingga ada pengaruh di *Augmented Reality* tersebut, misalnya *user* menggunakan *handphone* yang terdapat tombol-tombol untuk menjalankan atau memberi efek pada *Augmented Reality*. *Augmented Reality* seperti ini biasanya digunakan *Smartphone* untuk membuat *Game Augmented Reality* yang bersifat interaktif.

Benda-benda maya menampilkan informasi yang tidak dapat diterima oleh pengguna dengan inderanya sendiri. Hal ini membuat *Augmented Reality* sesuai sebagai alat untuk membantu persepsi dan interaksi pengguna dengan dunia nyata. Informasi yang ditampilkan oleh benda maya membantu pengguna melaksanakan kegiatan-kegiatan dalam dunia nyata.

Ada tiga karakteristik yang menyatakan suatu teknologi menerapkan konsep *Augmented Reality*, yaitu :

1. Mampu mengkombinasikan dunia nyata dan dunia maya.
2. Mampu memberikan informasi secara interaktif dan real-time.
3. Mampu menampilkan dalam bentuk 3D (tiga dimensi).

2.2. Sejarah Augmented Reality

Sejarah tentang augmented reality dimulai tahun 1957-1962, ketika seorang penemu yang bernama Morton Heiking, seorang sinematografer, menciptakan dan mematenkan sebuah simulator yang disebut sensorama dengan visual, getaran dan bau. Pada tahun 1966, Ivan Sutherland menemukan head-mounted display yang dia claim adalah, jendela ke dunia virtual. Tahun 1975 seorang ilmuwan bernama Myron Krueger menemukan videoplace yang memungkinkan pengguna, dapat berinteraksi dengan objek virtual untuk pertama kalinya. Tahun 1989, Jaron Lainer, memperkenalkan Virtual Reality untuk melakukan perbaikan pada pesawat boeing, dan pada tahun yang sama, LB Rosenberg mengembangkan salah satu fungsi sistem AR, yang disebut Virtual Fixtures, yang digunakan di Angkatan udara AS Amstrong Labs, dan menunjukkan manfaat pada manusia, dan pada tahun 1992 juga, Steve Feiner, Blair MacIntyre dan Doree Selingmann memperkenalkan untuk pertamakalinya Major Paper untuk perkembangan Prototype AR.

2.3. Markerless Augmented Reality

Salah satu metode *Augmented Reality* yang saat ini sedang berkembang adalah metode *Markerless Augmented Reality*, dengan metode ini pengguna tidak perlu lagi menggunakan sebuah *marker* untuk menampilkan elemen-elemen *digital*. Seperti yang saat ini dikembangkan oleh perusahaan *Augmented Reality* terbesar di dunia *Total Immersion* dan

Qualcomm, mereka telah membuat berbagai macam teknik *Markerless Tracking* sebagai teknologi andalan mereka, seperti *Face Tracking*, *3D Object Tracking*, dan *Motion Tracking*.

1. Face Tracking

Dengan menggunakan algoritma yang mereka kembangkan, komputer dapat mengenali wajah manusia secara umum dengan cara mengenali posisi mata, hidung, dan mulut manusia, kemudian akan mengabaikan objek-objek lain di sekitarnya seperti pohon, rumah, dan benda-benda lainnya. Teknik ini pernah digunakan di Indonesia pada. Pekan Raya Jakarta 2010 dan Toy Story 3 Event.

2. 3D Object Tracking

Berbeda dengan Face Tracking yang hanya mengenali wajah manusia secara umum, teknik 3D *Object Tracking* dapat mengenali semua bentuk benda yang ada disekitar, seperti mobil, meja, televisi, dan lain-lain.

3. Motion Tracking

Pada teknik ini komputer dapat menangkap gerakan, *Motion Tracking* telah mulai digunakan secara ekstensif untuk memproduksi film-film yang mencoba mensimulasikan gerakan. Contohnya pada film Avatar, di mana James Cameron menggunakan teknik ini untuk membuat film tersebut dan menggunakannya secara *realtime*.

2.4. Vuforia SDK

Vuforia adalah *Augmented Reality Software Development Kit* (SDK) untuk perangkat mobile yang memungkinkan pembuatan aplikasi *Augmented Reality*. Dulunya lebih dikenal dengan QCAR (*Qualcomm Company Augmentend Reality*). Ini menggunakan teknologi *Computer Vision* untuk mengenali dan melacak gambar planar (*Target Image*) dan objek 3D sederhana, seperti kotak, secara *real-time*. Kemampuan registrasi citra memungkinkan pengembang untuk mengatur posisi dan *virtual* orientasi objek, seperti model 3D dan media

lainnya, dalam kaitannya dengan gambar dunia nyata ketika hal ini dilihat melalui kamera perangkat *mobile*. Obyek maya kemudian melacak posisi dan orientasi dari gambar secara *real-time* sehingga perspektif pengguna pada objek sesuai dengan perspektif mereka pada *Target Image*, sehingga muncul bahwa objek *virtual* adalah bagian dari adegan dunia nyata.

SDK Vuforia mendukung berbagai jenis target 2D dan 3D termasuk Target Gambar 'markerless', 3D Multi target konfigurasi, dan bentuk *Marker Frame*. Fitur tambahan dari SDK termasuk Deteksi Oklusi lokal menggunakan 'Tombol virtual', *runtime* pemilihan gambar *target*, dan kemampuan untuk membuat dan mengkonfigurasi ulang set pemrograman

pada saat *runtime*. Vuforia menyediakan *Application Programming Interfaces* (API) di C++, Java, Objective-C. SDK mendukung pembangunan untuk IOS dan

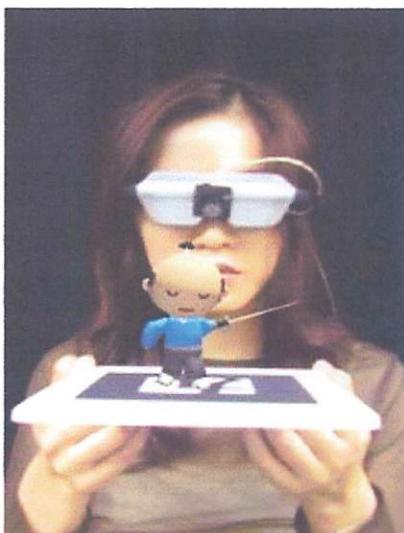
Android menggunakan Vuforia karena itu kompatibel dengan berbagai perangkat *mobile* termasuk iPhone (4/4S), iPad, dan ponsel Android dan *tablet* yang menjalankan Android OS versi 2.2 atau yang lebih besar dan prosesor ARMv6 atau 7 dengan FPU (*Floating Point Unit*)

kemampuan pengolahan. *Qualcomm Augmented Reality* memberikan beberapa keuntungan seperti :

1. Teknologi *computer vision* untuk menyelaraskan gambar yang tercetak dan *object* 3D.
2. Mendukung beberapa alat development seperti Eclipse, Android, Xcode. Selain itu, QCAR juga menawarkan *development* dan distribusi yang gratis (Qualcomm, 2012).

2.5. ARToolKit

ARToolKit adalah salah satu pustaka (*library*) perangkat lunak berbasis C yang menggunakan metoda *computer vision tracking* untuk menghitung posisi kamera dan orientasinya yang relatif terhadap *marker*. ARToolKit dikembangkan oleh Dr. Hirokazu Kato dari Universitas Osaka Jepang dan Mark Billinghurst dari *Human Interface Technology Laboratory* (HIT Lab). ARToolKit banyak digunakan untuk mengembangkan aplikasi AR.



Gambar 2.1 Penggunaan ARToolKit

Sumber: www.hitl.washington.edu/artoolkit/

Saat ini ARToolKit bisa berjalan pada Sistem Operasi SGI IRIX, PC Linux, Mac OS X, dan PC Windows (95/98/NT/2000/XP). Fungsi setiap versi ARToolKit adalah sama, tetapi performanya bisa berbeda-beda tergantung dari konfigurasi hardware yang digunakan.

2.6. Marker

Marker merupakan sebuah penanda khusus yang memiliki pola tertentu sehingga saat kamera mendeteksi *marker*, objek 3 dimensi dapat ditampilkan. Augmented reality saat ini melakukan perkembangan besar-besaran, salah satunya pada bagian marker. Marker pertama adalah marker based tracking. Marker Based Tracking ini sudah lama dikembangkan sejak 1980-an dan pada awal 1990-an mulai dikembangkan untuk penggunaan Augmented Reality. Kemudian Markerless, perkembangan terbaru marker ini merupakan salah satu metode Augmented Reality tanpa menggunakan frame marker sebagai obyek yang dideteksi. Dengan adanya Markerless Augmented Reality, maka, penggunaan marker sebagai tracking object yang selama ini menghabiskan ruang, akan digantikan dengan gambar, atau permukaan apapun yang berisi dengan tulisan, logo, atau gambar sebagai tracking object (obyek yang dilacak) agar dapat langsung melibatkan obyek yang dilacak tersebut sehingga dapat terlihat hidup dan interaktif, juga tidak lagi mengurangi efisiensi ruang dengan adanya.

2.7. Android

Android adalah sistem operasi untuk ponsel yang berbasis *Linux*. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak.

2.7.1. Versi Android

Sampai saat ini google telah meluncurkan banyak versi. Peluncuran versi-versi terbaik android lebih diutamakan untuk perkembangan dan perbaikan dari versi sebelumnya, versi android tersebut adalah :

1. Android Beta

Dirilis pada tanggal 5 November 2007, kemudian SDK (*Software Development Kit*) dirilis tanggal 12 November 2007.

2. Android 1.0

Versi ini merupakan android komersial yang pertama kali dan dirilis pada tanggal 23 November 2008. Perangkat yang pertama kali menggunakan versi ini adalah HTC Dream.

3. Android 1.1

Versi ini hanya dirilis untuk HTC Dream saja pada tanggal 9 februari 2009 dengan perbaikan bug dan beberapa fitur tambahan.

4. Android 1.5 (cupcake)

Pada pertengahan mei 2009, google kembali merilis telepon seluler dengan menggunakan android dan SDK (*Software Development Kit*) dengan versi 1.5 (cupcake). Terdapat beberapa pembaharuan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke youtube dan gambar ke picasa langsung dari telepon, dukungan bluetooth A2DP, kemampuan terhubung secara otomatis ke headset bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

5. Android 1.6 (Donut)

Donut (versi 1.6) dirilis pada september dengan menampilkan proses pencarian yang lebih baik di banding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang diintegrasikan; CDMA/EVDO, 802.1x, VPN, Gestares, dan Text-to-speech engine.

6. Android 2.0/2.1 (Eclair)

Pada 3 desember 2009 kembali diluncurkan ponsel android versi 2.0/2.1 (Eclair), perubahan yang dilakukan adalah pengoptimalan hardware, perubahan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML

5, daftar kontak yang baru, dukungan flash untuk camera 3,2 MP, digital Zoom dan bluetooth 2.1.

Dengan semakin berkembangnya dan semakin bertambahnya jumlah handset android, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka pada sistem operasi android. Aplikasi terkenal yang diubah ke dalam sistem operasi android adalah shazam, *backgrounds*, dan weatherBug. Sistem operasi android dalam situs internet juga dianggap penting untuk menciptakan aplikasi Android asli, contohnya oleh Myspace dan Facebook.

7. Android 2.2.x(Frozen Yoghurt/Froyo)

Pada 20 mei 2010, android versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umumnya terhadap umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi V8 JavaScript *engine* yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD card, kemampuan WIFI Hostpot portabel, dan kemampuan *auto update* dalam aplikasi Android Market.

8. Android 2.3.x (Gingerbread)

Pada 6 desember 2010, android versi 2.3 (Gingerbread) di luncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan fungsi copy paste, layar antarmuka (user interface) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (reverb, equalization, headphone virtualization, dan bass boost) dan dukungan jumlah kamera yang lebih dari satu.

9. Android 3.x (Honeycomb)

Android Honeycomb di rancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. User interface pada Honeycomb juga berbeda karena sudah di desain untuk tablet. Honeycomb juga mendukung multi prosesor dan akselerasi perangkat keras (*hardware*) untuk grafis.

10. Android 4.0.x (Ice Cream Sandwich)

Di umumkan pada tanggal 19 oktober 2011, membawa fitur Honeycomb untuk *smartphone* dan menambahkan fitur baru termasuk membuka kunci

dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan geografi, mencari email secara offline, dan berbagai informasi.

11. Android 4.1 (Jelly Bean)

Android Jelly Bean diluncurkan oleh google pada tanggal 27 juni 2012 pada acara Google I/O Conference. Versi ini berbasiskan linux kernel 3.0.1.

Google Now juga menjadi bagian yang harus di perbarui. Google Now memberikan informasi yang tepat pada waktu yang tepat pula. Salah satu kemampuannya adalah dapat mengetahui informasi cuaca, lalu lintas, atau hasil pertandingan olahraga. OS Android Jelly Bean 4.1 muncul pertama kali dalam produk tablet Asus, yakni Google Nexus 7.

2.8. Sistem Operasi Android



Gambar 2.2 Sistem Operasi Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008.

Antarmuka pengguna Android didasarkan pada manipulasi langsung, menggunakan masukan sentuh yang serupa dengan tindakan di dunia nyata, seperti menggesek, mengetuk, mencubit, dan membalikkan cubitan untuk memanipulasi obyek di layar. Android adalah sistem operasi dengan sumber terbuka, dan Google merilis kodenya di bawah Lisensi Apache. Kode dengan sumber terbuka dan lisensi perizinan pada Android memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator nirkabel, dan pengembang aplikasi. Selain itu, Android memiliki sejumlah besar komunitas pengembang aplikasi (apps) yang memperluas fungsionalitas perangkat, umumnya ditulis dalam versi kustomisasi bahasa pemrograman Java. Pada bulan Oktober 2012, ada sekitar 700.000 aplikasi yang tersedia untuk Android, dan sekitar 25 juta aplikasi telah diunduh dari Google Play, toko aplikasi utama Android. Sebuah survey pada bulan April-Mei 2013 menemukan bahwa Android adalah platform paling populer bagi para pengembang, digunakan oleh 71% pengembang aplikasi seluler.

Faktor-faktor di atas telah memberikan kontribusi terhadap perkembangan Android, menjadikannya sebagai sistem operasi telepon pintar yang paling banyak digunakan di dunia, mengalahkan Symbian pada tahun 2010. Android juga menjadi pilihan bagi perusahaan teknologi yang menginginkan sistem operasi berbiaya rendah, bisa dikustomisasi, dan ringan untuk perangkat berteknologi tinggi tanpa harus mengembangkannya dari awal. Akibatnya, meskipun pada awalnya sistem operasi ini dirancang khusus untuk telepon pintar dan tablet, Android juga dikembangkan menjadi aplikasi tambahan di televisi, konsol permainan, kamera digital, dan perangkat elektronik lainnya. Sifat Android yang terbuka telah mendorong munculnya sejumlah besar komunitas pengembang aplikasi untuk menggunakan kode sumber terbuka sebagai dasar proyek pembuatan aplikasi, dengan menambahkan fitur-fitur baru bagi pengguna tingkat lanjut atau mengoperasikan Android pada perangkat yang secara resmi dirilis dengan menggunakan sistem operasi lain.

Pada November 2013, Android menguasai pangsa pasar telepon pintar global, yang dipimpin oleh produk-produk Samsung, dengan persentase 64% pada bulan Maret 2013. Pada Juli 2013, terdapat 11.868 perangkat Android berbeda

dengan beragam versi. Keberhasilan sistem operasi ini juga menjadikannya sebagai target litigasi paten "perang telepon pintar" antar perusahaan-perusahaan teknologi. Hingga bulan Mei 2013, total 900 juta perangkat Android telah diaktifkan di seluruh dunia, dan 48 miliar aplikasi telah dipasang dari Google Play. Pada tanggal 3 September 2013, 1 miliar perangkat Android telah diaktifkan.

2.8.1. Fitur Antarmuka Android

Antarmuka pengguna pada Android didasarkan pada manipulasi langsung, menggunakan masukan sentuh yang serupa dengan tindakan di dunia nyata, misalnya menggesek (*swiping*), mengetuk (*tapping*), dan mencubit (*pinching*), untuk memanipulasi obyek di layar. Masukan pengguna direspon dengan cepat dan juga tersedia antarmuka sentuh layaknya permukaan air, seringkali menggunakan kemampuan getaran perangkat untuk memberikan umpan balik haptik kepada pengguna. Perangkat keras internal seperti akselerometer, giroskop, dan sensor proksimitas digunakan oleh beberapa aplikasi untuk merespon tindakan pengguna, misalnya untuk menyesuaikan posisi layar dari potret ke lanskap, tergantung pada bagaimana perangkat diposisikan, atau memungkinkan pengguna untuk mengarahkan kendaraan saat bermain balapan dengan memutar perangkat sebagai simulasi kendali setir.

Ketika dihidupkan, perangkat Android akan memuat pada layar depan (*homescreen*), yakni navigasi utama dan pusat informasi pada perangkat, serupa dengan desktop pada komputer pribadi. Layar depan Android biasanya terdiri dari ikon aplikasi dan widget; ikon aplikasi berfungsi untuk menjalankan aplikasi terkait, sedangkan widget menampilkan konten secara langsung dan terbaru otomatis, misalnya prakiraan cuaca, kotak masuk surel pengguna, atau menampilkan tiker berita secara langsung dari layar depan. Layar depan bisa terdiri dari beberapa halaman, pengguna dapat menggeser bolak balik antara satu halaman ke halaman lainnya, yang memungkinkan pengguna Android untuk mengatur tampilan perangkat sesuai dengan selera mereka. Beberapa aplikasi pihak ketiga yang tersedia di Google Play dan di toko aplikasi lainnya secara ekstensif mampu mengatur kembali tema layar depan Android, dan bahkan bisa

meniru tampilan sistem operasi lain, misalnya Windows Phone. Kebanyakan produsen telepon seluler dan operator nirkabel menyesuaikan tampilan perangkat Android buatan mereka untuk membedakannya dari pesaing mereka.

Di bagian atas layar terdapat status bar, yang menampilkan informasi tentang perangkat dan konektivitasnya. Status bar ini bisa "ditarik" ke bawah untuk membuka layar notifikasi yang menampilkan informasi penting atau pembaruan aplikasi, misalnya surel diterima atau SMS masuk, dengan cara tidak mengganggu kegiatan pengguna pada perangkat. Pada versi awal Android, layar notifikasi ini bisa digunakan untuk membuka aplikasi yang relevan, namun setelah diperbarui, fungsi ini semakin disempurnakan, misalnya kemampuan untuk memanggil kembali nomor telepon dari notifikasi panggilan tak terjawab tanpa harus membuka aplikasi utama. Notifikasi ini akan tetap ada sampai pengguna melihatnya, atau dihapus dan di nonaktifkan oleh pengguna.

2.8.2. Fitur Aplikasi Android

Android memungkinkan penggunanya untuk memasang aplikasi pihak ketiga, baik yang diperoleh dari toko aplikasi seperti Google Play, Amazon Appstore, ataupun dengan mengunduh dan memasang berkas APK dari situs pihak ketiga. Di Google Play, pengguna bisa menjelajah, mengunduh, dan memperbarui aplikasi yang diterbitkan oleh Google dan pengembang pihak ketiga, sesuai dengan persyaratan kompatibilitas Google. Google Play akan menyaring daftar aplikasi yang tersedia berdasarkan kompatibilitasnya dengan perangkat pengguna, dan pengembang dapat membatasi aplikasi ciptaan mereka bagi operator atau negara tertentu untuk alasan bisnis. Pembelian aplikasi yang tidak sesuai dengan keinginan pengguna dapat dikembalikan dalam waktu 15 menit setelah pengunduhan. Beberapa operator seluler juga menawarkan tagihan langsung untuk pembelian aplikasi di Google Play dengan cara menambahkan harga pembelian aplikasi pada tagihan bulanan pengguna. Pada bulan September 2012, ada lebih dari 675.000 aplikasi yang tersedia untuk Android, dan perkiraan jumlah aplikasi yang diunduh dari Play Store adalah 25 miliar.

Aplikasi Android dikembangkan dalam bahasa pemrograman Java dengan menggunakan kit pengembangan perangkat lunak Android (SDK). SDK ini terdiri

dari seperangkat perkakas pengembangan,[61] termasuk debugger, perpustakaan perangkat lunak, emulator handset yang berbasis QEMU, dokumentasi, kode sampel, dan tutorial. Didukung secara resmi oleh lingkungan pengembangan terpadu (IDE) Eclipse, yang menggunakan plugin Android Development Tools (ADT). Perkakas pengembangan lain yang tersedia di antaranya adalah Native Development Kit untuk aplikasi atau ekstensi dalam C atau C++, Google App Inventor, lingkungan visual untuk pemrogram pemula, dan berbagai kerangka kerja aplikasi web seluler lintas platform.

2.8.3. Fitur Pengelolaan Memori Android

Android dikembangkan secara pribadi oleh Google sampai perubahan terbaru dan pembaruan siap untuk dirilis, dan informasi mengenai kode sumber juga mulai diungkapkan kepada publik. Kode sumber ini hanya akan berjalan tanpa modifikasi pada perangkat tertentu, biasanya pada seri Nexus. Ada binari tersendiri yang disediakan oleh produsen agar Android bisa beroperasi.

Logo Android yang berwarna hijau awalnya dirancang untuk Google pada tahun 2007 oleh desainer grafis Irina Blok. Tim desain ditugaskan dengan sebuah proyek untuk membuat sebuah ikon universal yang mudah dikenali dengan menyertakan ikon robot secara spesifik dalam desain akhir. Setelah sejumlah perkembangan desain yang didasarkan pada tema-tema fiksi ilmiah dan film luar angkasa, tim akhirnya mendapat inspirasi dari simbol manusia yang terdapat di pintu toilet, dan memodifikasi bentuknya menjadi bentuk robot. Karena Android adalah perangkat lunak sumber terbuka, disepakati bahwa logo tersebut juga harus terbuka, dan sejak diluncurkan, logo hijau tersebut telah didesain ulang kembali dalam berbagai variasi yang tak terhitung jumlahnya.

2.9. Arsitektur Android

Android terdiri dari susunan arsitektur atau lapisan dari komponen Platform Android. Pada gambar 2.2 dapat dilihat lapisan-lapisan dari arsitektur Android.



Gambar 2.3 Arsitektur Platform Android

Sumber: http://elinux.org/Android_Architecture

Google mengibaratkan Android sebagai sebuah tumpukan software. Setiap lapisan dari tumpukan ini menghimpun beberapa program yang mendukung fungsi-fungsi spesifik dari sistem operasi. Berikut ini susunan dari lapisan-lapisan tersebut jika di lihat dari lapisan dasar hingga lapisan teratas.

1. Linux Kernel

Tumpukan paling bawah pada arsitektur Android ini adalah kernel. Google menggunakan kernel Linux versi 2.6 untuk membangun sistem Android, yang mencakup memory management, security setting, power management, dan beberapa driver hardware. Kernel berperan sebagai abstraction layer antara hardware dan keseluruhan software. Sebagai contoh, HTC GI dilengkapi dengan kamera. Kernel Android terdapat driver kamera yang memungkinkan pengguna mengirimkan perintah kepada hardware kamera.

Android dibangun di atas kernel Linux 2.6. Namun secara keseluruhan android bukanlah linux, karena dalam android tidak terdapat paket standar yang dimiliki oleh linux lainnya. Linux merupakan sistem operasi terbuka yang

handal dalam manajemen memori dan proses. Oleh karenanya pada android hanya terdapat beberapa servis yang diperlukan seperti keamanan, manajemen memori, manajemen proses, jaringan dan driver. Kernel linux menyediakan driver layar, kamera, keypad, WiFi, Flash Memory, audio, dan IPC (Interprocess Communication) untuk mengatur aplikasi dan lubang keamanan.

2. Android Run Time

Lapisan setelah Kernel Linux adalah Android Runtime yang berisi Core Libraries dan Dalvik. **Core Libraries** mencakup serangkaian inti library Java, artinya Android menyertakan satu set library-library dasar yang menyediakan sebagian besar fungsi-fungsi yang ada pada library-library dasar bahasa pemrograman Java. **Dalvik** adalah Java Virtual Machine yang memberi kekuatan pada sistem Android. Dalvik VM ini di optimalkan untuk telepon seluler. Setiap aplikasi yang berjalan pada Android berjalan pada processnya sendiri, dengan instance dari Dalvik Virtual Machine. Dalvik telah dibuat sehingga sebuah piranti yang memakainya dapat menjalankan multi Virtual Machine dengan efisien. Dalvik VM dapat mengeksekusi file dengan format Dalvik Executable (.dex) yang telah dioptimasi untuk menggunakan minimal memory footprint. Virtual Machine ini register-based, dan menjalankan class-class yang dicompile menggunakan compiler Java yang kemudian ditransformasi menjadi format .dex menggunakan “dx” tool yang telah disertakan. Dalvik Virtual Machine (VM) menggunakan kernel Linux untuk menjalankan fungsi-fungsi seperti threading dan low-level memory management.

3. Libraries

Bertempat di level yang sama dengan Android Runtime adalah Libraries. Android menyertakan satu set library-library dalam bahasa C/C++ yang digunakan oleh berbagai komponen yang ada pada sistem Android. Kemampuan ini dapat diakses oleh programmer melewati Android application framework. Sebagai contoh Android mendukung pemutaran format audio, video, dan gambar. Berikut ini beberapa core library tersebut :

1) Sistem C library

Diturunkan dari implementasi standard C sistem library (libc) milik BSD, dioptimasi untuk piranti embedded berbasis Linux.

2) Media Libraries

Berdasarkan PacketVideo's OpenCORE; library-library ini mendukung playback dan recording dari berbagai format audio and video populer, meliputi MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG.

3) Surface Manager

Mengatur akses pada display dan lapisan composites 2D and 3D graphic dari berbagai aplikasi.

4) LibWebCore

Web browser engine modern yang mensupport Android browser maupun embeddable web view.

5) SGL, the underlying 2D graphics engine.

6) 3D libraries

Implementasi berdasarkan OpenGL ES 1.0 APIs; library ini menggunakan hardware 3D acceleration dan highly optimized 3D software rasterizer.

7) FreeType, bitmap dan vector font rendering.

8) SQLite

Relational database engine yang powerful dan ringan tersedia untuk semua aplikasi.

Library-library tersebut bukanlah aplikasi yang berjalan sendiri, namun hanya dapat digunakan oleh program yang berada di level atasnya. Sejak versi Android 1.5, pengembang dapat membuat dan menggunakan pustaka sendiri menggunakan Native Development Toolkit (NDK).

4. Application Framework

Lapisan selanjutnya adalah application framework, yang mencakup program untuk mengatur fungsi-fungsi dasar smartphone. Application Framework merupakan serangkaian tool dasar seperti alokasi resource smartphone, aplikasi telepon, pergantian antar – proses atau program, dan pelacakan lokasi fisik telepon. Para pengembang aplikasi memiliki aplikasi penuh kepada tool-tool dasar tersebut, dan memanfaatkannya untuk menciptakan aplikasi yang lebih kompleks.

Programmer mendapatkan akses penuh untuk memanfaatkan API-API (Android Protocol Interface) yang juga digunakan core applications. Arsitektur aplikasi didesain untuk menyederhanakan pemakaian kembali komponen-komponen, setiap aplikasi dapat menunjukkan kemampuannya dan aplikasi lain dapat memakai kemampuan tersebut. Mekanisme yang sama.

2.10. *Android Software Development Kit (Android SDK)*

Android SDK merupakan *tools* bagi para *programmer* yang ingin mengembangkan aplikasi berbasis google android. Android SDK mencakup seperangkat alat pengembangan yang komprehensif. Android SDK terdiri dari *debugger*, *libraries*, *handset emulator*, dokumentasi, contoh kode dan tutorial. IDE yang didukung secara resmi adalah Eclipse 3.2 atau lebih dengan menggunakan *plugin Android Development Tools (ADT)*, dengan ini pengembang dapat menggunakan teks editor untuk mengedit file Java dan XML serta menggunakan peralatan *command line* untuk menciptakan, membangun, melakukan *debug* aplikasi Android dan pengendalian perangkat Android.

2.11. *ADT (Android Development Tools)*

Android Development Tools (ADT) adalah *plug-in* untuk *Eclipse IDE* yang dirancang untuk memberikan lingkungan yang *powerfull* dan terpadu untuk membangun aplikasi Android.

ADT memperluas kemampuan *Eclipse* untuk mempercepat dalam pembuatan project Android baru, membuat UI aplikasi, menambahkan komponen

berdasarkan Android Framework API, debug aplikasi menggunakan Android tool SDK, dan bahkan ekspor unsigned .apk file dalam rangka untuk mendistribusikan aplikasi. Mengembangkan aplikasi Android di Eclipse dengan ADT sangat dianjurkan dan merupakan cara tercepat untuk memulai membuat aplikasi Android. Dengan dipandu set-up project, serta integrasi peralatan, custom XML editor, dan debug panel output, ADT memberikan dorongan luar biasa dalam mengembangkan aplikasi Android.

2.12. Eclipse Integrated Development Environment (Eclipse IDE)

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent*). Berikut ini adalah sifat dari Eclipse:

1. **Multi-platform:** Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
2. **Multilanguage:** Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
3. **Multi-role:** Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.

Konsep Eclipse adalah IDE yang terbuka (*open*), mudah diperluas (*extensible*) untuk apa saja, dan tidak untuk sesuatu yang spesifik. Jadi, Eclipse tidak saja untuk mengembangkan program Java, akan tetapi dapat digunakan untuk berbagai macam keperluan, cukup dengan menginstal *plug-in* yang dibutuhkan. Apabila ingin mengembangkan program C/C++ terdapat *plug-in* CDT (*C/C++ Development Tools*). Selain itu, pengembangan secara visual

bukan hal yang tidak mungkin oleh Eclipse, *plug-in* UML2 tersedia untuk membuat diagram UML. Dengan menggunakan PDE setiap orang bisa membuat *plug-in* sesuai dengan keinginannya.

BAB III

ANALISA DAN DESAIN

Bab ini berisi penjelasan tentang desain sistem keseluruhan yang akan dibuat. Tahap-tahap dalam pengerjaan penelitian sesuai dengan metode penelitian. Berikut ini adalah tahap Analisa dan Perancangan sistem, diantaranya :

1. Deskripsi Sistem
2. Analisa Kebutuhan Perangkat Keras
3. Metode Pengumpulan Data
4. Proses Alur Kerja Aplikasi Mobile Museum Satwa Virtual Berbasis Android
5. Analisis Resiko
6. Perancangan
7. Flowchart
8. Antar Muka Aplikasi
9. Konstruksi Sistem
10. Penginstalan ARToolkit
11. Perancangan *Marker*

3.1. Deskripsi Sistem

Sistem aplikasi museum berbasis *augmented reality* ini, merupakan sebuah sistem dengan objek 3D sebagai *user manual* untuk menjelaskan satwa dari museum. Sehingga dapat memberikan kesan seperti dengan keadaan yang terdapat dalam dunia nyata, karena *augmented reality* adalah penggabungan objek virtual dengan objek nyata secara *real time*.

Aplikasi *Augmented Reality* (AR) ini akan menangkap objek melalui kamera *handphone* yang berupa kotak hitam atau yang disebut dengan *marker* yang kemudian ditandai untuk disimpan dalam *library*. Aplikasi ini dapat mengenali *marker* secara otomatis dengan cara menyorotkannya didepan kamera *webcam*, jika *marker* terbaca oleh kamera *webcam* maka akan muncul objek persegi sebagai tanda bahwa marker terbaca oleh kamera dan bisa disimpan dalam

library sebaliknya jika *marker* tidak terbaca oleh kamera maka tidak akan muncul objek persegi di objek *marker* sebagai tanda bahwa kamera tidak mengenali objek tersebut sebagai *marker*.

Setelah objek dikenali oleh kamera kemudian akan disimpan dalam bentuk (.patt) di *library Augmented Reality*, objek tersebut adalah sebagai *action* dalam pemanggilan objek 3D yang sebelumnya sudah dirancang dan kemudian dicocokkan dengan *marker*. Aplikasi *Augmented Reality* ini dapat menangkap objek *marker* yang telah tersimpan dalam *library*, dengan menyorotkan objek *marker* didepan kamera *handphone* maka akan dikenali secara otomatis apakah *marker* tersebut sudah tersimpan dalam *library* atau belum. Jika objek *marker* yang ditangkap sudah tersimpan dalam *library* maka akan muncul objek 3D sebagai hasil *output* nya, sebaliknya jika *marker* tidak dikenali oleh kamera maka tidak akan muncul objek 3D.

3.2. Analisa Kebutuhan Perangkat Keras

a. Perangkat keras yang dibutuhkan untuk membangun sistem ini adalah:

1. CPU dengan spesifikasi :
 - a. Processor intel® Core™ 2 Duo T6400.
 - b. Memory: 2GB DDR2 PC5300, up to 4GB.
 - c. Hard Drive: 500GB SATA.
 - d. Optical Drive: DVDRW±RW.
 - e. Chipset: Intel® 40GL Chipset.
 - f. Display: 14.1” WXGA BrightView TFT with 1280x800 Max Resolution.
2. *Smartphone Samsung Galaxy Tab 3.8* Android dengan spesifikasi:
 - a. Network: GSM 850/ 900/ 1800/ 1900/ HSDPA 850/ 900/ 1900/2100.
 - b. Tipe Layar: TFT capacitive touchscreen, 16M colors.
 - c. Dimensi: 800x1280 pixels 8.0 inch (189 ppi pixel density).
 - d. Fitur/ Audio: Vibration, MP3, WAV ringtones.
 - e. Memory: Internal 16Gb, Ram 1,5Gb.
 - f. Data: 4G, 3G,HSDPA, 21 Mbps, EDGE, GPRS.

- g. WLAN: Wi-Fi 802.11 a/b/g/n, dual-band, Wi-Fi Direct, DLNA, hotspot.
 - h. Bluetooth: v4.0, A2DP.
 - i. GPS: A-GPS, GLONASS.
 - j. USB: microUSB v2.0 (MHL TV-out), USB Host.
 - k. Kamera: primer 5 MP, 2560x1920 pixels, autofocus; Kamera Sekunder: 1.3 MP; 720p@30fps.
 - l. Baterai: Non-removable Li-Ion 4450 mAh battery; Up to 11 h talk time; Up to 96 h music play.
 - m. Operating System: Android OS, v4.2.2 (Jelly Bean).
 - n. CPU: Dual-core 1.5 GHz, Exynos 4212, Mali-400.
 - o. Browser: HTML5.
 - p. Messaging: SMS (theaded view), MMS, Email, Push Email, IM.
 - q. Java:No,MP4/DivX/Xvid/FLV/MKV/H.264 player- MP3/WAV/eAAC+/Flac player- Photo/video editor- Document viewer.
- b. Analisa kebutuhan perangkat lunak yang dibutuhkan untuk sistem ini adalah:
1. *Java Development*, untuk kompilasi kode program
 2. *Java Runtime Edition*, sebagai *platform* untuk menjalankan sistem
 3. *Eclipse*, sebagai perangkat lunak yang digunakan untuk membangun aplikasi *client* pada Android.
 4. Android SDK, sebagai pengembang aplikasi Android.
 5. *Android development tool*, sebagai *plugin* Android pada *eclipse*.

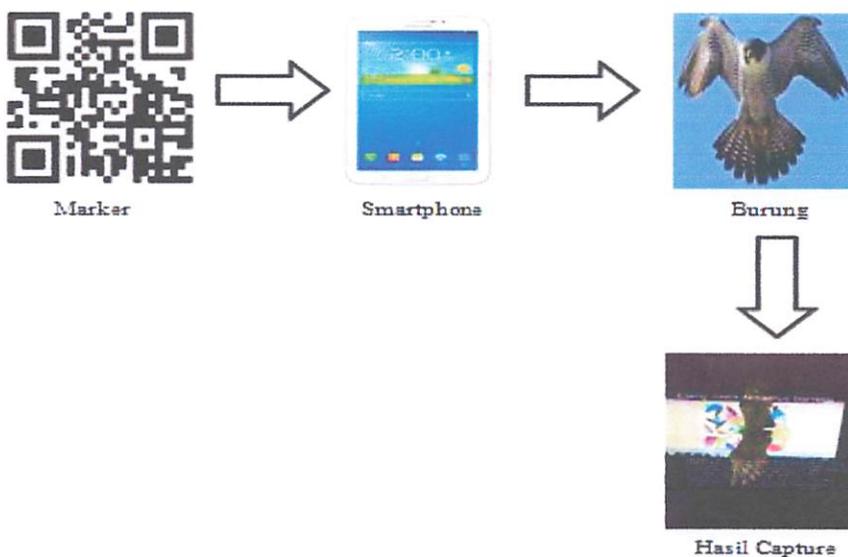
Android *virtual device*, sebagai *simulator* untuk menjalankan sistem Android.

3.3. Metode Pengumpulan Data

Langkah-langkah dalam penerapan *Augmented Reality* pada aplikasi mobile museum satwa berbasis *android* secara *virtual*, adalah sebagai berikut :

1. Melakukan proses tanya jawab dengan pihak museum.
2. Mencari bahan-bahan kepustakaan sebagai landasan teori yang ada hubungannya dengan permasalahan yang dijadikan obyek penelitian. Seperti literatur ataupun artikel-artikel yang ada kaitannya dengan obyek penelitian.
3. Melakukan analisis resiko yang mungkin terjadi, sehingga dapat meminimalisir resiko yang ada.
4. Melakukan perancangan sistem, dengan membuat desain 3D, *Flowcart*, desain Interface, dan desain *marker*.
5. Melakukan pengujian terhadap AR Museum dengan menggunakan handphone agar sesuai dengan kebutuhan terhadap dunia nyata.
6. Melakukan evaluasi terhadap aplikasi AR Museum.

3.4. Proses Alur Kerja Aplikasi Mobile Museum Satwa Virtual Berbasis Android



Gambar 3.1 Konsep Aplikasi Mobile Museum Satwa Virtual Berbasis Android

Proses informasi satwa museum berbasis AR diatas menjelaskan bahwa, *user* melakukan interaksi dengan cara mengarahkan kamrea ke pada *marker* AR

museum, yang memiliki *software library* ARToolkit. Interaksi yang terjadi kepada *user* adalah, *user* dapat melihat objek virtual satwa untuk mengetahui informasi satwa sebagai *user manual*. *User* dapat mengetahui informasi satwa sesuai dengan marker yang ditangkap oleh kamera.

3.5. Analisis Resiko

Analisis resiko yang dilakukan adalah, dengan menganalisis kemungkinan kesalahan teknis yang terjadi pada aplikasi AR Musium ini. Berikut hasil analisis resiko yang terjadi pada aplikasi informasi satwa musium berbasis AR ini:

1. Resiko objek 3D

Dalam pembuatan objek satwa AR sebagai *user manual* dalam bentuk 3D, tidak semua objek 3D dapat dirender oleh kamera. Hal ini disebabkan objek 3D yang terlalu rumit sehingga saat dirender oleh kamera, terkadang objek 3D tersebut hanya muncul setengahnya saja.

2. Resiko Marker

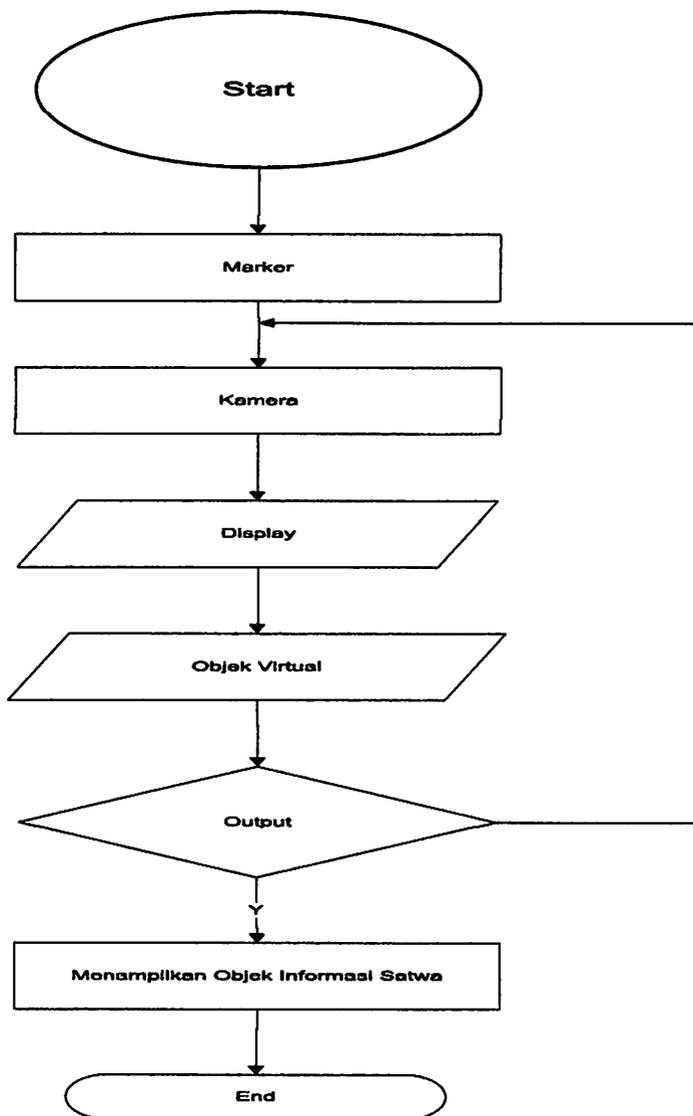
Dengan adanya banyak *marker* dalam AR Museum ini, harus adanya pembeda pada *marker* yang satu dengan yang lain, agar kamera *handphone* tidak salah dalam mendeteksi *marker*, sehingga objek yang harusnya dirender di *marker* A, maka objek tersebut dirender di *marker* A. Dan dalam ukuran *marker* tidak boleh terlalu besar agar kamera dapat mendeteksi *marker* dengan baik, *standart* ukuran *marker* agar mudah dideteksi oleh kamera *handphone*, cukup 631x 634 pixel.

3.6. Perancangan

Perancangan dalam penelitian ini adalah, penjelasan bagaimana sistem dalam aplikasi mobile museum satwa virtual berbasis *android* ini berkerja, yang digambarkan melalui *flow chart*. Berikut hasil dari perancangan:

3.6.1. Flowchart Aplikasi

Alur Aplikasi secara lengkap dapat digambarkan dengan flowchart pada gambar 3.2



Gambar 3.2 Flowchart Aplikasi Museum Satwa

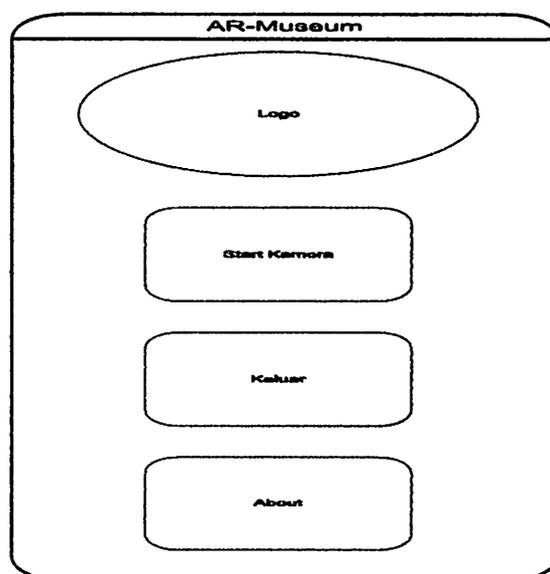
Berikut penjelasan *flow chart* aplikasi mobile museum satwa virtual berbasis android :

1. Sebelum melakukan simulasi *user* harus menyiapkan yang akan dipresentasikan sebagai *user manual*, dari objek satwa ini, mulai dari marker 1 sampai dengan semua *marker* agar sesuai dengan informasi satwa.
2. *User* mengarahkan *marker* satu persatu, mulai dari marker 1 sampai dengan semua *marker* secara bergantian, ke arah kamera *handphone* agar *marker* dapat terdeteksi sesuai dengan objek *virtual* yang telah ditentukan, sistem akan terus mendeteksi keberadaan *marker* untuk dioalah dan dimunculkan ke display sebagai hasil dari objek *virtual*.

3.6.2. Antar Muka Aplikasi

Didalam sub bab antar muka aplikasi ini akan dijelaskan tentang desain dari tampilan yang di gunakan didalam aplikasi.

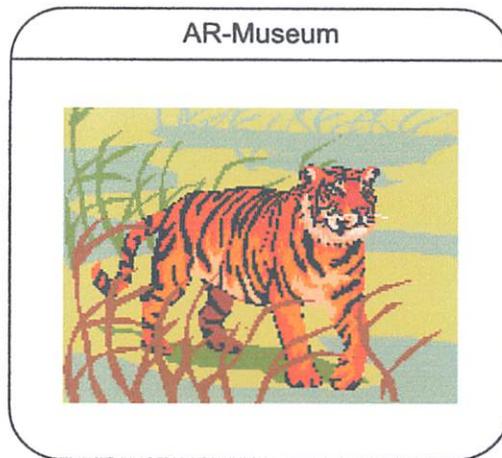
a. Antar Muka *Menu*



Gambar 3.3 Tampilan *Menu*

Pada gambar 3.3 menampilkan rancangan antarmuka *Menu* yang akan ditampilkan bagi pengguna.

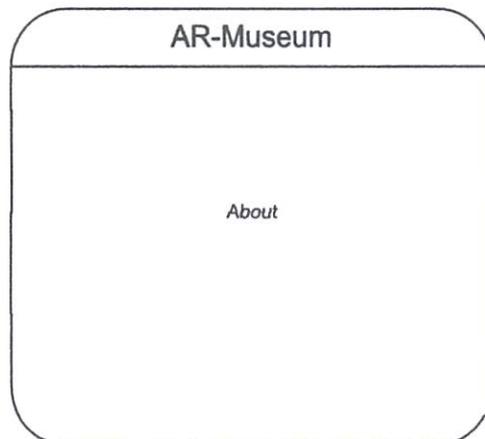
b. Antar Muka Kamera



Gambar 3.4 Antar Muka Kamera

Pada gambar 3.12 menampilkan rancangan antarmuka *kamera view*. Melalui fitur ini pengguna dapat melihat objek 3D dari hasil penangkapan marker. Selain iut juga informasi detail mengenai satwa hasil penangkapan marker.

c. Antar Muka List



Gambar 3.5 Antar Muka About

Pada gambar 3.5 menampilkan halaman tentang informasi dari aplikasi.

3.7. Konstruksi Sistem

Konstruksi sistem dalam tugas akhir ini adalah tahapan membangun informasi satwa museum *augmented reality* ini. Berikut hasil konstruksi sistem:

3.7.1. Penginstalan ARToolkit

ARToolkit adalah software library untuk membangun *augmented reality*. Cara penginstalan *ARToolkit* hanya, menyalin *ARToolkit* yang sudah di *download* di Internet. Adapun langkah-langkah instalasi *ARToolkit* diantaranya :

1. Buat folder AR di drive C.
2. Simpan *ARToolkit-2.72.1-bin-win32.zip* dan *glut-3.7.6-bin.zip* ke folder AR.
3. Ekstrak *ARToolkit-2.72.1-bin-win32.zip* ke drive C. Sehingga nanti ada folder *ARToolKit* di drive C.
4. Ekstrak *glut-3.7.6-bin.zip* pada folder AR sehingga ada folder *glut-3.7.6-bin* pada folder AR.
5. Copy file *glut32.dll* dari folder *glut-3.7.6-bin* ke folder *C:\ARToolKit\bin*.
6. Buat folder *C:\ARToolKit\include\GL*. Copy file *glut.h* dari folder *glut-3.7.6-bin* ke folder *C:\ARToolKit\include\GL*.
7. Copy file *glut32.lib* dari folder *glut-3.7.6-bin* ke folder *C:\ARToolKit\lib*. (sobana, 2010).

3.7.2. Perancangan Objek 3D

Pemodelan objek 3D dibagi dalam tiga model, yang merupakan, tiga objek terpenting, saat *user* menjalankan aplikasi mobile museum satwa berbasis AR. Objek-objek tersebut adalah objek 3D satwa museum, yang dibuat untuk tampilan awal satwa dan informasi dari satwa yang terdapat dalam museum, objek 3D tersebut yakni satwa : zebra, singa, harimau, buaya, dan lain-lain.

3.7.3. Perancangan Marker

Marker adalah, sebuah pola yang berbentuk kotak dan berbingkai hitam, yang digunakan untuk menampilkan objek 3D yang ada di *template memory* *ARToolkit*. Cara proses kerja *marker* adalah, kamera menyorot *marker*, kemudian kamera membaca pola yang ada didalam *marker* tersebut, bila pola terdeteksi

maka kamera mencocokkan pola *marker* yang disorot kamera dengan pola *marker* yang ada di *template memory* ArToolkit, selanjutnya bila pola cocok, maka objek dirender di pola *marker* yang disorot tersebut.

Dalam perancangan *marker*, digunakan dua *software* yaitu *software design grafis* bisa berupa *paint*, *adobe photoshop*, dll dan *mk_patt*. *Software design grafis* digunakan untuk membuat pola *marker*, dalam hal ini, *marker* dibuat dengan pola hitam-putih. Disini *marker* yang satu dengan yang lain diberi pembeda untuk lebih memudahkan sistem mendeteksi *marker*. *Software mk_patt* digunakan untuk mempol *marker*, kemudian *marker* yang sudah terpola olah *mk_patt* disimpan di dalam *template memory* ArToolkit, sehingga *marker* dapat dibaca oleh ArToolkit. Berikut adalah langkah-langkah dalam perancangan *marker*:

1. Membuat *Marker* dengan *Adobe Photoshop*. Berikut output gambar pola *marker* yang dihasilkan :
 - a. *Marker* objek 3D .



Gambar 3.6 Marker Objek 3D

BAB IV

IMPLEMENTASI

Bab ini menjelaskan tentang hasil yang diperoleh berdasarkan pengujian yang telah dilakukan. Tujuan dari dilakukannya pengujian adalah mengetahui apakah Aplikasi Mobile Museum Satwa Virtual Berbasis Android sudah berjalan dengan baik atau tidak. Berikut tahap-tahap pengujian yang dilakukan:

1. Implementasi sistem
2. Pengujian Aplikasi

4.1. Implementasi Perangkat Lunak

Berdasarkan hasil analisis dan perancangan sistem yang telah dilakukan, maka dilakukan implementasi/pengkodean ke dalam bentuk perangkat lunak. Pengkodean merupakan tahap menerjemahkan hasil perancangan perangkat lunak secara rinci ke dalam bahasa pemrograman. Implementasi dilakukan dengan menggunakan bahasa pemrograman java untuk mobile android, sehingga hasil dari aplikasi program ini dapat digunakan pada mobile device dengan system operasi android. System operasi yang digunakan minimal adalah android 4.0.

4.2. Interface Perangkat Lunak

Tampilan aplikasi mobile museum satwa virtual berbasis android yang dikembangkan dibuat sesederhana mungkin untuk memudahkan dalam pengopersian *software* aplikasi mobilie museum satwa virtual berbasis android. Berikut ini adalah tampilan menu utama dari *software* aplikasi mobile museum satwa virtual berbasis android.

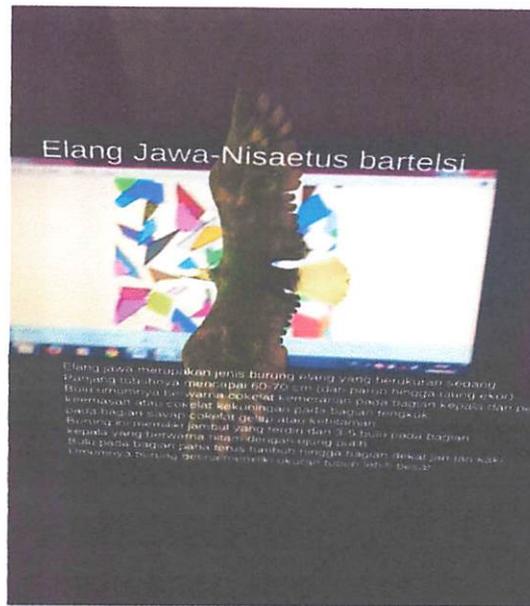


Gambar 4.1 Menu Utama Aplikasi

Dalam menu utama mobile museum satwa virtual berbasis AR, terdapat 3 menu pilihan yaitu Capture, About, dan Keluar. Berikut adalah uraian dari masing-masing option yang ada dalam *software* aplikasi mobile museum satwa virtual berbasis android.

4.3. Menu Capture

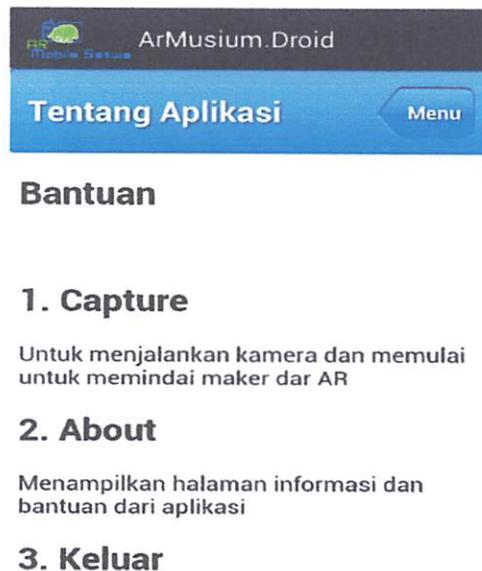
Capture digunakan untuk menjalankan kamera *handphone* untuk mengenali pola dari marker yang akan di proses. Hasil dari *capture* pada sistem ditunjukkan pada gambar 4.2 :



Gambar 4.2 Menu Capture Marker

4.4. Menu About

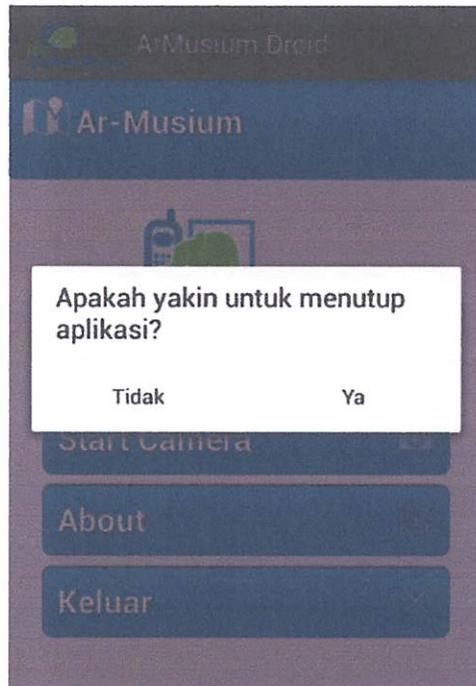
Pada menu ini berfungsi untuk menampilkan informasi dan bantuan mengenai aplikasi mobile museum satwa virtual berbasis android. Berikut adalah dari *menu* About pada aplikasi mobile museum satwa virtual berbasis android:



Gambar 4.3 Menu About

4.5. Menu Keluar

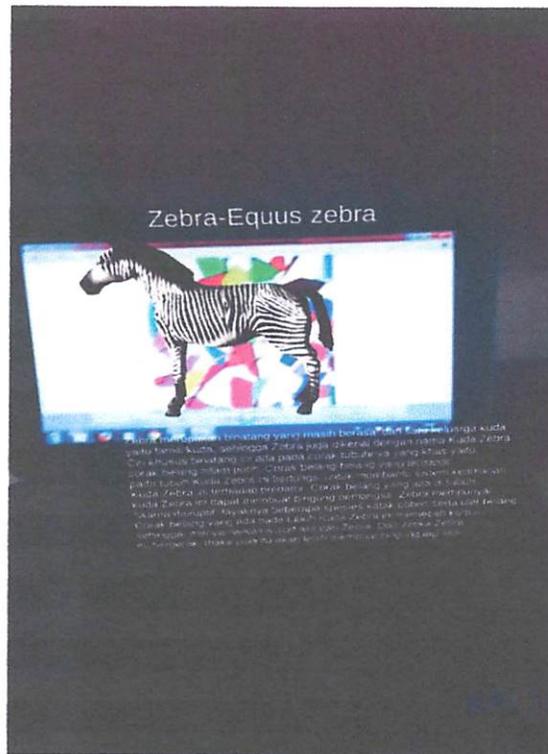
Pada menu ini berfungsi untuk mengakhiri aplikasi mobile museum satwa virtual berbasis android. Berikut adalah tampilan dari *menu* keluar pada aplikasi mobile museum satwa virtual berbasis android:



Gambar 4.4 Menu Keluar

4.6. Uji Sumber Marker

Pada menu deteksi aplikasi mobile museum satwa virtual berbasis android dilakukan proses pengujian terhadap marker pada setiap satwa. Dimana hal ini bertujuan untuk melakukan validasi pada marker yang telah dibuat untuk masing-masing satwa apakah berhasil atau tidak. Berikut adalah tampilan dari uji sumber marker:

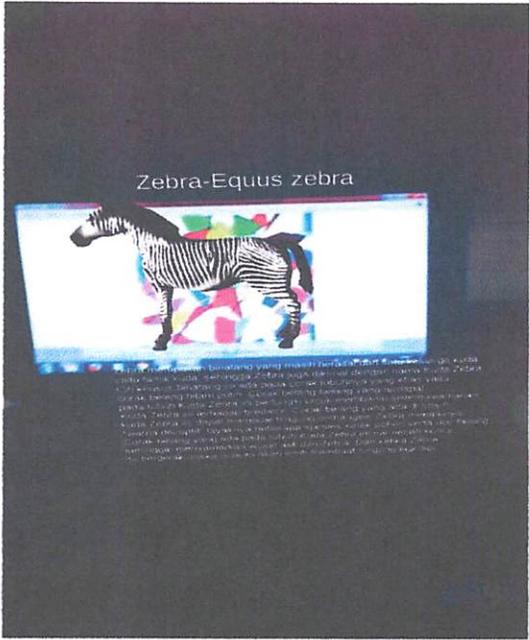
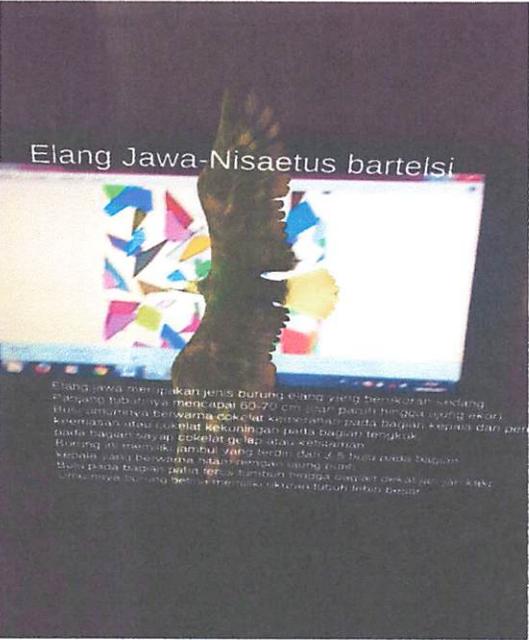


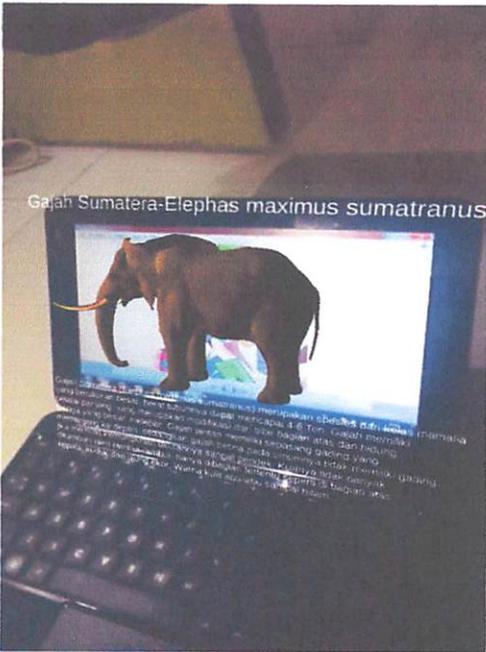
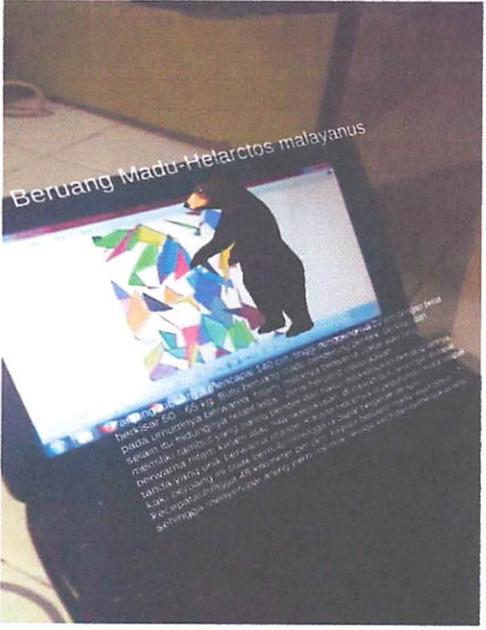
Gambar 4.5 Uji Sumber Marker

4.7. Pengujian Sistem dan Hasil

Setelah pengambilan langsung dari kamera *device mobile android* kemudian dilakukan pencocokan hasil informasi satwa yang disajikan pada masing-masing marker. Tahap pengujian selanjutnya adalah pengujian akurasi dari system. Pengujian ini bertujuan untuk memastikan bahwa perangkat lunak yang dikembangkan dapat memenuhi tujuan yang diinginkan dan memiliki tingkat kesalhan yang kecil dalam proses pengenalan marker dari aplikasi mobile museum satwa virtual berbasis android. Tabel hasil pengujian untuk akurasi bisa dilihat pada tabel 4.1.

Tabel 4.1 Hasil Pengujian Akurasi Sistem

No	Marker	Status
1		BENAR
2		BENAR

<p>3</p>		<p>BENAR</p>
<p>4</p>		<p>BENAR</p>

Dari hasil data pengujian akurasi sistem yang diperlihatkan pada Tabel 4.1 dapat di hitung nilai akurasi dengan menggunakan persamaan berikut:

$$\text{Akurasi (\%)} = (\text{Jumlah Benar}/\text{Total Data Uji}) \times 100\%$$

Sehingga dari persamaan tersebut diatas didapat hasil perhitungan sebagai berikut:

$$\text{Total Data} = 4$$

$$\text{Jumlah Benar} = 4$$

$$\text{Akurasi (\%)} = (4/4) \times 100\% = 1 \times 100\% = 100\%$$

Dengan hasil perhitungan diatas, diperoleh kesimpulan bahwa sistem memiliki nilai akurasi 100% dalam mengenali marker untuk setiap informasi satwa pada aplikasi mobile museum sawta berbasis android yang telah diujikan dan dengan demikian nilai kesalahan yang dimiliki sebesar 0%.

BAB V

PENUTUP

5.1. Kesimpulan

Setelah melakukan pembuatan dan perencanaan sistem kemudian dilakukan pengujian dan analisa, maka kesimpulannya yaitu sebagai berikut:

1. Aplikasi diterapkan pada smartphone android.
2. Dengan adanya aplikasi mobile museum satwa virtual berbasis android dapat memudahkan proses belajar mengenal satwa-satwa tanpa harus ke museum satwa.
3. Dari hasil pengujian dapat disimpulkan bahwa tingkat keberhasilan akurasi sistem selalu berhasil, hal tersebut dikarenakan marker yang dicoba dalam kondisi baik.
4. Hasil dari aplikasi ini hanya dapat menampilkan 4 hewan 3D, karena membuat hewan dalam bentuk 3D membutuhkan waktu yang tidak singkat.

5.2. Saran

Aplikasi mobile museum satwa virtual berbasis android ini dapat dikembangkan lebih jauh lagi karena dalam pembuatannya masih jauh dari sempurna dan masih banyak kekurangan. Adapun saran yang dapat dikemukakan agar aplikasi ini bisa berfungsi dengan optimal adalah :

1. Dapat dilakukan pengembangan dengan menggunakan metode lain.
2. Dapat dilakukan pengembangan untuk beberapa OS lainnya, seperti iOS dan Windows Phone.
3. Dan dapat ditambahkan beberapa hewan lagi untuk melengkapi kekurangan dari aplikasi yang saya buat.

DAFTAR PUSTAKA

- [1] Micheal Siregar, Ivan ST MT. *Membongkar Source Code Berbagai Aplikasi Android*, Penerbit Gava Media, Yogyakarta 2011.
- [2] Safaat, Nazaruddin H. *Pemrograman aplikasi mobile smartphone dan tablet Pc berbasis android*, Penerbit Informatika, Bandung 2012.
- [3] Mulyadi, ST., 2010, *Membuat Aplikasi untuk Android*. Multimedia Center Publishing, Yogyakarta.
- [4] Paul Deitel, Harvey Deitel, Abbey Deitel. *Android, How to Program*, Penerbit Deitel & Associates Inc, San Francisco 2013.
- [5] Vogella. 2014. "Source Code Android". <http://vogella.de.net>, diakses tanggal 04 juni 2015.

Lampiran

```
package me.atm.finder;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.io.UnsupportedEncodingException;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map.Entry;

import javax.net.ssl.HttpURLConnection;

import org.apache.http.HttpException;
import org.apache.http.util.EncodingUtils;

import me.atm.finder.MapsActivity.GettingLocation;
import me.atm.finder.R.id;

import android.app.Activity;
import android.app.AlertDialog;
import android.app.Dialog;
import android.app.ProgressDialog;
import android.base.org.sun.java.Utils;
import android.content.Context;
import android.content.DialogInterface;
```

```
import android.content.Intent;
import android.graphics.Bitmap;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.StrictMode;
import android.util.services.ApiParam;
import android.util.services.ApiServices;
import android.view.View;
import android.webkit.ConsoleMessage;
import android.webkit.WebChromeClient;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.AdapterView.OnItemClickListener;

public class ListDataActivity extends Activity implements LocationListener {
    //init var
    ApiService apiServices ;
    ListView lvData;
    Utils u ;
```

```

Stringbuilder FinalResult = new Stringbuilder();

boolean isFinish = false;

private LocationManager locationManager;

WebView webView ;

@Override

protected void onCreate(Bundle savedInstanceState) {
    // TODO Auto-generated method stub
    super.onCreate(savedInstanceState);
    setContentView(R.layout.list_data);
    u = new Utils(getApplicationContext(), this);
    lvData = (ListView) findViewById(id.lvData);
    //amil param
    Intent i = getIntent();
    ID = i.getStringExtra("ID");
    NAME = i.getStringExtra("NAME");
    setTitle(NAME);

    locationManager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);
    new GettingLocation().execute("");

    StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
    StrictMode.setThreadPolicy(policy);

    webView =(WebView) findViewById(R.id.webView1);
    webView.setWebChromeClient(new WebChromeClient() {
        public boolean onConsoleMessage(ConsoleMessage cmMsg)

```

```

        {
            if (cmmsg.message().startsWith("TAG"))
            {
                String searchResult =
cmmsg.message().substring(22).replace("</body>", "");
                FinalResult = new StringBuilder( searchResult);
                isFinish = true;
                System.out.println("Hasil : "+searchResult);
                return true;
            }

            return false;
        }
    });

```

```

webView.setWebViewClient(new WebViewClient() {
    @Override
    public void onPageStarted(WebView view, String url, Bitmap favicon) {
        // TODO Auto-generated method stub
        super.onPageStarted(view, url, favicon);
        isFinish = false;

        FinalResult= new StringBuilder("");
    }

    public void onPageFinished(WebView view, String address)
    {

view.loadUrl("javascript:console.log('TAG'+document.getElementsByTagName('html')[0
].innerHTML);");

```

```

    }
    @Override
    public void onReceivedError(WebView view, int errorCode,
        String description, String failingUrl) {
        // TODO Auto-generated method stub
        super.onReceivedError(view, errorCode, description,
failingUrl);

        FinalResult= new StringBuilder("-");
    }
});

```

```

        webView.getSettings().setLoadsImagesAutomatically(false);
webView.getSettings().setAppCacheEnabled(true);
webView.getSettings().setJavaScriptEnabled(true);

```

```

String postData = "CID="+u.getApiID()+"&" +
        "ID="+ID+"&" +
        "NAME="+NAME ;

```

```

webView.loadUrl(
"http://" + SplashScreenActivity.Uris + "api.php?type=get_tempat_by_jenis&&" +
(postData)); //,EncodingUtils.getBytes(postData, "BASE64");

```

```

}

```

```

@Override
public void onBackPressed() {
    // TODO Auto-generated method stub

    finish();
}

```

```

        super.onBackPressed();
    }

    boolean isFirst = true;

    class GettingLocation extends AsyncTask<String, Void, Void> {

private ProgressDialog Dialog = new ProgressDialog(ListDataActivity.this);
protected void onPreExecute() {
    //progress
    Dialog.setMessage(" Loading GPS Location .");
    Dialog.show();
    Dialog.setCancelable(true) ;
    locString = "";

    locationManager.requestLocationUpdates ( LocationManager.GPS_PROVIDER,
        1000, // 1 sec
        10, ListDataActivity.this);
}
protected Void doInBackground(String... urls) {
    while (locString.trim().equals("")) {
    }
    return null;
}
protected void onPostExecute(Void unused) {

    Dialog.dismiss();
}
}
}

```

```

String locString ="";

/***** Called after each 3 sec *****/

double lastLNG=0,lastLAT=0;

private String Kata;

public void onLocationChanged(Location location) {

    String str = "Lokasi Anda : \n-Latitude: "+location.getLatitude()+"\n-Longitude:
"+location.getLongitude();

    locString =location.getLatitude()+" "+location.getLongitude();

    if (isFirst){

        String lat2 = String.valueOf(location.getLatitude()).replace(".", ".");

        String lng2 =
String.valueOf(location.getLongitude()).replace(".", ".");

        isFirst = false;

        Toast.makeText(getApplicationContext(), str,
Toast.LENGTH_LONG).show();

        this.setPosition(location.getLatitude(), location.getLongitude());

        new GettingData().execute("");

    }

}

public double CalculationByDistance(double startLatitude, double startLongitude,
double endLatitude, double endLongitude ) {

    float [] dist = new float[1];

    Location.distanceBetween(startLatitude, startLongitude, endLatitude,
endLongitude, dist);

    System.out.println( " Dist => startLatitude : "+ startLatitude +" " +

```

```
        ", startLongitude : "+ startLongitude+", endLatitude :  
"+endLatitude+", endLongitude : "+endLongitude+", dist : " + dist[0]);
```

```
        return dist[0];
```

```
    }
```

```
    public void onProviderDisabled(String provider) {
```

```
        /***** Called when User off Gps *****/
```

```
        Toast.makeText(getBaseContext(), "Gps turned off",  
Toast.LENGTH_LONG).show();
```

```
    }
```

```
    public void onProviderEnabled(String provider) {
```

```
        /***** Called when User on Gps *****/
```

```
        Toast.makeText(getBaseContext(), "Gps turned on",  
Toast.LENGTH_LONG).show();
```

```
    }
```

```
    public void onStatusChanged(String provider, int status, Bundle extras) {
```

```
        // TODO Auto-generated method stub
```

```
    }
```

```
        public class GettingData extends AsyncTask<String, Void, Void> {
```

```
            private ProgressDialog Dialog = new ProgressDialog(ListDataActivity.this);
```

```
            protected void onPreExecute() {
```

```
                //progress
```

```
                Dialog.setMessage(" Loading data.");
```

```
                Dialog.show();
```

```
                Dialog.setCancelable(true) ;
```

```
                apiServices = new ApiServices(ListDataActivity.this, SplashScreenActivity.Url);
```

```
            }
```



```

    }else{
        u.Alert( "Koneksi dengan server gagal..");
        //    finish();
    }

}

}

ArrayList<HashMap<String, String>> dataListTempat ;
private String ID;
private String NAME;
double Curlat=0;
double Curlon=0;
public void setPosition(double lat, double lon) {
    Curlat = lat;
    Curlon = lon;
}

public String performPostCall(String requestURL,
HashMap<String, String> postDataParams) {

URL url;
String response = "";
try {
    url = new URL(requestURL);

    HttpURLConnection conn = (HttpURLConnection) url.openConnection();
    conn.setReadTimeout(15000);
    conn.setConnectTimeout(15000);
    conn.setRequestMethod("POST");

```

```

conn.setDoInput(true);
conn.setDoOutput(true);

OutputStream os = conn.getOutputStream();
BufferedWriter writer = new BufferedWriter(
    new OutputStreamWriter(os, "UTF-8"));
writer.write(getPostDataString(postDataParams));

writer.flush();
writer.close();
os.close();
int responseCode=conn.getResponseCode();

if (responseCode == HttpURLConnection.HTTP_OK) {
    String line;
    BufferedReader br=new BufferedReader(new
InputStreamReader(conn.getInputStream()));
    while ((line=br.readLine()) != null) {
        response+=line;
    }
}
else {
    response="";

    throw new HttpException(responseCode+"");
}
} catch (Exception e) {
    e.printStackTrace();
}

```

```

    }

    return response;
}

private String getPostDataString(HashMap<String, String> params) throws
UnsupportedEncodingException{
    StringBuilder result = new StringBuilder();
    boolean first = true;
    for(Entry<String, String> entry : params.entrySet()){
        if (first)
            first = false;
        else
            result.append("&");

        result.append(URLEncoder.encode(entry.getKey(), "UTF-8"));
        result.append("=");
        result.append(URLEncoder.encode(entry.getValue(), "UTF-8"));
    }

    return result.toString();
}

private boolean getData() {
    StringBuffer sb = new StringBuffer();

    String response="";
    try {
        //parameter yang dikirim ke api

```

```

        response = FinalResult.toString();
        System.out.println("Response :" + response);

        if (response.equals( "{\"items\":[]} ")) {

            u.Alert("Data tidak ditemukan..");

            //return false;
        }

        //parsing json dari api
        dataListTempat= apiServices.getResult(response, 9);

        lvData = (ListView) findViewById(id.lvData);

        ArrayList<HashMap<String, String>> TmpdataList = new
        ArrayList<HashMap<String, String>>();
        TmpdataList = new
        ArrayList<HashMap<String,String>>(dataListTempat);
        dataListTempat= new ArrayList<HashMap<String, String>>();
        ArrayList<Double> JarakPos = new ArrayList<Double>();
        for (int i = 0; i < TmpdataList.size(); i++) {
            if (Curlat!=0 ) {
                double d = Double.parseDouble(
                TmpdataList.get(i).get("E").replace(",","."));
                double c = Double.parseDouble(
                TmpdataList.get(i).get("F").replace(",","."));
                double jarak
                =CalculationByDistance(Curlat, Curlon, d, c);
                System.out.println("ATM
                :"+TmpdataList.get(i).get("C"));
                System.out.println("LAT :"+d);
                System.out.println("LNG :"+c);
            }
        }
    }
}

```

```

Meter ");
        System.out.println("Dist      :"+jarak+"

        JarakPos.add(jarak);
    }
    dataListTempat.add(TmpdataList.get(i)) ;
}

if (Curlat!=0) {
    double[] fixJarak = new double[JarakPos.size()];
    if (dataListTempat.size()!=0){
        double minDis = JarakPos.get(0);
        double[]      tmpJarak      =      new
double[JarakPos.size()];

        for (int i = 0; i < JarakPos.size(); i++) {
            tmpJarak[i] =JarakPos.get(i);
            fixJarak[i] =JarakPos.get(i);
        }
        double a=0;
        double b=0;
        double abTemp=0;
        HashMap<String, String> tmpDataList;
        for(int i=0;i<tmpJarak.length-1;i++)
        {
            for(int j=1;j<tmpJarak.length;j++)
            {
                a= tmpJarak[j] ;
                b= tmpJarak[j-1] ;
                if(a<b)
                {

```

```

        abTemp=tmpJarak[j];
        tmpDataList = dataListTempat.get(j);
        dataListTempat.set(j, dataListTempat.get(j-1));
        dataListTempat.set(j-1, tmpDataList);
        tmpJarak[j]=tmpJarak[j-1];
        tmpJarak[j-1]=abTemp;
    }
}

}

for (int i = 0; i < tmpJarak.length; i++) {
    fixJarak[i] = tmpJarak[i];
}

for (int i = 0; i < fixJarak.length; i++) {
    System.out.println("Fix   Jarak
:"+fixJarak[i] +" Meter ");
}

    TmpdataList           =           new
ArrayList<HashMap<String,String>>(dataListTempat);

    dataListTempat=           new
ArrayList<HashMap<String, String>>());

    for (int i = 0; i < fixJarak.length; i++) {
        if (fixJarak[i] <1000) {

dataListTempat.add(TmpdataList.get(i));
        }
    }

/*for (int i = 0; i < dataListTempat.size(); i++) {

```

```

String tmp =dataList.get(i).get("K");
tmp =tmp.replace("pcs", "pcs "+
String.format("##,###", tmpJarak[i]));

// dataList.get(i). = tmp;
}*/

}
}

if (dataListTempat.size()==0) {
    Toast.makeText(getApplicationContext(), "Data Tidak
ditemukan..", Toast.LENGTH_LONG);
}
} catch ( Exception e) {

    System.out.println(e.toString());
    return false;
}

return true;
}

// memilih salah satu list
protected void gotoItem(int position) {
    // TODO Auto-generated method stub
    //tampil detail
    /*Intent intent = new Intent(this, DetailDataActivity.class);
    intent.putExtra("ID", dataListTempat.get(position).get("A"));

```

```

        intent.putExtra("NAME",NAME);
        intent.putExtra("TITLE",dataListTempat.get(position).get("C"));
        intent.putExtra("DETAIL",dataListTempat.get(position).get("G"));
        intent.putExtra("ALAMAT",dataListTempat.get(position).get("D"));
        intent.putExtra("LAT",dataListTempat.get(position).get("E"));
        intent.putExtra("LNG",dataListTempat.get(position).get("F"));

        startActivity(intent);*/

        final Dialog dialog = new Dialog(this);
        dialog setContentView(R.layout.help);
        dialog.setTitle( dataListTempat.get(position).get("C"));
        dialog.setCancelable(true);

        final String mimeType = "text/html";
        final String encoding = "utf-8";

        final String html =
src="file:///android_asset/img/"+NAME.toLowerCase()+".png" width="100%"
height="80%" /></p>
        + "<b>Alamat : </b>" +
dataListTempat.get(position).get("D")+"<br>";

        WebView wv = (WebView) dialog.findViewById(R.id.webView1);
        wv.loadDataWithBaseURL("", html, mimeType, encoding, "");

//set up button
        Button button = (Button) dialog.findViewById(R.id.btnOk);
        button.setOnClickListener(new View.OnClickListener() {

@Override

        public void onClick(View v) {

```

```

        dialog.cancel();
    }
});
final String lat = dataListTempat.get(position).get("E").trim();
final String lng = dataListTempat.get(position).get("F").trim();
//set up button
Button button2 = (Button) dialog.findViewById(R.id.btnRute);
button2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(ListDataActivity.this, MapsActivity.class);

        intent.putExtra("LAT", lat);
        intent.putExtra("LNG", lng);
        startActivity(intent);
        dialog.cancel();
    }
});
dialog.show();
}

}

package me.atm.finder;
import java.util.ArrayList;
import me.atm.finder.R.id;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.Dialog;
import android.content.DialogInterface;

```

```
import android.content.Intent;
import android.graphics.Bitmap;
import android.os.Bundle;
import android.os.StrictMode;
import android.view.View;
import android.webkit.ConsoleMessage;
import android.webkit.WebChromeClient;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.Button;
import android.widget.ProgressBar;
import android.widget.TextView;
public class MainActivity extends Activity {
    //init view
    Button btn1,btn2,btn3,btn4,btn5,btn6;
    //init kode
    String[] ID_Tempat = new String[]{
    };
    //init jenis
    String[] Jenis_Tempat = new String[]{
    };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.menu);
        //init view
        btn1=(Button) findViewById(id.btn1);
        btn1.setOnClickListener(new View.OnClickListener() {
```

```
        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            show(0);
        }
    });

    btn2=(Button) findViewById(id.btn2);
    btn2.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            show(1);
        }
    });

    btn3=(Button) findViewById(id.btn4);
    btn3.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            show(3);
        }
    });

    btn4=(Button) findViewById(id.btn5);
    btn4.setOnClickListener(new View.OnClickListener() {
```

```

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            show(2);
        }
    });
}

protected void show(int i) {
    // TODO Auto-generated method stub
    //tampil list tempat
    Intent intent = new Intent(this, ListDataActivity.class);
    intent.putExtra("ID", ID_Tempat[i]);
    intent.putExtra("NAME",Jenis_Tempat [i]);
    startActivity(intent);
}

@Override
public void onBackPressed() {
    // TODO Auto-generated method stub
    close();
}

public void close(){
    //konfirmasi close
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setMessage("Apakah yakin untuk menutup aplikasi?")
        .setCancelable(false)
        .setPositiveButton("Ya",new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog,
                int id) {

```

```

        finish();
    }
})
.setNegativeButton("Tidak",new DialogInterface.OnClickListener()
{
    public void onClick(DialogInterface dialog,
        int id) {
        dialog.cancel();
    }
}
).show();
}

}

package me.atm.finder;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.List;
import java.util.Locale;
import org.apache.http.HttpRequest;
import org.apache.http.HttpRequestFactory;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import android.annotation.SuppressLint;
import android.annotation.TargetApi;

```

```
import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.base.org.sun.java.Utils;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.location.GpsStatus.Listener;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Build;
import android.os.Bundle;
import android.os.StrictMode;
import android.speech.tts.TextToSpeech;
import android.speech.tts.TextToSpeech.OnInitListener;
import android.util.Log;
import android.utils.services.ApiServices;
import android.view.View;
import android.webkit.WebChromeClient;
import android.webkit.WebView;
import android.webkit.GeolocationPermissions.Callback;
import android.widget.Button;
import android.widget.ListView;
import android.widget.Toast;

@SuppressLint("NewApi")
@TargetApi(Build.VERSION_CODES.GINGERBREAD)
```

```

public class MapsActivity extends Activity implements LocationListener {

    private LocationManager locationManager;

    private String LAT;
    private String LNG;
    WebView webView;

    @SuppressWarnings("SetJavaScriptEnabled")
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Utils.setFullScreen(this);
        setContentView(R.layout.infoview);
        Intent i = getIntent();
        LAT = i.getStringExtra("LAT");
        LNG = i.getStringExtra("LNG");
        System.out.println("MapsActivity.onCreate() : LAT : "+ LAT);
        System.out.println("MapsActivity.onCreate() : LNG : "+ LNG);
        webView = (WebView) findViewById(R.id.mapview);
        webView.getSettings().setJavaScriptEnabled(true);
        webView.setWebChromeClient(new WebChromeClient(){
            @Override
            public void onGeolocationPermissionsShowPrompt(final String
origin, final Callback callback) {
                final boolean remember = false;
                AlertDialog.Builder builder = new
AlertDialog.Builder(MapsActivity.this);
                builder.setTitle("Acces Permission..");
                builder.setMessage("Allow your application to access your
location? ")

```

```

        .setCancelable(true).setPositiveButton("Yes",
DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                callback.invoke(origin, true, remember);
            }
        }).setNegativeButton("No", new DialogInterface.OnClickListener()
{
            public void onClick(DialogInterface dialog, int id) {

                callback.invoke(origin, false, remember);
            }
        });
        AlertDialog alert = builder.create();
        alert.show();
    }
});

webView.getSettings().setGeolocationDatabasePath("/data/data/me.atm.finder");

        locationManager =
getSystemService(Context.LOCATION_SERVICE);
        new GettingLocation().execute("");
    }

@Override
public void onBackPressed() {
    // TODO Auto-generated method stub

    finish();
    super.onBackPressed();
}

```

```

    boolean isFirst = true;

    class GettingLocation extends AsyncTask<String, Void, Void> {
private ProgressDialog Dialog = new ProgressDialog(MapsActivity.this);
protected void onPreExecute() {
    //progress
    Dialog.setMessage(" Loading GPS Location .");
    Dialog.show();
    Dialog.setCancelable(true) ;
    locString = "";
    locationManager.requestLocationUpdates( LocationManager.GPS_PROVIDER,
        100, // 10 sec
        10, MapsActivity.this);
}
protected Void doInBackground(String... urls) {
    while (locString.equals("")){
    }
    return null;
}
protected void onPostExecute(Void unused) {
    Dialog.dismiss();
}
}

    class GettingNavigasi extends AsyncTask<String, Void, Void> {

private ProgressDialog Dialog = new ProgressDialog(MapsActivity.this);
protected void onPreExecute() {
    //progress
    Dialog.setMessage(" Loading data .");
    Dialog.show();
    Dialog.setCancelable(true) ;

```

```

    }
    protected Void doInBackground(String... urls) {
        return null;
    }
    protected void onPostExecute(Void unused) {
        Dialog.dismiss();
    }
}

String locString ="";

    /******* Called after each 3 sec *****/

    double lastLNG=0,lastLAT=0;

    private String Kata;

    public void onLocationChanged(Location location) {

        String str = "Lokasi Anda : \n-Latitude: "+location.getLatitude()+"\n-Longitude:
"+location.getLongitude();

        locString =location.getLatitude()+" "+location.getLongitude();

        if (isFirst){

            String lat2 = String.valueOf(location.getLatitude()).replace(".", ".");

            String lng2 =
String.valueOf(location.getLongitude()).replace(".", ".");

            /*LNG = lng2 ;

            LAT = lat2 ;*/

            isFirst = false;

            webView.loadUrl("file:///android_asset/wgx/index.hmx?lat2="+LAT.replace(".",
".")+ "&&lng2="+LNG.replace(".", ".")+ "&&lat="+lat2+"&&lng="+lng2);

            Toast.makeText(getApplicationContext(), str,
Toast.LENGTH_LONG).show();

        }

```

```

    }
    public void onProviderDisabled(String provider) {

        /******* Called when User off Gps *****/

        Toast.makeText(getApplicationContext(), "Gps turned off ",
        Toast.LENGTH_LONG).show();
    }

    public void onProviderEnabled(String provider) {

        /******* Called when User on Gps *****/

        Toast.makeText(getApplicationContext(), "Gps turned on ",
        Toast.LENGTH_LONG).show();
    }

    public void onStatusChanged(String provider, int status, Bundle extras) {
        // TODO Auto-generated method stub
    }
}

package me.atm.finder;

import java.io.File;

import android.app.Activity;

import android.content.Intent;

import android.os.Bundle;

import android.widget.ProgressBar;

public class SplashScreenActivity extends Activity {

    ProgressBar pbMain;

    //init alamat web services

//localhost

    //public static final String Urls = "10.0.2.2/atmadmin/inc/";

    //public static final String MainUrls = "10.0.2.2/atmadmin/";

//wifi

```

```

/*      public static final String Urls = "192.168.100.1/atmadmin/inc/";
        public static final String MainUrls = "192.168.100.1/atmadmin/";*/

//online

public static final String Urls = "apihosting.000space.com/atmadmin/inc/";
public static final String MainUrls = "apihosting.000space.com/atmadmin/";

@Override

public void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);
    setContentView(R.layout.splash);
    setTitle("Selamat Datang..");

    //ini view

    final int welcomeScreenDisplay =100;
    pbMain = (ProgressBar) findViewById(R.id.progressBar1 );
    pbMain.setMax(welcomeScreenDisplay);

    Thread threads = new Thread() {

        int wait = 0;

        @Override

        public void run() {

            try {

                super.run();

                //delay

                while (wait < welcomeScreenDisplay) {

                    sleep(30);

                    wait += 1;

                    pbMain.setProgress(wait);

                }

            } catch (Exception e) {

                System.out.println("Error:" + e);

```

```
        } finally {  
  
            newIntens();  
        }  
    }  
};  
threads.start();  
}  
protected void newIntens() {  
    // TODO Auto-generated method stub  
    //tampil menu utama  
    startActivity(new Intent( this,MainActivity.class ));  
    finish();  
}  
}
```



PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama : **GUNARDI AFRIDITA**
Nim : **08.12.517**
Jurusan : **Teknik Elektro**
Konsentrasi : **Teknik Komputer S-1**
Masa Bimbingan : **Semester Genap 2014-2015**
Judul : **APLIKASI MOBIEL MUSEUM SATWA VIRTUAL
BERBASIS ANDROID**

Dipertahankan dihadapan Tim Penguji Skripsi Jenjang Program Strata Satu (S-1)

Pada Hari : Selasa
Tanggal : 18 Agustus 2015
Dengan Nilai : 72,7 (B+)

Panitia Ujian Skripsi:

Ketua Majelis Penguji

M. Ibrahim Ashari, ST, MT
NIP.P. 1030100358

Sekretaris Majelis Penguji

Dr. Eng. I Komang Somawirata, ST, MT
NIP. Y. 1030100361

Anggota Penguji :

Dosen Penguji I

Dr. F. Yudi Limpraptono, ST, MT
NIP. Y. 1039500274

Dosen Penguji II

Yuli Wahyuni, ST, MT
NIP. P.1031200456



FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan Ujian Skripsi Jenjang Strata 1 Jurusan Teknik Elektro Konsentrasi Teknik Komputer, maka perlu adanya perbaikan skripsi untuk mahasiswa :

Nama : **GUNARDI AFRIDITA**
Nim : **08.12.517**
Jurusan : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer**
Masa Bimbingan : **Semester Genap 2014-2015**
Judul Skripsi : **APLIKASI MOBILE MUSEUM SATWA VIRTUAL BERBASIS ANDROID**

No	Penguji	Tanggal	Uraian	Paraf
1.	Penguji I	18 Agustus 2015	1. Aturan penulisan acuan tentang BAB II tulisan yang di ambil dari referensi dan di cantumkan kutipan. 2. Detail flowchart software.	
2	Penguji II	18 Agustus 2015	1. Perbaiki dengan revisi total	

Disetujui:

Penguji I

Dr. F. Yudi Limpraptono, ST, MT
NIP. Y. 1039500274

Penguji II

Yuli Wahyuni, ST, MT
NIP. P. 1031200456

Mengetahui:

Dosen Pembimbing I

Dr. Eng. Aryanto Soetedjo, ST, MT
NIP. Y. 1030800417

Dosen Pembimbing II

Ir. Ni Putu Agustini, MT
NIP. Y. 1030100371



INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
Jl. Raya Karangrejo, Km. 2 MALANG

Formulir Perbaikan Ujian Skripsi

Dalam Pelaksanaan Ujian Skripsi Jenjang Strata 1 Jurusan Teknik Elektro Konsentrasi T.Energi Listrik, /
T. Elektronika, /T. Komputer, / T.Telekomunikasi, Maka Perlu Adanya Perbaikan Skripsi Untuk Mahasiswa:

Nama : *BUMUDI AFRIDITA*
NIM : *0812517*
Perbaikan Meliputi :

- ① *Aturan penulisan acuan, kutipan dan sitasi
tulisan yg diambil dari referensi lain,
di cantumkan kutipan.*
- ② *Perbaikan Flow chart ~~per~~ software.*

Malang, *19-8-*.....20*19*

[Signature]
.....