

BAB II

LANDASAN TEORI

2.1 *Augmented Reality*

2.1.1 *Pengertian Augmented Reality*

Augmented reality merupakan teknologi yang menggabungkan benda maya dua dimensi dan ataupun tiga dimensi ke dalam sebuah lingkungan nyata, lalu memproyeksikan benda-benda maya tersebut secara *real-time*. Tiga dimensi biasa disingkat 3D atau disebut ruang dari benda yang memiliki panjang, lebar, dan tinggi. Benda-benda maya menampilkan informasi berupa label maupun obyek *virtual* yang hanya dapat dilihat dengan kamera *handphone* maupun dengan komputer. Sistem dalam *Augmented reality* bekerja dengan menganalisa secara *real-time* obyek yang ditangkap dalam kamera.

Azuma (1997) mendefinisikan *Augmented reality* sebagai penggabungan benda-benda nyata dan maya di lingkungan nyata, berjalan secara interaktif dalam waktu nyata, dan terdapat integrasi antar benda dalam tiga dimensi, yaitu benda maya terintegrasi dalam dunia nyata. Penggabungan benda nyata dan maya dimungkinkan dengan teknologi tampilan yang sesuai, interaktivitas dimungkinkan melalui perangkat-perangkat *input* tertentu, dan integrasi yang baik memerlukan penjejukan yang efektif.

Menurut Bimber dan Raskar (2005), *Augmented Reality* berarti mengintegrasikan informasi sintetis ke dalam lingkungan nyata.

Menurut penjelasan Haller, Billinghurst, dan Thomas (2007), riset *Augmented Reality* bertujuan untuk mengembangkan teknologi yang memperbolehkan penggabungan secara *real-time* terhadap *digital content* yang dibuat oleh komputer dengan dunia nyata. Tidak seperti teknologi *Virtual Reality* (VR) yang membawa user sepenuhnya ke dalam lingkungan buatan, *Augmented Reality* memperbolehkan pengguna melihat objek maya tiga dimensi yang diproyeksikan terhadap dunia nyata.

2.1.2 Sejarah *Augmented Reality*

Menurut Matt Bower, Cathie Howe, Nerida McCredie, Austin Robinson and David Grover pada *Educational Media International* (2014). Sejarah tentang *Augmented reality* dimulai dari tahun 1957-1962, ketika seorang penemu yang bernama Morton Heilig, seorang sinematografer, menciptakan dan memapatenkan sebuah simulator yang disebut Sensorama dengan visual, getaran dan bau. Pada tahun 1966, Ivan Sutherland menemukan *head-mounted display* yang dia akui 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 Lanier memperkenalkan *Virtual Reality* dan menciptakan bisnis komersial pertama kali di dunia maya, Tahun 1992 mengembangkan *Augmented 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 Armstrong Labs, dan menunjukkan manfaatnya pada manusia, dan pada tahun 1992 juga, Steven Feiner,

Blair MacIntyre dan dorée Seligmann, memperkenalkan untuk pertama kalinya Major Paper untuk perkembangan *Prototype AR*.

Pada tahun 1999, Hirokazu Kato, mengembangkan ArToolkit di HITLab dan didemonstrasikan di SIGGRAPH, pada tahun 2000, Bruce.H.Thomas, mengembangkan ARQuake, sebuah Mobile Game AR yang ditunjukan di *International Symposium on Wearable Computers*.

Pada tahun 2008, Wikitude AR Travel Guide, memperkenalkan Android G1 Telephone yang berteknologi AR. tahun 2009, Saqoosha memperkenalkan FLARToolkit yang merupakan perkembangan dari ArToolkit. FLARToolkit memungkinkan kita memasang teknologi AR di sebuah website, karena output yang dihasilkan FLARToolkit berbentuk Flash. Ditahun yang sama, Wikitude Drive meluncurkan sistem navigasi berteknologi AR di Platform Android. Tahun 2010, Acrossair menggunakan teknologi AR pada I-Phone 3GS.

2.1.2.1 Perbedaan antara *Augmented Reality* dan *Virtual Reality*

Menurut Jun Yeon Ma dan Jong Soo Choi (2007). Perbedaan Augmented Reality dengan Virtual Reality adalah:

Tabel 2.1 Tabel Perbedaan *Augmented Reality* dengan *Virtual Reality*

Augmented Reality	Virtual Reality
<ul style="list-style-type: none"> • AR menambahkan atau menggabungkan dunia nyata dengan objek virtual • <i>User</i> bisa berinteraksi dengan dunia nyata, pada saat yang sama dapat melihat dunia nyata dan virtual dan bisa 	<ul style="list-style-type: none"> • VR menggantikan dunia nyata dengan dunia virtual secara total • <i>User</i> masuk ke dunia virtual, putus dari dunia nyata, secara hipotesis, user tidak bisa membedakan dunia nyata dengan

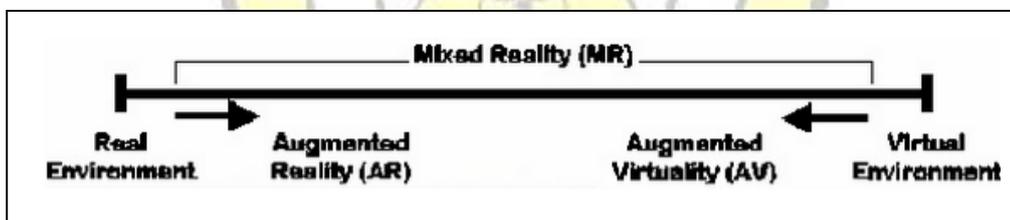
membedakannya <ul style="list-style-type: none"> • Membutuhkan mekanisme untuk menggabungkan dunia nyata dan virtual • AR menggunakan <i>Handheld Display smartphone / wearable device</i> 	dunia virtual <ul style="list-style-type: none"> • Membutuhkan mekanisme untuk mengisi seluruh dunia virtual untuk <i>user</i> • VR menggunakan <i>Head Mounted Display (HMD)</i>
--	---



Gambar 2.1 *Virtual Reality* (NASA, 2013)



Gambar 2.2 *Augmented Reality* (Vuforia Developer Portal, 2011)



Gambar 2.3 *Virtuality Continuum* oleh Milgram dan Kishino (1994)

Paul Milgram and Fumio Kishino merumuskan kerangka kemungkinan penggabungan dan peleburan dunia nyata dan dunia maya yang disebut (Milgram's Reality-Virtuality Continuum) pada tahun 1994. Dalam sisi yang paling kiri adalah lingkungan nyata yang hanya berisi benda nyata, dan sisi paling kanan adalah lingkungan maya yang berisi benda maya (Milgram, 1994).

2.1.3 Komponen *Augmented Reality*

Augmented Reality pada dasarnya adalah sebuah konsep yang mencitrakan gambar tiga dimensi yang seolah nyata. Proses ini bisa diperinci menjadi beberapa proses dan komponen. Untuk mencitrakan gambar tiga dimensi tersebut, sistem *Augmented Reality* terlebih dahulu harus melakukan penglihatan atau *vision* terhadap lingkungan yang padanya akan dicitrakan objek *virtual*. Kemudian, dilakukanlah proses *tracking* terhadap objek spesifik yang menentukan letak citraan objek *virtual* tersebut. Kemudian, objek tersebut akan dikenali, atau dianalisis. Setelah dikenali dan dianalisis posisi dan orientasinya, maka sistem akan melakukan proses pencitraan objek tersebut, dan akan tampak pada perlengkapan *display*. Menurut Silva, Oliveira, dan Giraldi (*Introduction to Augmented Reality*), komponen penting yang harus ada adalah:

a. *Scene Generator*

Scene Generator adalah komponen yang bertugas untuk melakukan *rendering* citra yang ditangkap oleh kamera. Objek *virtual* akan di tangkap kemudian diolah sehingga dapat kemudian objek tersebut dapat ditampilkan.

b. *Tracking System*

Tracking system merupakan komponen yang terpenting dalam *Augmented reality*. Dalam proses *tracking* dilakukan sebuah pendeteksian pola objek *virtual* dengan objek nyata sehingga sinkron diantara keduanya dalam artian proyeksi *virtual* dengan proyeksi nyata harus sama atau mendekati sama sehingga mempengaruhi validitas hasil yang akan didapatkan.

c. Display

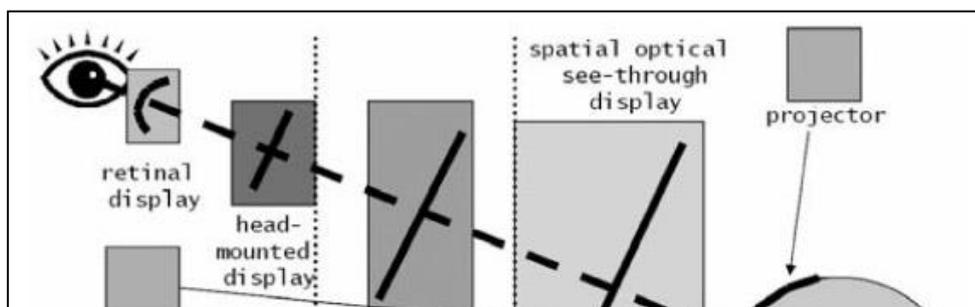
Dalam pembangunan sebuah sistem yang berbasis AR dimana sistem tersebut menggabungkan antara dunia virtual dan dunia nyata ada beberapa parameter mendasar yang perlu diperhatikan yaitu optik dan teknologi video. Keduanya mempunyai keterkaitan yang tergantung pada faktor resolusi, fleksibilitas, titik pandang, tracking area. Ada batasan-batasan dalam pengembangan teknologi Augmented reality dalam hal proses menampilkan objek. Diantaranya adalah harus ada batasan pencahayaan, resolusi layar, dan perbedaan pencahayaan citra antara citra virtual dan nyata.

d. AR Devices

Ada beberapa tipe media yang dapat digunakan untuk menampilkan objek berbasis Augmented reality yaitu dengan menggunakan optik, sistem retina virtual, video penampil, monitor berbasis AR dan proyektor berbasis AR.

2.1.4 Teknik Display Augmented Reality

Sistem *display* AR merupakan sistem manipulasi citra yang menggunakan seperangkat optik, elektronik, dan komponen mekanik untuk membentuk citra dalam jalur optik antara mata pengamat dan objek fisik yang akan digabungkan dengan teknik AR. Bergantung kepada optik yang digunakan, citra bisa dibentuk pada sebuah benda datar atau suatu bentuk permukaan yang kompleks (tidak datar) (Bimber, 2005). Gambar 2.4 mengilustrasikan kemungkinan citra akan dibentuk untuk mendukung AR, peletakan *display* bergantung dari pandangan pengguna dan objek, dan tipe citra seperti apa yang akan dihasilkan (*planar* atau *curved*).



Gambar 2.4 Pembentukan citra untuk *display augmented reality* (Bimber, 2005)

Secara garis besarnya ada tiga teknik display AR (Bimber, 2005), yaitu sebagai berikut:

1. *Head-Attached Display*
2. *Handheld Display*
3. *Spatial Display*

2.1.4.1 *Head-Attached Display*

Head-Attached Display merupakan teknik *display* yang mengharuskan penggunaannya untuk memakai sistem ini di kepala pengguna. Berdasarkan teknik citra yang terbentuk, *Head-Attached Display* terbagi tiga, yaitu sebagai berikut:

- *Head-Mounted Display.*
- *Head-Mounted Projectors.*

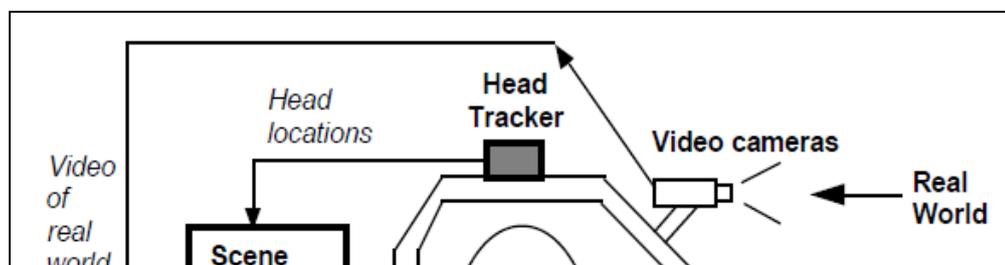
- *Virtual Retina Display.*

Kelebihan teknik display *Head-Attached Display* ini adalah lebih nyaman kepengguna, karena citra yang terbentuk mengikuti sudut pandang pengguna.

2.1.4.1.1 *Head-Mounted Display*

Head-Mounted Display (HMD) menggabungkan citra dari objek *virtual* dan objek nyata dan menampilkannya langsung ke mata pengguna melalui suatu alat yang dipasang di kepala pengguna. Terdapat dua tipe utama perangkat HMD yang digunakan dalam aplikasi realitas tertambah, yaitu *optical-see-through* HMD dan *video see-through* HMD. Keduanya digunakan untuk berbagai jenis pekerjaan dan memiliki keuntungan dan kerugian masing-masing. Dengan *optical-see-through* HMD, lingkungan nyata dilihat melalui cermin semi transparan yang diletakkan di depan mata pengguna. Cermin tersebut juga digunakan untuk merefleksikan citra yang dibentuk oleh komputer ke mata pengguna, menggabungkan lingkungan nyata dan *virtual*. Dengan *video see-through* HMD, lingkungan nyata direkam menggunakan dua kamera video yang terintegrasi ke alat, seperti gambar 2.8, dan citra yang dibentuk komputer digabung dengan video tadi untuk merepresentasikan lingkungan yang akan dilihat pengguna (Rolland, Holloway and Fuchs, 1994).

- *Video-see-through Head-Mounted Display*



Gambar 2.5 Diagram *Opaque* HMD (Azuma 1997)

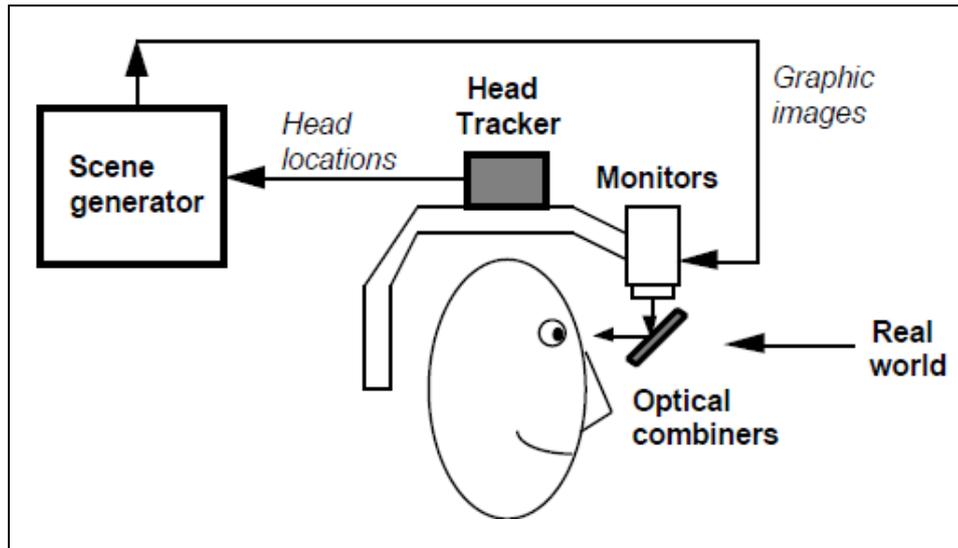
Video see-through HMD bekerja dengan menggabungkan sebuah *closed-view* HMD dengan satu atau dua *head-mounted* kamera video, melalui kamera video tersebut pengguna melihat ke lingkungan nyata. Video dari kamera dikombinasikan dengan citra yang dibuat oleh *scene generator*, dunia nyata dan virtual digabungkan. Hasilnya dikirimkan ke monitor yang terletak di depan mata pengguna. Gambar 2.5 menunjukkan konsep dari *video see-through* HMD, gambar 2.6 adalah contoh *video see-through* HMD, dengan dua video terintegrasi di bagian atas *Helm* (Azuma 1997).



Gambar 2.6 Contoh *Opaque HMD*, Courtesy Jannick Rolland, Frank Biocca, dan UNC Chapel Hill *Dept. of Computer Science*. Gambar dari Alex Treml. (Azuma 1997)

- ***Optical see-through Head-Mounted Display***

Tidak seperti penggunaan *video see-through HMD*, *optical see-through HMD* menyerap cahaya dari lingkungan luar, sehingga memungkinkan pengguna untuk secara langsung mengamati dunia nyata dengan mata (gambar 2.7). Selain itu, sebuah sistem cermin yang diletakkan didepan mata pengguna memantulkan cahaya dari pencitraan grafis yang dihasilkan komputer. Pencitraan yang dihasilkan merupakan gabungan optis dari pandangan atas dunia nyata dengan pencitraan grafis (Azuma 1997).



Gambar 2.7 Diagram *see-through* HMD (Azuma 1997)

2.1.4.1.2 *Head-Mounted Projectors*

Head-Mounted Projectors menggunakan proyektor atau panel LCD kecil dan mempunyai cahaya sendiri untuk menampilkan citra langsung ke lingkungan nyata (Bimber 2005). Seperti yang ditunjukkan oleh gambar 2.9



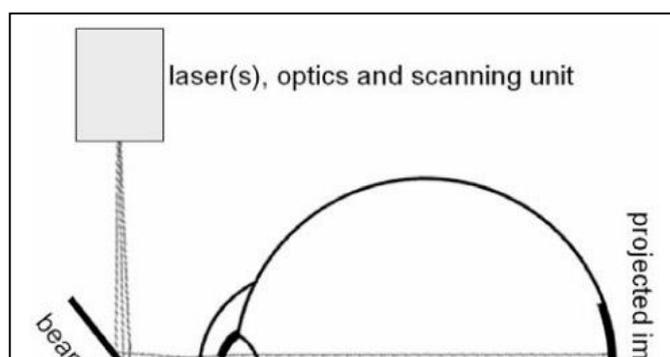
Gambar 2.8 Contoh *see-through* HMD, dibuat oleh Hughes Electronics (Azuma 1997)



Gambar 2.9 Ilustrasi penggunaan dua jenis perangkat HMPD yang digunakan untuk menampilkan data dan informasi tambahan. Gambar dari Institute fml (Günthner, 2009)

2.1.4.1.3 *Virtual Retina Display*

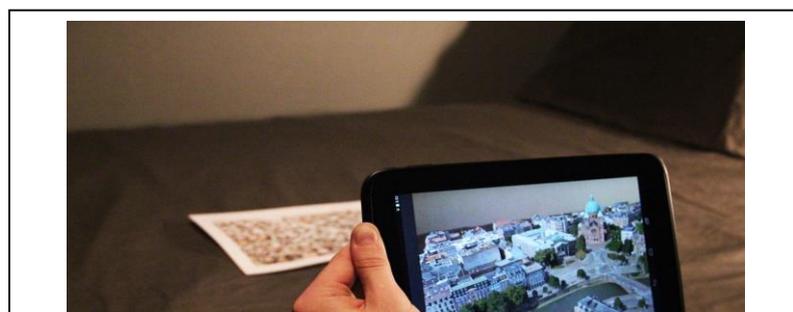
Virtual retina display (VRD), atau disebut juga dengan *retinal scanning display* (RSD), memproyeksikan cahaya langsung kepada retina mata pengguna (Haller, 2007). VRD dapat menampilkan proyeksi citra yang penuh dan juga tembus pandang tergantung pada intensitas cahaya yang dikeluarkan, sehingga pengguna dapat menggabungkan realitas nyata dengan citra yang diproyeksikan melalui sistem penglihatannya. VRD dapat menampilkan jarak pandang yang lebih luas daripada HMD dengan citra beresolusi tinggi (Jacko, 2010). Keuntungan lain VRD adalah konstruksinya yang kecil dan ringan. Namun, VRD yang ada kini masih merupakan *prototype* yang masih terdapat dalam tahap perkembangan, sehingga masih belum dapat menggantikan HMD yang masih dominan digunakan dalam bidang AR. Gambaran sederhana VRD ini dapat dilihat pada gambar 2.10.



Gambar 2.10 Diagram sederhana *virtual retina display*
(Bimber dan Raskar, 2005)

2.1.4.2 *Handheld Display*

Teknik ini menggunakan alat dengan *display* yang dengan mudah dapat digenggam pengguna (Tablet PC, PDA dan telepon genggam) seperti yang ditunjukkan pada gambar 2.11. Sensor dapat berupa GPS, kompas digital ataupun kamera yang ada pada *handheld* tersebut. Semua penerapan AR pada perangkat genggam menggunakan kamera untuk menggabungkan citra digital dengan lingkungan nyata, *Handheld AR* sangat menjanjikan untuk tujuan komersial. Dua kelebihan utama dari *Handheld AR* adalah mobilitas perangkat yang mudah dan salah satu perangkat genggam yang banyak digunakan (telepon genggam) telah banyak dilengkapi kamera.



Gambar 2.11 Contoh *Augmented Reality* dengan *handphone*. Hoff Research,
Electrical Engineering & Computer Science, Colorado School of Mines
(Kincade, 2014)

2.1.4.3 *Spatial Display*

Dalam *Spatial Augmented Reality* (SAR), objek nyata digabungkan langsung dengan citra yang terintegrasi langsung ke lingkungan nyata. Contohnya, citra diproyeksikan ke lingkungan nyata menggunakan proyektor digital atau tergabung dengan lingkungan menggunakan panel display. Perbedaan utama pada SAR dibanding teknik *display* sebelumnya adalah *display*-nya terpisah dengan pengguna. SAR memiliki kelebihan dari HMD dan *handheld*, sistem ini bisa digunakan oleh banyak orang pada waktu bersamaan tanpa perlu mengenakan suatu alat.

Ada tiga teknik display dalam SAR, yaitu sebagai berikut:

1. *Screen-Based Video See-Through Displays*

Screen-based AR menggabungkan citra dan lingkungan nyata yang ditampilkan ke sebuah monitor, seperti yang ditunjukkan pada gambar 2.12.



Gambar 2.12 Contoh *Screen-Based Video See-Through Displays*.
Gambar dari IEEE Computer tahun 2002 (Bimber, 2005)

2. *Spatial Optical See-Through Displays*

Sistem ini menghasilkan citra yang ditampilkan langsung ke lingkungan nyata. Komponen yang penting dalam sistem ini meliputi *spatial optical combiners* (*planar* atau *curved beam combiners*), layar transparan atau hologram.

3. *Projection-Based Spatial Displays*

Sistem ini memproyeksikan citra secara langsung pada permukaan objek fisik daripada menampilkannya pada sebuah bidang pencitraan dalam penglihatan pengguna. Sistem ini menggunakan banyak proyektor yang digunakan untuk meningkatkan wilayah tampilan serta meningkatkan kualitas citra.

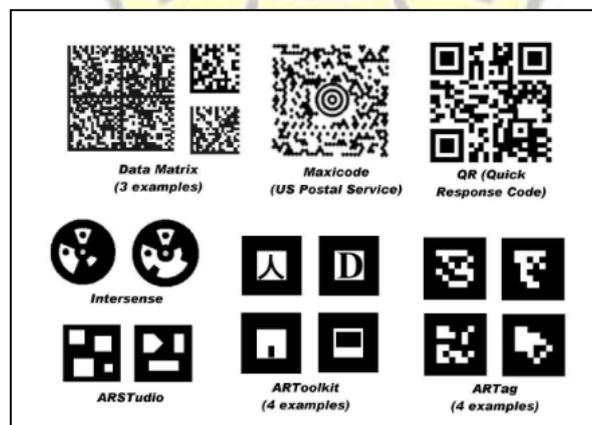
2.1.5 *Marker dan Markerless Augmented Reality*

Marker merupakan perangkat keras lainnya yang digunakan untuk membuat suatu aplikasi *Augmented Reality*. *Marker* diperlukan sebagai penanda untuk menampilkan suatu objek. *Marker* ini dicetak dengan menggunakan *printer*

untuk diarahkan langsung pada kamera. *Marker* yang dimaksud disini adalah pola yang dibuat, dalam bentuk gambar yang akan dikenali oleh kamera. Pada *Augmented Reality* terdapat dua metoda yaitu *Marker Augmented Reality (Marker Based Tracking)* dan *Markerless Augmented Reality*.

2.1.5.1 *Marker Augmented Reality (Marker Based Tracking)*

Marker biasanya merupakan ilustrasi hitam dan putih persegi dengan batas hitam tebal dan latar belakang putih. ARToolkit adalah sistem marker planar yang populer untuk *Augmented Reality*. Tahap pertama dari proses pengenalan adalah menemukan bingkai hitam pada *marker* mengorientasi marker untuk menghitung posisi kamera nyata dengan marker nyata, setelah itu matriks 3x4 akan diisi saat kamera nyata mengkoordinasi ke marker, matrik ini nantinya digunakan untuk mengatur posisi virtual kamera. Tahap Ketiga, ketika koordinat kamera virtual dan kamera nyata telah sama, maka grafik komputer kita akan menggambar dan melakukan overlay objek 3D (Ardhianto, Hadikurniawati dan Edy, 2012).



Gambar 2.13 Contoh *Marker Augmented Reality* (Fiala, 2005)

2.1.5.2 *Markerless Augmented Reality*

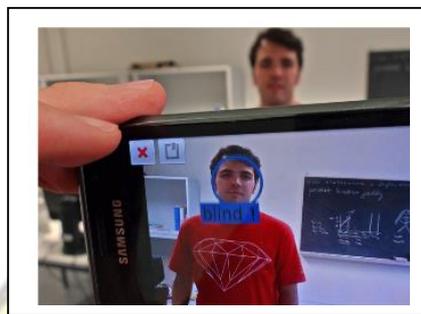
Salah satu metode *Augmented Reality* yang saat ini sedang berkembang adalah metode *Markerless Augmented Reality*, dengan metode ini pengguna tidak perlu lagi menggunakan sebuah marker untuk menampilkan elemen-elemen digital, dengan tool yang disediakan Qualcomm untuk pengembangan *Augmented Reality* berbasis *mobile device*, mempermudah pengembang untuk membuat aplikasi yang *markerless* (Qualcomm, 2012).

Pembuatan aplikasi yang *markerless* dapat menggunakan gambar yang diinginkan sebagai *Image Target*, gambar akan di-generate secara otomatis pada *Vuforia Target Manager*, lalu akan muncul tampilan *rating augmentable* gambar sebagai indikator bagus atau tidaknya gambar tersebut untuk dijadikan *Image Target*. Apabila *rating* bagus, maka gambar tersebut bagus untuk dijadikan *Image Target* dan sebaliknya apabila *rating augmentable* tidak bagus, maka akan sulit mendeteksi *Image Target*.

Namun, sekalipun dinamakan dengan *markerless*, aplikasi tetap berjalan dengan melakukan pemindaian terhadap objek, namun ruang lingkup yang dipindai lebih luas dibanding dengan *marker* AR. 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*.

2.1.5.2.1 Face Tracking

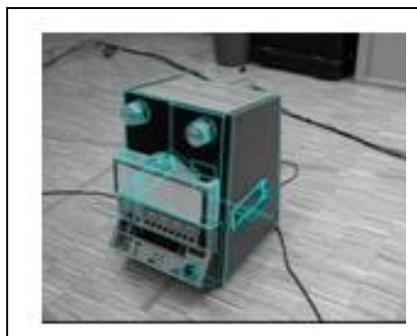
Banyak algoritma dan metode telah diterapkan untuk mencapai deteksi wajah. Hal-hal ini didukung oleh manfaat yang luas dari teknologi deteksi wajah. Aplikasi deteksi wajah dapat menggunakan media kamera video dan wajah pengguna sebagai masukan. Intinya adalah untuk mendeteksi wajah manusia, bukan serupa wajah manusia (Setyati, Alexandre, Widjaja, 2012).. Teknik ini pernah digunakan di Indonesia pada Pekan Raya Jakarta 2010 dan *Toy Story 3 Event*.



Gambar 2.14 Contoh Teknik *Face Tracking* (Dantone, 2011)

2.1.5.2.2 *3D Object Tracking*

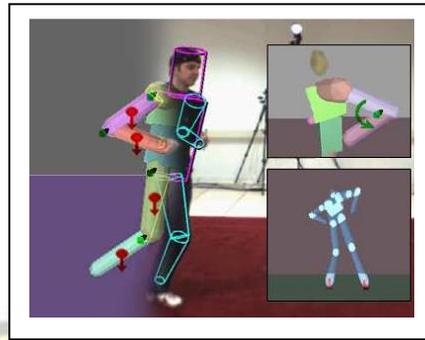
Teknik *3D Object Tracking* dapat mengenali semua bentuk benda tiga dimensi (3D) yang mempunyai ruang yaitu panjang lebar dan tinggi, seperti mobil, meja, televisi, dan lain-lain.



Gambar 2.15 Contoh Teknik *Object Tracking* (Vacchetti, 2004)

2.1.5.2.3 *Motion Tracking*

Pada teknik ini komputer dapat menangkap gerakan, Sistem computer grafis menggunakan *motion trackers* untuk lima tujuan utama yaitu *View Control*, *Navigation*, *Object Selection and Manipulation*, *Instrument Tracking*, dan *Avatar Animation* (Welch, 2002). Teknik ini sering dilakukan dalam produksi-produksi film yaitu film Avatar, teknik ini digunakan secara *real-time*.



Gambar 2.16 Contoh Teknik *Motion Tracking* (Vondrak, 2011)

2.1.5.2.4 GPS Based Tracking

Teknik *GPS Based Tracking* saat ini mulai populer dan banyak dikembangkan pada aplikasi *smartphone* (iPhone dan Android). Dengan memanfaatkan fitur GPS dan kompas yang ada didalam *smartphone*, aplikasi akan mengambil data dari GPS dan kompas kemudian menampilkannya dalam bentuk arah yang kita inginkan secara *real-time*, bahkan ada beberapa aplikasi menampilkannya dalam bentuk 3D. Salah satu pelopor *GPS Based Tracking* adalah aplikasi yang bernama Layar.

Untuk beberapa visi komputer aplikasi AR sendiri tidak dapat memberikan solusi pelacakan yang kuat dan metode sehingga metode *hybrid* telah dikembangkan, yang menggabungkan beberapa teknologi penginderaan. Misalnya, Azuma (Azuma, 1998) mengusulkan bahwa sistem AR berkembang yang bekerja di luar diperlukan sistem pelacakan berdasarkan GPS, inersia dan visi komputer

penginderaan. Salah satu aplikasi yang menggunakan metode ini adalah aplikasi android bernama Layar.



Gambar 2.17 Contoh Teknik GPS *Based Tracking* (Layar, 2009)

2.1.6 Penerapan *Augmented Reality*

Saat ini teknologi *Augmented Reality* telah dapat dirasakan. Banyak manfaat yang bisa di dapat dari *Augmented Reality* dan banyak bidang yang dapat dijadikan wadah penerapan teknologi *Augmented Reality*, diantaranya yaitu:

a. Kedokteran (*Medical*)

Pada tahun 1968, Sutherland menyarankan sebuah *head-mounted three dimensional display* sebagai yang memungkinkan menampilkan tampilan tiga dimensi untuk menampilkan ke *user* sebuah perspektif gambar yang berubah sebagaimana *user* bergerak dan menampilkan objek secara dekat ke *user* yang terlihat sangat jauh. Dua dekade kemudian, pada tahun 1986, Roberts menerapkan sistem *augmented reality* medis pertama. Contoh penggunaannya adalah pada pemeriksaan sebelum operasi, seperti CT Scan atau MRI, yang memberikan gambaran kepada ahli bedah mengenai anatomi internal pasien. Dari gambar-gambar ini kemudian pembedahan

direncanakan. *Augmented Reality* dapat diaplikasikan sehingga tim bedah dapat melihat data CT Scan atau MRI pada pasien saat pembedahan berlangsung.

Penggunaan lain adalah untuk pencitraan ultrasonik, di mana teknisi ultrasonik dapat mengamati pencitraan fetus yang terletak di abdomen wanita yang hamil (Bajura, 1992). Pembangunan sebuah mekanisme panduan sistem bantu *augmented reality* alami yang menggabungkan kelebihan dari aspek-aspek berikut: informasi panduan visual AR, pengalaman-pengalaman dokter bedah dan akurasi bantu operasi (Sielhorst, 2008)

Selain itu, pada tahun 2012, Blum menjelaskan langkah-langkah pertama menuju penglihatan X-ray seperti Superman dimana perangkat *brain-computer Interface* (BCI) dan sebuah gazetracker digunakan untuk memungkinkan *user* mengendalikan visualisasi *augmented reality*. Baru-baru ini, Wen pada tahun 2014, mengusulkan sistem operasi kooperatif, dipandu oleh gerakan tangan dan didukung oleh *augmented reality* berdasarkan bidang bedah.

b. Hiburan (*Entertainment*)

Augmented reality telah diterapkan di industri hiburan untuk membuat *game*, tetapi juga untuk meningkatkan visibilitas aspek pertandingan penting dalam penyiaran kehidupan olahraga. Dalam kasus ini di mana publik yang besar tercapai, *augmented reality* juga dapat berfungsi pengiklan untuk menampilkan iklan virtual dan penempatan produk. Kolam renang, lapangan sepak bola, trek balap dan lingkungan olahraga lainnya yang terkenal dan mudah disiapkan, yang *video see-through augmentation*-nya melalui pelacakan umpan-umpan kamera (Krevelen, 2012). Salah satu contoh adalah

sistem Fox-Trax (Cavallaro, 1997), digunakan untuk menyoroti lokasi dari sebuah *hockey puck* yang sulit dilihat yang bergerak cepat di atas es, tapi *augmented reality* juga diterapkan untuk menambahkan catatan atau keterangan mobil balap pertunjukan, lintasan bola snooker, penamapilan perenang, dan lain-lain (Azuma, 2006). Berkat lingkungan yang terprediksi (pemain berseragam di lapangan hijau, putih, dan coklat) dan teknik *chroma-keying*, penjelasan ditampilkan di lapangan dan bukan pada pemain (Cavallaro, 2011).

c. Pendidikan (*Education*)

Augmented Reality dalam bidang pendidikan berguna untuk meningkatkan pembelajaran sebagai sebuah media pembelajaran yang lebih menarik dalam mendapatkan informasi. *Augmented Reality* juga memungkinkan visualisasi dengan benda virtual atau informasi ke benda-benda fisik atau lingkungan (Arvanitis, 2009). Sistem *augmented reality* bisa mendukung peserta didik dalam memvisualisasikan konsep ilmu abstrak atau fenomena yang tidak teramati, seperti bidang aliran udara atau magnet, dengan menggunakan benda-benda maya termasuk molekul, vektor, dan simbol. Misalnya, *Augmented Chemistry* memungkinkan siswa untuk memilih unsur-unsur kimia, menyusun ke dalam model molekul 3D (Fjeld, 2002).

Pada tahun 2012, Clark mengusulkan buku mewarnai ditambah berbasis kertas dengan konten 3D dan memberikan anak-anak dengan pengalaman buku pop-up memvisualisasikan isi buku.

d. Militer (*Military*)

Augmented reality dapat digunakan untuk menampilkan adegan pertempuran nyata dan digabungkan dengan informasi penjelasan untuk memperlihatkan unit musuh yang tidak terlihat tanpa perlengkapan ini (Urban, 1999).

Beberapa HMD ini sudah diteliti dan dibangun oleh perusahaan Liteye untuk pemakaian militer. Dalam optik hibrida dan *inertial tracker* yang digunakan sensor-sensor miniatur MEMS (sistem mikro elektro-mekanik) yang dikembangkan untuk pelacakan kokpit helm (Foxlin, 2004). Teknik *augmented reality* untuk perencanaan pelatihan militer di medan perkotaan menggunakan teknik AR untuk menampilkan medan animasi, yang dapat digunakan untuk perencanaan intervensi militer, dikembangkan oleh perusahaan Arcane (Livingston, 2002). Sistem penglihatan malam helicopter dikembangkan oleh *Canada's Institute for Aerospace Research* (NRC-IAR) menggunakan *augmented reality* untuk memperluas selubung operasional pesawat dan meningkatkan kemampuan pilot untuk menavigasi dalam kondisi visual yang terdegradasi (Yu, 2010). HMD dikembangkan untuk tampilan yang dapat digabungkan dengan sistem informasi portabel di militer (Sanders, 2004).

Manfaat tambahan khusus untuk pengguna militer yaitu dapat berlatih dalam skenario pertempuran skala besar dan simulasi real-time tindakan musuh, seperti dalam sistem *Battlefield Augmented Reality* (BAR) (Julier, 2000). Sistem BAR juga menyediakan alat untuk penulis lingkungan dengan informasi 3D baru yang pengguna sistem lain melihat pada gilirannya (Baillot, 2001).

e. ***Manufacturing***

Tantangan di bidang manufaktur adalah untuk merancang dan mengimplementasikan sistem manufaktur *augmented reality* yang terintegrasi yang dapat meningkatkan proses manufaktur, serta produk dan proses pengembangan, yang mengarah ke waktu yang lebih pendek, mengurangi biaya dan meningkatkan kualitas (Caudell dan Mizell, 1992).

Tujuan utamanya adalah untuk menciptakan sebuah sistem yang baik seperti dunia nyata atau lebih baik dan lebih efisien. *Augmented reality* dapat meningkatkan persepsi seseorang dari dunia sekitarnya dan pemahaman tentang pengerjaan perakitan produk untuk dikerjakan (Reinhart dan Patron, 2003).

f. *Robotics dan Telerobotics*

Augmented reality adalah platform yang ideal untuk kolaborasi manusia-robot (Green, 2008). Terdapat robotika medis dan panduan gambar operasi berbasis *augmented reality* (Suzuki, 2008) dan layar prediksi untuk *telerobotics* sudah dirancang berdasarkan *augmented reality* (Kim, 1995).

Robot dapat menyajikan informasi yang kompleks dengan menggunakan teknik *augmented reality* untuk mengkomunikasikan informasi kepada manusia (Daily, 2003). Teknik *augmented reality* digambarkan untuk pengembangan dan eksperimen robot (Stilman, 2005). Penjelasan penggabungan teknik *augmented reality* dengan sistem robot bedah untuk bedah kepala (Worn, 2005).

g. *Marketing*

Augmented reality pertama kali digunakan untuk iklan dalam industri otomotif. Beberapa perusahaan mencetak brosur khusus yang secara otomatis dikenali oleh webcam, lalu sebuah objek tiga dimensi mobil yang diiklankan akan ditampilkan pada layar (Spies, 2009). Pendekatan ini kemudian menyebar ke berbagai *marketing*, dari permainan komputer dan film untuk sepatu dan *furniture* (Carmigniani, 2011). Contoh *augmented reality* yang lebih kompleks mencoba sepatu. Pengguna memakai sepasang kaus kaki khusus, kemudian berjalan di depan kamera dan melihat gambarnya di layar mengenakan sepasang sepatu yang diinginkan. Model, warna dan aksesoris dari sepatu bisa berubah dalam sekejap, memungkinkan pengguna untuk dengan mudah menemukan sepatu yang paling menarik (Papagiannakis, 2008). Gambar 2.18 merupakan contoh AR pada bidang *marketing/advertising* yang bernama *Virtual Show Fitting* pada Goertz (Nvidia Corporation, 2014).



Gambar 2.18 Contoh AR pada bidang *Marketing/Advertising* (Nvidia Corporation, 2014)

2.2 Virtual

Vuforia merupakan *software library/Augmented Reality Software Development Kit (SDK)* yang disediakan oleh Qualcomm untuk membantu para *developer* membuat aplikasi-aplikasi *Augmented Reality*, yang menggunakan sumber yang konsisten mengenai *computer vision* yang fokus pada *image recognition*. Vuforia adalah SDK untuk *computer vision based AR* untuk mengenali dan melacak gambar planar (Target Gambar) dan objek 3D sederhana, seperti kotak, secara *real-time*. Kemampuan *registrasi* citra ini memungkinkan pengembang untuk posisi dan orientasi objek *virtual*, seperti model 3D dan media lainnya, dalam kaitannya dengan gambar dunia nyata ketika hal ini dilihat melalui kamera dari perangkat *mobile*. Objek *virtual* kemudian melacak posisi dan orientasi dari gambar secara *real-time* sehingga perspektif *user* pada objek sesuai dengan perspektif mereka pada Target Gambar, sehingga tampak bahwa objek *virtual* adalah bagian dari adegan dunia nyata (Qualcomm Developer Network, 2010).

Gambaran sistem tingkat tinggi dari Vuforia SDK (Qualcomm Vuforia Developer Portal, 2011):

- *Callbacks* untuk *events* (Contoh: *A new camera image is available*)
- Akses tingkat tinggi untuk unit-unit hardware (Contoh: *Camera start/stop*)
- *Multiple trackables* (tipe-tipe tracking):
 - *Image Targets*
 - *Multi Targets*
 - *Cylinder Targets*
 - *Word Targets*
 - *Frame Markers*

- Interaksi dunia nyata
 - Virtual Buttons



Gambar 2.19 Gambaran sistem tingkat tinggi dari Vuforia SDK (Qualcomm Vuforia Developer Portal, 2011)

Tabel 2.2 Tabel pilihan pengembangan Vuforia

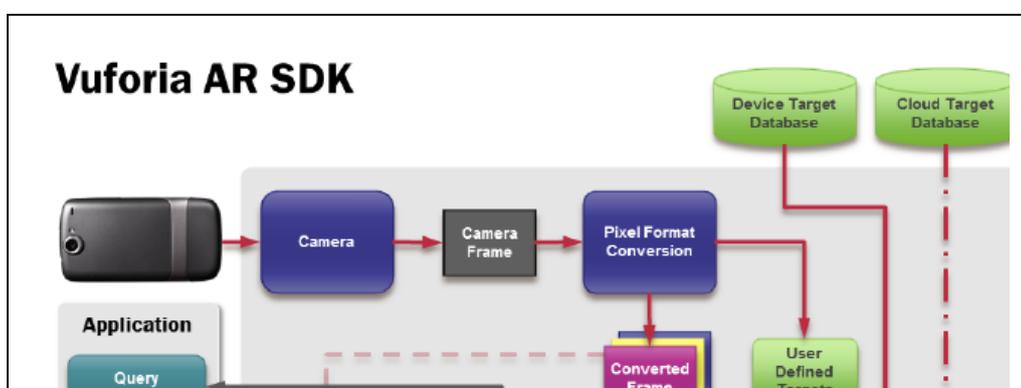
Development Environment	Development Platform			
	Native SDK		Unity Extension	
	Android	iOS	Android	iOS
Windows	Yes	--	Yes, multi-platform deployment	
MacOS	Yes	Yes	Yes, multi-platform deployment	
Linux	Yes	--	--	--

Vuforia mempunyai banyak fitur-fitur dan kemampuan, yang dapat membantu pengembang untuk mewujudkan pemikiran mereka tanpa adanya batas secara teknis. Vuforia SDK mendukung berbagai 2D dan 3D jenis sasaran termasuk markerless *Image Target*, konfigurasi 3D *Multi-Target*, dan bentuk beralamat *Fiduciary Marker* yang dikenal sebagai *Frame Marker*. Fitur tambahan dari SDK termasuk *Localized Occlusion Detection* menggunakan *Virtual Buttons*, pemilihan target gambar *run-time*, dan kemampuan untuk membuat dan mengkonfigurasi ulang target pengaturan pemrograman saat *run-time*.

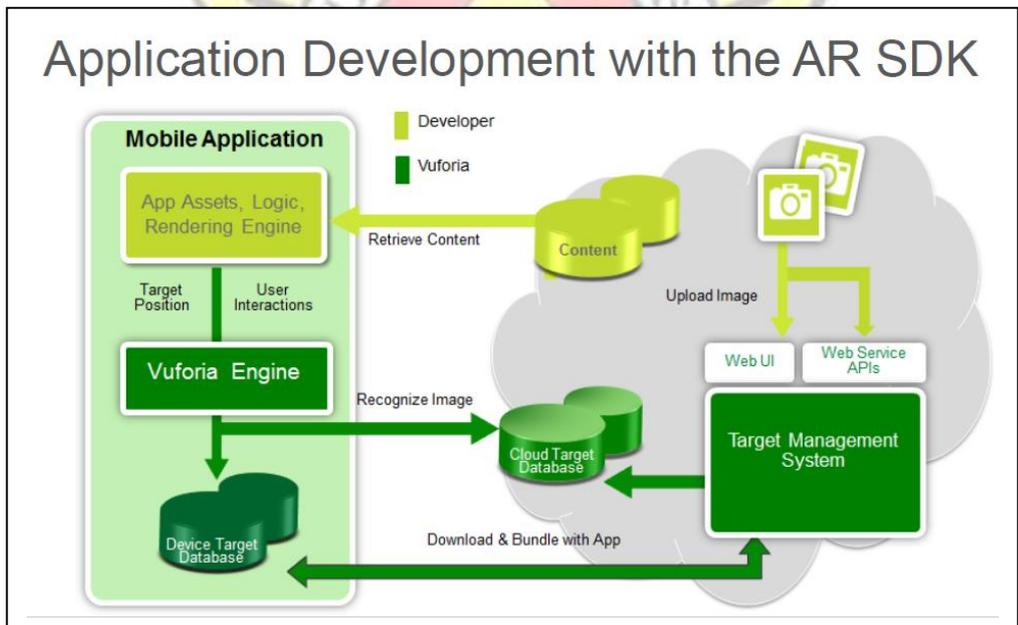
Vuforia SDK mendukung pengembangan aplikasi AR untuk iOS, Android, dan Unity3D, *platform* Vuforia mendukung para pengembang untuk membuat aplikasi yang dapat digunakan di hampir seluruh jenis *smartphone* dan *tablet*. (Oktari Permata Sari, Crisnapati, Antara Kesiman, Gede Sunarya, 2014) Pengembang juga diberikan kebebasan untuk mendesain dan membuat aplikasi yang mempunyai kemampuan antara lain:

1. Teknologi *computer vision* tingkat tinggi
2. Terus-menerus mengenali *multiple image*.
3. *Tracking* dan *Detection* tingkat lanjut.
4. Solusi pengaturan database gambar yang fleksibel.

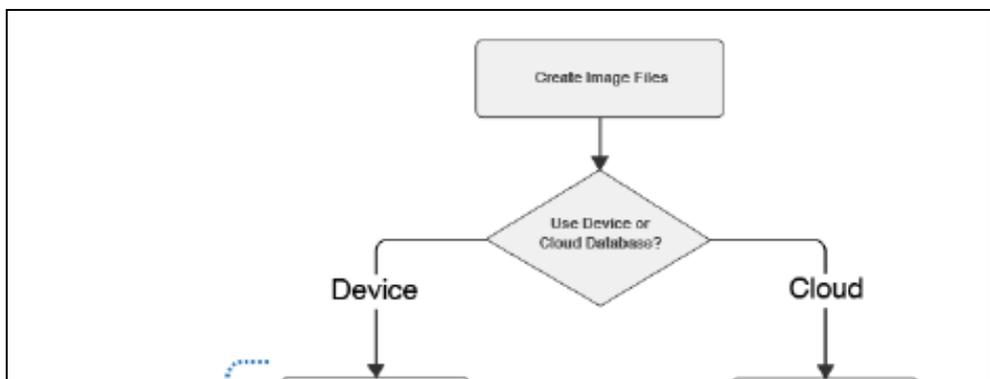
Vuforia menyediakan *Application Programming Interface* (API) di C++, Java, *Objective-C*, dan bahasa .Net. melalui sebuah *extension* ke Unity . Dengan cara ini , SDK mendukung pengembangan asli untuk iOS dan Android sementara juga memungkinkan pengembangan aplikasi AR di Unity yang mudah dibawa ke kedua platform tersebut. Aplikasi AR yang dikembangkan menggunakan Vuforia karena itu kompatibel dengan berbagai perangkat *mobile* termasuk iPhone (4/4S) , iPad , *smartphone/tablet* Android yang menjalankan OS Android versi 2.2 atau versi di atasnya dan ARMv6 atau ARMv7 prosesor dengan FPU (*Floating Point Unit*) kemampuan pemrosesan (Ibañez, 2013).



Gambar 2.20 Arsitektur Vuforia (Qualcomm Vuforia Developer Portal, 2011)



Gambar 2.21 Pengembangan Aplikasi dengan Augmented Reality SDK (Qualcomm Vuforia Developer Portal, 2011)





Gambar 2.22 Alur kerja *Developer* (Qualcomm Vuforia Developer Portal, 2011)

1. *Target*

Target adalah representasi dari objek dunia nyata yang dapat dideteksi dan dilacak. *Target* termasuk berbagai jenis objek, seperti:

- a. *Image targets*, misalnya, foto, papan permainan, halaman majalah, sampul buku, kemasan produk, poster, kartu ucapan. Ini memungkinkan *augmenting* gambar sederhana;
- b. *Frame markers*, yang merupakan jenis tertentu dari gambar bingkai 2D dengan pola visual khusus dan dapat digunakan sebagai bagian-bagian permainan dalam papan permainan;
- c. *Multi-targets*, misalnya, kemasan produk atau produk yang berbentuk persegi atau persegi panjang. Ini memungkinkan *augmenting* objek 3D sederhana;
- d. *Virtual buttons*, yang memungkinkan dalam menentukan tombol sebagai daerah persegi panjang pada target gambar.

2. *Image Target* yang dikenali oleh Vuforia

- a. *Image target* tidak perlu daerah hitam putih khusus atau kode-kode untuk dikenali. Seperti: markers, kode data matriks dan kode QR
- b. Vuforia menggunakan algoritma yang canggih untuk mendeteksi dan melacak fitur yang secara alami ditemukan dalam gambar itu sendiri.
- c. Dalam pembuatan *Image Target*, *image* sebaiknya kaya akan detail, kontrasnya bagus, tidak ada pola yang berulang, dan harus 8- atau 24-bit format PNG dan JPG, kurang dari 2 MB, JPG harus RGB atau Grayscale (bukan CMYK).

2.3 Android

Android (*/ˈæn.drɔɪd/; AN-droyd*) adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti *smartphone* dan

komputer tablet. Android adalah *mobile platform* yang paling populer di dunia dan android adalah basis yang paling banyak terinstal di berbagai *mobile platform* dan tumbuh cepat setiap hari. Gargenta (2011) mengatakan bahwa android adalah *open source platform*. Lisensi Android membolehkan orang lain untuk menggunakan dengan berbagai tujuan. Lebih dari itu, android di bawah lisensi yang ramah bisnis (Apache/MIT) sehingga orang lain dapat dengan bebas memperpanjang dan menggunakannya untuk berbagai tujuan. Bahkan beberapa pihak ketiga *open source libraries* yang dibawa ke dalam *stack* Android sudah ditulis ulang atas persyaratan lisensi baru. Sehingga, *developers* memiliki akses ke seluruh *source code platform*. Hal ini memungkinkan untuk melihat bagaimana proses kerja sistem operasi android. Produsen dapat dengan mudah memasukkan sistem operasi Android ke *hardware* tertentu. Dan juga bisa menambahkan tambahan hak patennya sendiri, dan tidak perlu dikembalikan ke komunitas pengembangan.

2.3.1 Sejarah Android

Sistem operasi Android didasarkan pada modifikasi kernel Linux versi 2.6 (Johnson, 2005). Dibandingkan kernel Linux versi 2.6, beberapa *drivers* dan *libraries* telah dimodifikasi atau dikembangkan agar Android dapat berjalan dengan efisien dan efektif pada perangkat *mobile* (seperti *smartphones* atau *internet tablets*). Beberapa dari *libraries* ini memiliki akar dalam proyek *open source*. Karena beberapa masalah perizinan, komunitas Android memutuskan untuk menerapkan *c library* mereka sendiri (*Bionic*), dan untuk mengembangkan Android khusus *Java runtime engine* (Dalvik Virtual Machine-DVM). Dengan

Android, fokus sudah selalu mengoptimalkan infrastruktur berdasarkan pada sumber daya terbatas yang tersedia di perangkat *mobile* (Liang, 2010).

Sejarah Android menarik dan memberikan beberapa perspektif tentang masa depan apa yang mungkin terjadi. Ini adalah peristiwa penting dari beberapa tahun terakhir (Gargenta, 2011):

- Pada tahun 2005, Google membeli Android, Inc. Dunia berpikir sebuah "Gphone" akan keluar.
- Semuanya berjalan tenang untuk sementara waktu.
- Pada tahun 2007, Open Handset Alliance diumumkan. Android adalah *open source* resmi.
- Pada tahun 2008, SDK Android 1.0 dirilis. Telepon G1, yang diproduksi oleh HTC dan dijual oleh operator nirkabel T-Mobile USA, berikut tak lama kemudian.
- Pada tahun 2009 melihat proliferasi perangkat berbasis Android. Versi baru dari sistem operasi yang dirilis: Cupcake (1.5), Donut (1.6), dan Eclair (2.0 dan 2.1). Lebih dari 20 perangkat menjalankan Android.
- Pada tahun 2010, Android berada diposisi kedua setelah Blackberry sebagai *smartphone platform* terlaris. Froyo (Android 2.2) dirilis dan lebih dari 60 perangkat yang menjalankannya.

2.3.2 Arsitektur Android



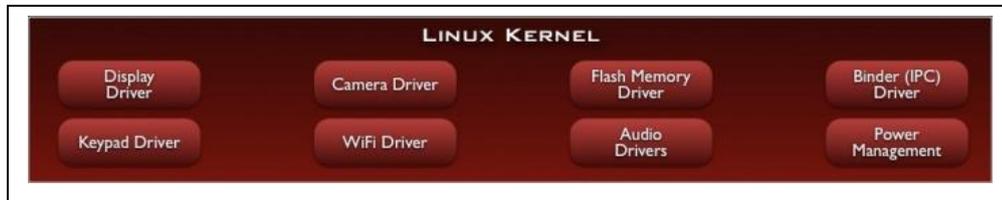
Gambar 2.23 Arsitektur Android (Felker, 2011)

1. Kernel Linux versi 2.6

Menurut Felker (2011), android diciptakan atas *open source* kernel Linux versi 2.6. Tim android memilih untuk menggunakan kernel ini karena memberikan fitur-fitur inti yang menjamin untuk mengembangkan sistem operasi Android. Fitur dari kernel Linux versi 2.6 termasuk (namun tidak terbatas pada) sebagai berikut:

- *Security model*: Kernel Linux menangani keamanan antara aplikasi dan sistem.
- *Memory management*: Kernel menangani manajemen memori untuk *developers*, sehingga *developers* bebas untuk mengembangkan aplikasi mereka.
- *Process management*: Kernel Linux mengelola proses dengan baik, mengalokasikan sumber daya untuk memproses apa yang dibutuhkannya.
- *Network stack*: Kernel Linux juga menangani komunikasi jaringan.

- *Driver model*: Tujuan dari Linux adalah untuk memastikan bahwa semuanya berjalan. Produsen perangkat keras dapat membangun *driver* mereka ke *Linux build*.



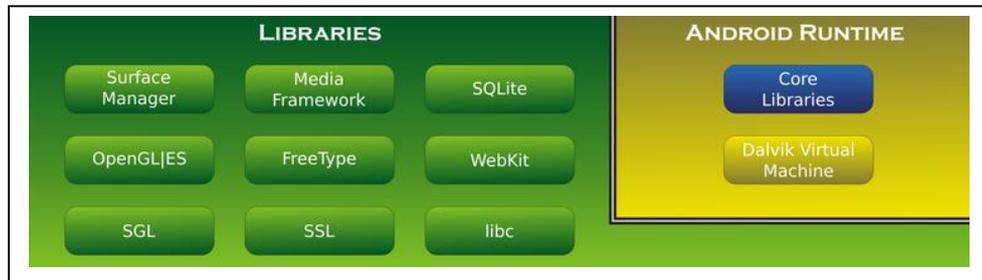
Gambar 2.24 Kernel Linux versi 2.6 (Felker, 2011)

2. *Android Framework*

Di atas Linux kernel 2.6, *Android Framework* dikembangkan dengan berbagai fitur. Fitur-fitur ini diambil dari berbagai sumber terbuka proyek. *Output* dari proyek-proyek ini menghasilkan hal-hal berikut:

- *The Android run time*: *The Android run time* terdiri dari *Java core libraries* dan *Dalvik Virtual Machine*.
- *Open GL (graphics library)*: lintas bahasa, lintas platform *Application Program Interface (API)* ini digunakan untuk menghasilkan komputer grafis 2D dan 3D.
- *WebKit*: mesin *open source Web browser* ini menyediakan fungsionalitas untuk menampilkan konten Web dan mempermudah *loading page*.
- *SQLite*: mesin *open source relational database* ini dirancang untuk disisipkan di dalam *device*.
- *Media frameworks: libraries* ini memungkinkan untuk *play* dan *record audio/video*.

- *Secure Sockets Layer (SSL)*: *libraries* ini bertanggung jawab untuk keamanan Internet.



Gambar 2.25 *Android Framework* (Felker, 2011)

3. *Application Framework*

Semua *open source framework* ini tersedia untuk *developers* melalui Android. *Developers* tidak perlu khawatir tentang bagaimana Android berinteraksi dengan *SQLite* dan *surface manager*; *Developers* hanya menggunakannya sebagai *tools* dalam Android mereka. Tim Android telah membangun satu set *libraries* yang terjamin dan telah memberikannya kepada *developers*, semua diekspos melalui Android *interface*. *Interfaces* ini membungkus berbagai *libraries* dan membuatnya berguna untuk platform Android begitu juga berguna bagi *developers*. Android memiliki semua *libraries* ini dibangun di latar belakang dan mengekspos fitur-fitur ini kepada *developers* tanpa harus membangun salah satu fungsionalitas yang mereka berikan:



Gambar 2.26 *Application Framework* (Felker, 2011)

Dari kernel ke aplikasi, sistem operasi Android telah dikembangkan dengan teknologi-teknologi terbukti *open source*. Hal ini memungkinkan *developers*, untuk membangun aplikasi yang kaya yang telah dibina dalam komunitas *open source*.

Aplikasi Android dikembangkan dalam bahasa pemrograman Java dengan menggunakan Software Development Kit(SDK). Android SDK ini terdiri dari seperangkat *development tools*, termasuk *debugger*, *software libraries*, sebuah perangkat *emulator*, dokumentasi, sampel *code*, dan tutorial (Felker, 2011). Android SDK menyediakan akses ke Android *libraries* dan memungkinkan *developers* untuk mengembangkannya untuk Android.

Inti dari Android SDK adalah *Application Programming Interface* (API). API adalah kumpulan fungsi, metode, *peoperties*, *classes*, dan *libraries* yang digunakan oleh *developers* aplikasi untuk membuat program yang bekerja pada *platform* tertentu. Android API berisi semua informasi spesifik yang *developers* butuhkan untuk membuat aplikasi yang dapat bekerja dan berinteraksi dengan aplikasi berbasis Android (DiMarzio, 2008).

Developers tidak bisa mengembangkan aplikasi Android tanpa Android SDK, sebelum melakukan instalasi Android SDK, *developers* harus melakukan instalasi *Java Development Kit* (JDK), menempatkan fondasi untuk Android SDK (Felker, 2011).

2.4 Unity

Unity Technologies telah mengubah wajah *games* dengan menciptakan *games* yang lebih menarik dan interaktif secara 3D dan 2D. Riccitiello (2014) mengatakan bahwa Unity merupakan suatu aplikasi yang digunakan untuk mengembangkan *game multi platform* (PC, Mac, Linux Standalone, iOS, Blackberry, Windows Phone, Windows Store, Xbox 360, Xbox One, PS Vita, PlayStation 3, PlayStation 4, Playstation Mobile dan Android).

Unity Technologies adalah penyedia platform pengembangan Unity yang memimpin pasar untuk konten 3D yang interaktif pada perangkat *mobile*. Dengan menambahkan platform *Augmented Reality* dari Qualcomm ke Unity untuk Android, *developers* Unity dengan cepat dan mudah membuat *game* dan aplikasi *augmented reality* untuk Android. (Qualcomm, June 2010).

AR *Extension* dari Qualcomm untuk Unity akan memungkinkan *developers* untuk menggunakan Unity untuk membangun aplikasi *augmented reality* dengan cepat untuk *smartphone* Android. Membangun aplikasi atas platform AR Qualcomm yang sudah ada untuk Android, *Extension*-nya, dikombinasikan dengan platform pengembangan Unity, memperluas pilihan yang tersedia dan memungkinkan *developers* untuk membangun *game* dan aplikasi interaktif yang lebih kaya dan dengan waktu pengembangan yang lebih pendek (Qualcomm, Nov 2010). Pada tahun 2010, Unity telah memperoleh *Technology Innovation Award* yang diberikan oleh Wall Street Journal.

Unity memiliki dua versi yaitu *Unity Pro* dan *Unity Personal*. *Unity Pro* hanya dapat dibeli dan *Unity Personal* dapat didapatkan secara gratis, Unity

tersedia untuk berbagai penggunaan untuk individu atau perusahaan dengan kurang dari US\$ 100,000 pendapatan kotor tahunan. Pada tanggal 3 Maret tahun 2015 dengan merilis Unity 5.0, Unity Teknologi membuat mesin lengkap yang tersedia secara gratis untuk semua fitur, *source code* yang lebih sedikit dan *supports*. Meskipun *Unity Personal* memiliki sebagian besar fitur dan fungsionalitas yang dimiliki Unity Pro. *Unity Personal* memiliki fitur tertentu yang tidak dapat dinonaktifkan ketika aplikasi dijalankan, termasuk *startup screen* merek Unity dan pengumpulan analisis data tertentu. *Unity Personal* tidak menyediakan fungsionalitas kolaborasi tim dan *users* akan diminta untuk melengkapi sebuah survei *user* untuk mengaktifkan *software*-nya. *Unity Pro* yang tidak memenuhi syarat untuk menggunakan *Unity Personal* mungkin tidak dapat mengembangkan dan mempublikasikan Konten Lisensi untuk platform iOS dan Android tanpa membeli lisensi produk *Unity Pro Add-On* yang berlaku (Unity Technologies, 2005).

2.4.1 Sejarah Unity dan Perkembangannya

Pada tanggal 21 Mei 2002, Nicholas Francis, seorang programmer Denmark, memposting di papan Mac OpenGL meminta bantuan dengan sistem *shader* yang sedang ia coba terapkan ke dalam *game engine*-nya. Sebuah *shader* yang merupakan apa yang memberikan objek 3D tampilan dan nuansanya. Beberapa jam kemudian, Joachim Ante, yang tinggal di Jerman pada waktu itu, merespons posting Francis. Percakapan mereka menghasilkan kolaborasi dua *developers* untuk menciptakan sistem *shader* yang akan bekerja untuk kedua mesin mereka yang terpisah. Francis mengatakan selama wawancara tentang asal-

usul Unity, mereka memutuskan untuk membuat sebuah mesin bersama-sama. David Helgason mendengar tentang proyek itu dan berpikir mereka akan menghasilkan sesuatu yang luar biasa, sehingga ia bergabung menjadi *developer* ketiga. Pada awalnya, mereka ingin mengembangkan sebuah *game* untuk bertahan hidup, tapi mereka melihat akan kebutuhan untuk teknologi yang lebih baik. Mereka berpikir untuk membuat sebuah *game* dan kemudian mematenkan teknologi tersebut dan bahwa *game* itu perlu membuktikannya. Pada akhirnya, mereka tidak membuat *game*, melainkan membuat alat untuk membuat *game* (*game engine*) yaitu *Unity* (Fear , 2009), (Haas, 2014).

Dalam beberapa tahun perkembangannya, sebelum dirilis, Unity telah diluncurkan pertama kali sebagai versi pra-rilis dengan GooBall, sebuah *video game* yang didesain khusus untuk Apple Macintosh.

GooBall, dengan unity pra- rilis telah diluncurkan atau diumumkan pada bulan Maret tahun 2005, sementara itu Unity diluncurkan secara resmi sebagai aplikasi yang bersifat komersial pada dua bulan setelahnya yaitu bulan Juni tahun 2005. Satu tahun kemudian yaitu tahun 2006, aplikasi pengembang *game* ini telah menjadi nominasi untuk *Apple Design Awards* dalam kategori “*Best OS X Graphics*”.

Sejak Unity secara resmi dirilis sebagai Unity versi 1.0.1, banyak pembaharuan (*update*), *upgrades* dan fitur yang telah ditambahkan selama tahun perilisannya tersebut dan Unity terus berkembang secara terus-menerus hingga saat ini yaitu Unity 5.0. Selama beberapa tahun banyak *game* yang dikembangkan dan dibuat berjalan di Unity. WolfQuest, di rilis pada tahun 2007, Tiger Woods PGA Tour Online dibuat pada bulan April tahun. Kemudian Unity dapat

mengembangkan AR *game* dan aplikasi untuk berbagai *platform* dengan sebuah AR extension.

2.4.2 Fitur-fitur Unity

- **Rendering**

Graphics engine yang digunakan adalah Direct3D (Windows, Xbox 360), OpenGL (Mac, Windows, Linux, PS3), OpenGL ES (Android, iOS), dan *proprietary APIs* (Wii). Ada pula kemampuan untuk *bump mapping*, *reflection mapping*, *parallax mapping*, *screen space ambient occlusion* (SSAO), *dynamic shadows using shadow maps*, *render-to-texture* and *full-screen post-processing effects*.

Unity tidak dirancang untuk proses desain atau *modelling*, dikarenakan unity bukan *tool* untuk mendesain. 3D *editor* lain digunakan untuk mendesain akan tetapi Unity dapat mengambil format desain dari 3dsMax, Maya, Softimage, Blender, modo, ZBrush, Cinema 4D, Cheetah3D, Adobe Photoshop, Adobe Fireworks and Allegorithmic Substance. Asset tersebut dapat ditambahkan ke *game project* dan diatur melalui *graphical user interface* Unity.

ShaderLab adalah bahasa yang digunakan untuk *shaders*, dimana mampu memberikan deklaratif “programming” dari *fixed-function pipeline* dan program *shader* ditulis dalam GLSL atau Cg. Sebuah *shader* dapat menyertakan banyak varian dan sebuah spesifikasi *fallback declarative*, dimana membuat Unity dapat mendeteksi berbagai macam *video card* terbaik saat ini, dan jika tidak ada yang kompatibel, maka akan

dilempar menggunakan shader alternatif yang mungkin dapat menurunkan fitur dan performa.

- ***Scripting***

Script game engine dibuat dengan Mono, sebuah implementasi open-source dari .NET Framework. Programmer dapat menggunakan UnityScript (bahasa terkustomisasi yang terinspirasi dari syntax ECMAScript, dalam bentuk JavaScript), C#, atau Boo (terinspirasi dari syntax bahasa pemrograman python). Dimulai dengan dirilisnya versi 3.0, Unity menyertakan versi MonoDevelop yang terkustomisasi untuk *debug script*.

- ***Asset Tracking***

Unity juga menyertakan *Server Unity Asset* – sebuah solusi terkontrol untuk *developer game asset* dan *script*. Server tersebut menggunakan PostgreSQL sebagai *backend*, sistem *audio* dibuat menggunakan FMOD library (dengan kemampuan untuk memutar *Ogg Vorbis compressed audio*), *video playback* menggunakan *Theora codec*, engine daratan dan vegetasi (dimana mendukung *tree billboarding*, *Occlusion Culling* dengan Umbra), *built-in lightmapping* dan *global illumination* dengan Beast, *multiplayer networking* menggunakan RakNet, dan navigasi *mesh* pencari jalur *built-in*.

- ***Platforms***

Unity mendukung pengembangan ke berbagai *platform*. Didalam *project*, *developer* memiliki kontrol untuk mengirim perangkat *mobile*, *web*

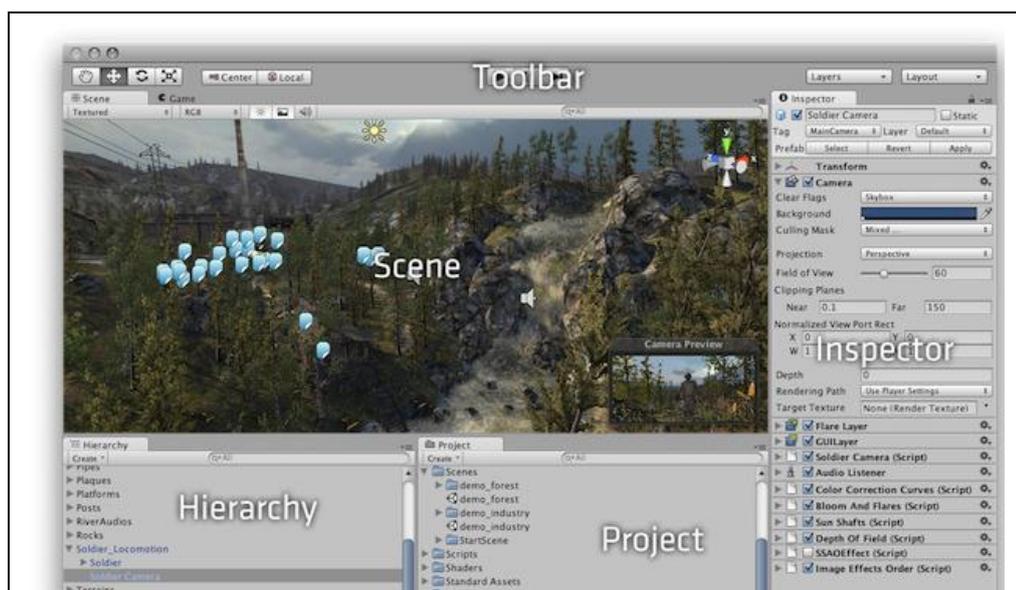
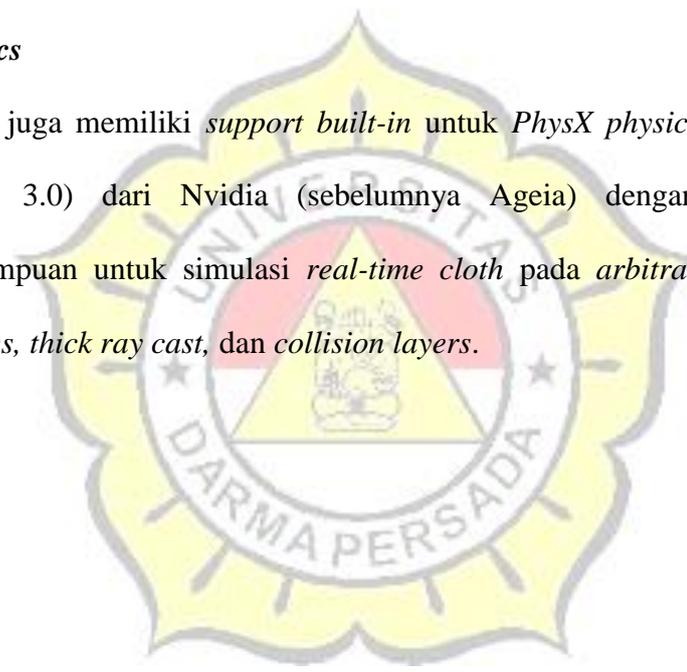
browser, desktop, dan console. Unity juga mengizinkan spesifikasi kompresi *texture* dan pengaturan resolusi di setiap *platform* yang didukung.

- ***Asset Store***

Diluncurkan November 2010, *Unity Asset Store* adalah sebuah *resource* yang hadir di *Unity editor*. *Asset store* terdiri dari koleksi lebih dari 4,400 *asset packages*, beserta *3D models, textures dan materials*, sistem *particle*, musik dan efek suara, tutorial dan *project, scripting package, editor extensions* dan servis *online*.

- ***Physics***

Unity juga memiliki *support built-in* untuk *PhysX physics engine* (sejak Unity 3.0) dari Nvidia (sebelumnya Ageia) dengan penambahan kemampuan untuk simulasi *real-time cloth* pada *arbitrary* dan *skinned meshes, thick ray cast, dan collision layers*.



Gambar 2.27 Lembar kerja Unity (Unity Technologies, 2005)

- **Projects**

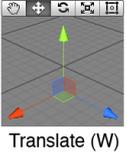
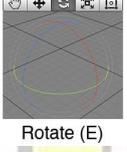
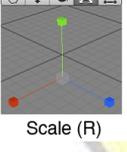
Bagian *projects* untuk mengakses dan memanajemen *assets* yang digunakan dalam sebuah *project*.

- **Scenes**

Scene View untuk memilih dan mengkoordinasikan lingkungan dan semua *GameObjects*. Memanuver dan memanipulasi objek dalam *Scene View* yang merupakan beberapa fungsi yang paling penting dalam Unity. Unity memberikan *keystrokes* untuk pengerjaan yang paling umum. Setiap project terdiri dari satu atau lebih *scene*. Satu buah *scene* mewakili satu buah level atau tampilan dalam suatu *game/aplikasi*.

Tabel 2.3 Tabel *Tools* pada *Scene View*

Scene View Navigation	
	Klik-drag untuk drag kamera.
	Tahan <i>Alt</i> dan klik- <i>drag</i> untuk mengorbit kamera sekitar <i>pivot point</i> .

	<p>Tahan <i>Alt</i> dan klik kanan-<i>drag</i> untuk zoom <i>Scene View</i>. Pada Mac Anda juga dapat menyimpan <i>Control</i> dan klik-<i>drag</i> sebagai gantinya.</p>
	<p>Merupakan <i>Scene Gizmo</i> untuk menampilkan orientasi <i>camera</i> yang di set dan mengubah angle tampilan sesuai keinginan dengan mudah dan cepat.</p>
<p><i>Rec Tool</i></p>	
	<p>untuk melihat dan mengedit penanganan <i>Rect</i> pada objek 2D dan 3D</p>
<p><i>Positioning GameObjects</i></p>	
 <p>Translate (W)</p>	<p>Mengatur posisi objek</p>
 <p>Rotate (E)</p>	<p>Merotasi objek</p>
 <p>Scale (R)</p>	<p>Mengatur skala objek</p>
<p><i>Scene View Control Bar</i></p>	
	
<p>Kontrol bar memungkinkan untuk memilih berbagai pilihan untuk <i>scene view</i>. Dapat mengaktifkan atau menonaktifkan audio, pencahayaan dan modus 2D dan juga mengontrol bagaimana benda, efek gambar dan gizmos akan ditampilkan.</p>	

- ***Game***

Game View di-render dari *Camera(s)* dalam sebuah *project*. Ini menampilkan gambaran hasil *project* yang telah dibuat dan akan perlu menggunakan satu atau lebih *camera* untuk mengontrol apa yang benar-benar dilihat *user* ketika mereka menjalankan *game/aplikasi* yang telah dibuat.

Tabel 2.4 Tabel *Tools* pada *Game View*

<i>Play Mode</i>	
	untuk mengontrol <i>Editor Play Mode</i> dan melihat bagaimana <i>game/aplikasi</i> yang ditampilkan. Sementara di <i>Play Mode</i> , setiap perubahan yang dibuat bersifat sementara, dan akan diatur ulang saat keluar dari <i>Play Mode</i> . <i>Editor UI</i> akan terlihat gelap untuk mengingatkan akan hal ini.
<i>Game View Control Bar</i>	
	
dapat mengubah rasio aspek <i>Game View window</i> ke nilai yang berbeda. Hal ini dapat digunakan untuk mengetes bagaimana <i>game/aplikasi</i> akan terlihat pada layar dengan rasio aspek yang berbeda.	

- ***Hierarchy***

Hierarchy berisi setiap *GameObject* di *scene* yang berjalan. Contoh yang diambil dari *assets* seperti model 3D, dan yang lainnya adalah contoh dari *prefabs*, berbagai objek yang telah ada. Dapat memilih objek pada *Hierarchy* dan *drag* satu objek ke yang lainnya yang disebut dengan *Parenting*. Ini digunakan untuk membuat sebuah *GameObject* menjadi anak dari *GameObject* yang lain. Menarik sebuah *gameobject* dan dipindahkan tepat di atas tulisan *gameObject* yang akan dijadikan *parent* dalam *hierarchy*. *GameObject* yang terdapat dalam sebuah *gameObject* lainnya akan mengikuti perpindahan dan perputaran ketika *gameObject parent* mengalami perubahan

posisi. Objek yang ditambahkan dan dihapus dalam *scene*, mereka akan muncul dan menghilang juga dari *Hierarchy*

- **Toolbar**

Tabel 2.5 Tabel *Tools* pada *Toolbar*

Five Basic Controls	
	<i>Transform Tools</i> – digunakan dengan <i>Scene View</i>
	<i>Transform Gizmo Toggles</i> – mempengaruhi tampilan <i>Scene View</i>
	Tombol <i>Play/Pause/Step</i> – digunakan dengan <i>Game View</i>
	<i>Layers Drop-down</i> – mengontrol objek mana yang ditampilkan di <i>Scene View</i>
	<i>Layout Drop-down</i> – mengatur susunan semua <i>Views</i>

- **Inspector**

Inspector menampilkan informasi rinci tentang *GameObject* sedang dipilih, termasuk semua komponen terpasang dan *property*-nya. Dapat memodifikasi fungsionalitas *GameObjects* di *scene*.

- **GameObjects**

GameObject adalah bagian terpenting di Unity. *GameObject* adalah container untuk menampung fungsionalitas yang disebut komponen. *GameObject* biasanya terdiri dari lebih dari satu komponen.

- **Components**

Komponen adalah komponen pembangun dari *GameObject*, dimana tanpa komponen, maka *GameObject* tidak akan berarti apapun. Komponen merepresentasikan *entity*, material data, *script* dan lain-lain. Komponen selalu terpasang di *GameObject* dan tidak bisa berdiri sendiri.

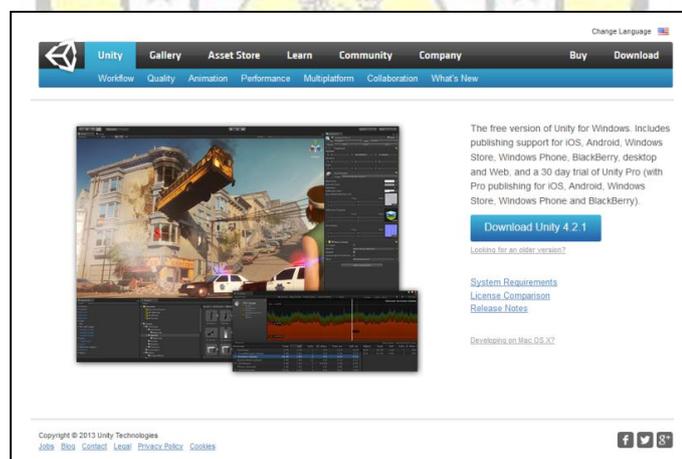
- **Assets**

Terdiri dari material, *texture*, file audio maupun *Prefab*.

- **Prefab**

Prefab adalah *asset* yang sudah didefinisikan menjadi *template*. Ketika meletakkan *prefab* kedalam *scene*, maka sama dengan melakukan proses instantiasi.

Untuk mendapatkan *software* ini secara gratis, dapat mengunjungi situs resminya di unity3d.com



Gambar 2.28 Situs resmi *Download Unity* (Unity Technologies, 2005)

2.5 Bahasa Pemrograman C#

C-Sharp (C#) merupakan sebuah bahasa pemrograman yang berorientasi objek (OOP) dan dirancang untuk *Common Language Infrastructure* yaitu sebuah spesifikasi terbuka yang dikembangkan oleh Microsoft dan distandarisasi oleh

ISO dan ECMA yang memungkinkan beberapa bahasa tingkat tinggi untuk digunakan pada platform komputer yang berbeda tanpa ditulis ulang untuk arsitektur tertentu.

C# merupakan bahasa pemrograman yang sederhana, modern dan mudah digunakan karena C# merupakan modifikasi perintah yang sepenuhnya sama dengan C++ yang kaya akan fitur yang lengkap tetapi dengan gaya bahasa yang lebih diperhalus dan dapat memberitahu dimana letak kesalahan bila ada kesalahan dalam aplikasi sehingga dapat mengurangi waktu dalam mencari error. Selain itu, C # juga mendukung tipe data Boolean, aturan dan fitur *namespace* pada C# sangat mirip dengan Java dan C++ sehingga cocok untuk membuat aplikasi *hosted* dan *embedded systems*, mulai dari yang sangat besar yang menggunakan sistem operasi canggih sampai ke fungsi yang sangat kecil dan dapat beradaptasi pada *software* komputer ke bahasa yang berbeda, proses mendesain aplikasi *software* yang berpotensi dapat disesuaikan dengan berbagai bahasa dan tempat tanpa perubahan mesin (Ecma International, 2006)

2.6 Instalasi Unity dan Vuforia dalam membuat aplikasi *Augmented Reality*

2.5.1 Registrasi akun *Developer*

Developer akan membutuhkan akun Unity dan Vuforia. Sebagai tambahan, jika meluncurkan dari OS X ke sebuah iOS device, maka perlu bergabung ke

Apple Developer program (free) dan *iOS Developer program (\$99)*. Sebagaimana pada bulan Februari 2014, para pelajar di COMS W4172 akan diijinkan peluncuran ke sebuah *iOS device* secara gratis melalui *iOS Developer University Program* sebagai anggota dari tim pelajar pengembangan untuk 4172.

- Unity (<https://accounts.unity3d.com/sign-up>)
- Vuforia (<https://developer.vuforia.com/user/register>)
- Apple
 - *Apple Developer Account (free)*
(<https://developer.apple.com/register/index.action>)
 - *iOS Developer (\$99)*
(<https://developer.apple.com/programs/ios/>)

2.5.2 Instalasi Unity

- *Download* dan *install* Unity.
- Mengikuti instruksi di instalasi *wizard*.
- Setelah instalasi selesai, jalankan Unity.
- Akan diminta untuk memilih lisensi. Silakan pilih: ***Activating Unity Free***
- Memasukkan kredensial Unity (lihat langkah 1 jika tidak mempunyai akun Unity).
- Aplikasi harus terbuka pada jendela *Create Project dialog*.
- Memilih sebuah lokasi dan membuat sebuah nama untuk *project* yang akan dibuat.
- Tekan ***Create Project***.
- Jika aplikasi *crash*, dapat berkonsultasi pada <http://docs.unity3d.com/Documentation/Manual/TroubleShooting.html>.

2.5.3 Instalasi Android *Development Tools*

Jika diaplikasikan pada sebuah Android *device*:

- Mengunjungi situs Android SDK dan *download* SDK. Mengikuti hal berikut:
 - Bisa menginstalasi *Android Development Tools (ADT) Bundle* atau hanya SDK. Merekomendasikan *ADT Bundle* jika mempunyai memori yang cukup.
 - Jika mempunyai memori yang terbatas pada *hard drive*, maka hanya dapat menginstalasi Android SDK: memilih “*Use an existing IDE*” pada halaman instalasi dan kemudian pilih “*Download the SDK Tools.*”
- Jika menginstalasi *ADT Bundle*, mengikuti instruksi di *Setting up the ADT Bundle*
- Jika menginstalasi *SDK Tools*, mengikuti instruksi di *Setting up an Existing IDE* dan *Adding Platforms and Packages*
- Akan perlu untuk menginstalasi *driver* yang tepat untuk Android *device*.
 - Jika pada Windows, dapat mengunjungi *OEM USB Drivers*.
 - Jika pada OS X, dapat melewati langkah ini.
- Juga harus *enable USB Debugging* pada Android *device*.
 - Untuk *device* yang menjalankan Android 3.2 atau dibawahnya:
Settings*→*Applications*→*Development

- Untuk *device* yang menjalankan Android 4.0 dan di atasnya:
Settings*→*Developer Options
 - Untuk *device* yang menjalankan Android 4.2 dan di atasnya:
Settings*→*Developer Options tersembunyi oleh *default*.
Untuk menampilkan *Developer options*, masuk ke
Settings*→*About Phone dan tap ***Build Number*** tujuh kali.
- Menggunakan *Using Hardware Devices* untuk informasi lebih lanjut.
 - Ketika membuat Android *project* pertama kali di Unity, maka akan diminta untuk memberikan *path* ke *Android SDK*. Memberikan *path* ke *root folder* dari instalasi SDK.
 - Dapat melihat *Android SDK Setup* untuk detail tambahan.

2.5.4 Mengetes Unity dan Android

- Jika diaplikasikan pada Android:
 - *Attach Android device via USB*.
 - Membuka Unity
 - Membuat *project* baru.
 - Memilih ***File*→*Build Settings***.
 - Memilih Android dari daftar *platform* yang tersedia.
 - Memilih ***Build and Run***
 - Unity mungkin akan menanyakan *path* pada instalasi Android.
 - Memilih ***Browse*** dan lokasi *root folder* dari direktori Android SDK.
 - Ketika proses pembuatan selesai, Android *device* harus menjalankan aplikasi Unity.

- Dapat mengunjungi <http://docs.unity3d.com/Documentation/Manual/TroubleShooting.html#AndroidTroubleShooting> untuk *troubleshooting tips*.

2.5.5 Instalasi *Vuforia Unity Extension*

- *Download Vuforia Unity Extension* dari situs *Vuforia developer*.
- *Extract* arsip dan *copy* isinya ke *Standard Packages directory* dari instalasi Unity.
- Jika mengembangkan pada Windows 7 (64-bit), harus *apply hotfix* dari Microsoft di <http://support.microsoft.com/kb/976038>.
 - Memperhatikan petunjuk instalasi *Vuforia Unity Extension* untuk lebih detilnya.
- **(Pilihan)** Mengikuti instruksi di *Compiling a simple project* untuk memastikan semua komponen berjalan dengan tepat.

2.7 Gigi Berlubang

2.6.1 Gigi

Gigi adalah bagian keras yang terdapat di dalam mulut dari banyak vertebrata. Mereka memiliki struktur yang bervariasi yang memungkinkan mereka untuk melakukan banyak tugas. Fungsi utama dari gigi adalah untuk merobek dan mengunyah makanan dan pada beberapa hewan, terutama *karnivora*, sebagai senjata. Akar dari gigi tertutup oleh gusi. Gigi memiliki struktur pelindung yang disebut email gigi, yang membantu mencegah lubang di gigi. *Pulp* dalam gigi menciut dan *dentin* terdeposit di tempatnya.

Gigi merupakan bagian paling membedakan di jenis mamalia yang berbeda, dan salah satu yang bisa menjadi fosil dengan baik. Paleontologis

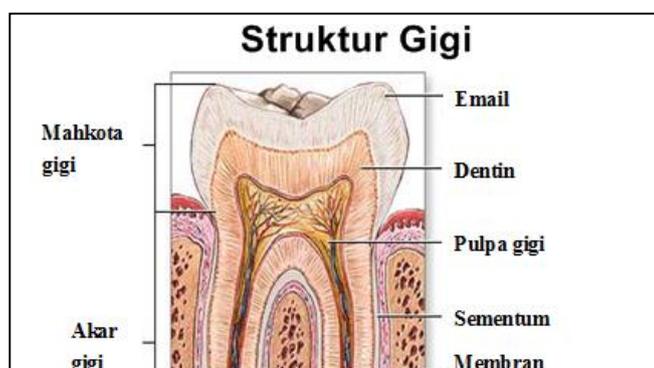
menggunakannya untuk mengidentifikasi jenis fosil dan seringkali hubungan di antaranya. Analisis perbandingan gigi juga sangat membantu untuk mendapatkan gambaran perilaku evolusioner selama perkembangan suatu spesies.

Bentuk gigi berhubungan dengan jenis makanan hewan. Misalnya *herbivora* memiliki banyak gigi geraham untuk mengunyah karena rumput sulit untuk dicerna. *Karnivora* membutuhkan taring untuk membunuh dan merobek, dan karena daging mudah untuk dicerna, maka mereka dapat menelan makanan tersebut tanpa membutuhkan geraham untuk mengunyah makanan tersebut terlebih dahulu.

2.6.2 Bagian-bagian Gigi dan Fungsinya

Gigi juga berfungsi sebagai alat pencernaan mekanis. Di sini, gigi membantu memecah makanan menjadi potongan-potongan yang lebih kecil. Hal ini akan membantu enzim-enzim pencernaan agar dapat mencerna makanan lebih efisien dan cepat. Selama pertumbuhan dan perkembangan, gigi manusia mengalami perubahan, mulai dari gigi susu dan gigi tetap (permanen).

Gigi pertama disebut gigi susu (*dens lakteus*). Pada anak berusia 6 tahun, gigi berjumlah 20; gigi seri (*dens insisivus*) 8 buah, gigi taring (*dens caninus*) 4 buah, gigi geraham kecil (*dens premolare*) 8 buah. Ketika usia anak berkisar antara 6 tahun hingga 14 tahun, gigi susu mulai tanggal dan kemudian digantikan oleh gigi tetap (permanen). Gigi permanen berjumlah 32 buah, yang berarti ada penambahan geraham besar yang berjumlah 12 buah (Nelson, 2010).



Gambar 2.29 Struktur Gigi (ADAM, 2014)

Pada bagian gigi manusia terstruktur / tersusun atas beberapa jaringan. Mahkota gigi atau *corona*, merupakan bagian yang tampak di atas gusi. Terdiri atas (Ross, 2011):

- Lapisan *email*, merupakan jaringan yang berfungsi untuk melindungi tulang gigi dengan zat yang sangat keras yang berada di bagian paling luar gigi manusia.
- Tulang gigi (*dentin*), di dalamnya terdapat saraf dan pembuluh darah dan merupakan lapisan yang berada pada lapisan setelah email yang dibentuk dari zat kapur.
- Rongga gigi (*pulpa*), merupakan bagian antara corona dan radiks, rongga yang di dalamnya terdapat pembuluh darah kapiler dan serabut-serabut syaraf.
- Leher gigi atau *kolum*, merupakan bagian yang berada di dalam gusi.
- Akar gigi atau *radiks*, merupakan bagian yang tertanam pada tulang rahang. Akar gigi melekat pada tulang rahang dengan perantaraan semen gigi.

- Semen gigi melapisi akar gigi dan membantu menahan gigi agar tetap melekat pada gusi. Terdiri atas:
 - Lapisan semen, merupakan pelindung akar gigi dalam gusi.
 - Gusi, merupakan tempat tumbuh gigi.

Fungsi gigi antara lain (Kokten, 2003):

1. Mengunyah makanan,
2. Untuk berbicara,
3. Fungsi kecantikan/ estetika / keindahan.

Pada manusia dapat ditemui 4 macam gigi yang terdapat pada mulut disertai dengan fungsinya yaitu (Ramadhan, 2010) :

1. **Gigi seri**, yang mempunyai satu akar, bentuknya seperti pahat fungsi untuk memotong makanan,
2. **Gigi taring**, yang mempunyai satu akar, bentuk seperti ujung tombak fungsi untuk merobek makanan,
3. **Gigi geraham**,
 - a. **Gigi geraham kecil**, yang mempunyai dua akar, bentuk seperti lumpang kecil fungsi untuk menghancurkan makanan,
 - b. **Gigi geraham besar**, yang mempunyai tiga akar, bentuk seperti lumpang besar fungsi untuk melumatkan dan menghaluskan makanan.



Gambar 2.30 Penampang Gigi Susu dan Gigi Tetap
(Dr. Che Noor Aini dan Dr Laila, 2012)

2.6.3 Pengertian Gigi Berlubang (Karies Gigi)

Karies gigi (Gigi Berlubang) dalam bahasa Yunani, kata “*ker*” artinya kematian. Dalam bahasa latin berarti kehancuran. Pembentukan lubang pada permukaan gigi disebabkan oleh kuman yang dikenal sebagai lubang (Srigupta, 2004).

Pembusukan terjadi di dalam lapisan gigi yang paling luar dan keras, tumbuh secara perlahan. Setelah menembus pada lapisan kedua (dentin, lebih lunak), pembusukan akan menyebar lebih cepat dan masuk ke dalam pulpa (lapisan gigi paling dalam yang mengandung saraf dan pembuluh darah) (Marsh, 2009).

2.6.4 Penyebab Gigi Berlubang (Karies Gigi)

Bowden (2000) menjelaskan bahwa keberadaan bakteri dalam mulut merupakan suatu hal yang normal. Bakteri dapat mengubah semua makanan,

terutama gula, menjadi asam. Bakteri, asam, sisa makanan, dan ludah akan membentuk lapisan lengket yang melekat pada permukaan gigi. Lapisan lengket inilah yang disebut plak. Plak akan terbentuk setelah makan. Zat asam dalam plak akan menyebabkan jaringan keras gigi larut dan terjadilah karies.

Setelah jaringan keras mengalami demineralisasi (mineral email gigi larut dalam asam) maka tanda-tanda paling awal yaitu adanya *white spot* (bercak putih di bawah jaringan email). Jadi karies itu mula-mula terlihat di bawah email, lambat laun asam tersebut akan mendemineralisasi jaringan email di sekitarnya dan mendemineralisasi dentin juga (Bowden, 2000). Kalau sampai pada keadaan yang parah bisa terkena *pulpa* di mana banyak terdapat serabut saraf sehingga terasa sakit.

Bakteri merupakan penyebab gigi berlubang yang cukup serius. Bakteri yang terdapat dalam mulut dapat menjadi penyebab adanya pengikisan email yang dapat membuat gigi berlubang. Gula adalah zat yang sering membuat gigi cepat berlubang. Dimana bakteri yang ada dalam mulut akan mengubah sisa gula yang terdapat di mulut menjadi asam. Asam inilah yang menjadi penyebab gigi berlubang (Stephan dan Miller, 1943)

1. Faktor Dalam

Bakteri penyebab gigi berlubang yaitu (Bowden, 2000):

- *Lactobacillus acidophilus* yang terdapat di dalam lubang dan celah pada bagian atas permukaan gigi,
- 6 jenis bakteri *Streptococcus* yang menyerang permukaan lunak pada bagian samping gigi. Biasanya terjadi pada daerah yang bersentuhan

dengan gigi lainnya. Lubang yang terjadi pada bagian ini sulit terlihat oleh mata sehingga membutuhkan foto *rontgen*,

2. Faktor Luar

- Cara menyikat gigi yang salah,
- Memilih sikat gigi yang salah,
- Waktu menyikat gigi yang salah,
- Malas menyikat gigi,
- Terlalu banyak mengonsumsi makanan manis (Stephan dan Miller, 1943),
- Merokok (Johnson, 2000),
- Kurangnya ilmu pengetahuan mengenai pentingnya menjaga kesehatan dan kebersihan gigi.

2.6.5 Gejala Gigi Berlubang (Karies Gigi)

Tanda-tanda dan gejala gigi berlubang dan kerusakan gigi bervariasi tergantung pada luas dan letaknya. Bila gigi berlubang baru saja dimulai, mungkin tidak mengalami gejala sama sekali.

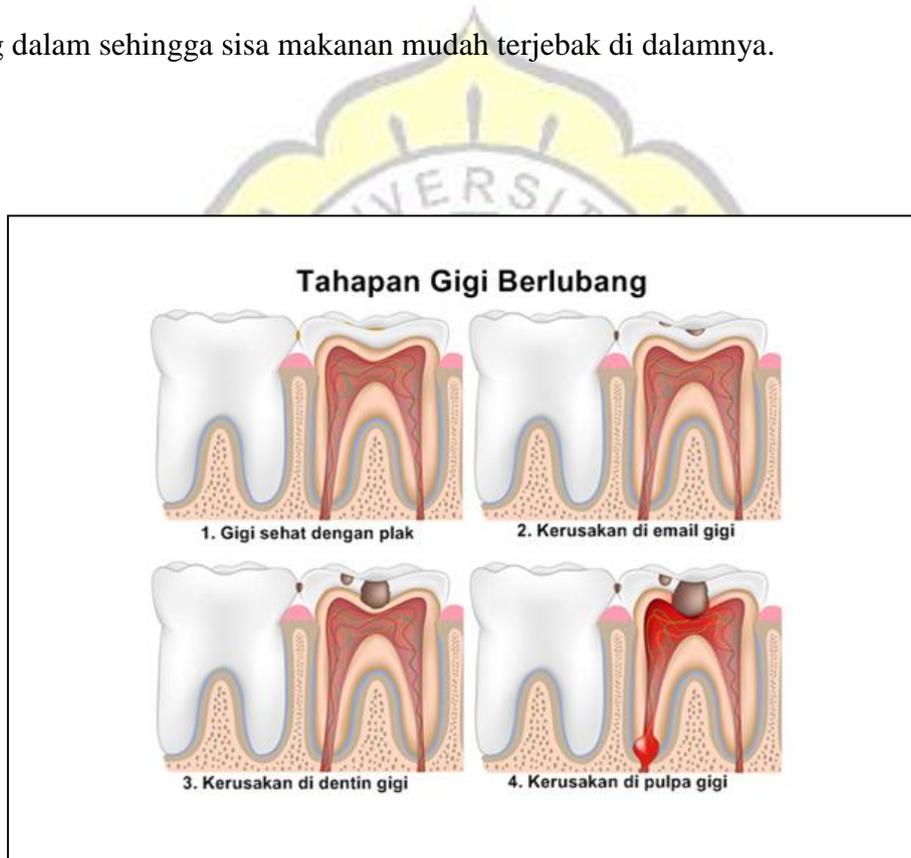
Jika pembusukan semakin besar akibat gigi berlubang, dapat menyebabkan tanda dan gejala seperti (Benarjee dan Watson, 2012):

- Sakit gigi
- Demam
- Nyeri ringan sampai nyeri yang menusuk ketika makan atau minum sesuatu yang manis, panas atau dingin
- Nyeri ketika menggigit

- Timbul nanah sekitar gigi, terutama ketika menekan pada gusi

2.6.6 Proses Gigi Berlubang (Karies Gigi)

Gigi yang paling rentan dan paling sering terkena karies atau gigi berlubang adalah gigi geraham. Hal ini juga diungkapkan oleh sebuah survei dari Adult Dental Health Survey (2000) di Inggris bahwa gigi bagian belakang juga memiliki kemungkinan dua kali lebih besar terserang karies. Hal tersebut disebabkan karena secara anatomis gigi geraham memiliki ceruk atau lekukan yang dalam sehingga sisa makanan mudah terjebak di dalamnya.



Gambar 2.31 Proses Gigi Berlubang (Prestonwood Dental, 2011)

1. Gigi sehat dengan plak

Plak adalah lapisan tipis yang melekat di permukaan gigi yang membentuk asam, berisi kumpulan protein air ludah yang disebut *pelikel*, sisa makanan dan bakteri.

Plak adalah lapisan tipis dari mikroorganisme, sisa makanan dan bahan organik yang terbentuk di gigi, kadang-kadang juga ditemukan pada gusi dan lidah. Plak merupakan agregat sejumlah besar dan berbagai macam mikroorganisme pada permukaan gigi mulai erupsi dengan cepat akan dilindungi lapisan tipis glikoprotein yang disebut *aequired pellicle*. Glikoprotein di dalam air ludah akan diserap dengan spesifik pada hidroksiaptit dan melekat erat pada permukaan gigi (Roeslan, 2002). Beberapa bakteri yang hidup di dalam rongga mulut mempunyai kemampuan membentuk koloni pada permukaan gigi dan membentuk plak secara berkesinambungan.

Menurut Roeslan (2002) proses pembentukan plak gigi terjadi setelah *S.mutans* serotipe c membentuk dekstran ekstraseluler baru terjadi perlekatan dan agregasi kuman pada permukaan enamel yang diikuti peningkatan kolonisasi. Agregasi kuman terjadi karena adanya reseptor dekstran pada permukaan sel sehingga terjadi interaksi antarsel selama pembentukan plak gigi. Kuman lain seperti *S.samguis* juga mampu mensintesis dekstran ekstraseluler dari sukrosa, namun yang dibentuk adalah dekstran ikatan $\alpha(1-6)$ yang mudah larut dalam air dan kuman ini tidak mempunyai reseptor dekstran pada permukaan selnya. *S.mutans* serotipe c lebih banyak mensintesis dekstran ikatan $\alpha(1-3)$ yang tidak larut di dalam air sehingga lebih efisien dalam membentuk plak gigi. *S.mutans*

juga membentuk levan walau tidak sebanyak yang dibentuk oleh *S.salivarius* dan *A.viscocus*. Levan akan dihidrolisi oleh beberapa kuman di dalam plak, oleh karena itu levan tidak seefisien dekstran dalam membentuk plak gigi.

Plak dalam mulut tidak akan bisa hilang, karena setiap beberapa jam setelah sikat gigi plak akan muncul kembali di dalam mulut. Jika plak di biarkan menumpuk di sekitar jaringan mulut akan menyebabkan kalkulus atau karang gigi bahkan akan merusak jaringan keras maupun lunak di dalam rongga mulut. Rusaknya jaringan keras inilah disebut karies gigi atau gigi berlubang. Jika dibiarkan akan menyebabkan matinya jaringan gigi, abses pada mulut, dan kadang-kadang jika karies hanya pada satu gigi bisa merambat ke gigi sebelahnya.

2. Kerusakan di *email* gigi

Email merupakan permukaan terluar gigi. Pada tahap ini zat asam mulai melarutkan *email*, proses ini dikenal dengan karies, tidak akan dirasakan gejala atau rasa apapun karena tidak ada syaraf pada *email* gigi, ngilu akan terasa jika sudah sampai pada batas pertemuan antara *email* dan *dentin*. Jika lubang tidak ditambal maka akan berlanjut ke lapisan *dentin*.

3. Kerusakan di *dentin* gigi

Pada tahap ini, zat asam mulai melakukan penetrasi karies pada *dentin*, ini baru akan terasa ngilu karena pada *dentin* terdapat serat syaraf tome's. Tingkatan rasa ngilu juga berbeda. Jika lubang sudah mencapai hampir

batas pertemuan antara *dentin* dan *pulpa*, biasanya ngilu akan terasa lebih hebat. Jika lubang pun tidak ditambal, maka akan berlanjut ke *pulpa*.

4. Kerusakan di pulpa gigi

Pada tahap ini bukan ngilu lagi yang dirasakan tapi sakit berdenyut bahkan sampai tidak bisa tidur. Karena pada pulpa berkumpul pembuluh darah dan pembuluh syaraf. Jika lubang tidak ditambal juga, maka gigi bisa menjadi *gangraena* (mati). Gigi mati yang dibiarkan akan berakibat terbentuknya abses (kumpulan nanah pada sekitar akar gigi).

2.6.7 Akibat Gigi Berlubang (Karies Gigi)

Menurut *World Health Organization (WHO)*, pada tahun 2012 ada sekitar 60-90 persen penduduk sebuah negara yang mengalami gigi berlubang. Dan gigi berlubang adalah “investasi untuk penyakit-penyakit kronis”. Gigi berlubang yang tidak segera ditangani akan menyebabkan infeksi menyebar ke bagian organ lain, tidak hanya menyebabkan bau mulut, kista periapikal, sinusitis, gusi bengkak dan berdarah. Infeksi di rongga syaraf gigi dapat menyebabkan komplikasi ke beberapa penyakit lainnya bahkan dapat menyebabkan kematian, antara lain:

1. Infeksi ke otak (Wu, 2000)
2. Diabetes Mellitus (Lux, 2004)
3. Peradangan jantung yang terjadi pada lapisan jantung dan otot jantung dan katup jantung (Lux, 2004)
4. Peradangan paru-paru (Lux, 2004)

Pada gigi berlubang terdapat bakteri gigi berlubang. Bakteri ini menggerogoti email gigi yang akhirnya menyebabkan gigi berlubang dan menimbulkan infeksi gigi. Bermula dari infeksi gigi, bakteri ini akan terbawa oleh pembuluh darah, sehingga penyebarannya sangat mudah. Saat terbawa di dalam pembuluh darah, bakteri ini akan sangat mudah bersarang di jantung, terutama bila kondisi tubuh sedang dalam keadaan yang lemah. Karena mengikuti aliran pembuluh darah, kuman tersebut tidak hanya akan menyerang jantung tetapi juga otak, paru-paru, bahkan memperparah kondisi gula darah pada penderita diabetes mellitus (Khrisnan, 2012).

Pada penyakit jantung (Lux, 2004), terdapat sejumlah penelitian menunjukkan, bakteri yang terikut aliran darah bisa memproduksi sejenis enzim yang mempercepat proses pengerasan dinding pembuluh darah, sehingga pembuluh darah menjadi tidak elastis (*aterosklerosis*). Bakteri juga bisa menempel pada lapisan lemak di pembuluh darah. Akibatnya, plak yang terbentuk menjadi makin tebal. Semua kondisi ini menghambat aliran darah ke jantung. Hal ini berarti penyaluran sumber makanan dan oksigen ke jantung juga tersendat. Jika berlangsung terus, jantung tak akan mampu berfungsi secara baik. Maka terjadilah penyakit jantung yang ditakutkan banyak orang.

Pada penyakit kista periapikal (Benarjee dan Watson, 2011), gigi berlubang yang dibiarkan akan menyebabkan peradangan pada jaringan pulpa gigi (*pulpitis*) kemudian terjadi kematian saraf pada gigi dan terbentuk kista periapikal pada ujung akar gigi tersebut. Bila kista membesar, wajah pun menjadi tidak simetri dan bahkan dapat menyebabkan parestesi karena tertekannya syaraf oleh kista tersebut.

Pada penyakit diabetes (Lux, 2004), kerusakan gigi yang parah, bakteri dapat masuk ke aliran darah dan mengganggu sistem kekebalan tubuh. Sel sistem kekebalan tubuh yang rusak melepaskan sejenis protein yang disebut *cytokines*. Unsur itu menyebabkan kerusakan sel pankreas penghasil insulin, hormon yang memicu diabetes.

Pada penyakit sinusitis (Abraham, 1996), sinusitis bisa disebabkan, infeksi gigi berlubang yang tidak ditangani dan meluas hingga menyebabkan *sinusitis maksilaris* yakni radang pada rongga sinus yang letaknya di pipi. Akan tetapi tidak semua gigi berlubang mengakibatkan sinusitis maksilaris. Hanya gigi geraham pertama dan kedua pada bagian atas yang berpotensi. Sebab, ujung akar giginya dekat sekali dengan saluran sinus.

Banyak kasus kematian akibat infeksi karena gigi berlubang, apabila daya tahan tubuh lemah maka infeksi akan sangat mudah menyebar dan akan lebih cepat menyebar jika terdapat komplikasi penyakit lainnya.

2.6.8 Merawat Gigi Berlubang (Karies Gigi)

Cara merawat gigi berlubang yaitu (Benarjee dan Watson, 2012):

1. Tambal gigi berlubang

Menambal gigi berlubang adalah pengisian materi yang menggantikan gigi pada area yang berlubang setelah kerusakan menjadi permanen. Tambalan gigi, kadang-kadang disebut restorasi, adalah pilihan pengobatan gigi berlubang yang utama saat pembusukan telah melampaui tahap enamel, kerusakan awal. Dokter gigi mempelajari bagian yang membusuk pada gigi Anda, kemudian mengisi pada celah untuk mengembalikan bentuk gigi.

Tambalan terbuat dari berbagai macam bahan, resin komposit, porselen, atau kombinasi dari beberapa bahan. Bahan tambal yang digunakan dapat bermacam-macam, misalnya resin komposit (penambalan dengan sinar dan bahannya sewarna gigi), glass ionomer cement, kompomer, atau amalgam.

2. Mahkota gigi.

Jika mengalami kerusakan gigi yang luas atau melemah, mungkin perlu sebuah mahkota gigi, daripada mengisi gigi bolong untuk mengobati dan mengembalikan gigi. Sebuah mahkota gigi adalah penutup kustom dipasang yang menggantikan seluruh mahkota gigi alami. Pada karies yang sudah mengenai jaringan pulpa, perlu dilakukan perawatan saluran akar/syaraf.

3. Mencabut gigi berlubang

Bila lubang terlalu besar dan tidak memungkinkan untuk ditambal, berarti gigi harus dicabut. Beberapa gigi dapat menjadi busuk begitu parah dan tidak dapat dikembalikan dengan pengobatan apapun, gigi ini harus dihilangkan/dicabut dan bisa dipasang dengan gigi palsu jika ingin. Akan tetapi sama seperti proses penambalan gigi, maka gigi juga tidak dapat langsung dicabut saat gigi masih terasa sakit. Hal ini disebabkan saat merasakan sakit gigi, maka obat anestesi (obat kebal agar tidak terasa sakit saat gigi dicabut) tidak dapat menembus akar gigi, sehingga saat dicabut akan menyebabkan sakit yang luar biasa. Proses pencabutan gigi baru bisa dilakukan saat gigi sudah tidak terasa sakit dan untuk menghilangkan rasa sakit dokter akan mematikan saraf gigi.

2.6.9 Mencegah Gigi Berlubang (Karies Gigi)

Cara mencegah gigi berlubang yaitu (Benarjee dan Watson, 2012):

1. Menyikat gigi yang benar

Beberapa pakar menyebutkan bahwa menyikat gigi sebaiknya dilakukan antara satu sampai dua menit. Cara menyikat gigi yang benar adalah dari atas ke bawah bukan dari kiri ke kanan atau sebaliknya. Usahakan menyikat gigi di depan cermin agar kotoran terlihat dan lebih mudah membersihkannya. Khusus untuk gigi geraham bersihkan secara seksama, karena sisa-sisa makanan biasanya lebih banyak menempel di gigi geraham.

2. Mengurangi bakteri penyebab gigi berlubang

Salah satu penyebab gigi berlubang adalah kondisi keasaman rongga mulut akibat bakteri. Untuk mencegah keasaman mulut, berkumurlah beberapa detik setelah makan untuk mengurangi sisa-sisa makanan yang menempel di sela gigi. Gunakan obat kumur yang mengandung anti-bakteri untuk mengurangi perkembangan bakteri penyebab gigi berlubang.

3. Waktu menyikat gigi yang tepat

Menyikat gigi minimal dua kali sehari, pagi dan malam hari. Menyikat gigi setelah makan dan sebelum tidur, bahkan biasakan membawa sikat gigi kemanapun pergi.

4. Memilih sikat gigi yang benar

Sikat gigi yang benar yaitu yang memiliki bulu lembut, tidak kasar. Jika bulu sikat gigi kasar, maka dapat membuat lapisan email cepat mengikis dan membuka lapisan tengah gigi terbuka. Jika sudah terbuka, akan lebih berpotensi untuk berlubang. Selain itu, perhatikan juga kebersihan sikat gigi Anda. Ini juga merupakan penyebab gigi berlubang. Jangan digunakan terlalu lama, setidaknya dalam waktu 3 bulan sudah harus menggantinya. Memilih sikat gigi sesuai dengan karakter rongga mulut dan bisa menjangkau gigi paling belakang (gigi geraham) sesuai dengan struktur mulut.

5. Mengurangi konsumsi makanan dan minuman manis

Gula adalah zat yang dapat menjadi penyebab gigi mudah berlubang. Makanan manis umumnya menggunakan gula sebagai pemanis. Makanan manis dapat memicu keasaman rongga mulut pemicu gigi berlubang. Sebaiknya segera berkumur setelah mengonsumsi makanan manis. Makan makanan yang cukup sehat dan bergizi seperti buah-buahan yang mengandung serat dan air serta mengonsumsi makanan yang mengandung kalsium, fosfor, vitamin A, D, dan C yang cukup dapat meningkatkan kesehatan dan kekuatan *enamel*.

6. Kumur setelah makan

Berkumur-kumur dapat dilakukan setelah makan agar sisa makanan tidak terus menempel dan mengurangi keadaan asam dalam gigi.

7. Memilih pasta gigi yang tepat

Pasta gigi mengandung *Fluoride* dapat memperkuat lapisan enamel dan denting, khususnya pada anak-anak sebelum gigi permanen keluar. *Fluoride topikal* dapat mengisi rongga-rongga pada enamel yang belum terbentuk sempurna dan menutup lubang gigi yang kecil serta mengurangi pembentukan lubang pada gigi permanen.

8. Sealant (Penutup gigi)

Merupakan suatu lapisan plastik yang diletakkan pada lubang dan celah di bagian atas permukaan gigi geraham untuk mencegah terbentuknya lubang pada gigi bayi dan orang dewasa yang mudah berlubang.

8. Melakukan *flossing*

Flossing dilakukan sekali dalam sehari untuk mengangkat plak dan sisa makanan yang tersangkut dicelah gigi, terutama di gigi bagian depan. Menggunakan benang gigi relatif lebih aman dibanding tusuk gigi karena tidak berujung lancip yang berpotensi melukai gusi. Menggunakan tusuk gigi juga dapat menyebabkan celah gigi semakin melebar sehingga makanan mudah terselip.

9. Pemeriksaan gigi

Memeriksa gigi secara teratur ke dokter gigi tiap 6 bulan sekali meskipun tidak merasakan sakit gigi. Hal ini diperlukan agar dokter dapat mendeteksi lubang kecil yang terjadi pada gigi dan dapat ditangani segera agar lubang

tidak semakin besar. Dapat juga dideteksi bagian gigi yang tidak rata atau berlekuk yang dapat menyebabkan gigi sulit dibersihkan.

2.7 Metode *Waterfall*

Metode *Waterfall* sering disebut dengan *classic life cycle* (Pressman, 2010). Metode ini merupakan metode pengembangan perangkat lunak terstruktur yang paling dikenal dan banyak digunakan secara luas. Oleh sebab itu, metode ini hampir selalu digunakan untuk pengembangan perangkat lunak.

Metode *Waterfall* pertama kali diperkenalkan oleh Royce pada tahun 1970 (Petersen et.al., 2009) dengan 7 (tujuh) tahapan yang berurut walaupun juga mempunyai *feedback loop* antar tahapan jika diperlukan, seperti terlihat pada gambar 2.32 (Royce, 1970). Metode ini mengalami banyak perbaikan dan perubahan diantaranya adalah perubahan langkah dari 7 (tujuh) menjadi 5 (lima) tahapan (Pressman, 2010), (Sommerville, 2011) seperti terlihat pada gambar 2.33 dan 2.34.

Pertimbangan menggunakan versi Sommerville karena metode ini merupakan versi terbaru dari metode *Waterfall*. Metode ini mempunyai tahapan-tahapan sebagai berikut:

a. *Requirements analysis and definition*

Layanan sistem, kendala, dan tujuan ditetapkan oleh hasil konsultasi dengan pengguna yang kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.

b. *System and software design*

Tahapan perancangan sistem mengalokasikan kebutuhan-kebutuhan sistem baik perangkat keras maupun perangkat lunak dengan membentuk arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan penggambaran abstraksi sistem dasar perangkat lunak dan hubungannya.

c. *Implementation and unit testing*

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.

d. *Integration and system testing*

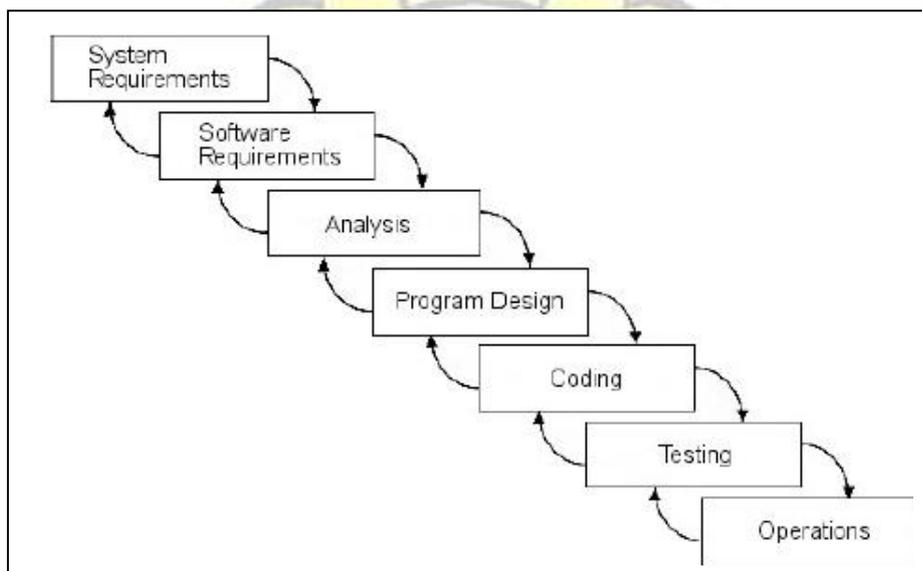
Unit-unit individu program atau program digabung dan diuji sebagai sebuah sistem lengkap untuk memastikan apakah sesuai dengan kebutuhan perangkat lunak atau tidak. Setelah pengujian, perangkat lunak dapat dikirimkan ke pengguna.

e. *Operation and maintenance*

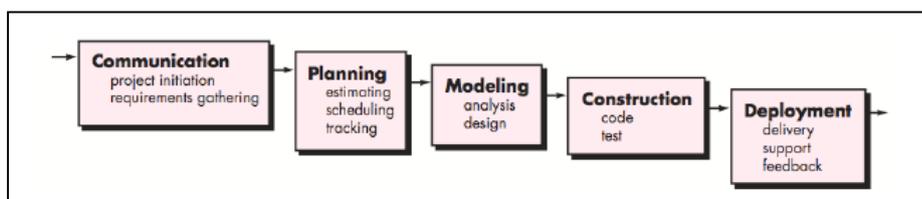
Biasanya (walaupun tidak selalu), tahapan ini merupakan tahapan yang paling panjang. Sistem dipasang dan digunakan secara nyata. *Maintenance* melibatkan pembetulan kesalahan yang tidak ditemukan pada tahapan-tahapan sebelumnya, meningkatkan implementasi dari unit sistem, dan meningkatkan layanan sistem sebagai kebutuhan baru.

Pada prinsipnya, setiap tahapan di metode *Waterfall* menghasilkan satu atau lebih dokumen yang sudah disetujui. Tahap berikutnya tidak dapat dimulai

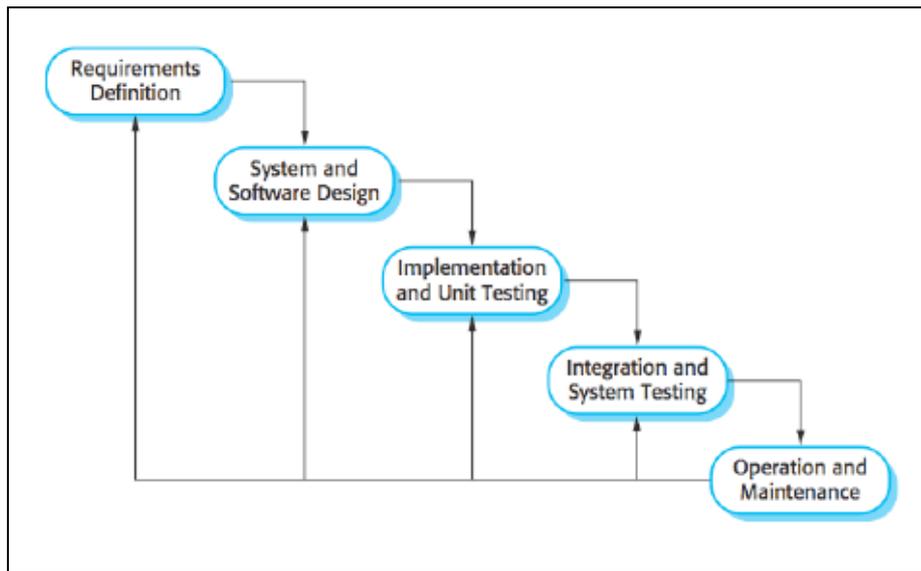
sebelum tahapan sebelumnya selesai. Dalam tataran praktis, tahapan-tahapan tersebut saling tumpang tindih (*overlap*) dan memberikan informasi satu sama lain. Pada waktu perancangan (*design*), masalah-masalah dengan persyaratan diidentifikasi. Pada waktu pengkodean (*coding*), dapat ditemukan masalah perancangan, walaupun juga masalah lainnya. Proses pengembangan perangkat lunak bukan merupakan model linier yang sederhana karena juga melibatkan umpan balik (*feedback*) dari satu tahapan ke tahapan lainnya. Dokumen yang dihasilkan pada setiap tahapan ada kemungkinan harus diubah supaya sesuai dengan perubahan yang sudah dibuat (Sommerville, 2011).



Gambar 2.32 Metode *Waterfall* versi Royce (1970)



Gambar 2.33 Metode *Waterfall* versi Pressman (2010)



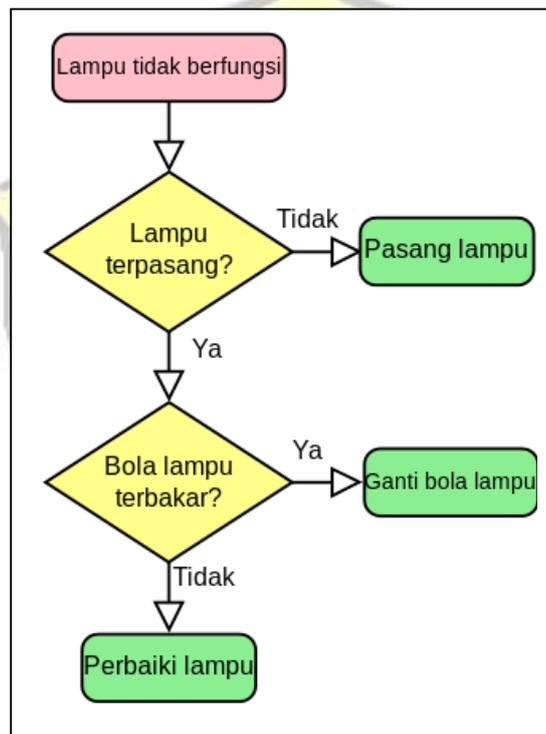
Gambar 2.34 Metode *Waterfall* versi Sommerville (2011)

Pada tahapan akhir (*operation and maintenance*) perangkat lunak mulai digunakan. Kesalahan dan kelalaian pada *software requirements* awal dapat ditemukan. Beberapa kesalahan perancangan atau yang lainnya terjadi dan kebutuhan akan fungsi baru teridentifikasi. Oleh karena itu, sistem harus berevolusi (membuat perubahan) untuk tetap dapat digunakan. Membuat perubahan ini (*maintenance software*) dapat melibatkan pengulangan tahapan sebelumnya (Sommerville, 2011).

Metode *Waterfall* merupakan contoh dari tahapan yang *plan-driven*, yaitu secara prinsip pengembang harus merencanakan dan menjadwalkan semua kegiatan tahapan sebelum mulai mengembangkan perangkat lunak. Komitmen dibuat di awal tahapan (Sommerville, 2011).

2.8 Flowchart Diagram

Flowchart atau diagram alir merupakan sebuah diagram dengan simbol-simbol grafis yang menyatakan aliran algoritma atau proses yang menampilkan langkah-langkah yang disimbolkan dalam bentuk kotak, beserta urutannya dengan menghubungkan masing masing langkah tersebut menggunakan tanda panah (IBM, 1969). Diagram ini bisa memberi solusi selangkah demi selangkah untuk penyelesaian masalah yang ada di dalam proses atau algoritma tersebut.

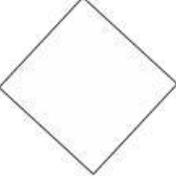


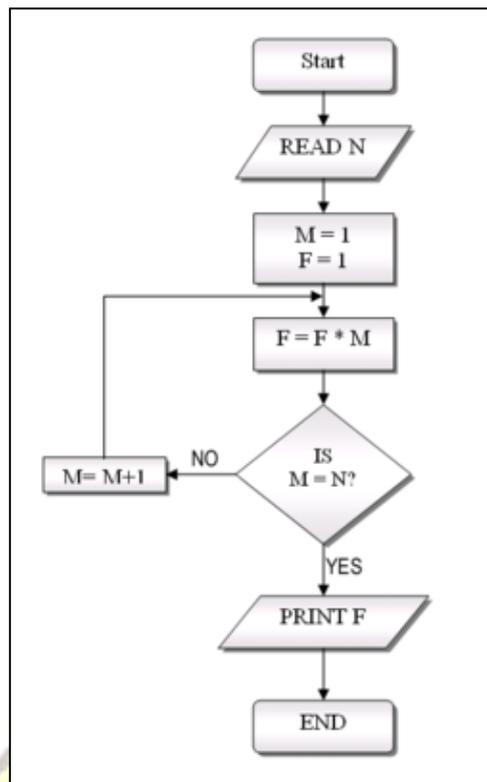
Gambar 2.35 Contoh diagram alir proses yang berhubungan dengan kerusakan lampu (Wikipedia, 2014)

2.8.1 Simbol-simbol *Flowchart Diagram*

Gambar berikut adalah simbol *flowchart* yang umum digunakan.

Tabel 2.6 Tabel Simbol *Flowchart*

NO	GAMBAR	NAMA	KETERANGAN
1		Proses / Langkah	Menyatakan kegiatan yang akan ditampilkan dalam diagram alir.
2		Titik Keputusan	Proses / Langkah dimana perlu adanya keputusan atau adanya kondisi tertentu. Di titik ini selalu ada dua keluaran untuk melanjutkan aliran kondisi yang berbeda.
3		Masukan / Keluaran Data	Digunakan untuk mewakili data masuk, atau data keluar.
4		Terminasi	Menunjukkan awal atau akhir sebuah proses.
5		Garis alir	Menunjukkan arah aliran proses atau algoritma.
6		Kontrol / Inspeksi	Menunjukkan proses / langkah dimana ada inspeksi atau pengontrolan.



Gambar 2.36 Contoh diagram alir untuk menghitung factorial N (N!) (Wikipedia, 2014)

2.8.2 Jenis-jenis *Flowchart Diagram*

Sterneckert (2003) menyarankan untuk membuat model diagram alir yang berbeda sesuai dengan perspektif pemakai (*managers, system analysts and clerks*) sehingga dikenal ada 4 jenis diagram alir secara umum:

- **Diagram Alir Dokumen**, menunjukkan kontrol dari sebuah sistem aliran dokumen.
- **Diagram Alir Data**, menunjukkan kontrol dari sebuah sistem aliran data.
- **Diagram Alir Sistem**, menunjukkan kontrol dari sebuah sistem aliran secara fisik.

- **Diagram Alir Program**, menunjukkan kontrol dari sebuah program dalam sebuah sistem.

2.9 Pemodelan UML

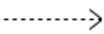
Unified Modeling Language (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat *tool* untuk mendukung pengembangan sistem tersebut. (Kroenke, 2005).

Beberapa contoh diagram pada UML yang digunakan untuk memodelkan atau membangun aplikasi berorientasi objek yaitu *Use Case Diagram* dan *Activity Diagram*.

2.9.1 Use Case Diagram

Use case adalah gambaran fungsionalitas dari suatu sistem, sehingga *customer* atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun (Object Management Group, 2007).

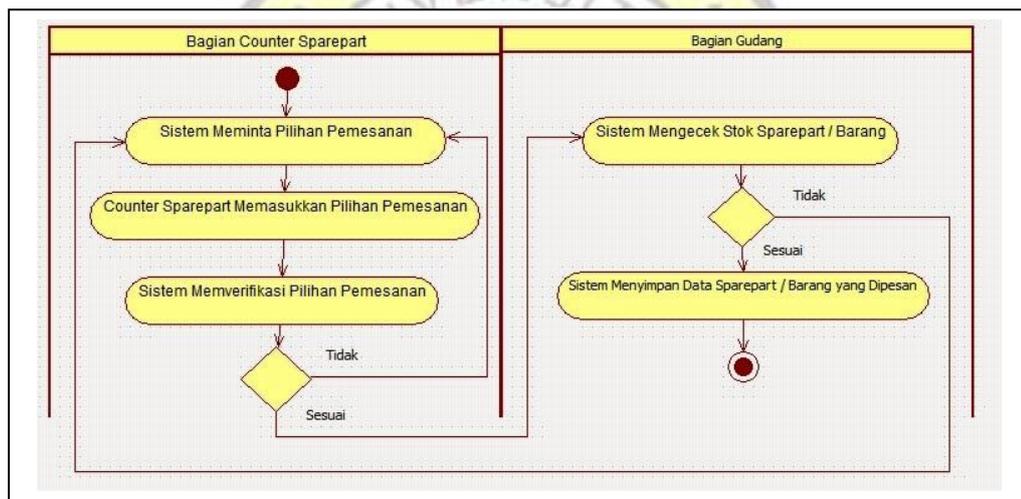
Tabel 2.7 Tabel Notasi Gambar *Use Case*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang

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

2.9.2 Activity Diagram

Activity Diagram digunakan untuk memodelkan prosedur seluruh tahapan alur kerja yang merupakan bagian dari aktivitas yang lebih besar (Bell, 2003). Diagram ini mengandung aktivitas, pilihan tindakan, perulangan dan hasil dari aktivitas tersebut. *Activity Diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi untuk membantu memahami proses secara keseluruhan. *Activity diagram* dibuat berdasarkan sebuah atau beberapa *use case* pada *use case diagram*.



Gambar 2.37 Contoh *Activity Diagram* untuk Administrasi Dealer Motor (Education For You, 2014)

2.10 Blender

Blender adalah sebuah software 3D modeling dan animasi gratis dan *open source*. Blender adalah program komputer grafis yang memungkinkan untuk menghasilkan gambar dan animasi kualitas tinggi menggunakan geometri tiga dimensi. Salah satu hal yang membuat Blender yang berbeda dan istimewa dibandingkan dengan software 3D sebanding lainnya adalah fakta bahwa Blender adalah gratis dan *open source* (Gumster, 2009). Pada tugas akhir ini, Blender digunakan untuk membuat objek 3D yang ditampilkan sebagai objek AR.

2.10.1 Sejarah Blender

Studio animasi Belanda Neo Geo dan *Not a Number Technologies* (NaN) mengembangkan Blender sebagai aplikasi pribadi, dengan penulis utama adalah Ton Roosendaal. Nama Blender terinspirasi oleh lagu oleh Yello, dari album Baby. Ton Roosendaal mendirikan NaN pada bulan Juni 1998 untuk lebih mengembangkan dan mendistribusikan program. Mereka awalnya mendistribusikan programnya secara gratis untuk dievaluasi (*shareware*) sampai NaN bangkrut pada tahun 2002.

Pada tanggal 18 Juli 2002, dalam menghadapi kebangkrutan Roosendaal memulai kampanye "Free Blender", sebagai orang pertama yang membuat sebuah proyek kecil untuk mendapatkan sedikit uang dari sebagian banyak orang. Kampanye bertujuan untuk *open-sourcing* Blender untuk pembayaran satu kali yang dikumpulkan dari komunitas. Pada tanggal 7 September 2002, diumumkan bahwa mereka telah mengumpulkan cukup dana dan akan merilis kode sumber Blender. Sekarang, Blender adalah *software gratis*, *open source* dan bagian

karyawan paruh waktu dan karyawan tetapnya dari Blender Institute yang dikembangkan oleh komunitas. Blender dibawah GNU *General Public License* pada 13 Oktober 2002.

2.10.2 Fitur-fitur Blender

Fitur-fitur Blender menghasilkan rendering dengan hasil yang nyata/realistis (*Photorealistic Rendering*) yang disebut *Cycles* dan dapat membuat model dengan cepat (*Fast Modeling*) yang dapat membuat, menciptakan, mengubah dan mengedit model dengan mudah. Material yang digunakan terlihat nyata/realistis (*Realistic Materials*) dan transformasi model menjadi karakter dapat berpose dengan mudah (*Fast Rigging*). Blender juga dapat mengubah karakter menjadi animasi yang mengesankan dengan menggunakan *Animation Toolset* dan *Sculpting*, serta dapat menggunakan tekstur gambar atau warna sendiri langsung ke model dengan cepat (*Fast UV Unwrapping*).

Selain itu, Blender dilengkapi dengan kompositor penuh (*Full Compositor*) dibangun tepat didalamnya dan tidak ada lagi mengekspor ke program pihak ketiga, dapat melakukan semuanya tanpa meninggalkan program. Blender juga menghasilkan simulasi yang menakjubkan (*Amazing Simulation*) dengan menggunakan *Fluid* (cairan), *Smoke* (asap), *Hair* (rambut), *Cloth* (kain), *Rigid Body Physic* (Fisik Tubuh), *Particles* (partikel). Dengan *game engine* yang lengkap, Blender dapat digunakan untuk membuat *game* 3D dengan fitur lengkap (*Game Creation*) dan dapat mengimpor rekaman mentah, melacak rekaman, *mask area* dan melihat gerakan *live* kamera dalam adegan 3D dengan pelacakan kamera dan objek (*Camera and Object Tracking*).

Ekstensi library (*Library of Extensions*) pada Blender dapat diaktifkan atau dinonaktifkan dengan mudah, Blender juga dilengkapi dengan *Video Editing* dengan adanya *Video Editor* untuk melakukan tindakan dasar seperti pemotongan video dan *splicing*, serta tugas-tugas yang lebih kompleks seperti *video masking*. *User* juga dapat menyesuaikan tata letak mereka sepenuhnya karena Antar Muka Blender yang fleksibel (*Flexible Interface*).

Blender mendukung dukungan format file impor/ekspor untuk berbagai program yang berbeda, termasuk:

- **Gambar**

JPEG, JPEG2000, PNG, TARGA, OpenEXR, DPX, Cineon, Radiance HDR, SGI Iris, TIFF

- **Video**

AVI, MPEG dan Quicktime (di OSX).

- **3D**

3D Studio (3DS), COLLADA (DAE), Filmbox (FBX), Autodesk (DXF), Wavefront (OBJ), DirectX (x), Lightwave (LWO), Motion Capture (BVH), SVG, Stanford ply, STL, VRML , VRML97, X3D.

2.11 Klinik Gigi Kementerian Sekretariat Negara RI

Berdasarkan Peraturan Menteri Sekretaris Negara RI Nomor 3 Tahun 2015 tentang Organisasi dan Tata Kerja Kementerian Sekretariat Negara, Bagian Pelayanan Kesehatan mempunyai tugas melaksanakan pelayanan kesehatan bagi pejabat, pegawai beserta keluarganya di lingkungan Kementerian Sekretariat

Negara, dan pengoordinasian kegiatan kenegaraan pada bidang kesehatan dengan instansi terkait bekerjasama dengan dokter kepresidenan.

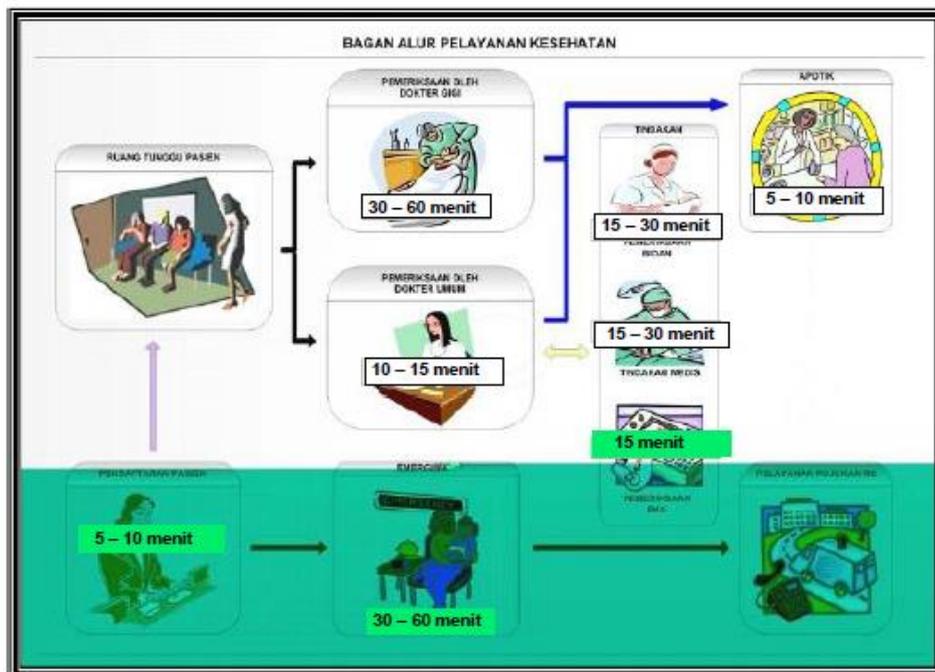
Dalam melaksanakan tugas sebagaimana dimaksud dalam Pasal 123, Bagian Pelayanan Kesehatan menyelenggarakan fungsi:

- a. pelaksanaan pelayanan klinik umum, gigi, dan spesialis;
- b. pelaksanaan pelayanan pemberian rujukan rumah sakit;
- c. pencatatan medik dan pelaporan;
- d. pelaksanaan dukungan administrasi layanan medis yang meliputi aspek promotif, preventif, kuratif, dan rehabilitatif;
- e. koordinasi pelayanan kesehatan di Istana-istana Kepresidenan di Daerah;
- f. koordinasi kegiatan kenegaraan pada bidang kesehatan dengan instansi terkait bekerja sama dengan dokter kepresidenan; dan
- g. pelaksanaan urusan administrasi pelayanan kesehatan.

Bagian Pelayanan Kesehatan terdiri atas:

- a. Subbagian Tata Usaha Pelayanan Kesehatan;
mempunyai tugas melakukan pencatatan medik, pelaporan, dan urusan administrasi pelayanan kesehatan.
- b. Subbagian Pelayanan Medis; dan
mempunyai tugas memberikan dukungan administrasi pelayanan medis yang meliputi aspek promotif, preventif, kuratif dan rehabilitatif bagi pelaksanaan pelayanan Kesehatan di lingkungan Kementerian Sekretariat Negara RI.
- c. Kelompok Jabatan Fungsional Tenaga Medis dan Paramedis.

Dasar hukum pelayanan kesehatan dasar di lingkungan Kementerian Sekretariat Negara RI yaitu Peraturan Menteri Sekretaris Negara Nomor 2 Tahun 2011 yang diperbaharui menjadi Peraturan Menteri Sekretaris Negara RI Nomor 3 Tahun 2015 tentang Organisasi dan Tata Kerja Kementerian Sekretariat Negara dan Peraturan Menteri Sekretaris Negara Nomor 21 tahun 2011 tentang Petunjuk Pelaksanaan Penyusunan dan Evaluasi *Standard Operating Procedures* Kementerian Sekretariat Negara.



Gambar 2.38 Bagan Alur Pelayanan Kesehatan (Permensesneg Nomor 21, 2011)

Uraian Prosedur:

1. Pendaftaran pasien (5- 10 menit)
 - a. Setiap pasien yang akan mendapatkan pelayanan kesehatan wajib mendaftarkan diri dengan mengisi buku register di Bagian Pendaftaran.
 - b. Khusus untuk pelayanan pemeriksaan gigi, pasien akan dijadwalkan terlebih dahulu untuk mendapatkan pelayanan gigi.
2. Persiapan pemeriksaan pasien (waktu tergantung banyaknya pasien)

Pasien menunggu di ruang tunggu pasien sebelum mendapatkan pelayanan kesehatan.
3. Pelayanan pemeriksaan oleh Dokter Umum (10-15 menit)
 - a. Sebelum pasien diperiksa oleh dokter umum, status *medical record* pasien harus sudah ada di ruangan dokter tersebut.
 - b. Pasien akan dipanggil masuk ke ruangan pemeriksaan dokter sesuai nomor antri/urut pendaftarannya.
 - c. Dokter memberikan pelayanan, mendengarkan keluhan pasien, dan melakukan pemeriksaan dengan seksama, membuat diagnosis, dan memberikan terapi sesuai dengan diagnosa yang dibuat, serta memberisaran kepada pasien sesuai dengan penyakitnya.
 - d. Dokter menuliskan keluhan, hasil pemeriksaan, diagnosa dan terapi pada status *medical record* pasien.
4. Pelayanan pemeriksaan gigi oleh Dokter Gigi (30-60 menit/tergantung tindakan/terapi yang diberikan)
 - a. Pasien yang akan diperiksa atau dilakukan tindakan sebelumnya sudah dijadwalkan terlebih dahulu di bagian pendaftaran.

- b. Sebelum pasien diperiksa oleh dokter gigi, status *medical record* pasien harus sudah ada diruangan dokter gigi tersebut.
 - c. Dokter gigi wajib memberikan pelayanan , mendengarkan keluhan pasien dan melakukan pemeriksaan dengan seksama, membuat diagnosis dan memberikan tindakan /terapi sesuai dengan diagnosa yang di buat serta memberi saran kepada pasien sesuai dengan penyakitnya. Memberikan terapi sesuai dengan diagnosa yang dibuat, serta memberi saran kepada pasien sesuai dengan penyakitnya.
 - d. Dokter menuliskan keluhan, hasil pemeriksaan, diagnosa dan terapi pada status *medical record* pasien.
5. Pelayanan pemeriksaan kebidanan (15-30 menit/tergantung tindakan/terapi yang diberikan.
- a. Bidan memberikan pelayanan kebidanan, mendengarkan keluhan pasien dan melakukan pemeriksaan dengan seksama.
 - b. Bidan mengkonsultasikan dengan Dokter Umum penanggungjawab hasil pemeriksaanya.
 - c. Dokter umum bersama bidan memberikan tindakan/terapi sesuai dengan diagnosa yang di buat serta memberi saran kepada pasien sesuai dengan penyakitnya.
 - d. Bidan memberikan penjelasan kepada pasien mengenai pemeriksaan kebidanan yang akan dilakukan.
 - e. Bidan menuliskan keluhan, hasil pemeriksaan, diagnosa, tindakan/terapi pada status *medical record* pasien setelah di konsultasikan dengan dokter umum penanggung jawab.

6. Pelayanan tindakan medis (15-30 menit/tergantung tindakan/terapi yang diberikan)
 - a. Tindakan medis pada pasien dapat dilakukan dengan penjadwalan terlebih dahulu atau segera dilakukan tindakan medis jika dalam keadaan darurat.
 - b. Sebelum tindakan medis dilakukan kepada pasien, Dokter Umum menjelaskan kepada pasien tentang tindakan yang akan dilakukan.

7. Pelayanan pemeriksaan EKG (15 menit)
 - a. Petugas EKG melakukan pemeriksaan EKG atas permintaan Dokter Umum.
 - b. Petugas EKG memberikan penjelasan mengenai hal-hal yang akan dilakukan dalam pemeriksaan kepada pasien.
 - c. Petugas memberikan hasil EKG langsung kepada Dokter Umum yang melakukan permintaan pemeriksaan EKG.
 - d. Dokter Umum memberitahukan hasil pemeriksaan EKG kepada pasien.

8. Pelayanan *emergency*
 - a. Pasien yang mendapatkan pelayanan *emergency* harus sesegera mungkin diberikan pertolongan pertama.
 - b. Dokter Umum dengan cepat melakukan pemeriksaan, penegakkan diagnosis sementara dan memberikan pertolongan pertama.
 - c. Kondisi pasien selalu dipantau selama dalam pelayanan *emergency*.
 - d. Jika pasien kondisi membaik diberikan terapi dan saran mengenai penyakitnya.

- e. Jika kondisi pasien memburuk segera di berikan pelayanan rujukan rumah sakit tertentu sesuai dengan penyakit yang diderita pasien.
9. Pelayanan rujukan pasien
- a. Petugas memonitor dan memberikan pertolongan pertama sesuai prosedur kepada pasien selama dalam perjalanan menuju rumah sakit.
 - b. Petugas memberikan laporan kepada dokter jaga yang ada di rumah sakit rujukan mengenai kondisi yang dialami pasien.
 - c. Petugas memberikan laporan kepada Kepala Bagian Pelayanan Kesehatan setelah selesai merujuk pasien.
 - d. Petugas memonitor/memantau perkembangan dari pasien yang dirujuk ke Rumah Sakit.
10. Pelayanan apotik (5-10 menit)
- a. Petugas mengeluarkan obat sesuai resep yang ditulis oleh Dokter Umum/Gigi.
 - b. Petugas menanyakan kepada dokter yang menulis resep jika ada tulisan pada resep yang kurang jelas atau jika ada obat yang tidak tersedia di apotik pada Bagian Pelayanan Kesehatan.
 - c. Obat yang diberikan kepada pasien oleh petugas dijelaskan aturan minum atau cara pemakaiannya.
 - d. Petugas menuliskan pada obat yang diberikan identitas pasien dengan jelas, Nama, NIP pasien.
 - e. Pasien menunggu obat yang akan diberikan oleh apotik diruang tunggu pasien.

Berdasarkan Laporan Kunjungan Pasien ke Dokter Gigi pada tahun 2014 selama satu tahun, total pasien yang berkunjung ke Dokter gigi (PNS dan Non PNS) adalah 1057 orang dan drg. Wibowo mendapatkan kunjungan pasien terbanyak yaitu sebanyak 571 orang. Menurut drg. Wibowo yang merupakan Dokter Gigi pada Kementerian Sekretariat Negara RI mengatakan bahwa banyak sekali masyarakat dan pasien yang mengalami masalah gigi berlubang dan masih banyak yang kurang memahami bahkan tidak mengetahui informasi mengenai masalah gigi berlubang. Selain itu, pada umumnya informasi mengenai masalah gigi berlubang disampaikan hanya secara lisan dengan menggunakan replika gigi yang tidak bisa menampilkan tahapan terjadinya gigi berlubang.

