

BAB 2 LANDASAN TEORI

2.1. Landasan Teori

Dalam bab ini menjelaskan landasan teori yaitu kumpulan beberapa teori yang mendefinisikan tentang sesuatu dan dikemukakan oleh beberapa para ahli. Dan komponen-komponen apa saja yang mendukung dalam pembangunan sistem perusahaan dengan pemanfaatan teknologi *Augmented Reality*.

2.1.1 Definisi *Augmented Reality*

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. (Ronald T. Azuma, 2008)

Augmented Reality menawarkan efek ajaib dari pencampuran dunia fisik dengan dunia maya dan membawa aplikasi dari layar pengguna ke tangan pengguna. (Jens Grubert dan Dr. Raphael Grasset, 2013)

Augmented Reality dapat diklasifikasikan menjadi dua berdasarkan adanya tindakan pengguna marker yaitu:

a. *Marker Based Tracking Augmented Reality*

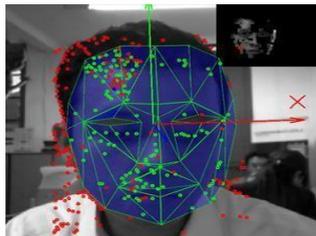
Sebuah metode yang memanfaatkan marker yang biasanya berupa ilustrasi hitam putih berbentuk persegi atau lainnya dengan batas hitam tebal dan latar belakang putih. Melalui posisi yang diharapkan pada sebuah kamera komputer atau *smartphone*, maka komputer atau *smartphone* akan melakukan proses menciptakan dunia *virtual 2D* atau *3D*.

b. *Markerless Augmented Reality*

Dengan metode ini pengguna tidak menggunakan sebuah *marker* untuk menampilkan objek 3D atau yang lainnya, sekalipun dinamakan dengan *markerless* namun aplikasi tetap berjalan dengan melakukan pemindaian terhadap objek, namun ruang lingkup yang akan dipindai lebih luas dibandingkan dengan *marker based tracking*. Adapun beberapa teknik yang digunakan dalam *Markerless Augmented Reality* adalah sebagai berikut :

1) *Face Tracking*

Dengan menggunakan algoritma yang banyak dikembangkan komputer dapat mengenali wajah manusia secara umum dengan cara mengenali posisi mata, hidung, dan mulut manusia, kemudian akan mengabaikan objek – objek disekitarnya seperti pohon, rumah, dan benda – benda lainnya.

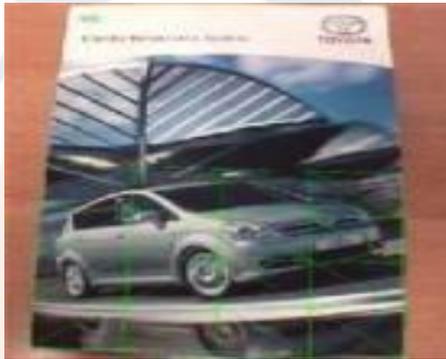


Gambar Error! No text of

(sumber : <http://www.saungit.org>)

2) *3D Object Tracking*

Teknik *3D object tracking* dapat mengenali semua bentuk benda yang ada di sekitar, seperti mobil, meja, televisi, dan lain-lain.



Gambar Error! No text of specified style in

document..2 *3D Object Tracking*

(Sumber : <http://www.saungit.org>)

3) *Motion Tracking*

Pada teknik ini komputer dapat menangkap gerakan, *motion tracking* telah mulai menggunakan ekstensif untuk memproduksi film-film yang akan mencoba menyimulasikan gerakan.



Gambar Error! No text of specified style in document..3 *Motion Tracking*

(Sumber : <http://www.saungit.org>)

Implementasi AR dengan menggunakan marker khusus yang telah dikenali oleh perangkat milik pengguna. Terdapat beberapa jenis marker yang pada umumnya digunakan pada implementasi AR, antara lain:

1. *Barcode Standards*. *2D-barcode* umumnya berbentuk persegi atau persegi panjang (garis). Terdiri dari dua warna yaitu hitam dan putih. Hitam menyimpan data bit 0, dan putih menyimpan data bit 1. *2D-barcode* biasanya terdiri dari *object* dengan garis putih dan hitam yang tebal dan tipis untuk mempermudah pembacaan. Bentuk lain *2D-barcode* adalah *QR Code* yang terdiri dari penanda object berbentuk persegi berwarna hitam putih.
2. *Circular Marker*. Sesuai dengan namanya, penanda yang berbentuk lingkaran pada sisi luarnya, semakin banyak lingkaran di dalam *Circular Marker* (CM) semakin tepat posisi objek AR yang digambarkan. Pada umumnya CM digunakan pada implementasi AR yang membutuhkan akurasi tinggi, tetapi memungkinkan pemrosesan informasi secara *offline*. Sama seperti *Barcode*, CM juga terdiri dari warna hitam dan putih.

3. *Image Marker*. Penanda berupa gambar yang menggunakan warna natural sebagai *marker*. *Image marker* biasanya dikelilingi *frame* tertentu yang membantu deteksi dan rotasi *marker* tersebut. Pada umumnya digunakan jika *marker* tidak ingin dihilangkan pada saat objek AR ditanamkan ke dalam foto yang telah diproses.

2.1.1.1 Keunggulan Marker Based Tracking AR

1. Dari aspek akurasi, AR berbasis Marker lebih akurat dibandingkan dengan Markerless AR. Ini disebabkan karena GPS dan GS memiliki lebih banyak distraksi seperti *microwave radio signal*, ketidakakuratan jam, dan gangguan sensor *magnetic* pada perangkat pengguna yang dapat menyebabkan ketidakakuratan posisi objek yang ditampilkan. Tetapi penggunaan marker sebagai substitusi GPS dan GS, membuat tingkat akurasi penempatan object lebih tinggi dari *Markerless AR*.
2. Dari aspek metode pemrosesan data, tidak seperti *Markerless AR* dimana data diproses secara *online*, AR berbasis *Marker* memungkinkan data diproses secara *offline*. Seperti yang sudah kita ketahui bersama, pemrosesan data yang bersifat *offline* tidak akan memberatkan kapasitas *memory* pada perangkat milik pengguna. Tetapi jika data diproses secara *online*, akan membutuhkan waktu dan *bandwidth* untuk mengirim data menuju server untuk melayani permintaan user.
3. Dari aspek interaksi dengan *object*, akan lebih mudah dilakukan karena dapat disisipkan beberapa

marker tambahan di bagian tubuh pengguna. *Marker* ini nantinya akan memudahkan perangkat milik pengguna membaca interaksi yang dilakukan oleh pengguna. Terlebih lagi, dikarenakan objek yang ditampilkan lebih spesifik berdasarkan *marker* yang dibaca oleh perangkat milik pengguna, hal ini menyebabkan dua objek tidak berada pada satu posisi tertentu. Hal ini tentunya akan menguntungkan AR berbasis *Marker* karena ambiguitas yang ditimbulkan dalam pembacaan objek dapat dihilangkan. (Mustika, 2015)

Jadi, *Augmented Reality* adalah teknologi yang menggabungkan benda maya dua atau tiga dimensi ke dalam lingkungan nyata dan benda maya tersebut diproyeksikan secara real dalam waktu yang nyata.

2.1.2 Definisi Kinerja

Kinerja adalah suatu keadaan yang berkaitan dengan keberhasilan organisasi dalam menjalankan misi yang dimilikinya yang dapat diukur dari tingkat produktivitas, kualitas layanan, responsivitas, responsibilitas, dan akuntabilitas. (Tangklisan, 2008).

Kinerja atau *performance* merupakan gambaran mengenai tingkat pencapaian pelaksanaan suatu program kegiatan atau kebijakan dalam mewujudkan sasaran, tujuan, visi dan misi organisasi yang dituangkan melalui perencanaan strategis suatu organisasi. (Moeheriono, 2012).

Kinerja berasal dari kata *performance* yang berarti hasil pekerjaan atau prestasi kerja. Namun perlu dipahami bahwa kinerja itu bukan sekedar hasil pekerjaan atau prestasi kerja, tetapi juga mencakup bagaimana proses pekerjaan itu berlangsung. (Wibowo, 2008).

2.1.3 Definisi Party Planner

Party Planner merupakan bagian dari usaha penyelenggara acara atau dengan istilah penyedia jasa profesional penyelenggara acara. Pada umumnya disebut dengan EO (*Event Organizer*). Tugas dari penyelenggara acara adalah membantu klien untuk dapat menyelenggarakan acara yang diinginkan. (Natoradjo ,2012).

Party Planner adalah usaha yang bergerak di bidang jasa yang khusus ahli dalam perencanaan acara pesta ulang tahun termasuk acara *gathering*, arisan dan persiapan acara pernikahan. *Party planner* dalam konteks sebagai sebuah bisnis memiliki definisi sebagai usaha dalam bidang jasa yang secara sah ditunjuk oleh kliennya, guna mengorganisasikan seluruh rangkaian acara, mulai dari perencanaan, persiapan, eksekusi hingga evaluasi, dalam rangka membantu mewujudkan tujuan yang diharapkan klien dengan membuat acara. (Amelia, 2016)

2.1.3.1 Konsep Acara

Dalam membuat konsep acara, dibagi mejadi tiga bagian :

1. Konsep Baku, dimana konsep ini sebelumnya telah dibuat oleh perusahaan yang bersangkutan, sehingga tinggal melaksanakan saja.
2. Konsep Baru, dimana semua konsep acara dirancang sendiri kemudian dipresentasikan kepada klien, dan hasil kesepakatan akan dilaksanakan.
3. Konsep Kombinasi, dimana pihak party planner dank klien mengadakan sharing bagaimana acara akan dibuat.

2.1.4 *Android*

Android adalah sistem operasi berbasis *Linux* bagi telepon seluler seperti telepon pintar dan komputer *tablet*. *Android* juga menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk berbagai macam piranti gerak. Awalnya, *Google Inc.* membeli *Android Inc.*, pendatang baru yang membuat piranti lunak untuk ponsel. kemudian dalam pengembangan *Android*, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan piranti keras, piranti lunak, dan telekomunikasi, termasuk *Google*, *HTC*, *Intel*, *Motorola*, *Qualcomm*, *T-Mobile*, dan *Nvidia*.

a) Kelebihan *Android*

- *Multitasking*
- Kemudahan dalam notifikasi
- Akses mudah terhadap ribuan aplikasi *Android* lewat *Google Android*
- *App Market*
- Pilihan ponsel yang beranekaragam
- Bisa menginstal ROM yang dimodifikasi
- Widget
- *Google Maniak* (layanan *Google* mulai dari *Gmail* sampai *Google Reader*, ponsel *Android* telah terintegrasi dengan layanan *Google*, sehingga bisa dengan cepat mengecek email dari *Gmail*.)

b) Kelemahan *Android*

- Koneksi *Internet* yang terus menerus
- Iklan

Android menggunakan sistem *open source* sehingga para pembuat aplikasi dapat dengan bebas memodifikasi dan 18 mengembangkan aplikasi pada *OS android*. Sehingga tidak mengherankan *Android* telah memiliki lebih dari 700 ribu aplikasi dan lebih dari 25 juta aplikasi diunduh melalui Google *Play Store*. Berikut adalah daftar *android* yang telah rilis:

- *Android* versi 1.6 (Donut)
- *Android* versi 2.0 (Eclair)
- *Android* versi 2.2 (Froyo: Frozen Yoghurt)
- *Android* versi 2.3 (Gingerbread)
- *Android* versi 3.0 (Honeycomb)
- *Android* versi 4.0 (Ice Cream Sandwich)
- *Android* versi 4.1 (Jelly Bean)
- *Android* versi 4.4 (KitKat)
- *Android* versi 5.0 (Lollipop)
- *Android* versi 6.0 (Marshmallow)

Android, sebagai sistem operasi telepon pintar yang paling banyak digunakan di dunia, mengalahkan *Symbian* pada tahun 2010. *Android* juga menjadi pilihan bagi perusahaan teknologi yang menginginkan sistem operasi berbiaya rendah, bisa dikustomisasi, dan ringan untuk perangkat berteknologi tinggi tanpa harus mengembangkannya dari awal. (Safaat, 2012)

2.1.5 Interaksi Manusia Komputer

Interaksi Manusia dan Komputer adalah ilmu yang melibatkan perancangan dengan mengumpulkan data dan kerangka intelektual yang dikembangkan dari ilmu komputer. Shneiderman dan Plaisant menyatakan bahwa perancangan sebuah sistem yang *user-friendly*,

efisien, dan dapat memenuhi kebutuhan. (Shneiderman dan Plaisant,2010).

Interaksi Manusia Komputer sebagai berikut :

1. Sekumpulan proses, dialog dan kegiatan dimana melalui pengguna memanfaatkan dan berinteraksi dengan komputer.
2. Suatu disiplin ilmu yang menekankan pada aspek desain, evaluasi, dan implementasi dari sistem komputer interaktif untuk kegunaan manusia dengan mempertimbangkan fenomena-fenomena disekitar manusia itu sendiri.
3. Suatu studi ilmiah tentang masyarakat didalam lingkungan kerjanya. (Sudarmawan,2007)

Dari definisi diatas diambil kesimpulan bahwa interaksi manusia komputer adalah interaksi yang dapat membuat manusia berada disekitar lingkungan manusia itu sendiri dengan memanfaatkan komputer. Adapun faktor yang memengaruhi interaksi manusia komputer:

a. Faktor Rekayasa Perangkat Lunak

Rekayasa perangkat lunak (*software engineering*) bisa didefinisikan sebagai berikut :

- 1) Instruksi-instruksi yang bila dieksekusi akan memberikan fungsi dan kerja yang diinginkan, atau
- 2) Struktur data yang memungkinkan suatu program memanipulasi informasi secara proposional
- 3) Dokumen yang menggambarkan operasi dan kegunaan program

Faktor rekayasa perangkat lunak merupakan suatu faktor yang bisa menciptakan suatu program yang efektif,efisien serta *user friendly*

sehingga darinya dapat dihasilkan suatu mesin yang betul-betul diinginkan oleh user.

b. Faktor Kecerdasan Buatan

Kecerdasan buatan (*artificial intelligence*) merupakan suatu bagian ilmu komputer yang bertujuan agar komputer dapat melakukan pekerjaan seperti dan sebaik yang dilakukan manusia serta menggabungkan cara berfikir manusia dengan mesin, untuk mengambil keputusan, memecahkan masalah dan pembelajaran.

Faktor ini mempunyai peranan yang sangat penting dalam pengembangan HCI untuk masa yang akan datang, di mana diharapkan akan tercipta suatu sistem yang betul-betul andal, canggih dan menyerupai pola pikir manusia.

Dengan kecerdasan buatan maka manusia bisa menciptakan robot yang memiliki struktur tubuh mirip manusia, dengan kaki, tangan, kepala.

c. Faktor Linguistik Komputasional

Peran bahasa yang digunakan sebagai antarmuka sangatlah penting, agar *user* lebih mudah menggunakan sistem itu dan tidak terjadi kesalahpahaman di dalam menggunakannya. Dibutuhkan suatu komposisi suatu komposisi bahasa yang sangat baik, yang dapat dengan mudah dimengerti oleh *user*.

d. Faktor Psikologi

Pemahaman akan psikologi orang akan menggunakan *software* sangatlah dibutuhkan dalam interaksi manusia dan komputer mengingat setiap *user* memiliki sifat dan kelakuan yang berbeda. Di

dalam merancang suatu program, faktor ini harus difikirkan terlebih dahulu, seperti siapa target pengguna program, bagaimana suasana lingkungan target, bagaimana perilaku mereka secara umum, dan masih banyak faktor psikologi lain yang harus difikirkan agar program yang dirancang itu lebih *user friendly*. Faktor psikologi juga mencakup pengetahuan dan keahlian pengguna di dalam mempersepsikan dan memecahkan masalah (*problem solving*).

e. Faktor Multimedia

Konteks multimedia digunakan sebagai sarana dialog yang sangat efektif antara manusia dan komputer. Saat ini multimedia berpengaruh besar dalam interaksi manusia dan komputer. Dengan konteks multimedia, tampilan dari suatu inetraksi lebih menarik dan lebih muda dimengerti oleh pengguna.

Pada awal perkembangan komputer, konteks multimedia belum digunakan sehingga hanya orang-orang tentu yang dapat berinteraksi dengan komputer. Kenyataanya sekarang, anak berumur 3 tahun saja sudah bisa memainkan *game* dengan menggunakan komputer karena banyak konteks multimedia didalamnya, seperti gambar, suara, *teks*, grafik dan sebagainya.

f. Faktor Ergonomik

Faktor ergonomik berhubungan dengan aspek fisik untuk mendapatkan lingkungan kerja yang nyaman. Karena manusia yang bekerja didepan komputer biasanya menghabiskan waktu yang cukup lama maka letak meja, kursi, monitor, mouse, posisi duduk, pengaturan cahaya, kebersihan sangat berpengaruh terhadap interaksi manusia dengan komputer yang digunakannya.

g. Faktor Sosiologi

Pada intraksi manusia komputer, sosiologi berhubungan dengan pengaruh sistem komputer dalam struktur sosial. Pada intinya faktor ini merupakan konteks dari suatu interaksi.

h. Teknik Penulisan

Suatu produk (program) yang dibuat membutuhkan manual agar orang yang belum bisa menggunakan produk tersebut bisa mempelajarinya terlebih dahulu agar kemudian dapat terjadi interaksi yang baik antara orang itu dengan komputer.

i. Faktor Matematika

Yang dimaksud faktor matematika dalam interaksi manusia komputer adalah bahwa pembuatan suatu produk (*software*) haruslah efisien dalam perhitungan matematika. Jika suatu *software* bisa dijalankan oleh semua orang, baik yang baru belajar komputer maupun yang telah pakar, unsur kemudahan dalam segi hitungan matematika akan membuat *software* disukai banyak orang.

j. Faktor Bisnis

Faktor bisnis mempengaruhi perkembangan interaksi manusia dan komputer. Produk-produk yang dibuat, baik itu produk perusahaan, kelompok atau perorangan, semakin memudahkan terjadinya interaksi manusia dan komputer. Persaingan bisnis yang semakin hari semakin ketat membuat mereka berlomba untuk merebut hati konsumen dengan produk yang lebih murah digunakan. (Sudarmawan,2007)

2.1.6 Proses Bisnis

Proses bisnis adalah sekumpulan kegiatan yang mengambil salah satu atau banyak masukan dan menciptakan sebuah keluaran yang berguna bagi pelanggan. (Hammer dan Champy dalam Weske, 2007).

Proses bisnis adalah sekumpulan kegiatan dalam bisnis untuk menghasilkan produk dan jasa. Kegiatan proses bisnis ini dapat dilakukan baik secara manual maupun dengan bantuan sistem informasi. Dalam sebuah proses bisnis, harus mempunyai :

1. Tujuan yang jelas
 2. Adanya masukan
 3. Adanya keluaran
 4. Menggunakan *resource*
 5. Mempunyai sejumlah kegiatan dalam beberapa tahapan
 6. Dapat mempengaruhi lebih dari satu unit dalam organisasi dan
 7. Dapat menciptakan nilai atau value bagi konsumen
- (Sudarmawan,2007)

2.2 Metode Analisis dan Pembuatan Sistem

2.2.1 Metode Analisis Sistem

2.2.1.1 Analisis SWOT

Metode analisis SWOT digunakan sebagai dasar untuk memperoleh pokok-pokok permasalahan yang lebih spesifik. Dalam penelitian proses bisnis Biqi *Party Planner* menggunakan metode Analisis SWOT. Metode Analisis SWOT membandingkan antara faktor eksternal peluang dan ancaman dengan faktor internal kekuatan dan kelemahan. Faktor internal dimasukkan kedalam matrik yang disebut

matrik faktor strategi internal atau IFAS (*Internal Strategic Factor Analisis Summary*). Faktor eksternal dimasukkan kedalam matrik yang disebut matrik faktor strategi eksternal EFAS (*Eksternal Strategic Factor Analisis Summary*). (Nisak, 2013)

Alat yang dipakai untuk menyusun faktor-faktor strategis perusahaan adalah matrik SWOT. Matrik ini dapat menggambarkan secara jelas bagaimana peluang dan ancaman eksternal yang dihadapi perusahaan dapat disesuaikan dengan kekuatan dan kelemahan yang dimilikinya. Matrik ini dapat menghasilkan 4 set kemungkinan alternatif strategis. (Nisak, 2013)

Tabel Error! No text of specified style in document..1 Matrik SWOT

| | | | |
|------------------|--------------------|---|--|
| EKSTERNAL | | KEKUATAN (S) | KELEMAHAN (W) |
| | PELUANG (O) | Strategi untuk Memanfaatkan peluang untuk mendayagunakan kekuatan. (Strategi S-O) | Strategi untuk Memanfaatkan peluang untuk mengatasi kelemahan (Strategi W-O) |
| | ANCAMAN(T) | Strategi untuk Mengatasi ancaman dengan jalan mendayagunakan kekuatan. (Strategi S-T) | Strategi untuk Menghindari ancaman sekaligus melindungi kelemahan (Strategi W-T) |

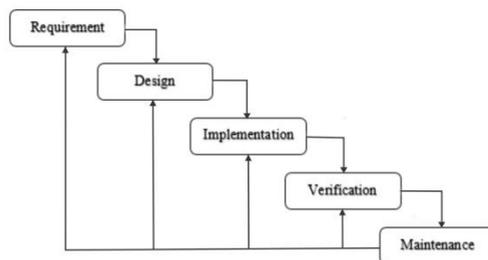
(Sumber: Nisak, 2013)

2.2.2 Metode Pembuatan Sistem

2.2.2.1 Metode Waterfall

Pengembangan proses bisnis dalam hal perancangan konsep dekorasi terhadap pelanggan di *Biqi Party Planner* dalam tugas akhir ini dirancang dengan model SDLC *waterfall* dikarenakan semua kebutuhan sistem dapat didefinisikan secara utuh dan lengkap di awal *project* sehingga proses pembuatan aplikasi dapat berjalan secara dengan baik tanpa ada masalah. Proses pembuatan sistem yang dibuat secara bertahap mampu meminimalkan masalah yang terjadi pada tahap selanjutnya, selain itu pengaplikasian dengan model ini mudah untuk dilakukan.

Model SDLC *waterfall* sering juga disebut model sekuensial linier (*sequential linier*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian dan tahap pendukung (*support*). (Rosa dan Shalahuddin, 2014)



Gambar Error! No text of specified style in document..4

Model *Waterfall*

Model SDLC air terjun (*waterfall*) menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari:

- a. Analisis kebutuhan perangkat lunak
Adalah proses pengumpulan kebutuhan yang dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*.
- b. Desain
Adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka dan prosedur pengkodean.
- c. Pembuatan kode program
Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap sistem.
- d. Pengujian
Pengujian fokus pada perangkat lunak secara dari segi logis dan fungsional untuk memastikan bahwa semua bagian sudah diuji.
- e. Pendukung (*support*) atau pemeliharaan (*maintenance*)
Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk

membuat perangkat lunak baru. (Sukamto dan Shalahuddin, 2013)

2.2.2.2 Kelebihan Metode Waterfall

Kelebihan menggunakan metode air terjun (*waterfall*) adalah metode ini memungkinkan untuk departementalisasi dan kontrol. proses pengembangan model fase *one by one*, sehingga meminimalis kesalahan yang mungkin akan terjadi. Pengembangan bergerak dari konsep, yaitu melalui desain, implementasi, pengujian, instalasi, penyelesaian masalah, dan berakhir di operasi dan pemeliharaan. (Pressman, 2012)

2.2.2.3 Kekurangan Metode Waterfall

Kekurangan menggunakan metode *waterfall* adalah metode ini tidak memungkinkan untuk banyak revisi jika terjadi kesalahan dalam prosesnya. Karena setelah aplikasi ini dalam tahap pengujian, sulit untuk kembali lagi dan mengubah sesuatu yang tidak terdokumentasi dengan baik dalam tahap konsep sebelumnya. (Pressman, 2012).

2.2.3 Metode Perancangan Sistem

2.2.3.1 Unified Modelling Language(UML)

Unified Modelling Language adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berpradigma berorientasi objek". Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-

permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. (Adi Nugroho,2010).

Dari definisi di atas, dapat di simpulkan bahwa UML adalah sebuah bahasa yang digunakan untuk menggambarkan sistem informasi dan sistem perangkat lunak yang berkaitan dengan objek yang digunakan untuk mendokumentasikan dan menjelaskan suatu informasi.

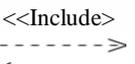
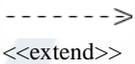
2.2.3.2 *Macam-macam Diagram UML*

2.2.3.2.1 *Use Case Diagram*

Use case merupakan pemodelan untuk kelakuan (*behavior*) system informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antar satu atau lebih aktor dengan sistem informasi yang akan dibuat (Verdi Yasin, 2012).

Tabel Error! No text of specified style in document..2 *Use Case Diagram*

| N O | GAMBAR | NAMA | KETERANGAN |
|--------|---|-------------------|---|
| 1 |  | <i>Actor</i> | Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem itu sendiri. |
| 2 |  | <i>Dependency</i> | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri |

| | | | |
|---|---|-----------------------|---|
| 3 |  | <i>Generalization</i> | Hubungan generalisasi dan spesialisasi (umum-khusus) antar dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya. |
| 4 |  | <i>Include</i> | Relasi use case tambahan ke sebuah use case, dimana use case yang ditambahkan dapat berdiri sendiri. |
| 5 |  | <i>Extend</i> | Relasi use case tambahan ke sebuah use case, di mana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini. |
| 6 |  | <i>Association</i> | Komunikasi antar aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor. |
| 7 |  | <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 |  | Note | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi |

(Sumber: Verdi Yasin, 2012)

2.2.3.2.2 Use Case Description (Fully Develop)

Menurut (Satzinger, Jackson, dan Burd, 2010), *Use Case Description (Fully Developed)* merupakan metode formal yang digunakan untuk mendokumentasikan suatu *use case*. Metode yang dimaksud bertujuan untuk mendefinisikan alur dari kegiatan *use case*, dan memungkinkan *user* untuk memahami alur proses bisnis dan bagaimana sistem harus mendukung.

Table Error! No text of specified style in document..3 Use Case Description

| | | |
|-------------------------------|--|--|
| Use Case Name : | Create customer account | |
| Scenario : | Create online customer account | |
| Triggering Event : | New customer wants to set up account online. | |
| Brief Description : | Online customer creates customer account by entering basic information and then following up with one or more | |
| Actors : | Customer | |
| Related Use Case : | Might be involved by the check out shopping cart use case. | |
| Stakeholders: | Accounting, Marketing, Sales. | |
| Preconditions : | Customer account subsystem must be available. Credit/debit authorization services must be available. | |
| Postconditions : | Customer must be created and saved. One or more Addresses must be created and saved. Credit/debit card information must be validated. Account must be created and saved. Address and Account must be associated with Customer. | |
| Flow of Activities : | Actor 1. Customer indicates desire to create customer account and enters basic customer information. 2. Customer enters one of more addresses. 3. Customer enters credit/debit card information. | System 1.1 System creates a new customer. 1.2 System prompts for customer addresses 2.1 System creates addresses. 2.2 System prompts for credit/debit card. 3.1 System creates account 3.2 System verifies authorization for credit/debit card 3.3 System associates customer, address, and account. 3.4 System returns valid customer account details. |
| Exception Conditions : | 1.1 Basic customer data are incomplete. 2.1 The address isn't valid 3.2 Credit/debit information isn't valid. | |

(Sumber: Satzinger, Jackson, dan Burd 2012)

2.2.3.2.3 Activity Diagram

Activity diagram adalah gambaran berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. (Verdi Yasin, 2012)

Tabel Error! No text of specified style in document..4 Simbol – simbol *Activity Diagram*

| SIMBOL | KETERANGAN |
|---|--|
|  | Titik Awal |
|  | Titik Akhir |
|  | Activity |
|  | Pilihan untuk pengambilan keputusan |
|  | Fork ; digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu |
|  | Rake ; menunjukkan adanya dekomposisi |

| | |
|---|---------------------------|
|  | Tanda waktu |
|  | Tanda pengiriman |
|  | Tanda penerimaan |
|  | Aliran akhir (Flow Final) |

(Sumber: Verdi Yasin, 2012)

2.2.3.2.4 *Class Diagram*

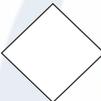
Class diagram menggambarkan struktur dan dekripsi *class*, *package*, dan objek beserta hubungan satu sama lain seperti containment, pewaris, asosiasi, dan lain-lain.

Class diagram berfungsi untuk menjelaskan tipe dari objek sistem dan hubungannya dengan objek yang lain.

Objek adalah nilai tertentu dari setiap attribute kelas entity. *Class* adalah sebuah spesifikasi yang jika diinstansikan akan menghasilkan sebuah objek dan merupakan inti dari pengembangan berorientasi objek.

Class menggambarkan keadaan (*attribute/property*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). (Verdi Yasin, 2012)

Tabel Error! No text of specified style in document..5 Simbol *Relationship Class Diagram*

| NO | GAMBAR | NAMA | KETERANGAN |
|----|---|-------------------------|---|
| 1 |  | <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>). |
| 2 |  | <i>Nary Association</i> | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek. |
| 3 |  | <i>Class</i> | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama. |
| 4 |  | <i>Collaboration</i> | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 5 |  | <i>Realization</i> | Operasi yang benar-benar dilakukan oleh suatu objek. |

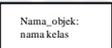
(Sumber: Verdi Yasin, 2012)

- a. Relasi Generalisasi digunakan dalam hubungan antara kelas induk dengan kelas turunan (*inherited*).
- b. Relasi Agregasi digunakan ketika satu kelas dibentuk terdiri dari kelas kelas lain.
- c. *Relationship Multiplicity*. *Multiplicity* atau multiplisitas menunjukkan jumlah suatu objek yang bisa berhubungan dengan objek lain.

2.2.3.2.5 Sequence Diagram

Sequence diagram merupakan gambaran interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display* dan sebagainya) berupa *message* yang digambarkan terhadap waktu (Verdi Yasin, 2012).

Tabel Error! No text of specified style in document..6 Sequence Diagram

| NO | GAMBAR | NAMA | KETERANGAN |
|----|---|-----------------|---|
| 1 |  | <i>Actor</i> | Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi itu sendiri. |
| 2 |  | <i>LifeLine</i> | Objek <i>entity</i> , antarmuka yang saling berinteraksi. |
| 3 |  | <i>Objek</i> | Menyatakan objek yang berinteraksi oleh pesan. |
| 4 |  | <i>Message</i> | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi |
| 5 |  | <i>Message</i> | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi |

| | | | |
|---|--|--------------------------|---|
| 6 | Keluaran  | <i>Pesan tipe return</i> | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian. |
| 7 | Masukan  | <i>Pesan tipe send</i> | Menyatakan bahwa suatu objek mengirim data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim. |
| 8 | I:nama_metode()  | <i>Pesan tipe call</i> | Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri. |

(Sumber: Rosa A.S dan M. Shalahuddin, 2013)

2.3 Metode Pembuatan Perangkat Lunak

2.3.1 Marker Based Tracking Augmented Reality

Metode *marker based tracking* bekerja pada perangkat pengguna mengenali suatu objek, objek tersebut harus dibuatkan tanda. Sistem *Augmented Reality* menggunakan dua jenis tanda, yaitu AR berbasis *Marker*, dan AR tanpa *Marker*. AR berbasis *Marker*, yaitu objek tersebut harus dibuat menjadi marker khusus yang dikenali perangkat. *Marker* yang baik adalah *Marker* yang mudah dikenali oleh perangkat milik pengguna. (Jens Grubert dan Dr. Raphael Grasset, 2013).

Sedangkan pada *Augmented Reality* tanpa *Marker* atau biasa disebut *markerless*, tanda bisa menggunakan GPS dan *Geomagnetic Sensor*. Implementasi *Augmented Reality* dengan menggunakan marker khusus yang telah dikenali oleh perangkat milik pengguna (*Image Marker*) atau biasa disebut *Marker Based Tracking*. Penanda berupa gambar yang menggunakan warna natural sebagai *marker*. *Image marker* biasanya dikelilingi *frame*

tertentu yang membantu deteksi dan rotasi *marker* tersebut. Pada umumnya digunakan jika *marker* tidak ingin dihilangkan pada saat objek *Augmented Reality* ditanamkan ke dalam foto/gambar yang telah diproses. (Karina Aprilia, 2014)

2.4 Perangkat Lunak dan Bahasa Pemrograman

2.4.1 Blender

Blender adalah *software open source* untuk membuat kreasi pembuatan model 3 dimensi, selain untuk model 3 dimensi *Blender* juga digunakan untuk *texturing* pada model 3 dimensi, *lightning* (pencahayaan) pada model 3 dimensi dan animasi, *Blender* memiliki fitur untuk membuat tiruan pergerakan manusia berupa simulasi 3 dimensi. (T. C. Ferreira Fonseca, R. Bogaerts, John Hunt and F. Vanhavere ,2014)

2.4.2 Vuforia SDK

Vuforia SDK Vuforia adalah *Augmented Reality Software Development Kit (SDK)* untuk perangkat bergerak yang memungkinkan pembuatan aplikasi *Augmented Reality*. *Vuforia* menggunakan teknologi *Computer Vision* untuk mengenali dan melacak *marker* atau *image target* dan objek 3D sederhana , seperti kotak , secara *real-time*. (Harvey Tjahjono, Liliana dan Kartika Gunadi,2015)

2.4.3 Unity 3D

Unity 3D merupakan *game engine* yang memiliki potensi yang tinggi dalam proses pengembangan *video game* yang lebih efisien. Dasar bahasa pemrograman yang digunakan *Unity 3D* yaitu *C#* dan *javascript*, pengembang bisa memilih salah satu bahasa

pemrograman yang ingin digunakan, aplikasi video *game* yang dikembangkan menggunakan Unity 3D bisa dibangun ke beberapa platform seperti IOS, Android, Windows Phone, Tizen, Windows, Windows Store Apps, MAC, Linux, Web GL, Playstation 4, Playstation Vita, XBOX One, XBOX 360, Nintendo WII U, Nintendo 3DS dan aplikasi virtual reality. (Harvey Tjahjono, Liliana dan Kartika Gunadi,2015).

Proses pembangunan *video game* edukasi *grammar* bahasa Inggris menggunakan *software Unity 3D*, karena *Unity 3D* mendukung pembangunan fitur *virtual reality* jenis *Google Carboard* atau *VR Box.*, yaitu Unity Web Player seperti Flash Player pada browser. (Harvey Tjahjono, Liliana dan Kartika Gunadi, 2015)

2.4.4 C# (C-Sharp)

C# merupakan bahasa pemrograman yang dibuat khusus untuk platform .NET. mendefinisikan C# sebagai pemrograman *object oriented*, C# sebagai *event driven* karena di dalam C# dapat merespon pengguna dengan *event klik* atau dengan penekanan tombol, dan C# sebagai bahasa pemrograman visual. Dari penjelasan di atas dapat disimpulkan bahwa bahasa pemrograman C# merupakan cabang dari bahasa C, C++ dan java, di mana Bahasa pemrograman C# khusus digunakan untuk *platform .NET* oleh microsoft. (Deitel dan Deite, 2011)

2.4.5 Testing

Testing merupakan pengujian yang berfokus pada perangkat lunak dari segi logika, fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir

kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan. (Rosa dan Shalahuddin, 2016)

Tujuan dari pengujian adalah untuk menemukan dan memperbaiki sebanyak mungkin kesalahan dalam program sebelum menyerahkan program kepada *customer*. Salah satu pengujian yang baik adalah pengujian yang memiliki probabilitas tinggi dalam menemukan kesalahan. (Pressman, 2010)

2.4.6 XAMPP

XAMPP adalah paket program web lengkap yang dapat dipakai untuk belajar pemrograman web, khususnya *PHP* dan *MySQL*. (Nugroho, 2013).

Dibawah folder utama *xampp*, terdapat beberapa folder penting yang perlu diketahui setiap fungsinya, yaitu:

1. *Apache* : Folder utama dari *Apache Webserver*.
2. *Htdocs* : Folder utama untuk menyimpan data data latihan web, baik PHP maupun HTML biasa. Pada folder ini dapat membuat subfolder sendiri untuk mengelompokkan file latihannya. Semua folder dan file program di *htdocs* bisa diakses dengan mengetikkan alamat <http://localhost/> di browser.
3. *Manual* : Berisi subfolder yang di dalamnya terdapat manual program dan *database*, termasuk manual PHP dan MySQL.
4. *Mysql* : Folder utama untuk *database MySQL server*. Di dalamnya terdapat subfolder data (lengkapnya: C:\xampp\mysql\data) untuk merekam semua nama *database*, serta subfolder bin yang berisi *tools klien* dan *server MySQL*.

5. *Php* : Folder utama untuk program *PHP*.

2.4.7 HTML

HTML mempunyai kepanjangan *Hyper Text Markup Language*, yaitu suatu bahasa pemrograman *hyper text*. *HTML* ini memiliki fungsi untuk membangun kerangka ataupun format web berbasis *html*. *HTML* bisa disebut bahasa yang digunakan untuk menampilkan dan mengelola *hypertext*. *HTML* digunakan untuk menampilkan berbagai informasi di dalam sebuah penjelajah web internet dan *formatting hypertext* sederhana yang ditulis ke dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. (Saputra, 2012).

2.4.8 PHP (Hypertext Preprocessor)

PHP atau yang memiliki kepanjangan *Hypertext Preprocessor*, merupakan suatu bahasa pemrograman yang difungsikan untuk membangun suatu website dinamis. *PHP* menyatu dengan kode *HTML*, maksudnya adalah beda kondisi, *HTML* digunakan sebagai pembangun atau pondasi dari kerangka *layout* web, sedangkan *PHP* difungsikan sebagai prosesnya, sehingga dengan adanya *PHP* tersebut, sebuah *web* akan sangat mudah di-*maintenance*. (Saputra, 2012).

2.4.9 MySQL

MySQL bekerja menggunakan *SQL Language (Structure Query Language)*. Itu dapat diartikan bahwa *MySQL* merupakan standar penggunaan *database* di dunia untuk pengolahan data. Pada umumnya perintah yang paling sering digunakan dalam *MySQL* adalah *SELECT* (mengambil), *INSERT* (menambah), *UPDATE*

(mengubah), dan *DELETE* (menghapus). Selain itu, *SQL* juga menyediakan perintah untuk membuat *database*, *field* ataupun *index* untuk menambah dan menghapus data. (Saputra, 2012).

2.4.10 Pengujian Black-Box

Pengujian *Black-Box* dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian *BlackBox* harus dibuat dengan kasus benar dan kasus salah. Pengujian *Black-Box* juga disebut pengujian tingkah laku, memusat pada kebutuhan fungsional perangkat lunak.

Teknik pengujian *Black-Box* memungkinkan mendapat serangkaian kondisi masukan yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Beberapa jenis kesalahan yang dapat diidentifikasi adalah fungsi tidak benar atau hilang, kesalahan antar muka, kesalahan pada struktur data (pengaksesan *database*), kesalahan performasi, kesalahan inisialisasi dan akhir program. (Agisa Rusaldi Wildan, 2013)

