Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

Team project ©2017
Dony Pratidana S. Hum | Bima Agus Setyawan S. IIP

BAB II

LANDASAN TEORI

2.1 *Mobile Application*

Sebuah aplikasi *mobile*, atau yang lebih biasa disebut aplikasi adalah sebuah aplikasi *software* yang dikhususkan untuk dijalankan di perangkat *mobile* seperti *smartphone* atau *tablet computer*. *Mobile Application* dirancang dengan memanfaatkan teknologi yang ada pada perangkat *smartphone* atau *tablet* seperti kamera, *gyroscope*, *accelerometer*, *gps*, dan *touchscreen.*

Kebanyakan *Mobile Application* dirancang dengan tujuan penggunaannya masing-masing, ada yang bertujuan sebagai alat pembantu seperti *notes* dan *reminder*; ada yang dikhususkan sebagai *software* untuk *social media*; bermain *game*; dan juga sebagai alat komunikasi. Untuk penyebarannya sendiri *mobile application* sudah memiliki 'pasar' sesuai dengan sistem operasi dari *smartphone* atau *tablet* yang digunakan. Untuk sistem operasi *Android* ada *Google Play* dan untuk *iOS* ada *App Store*.

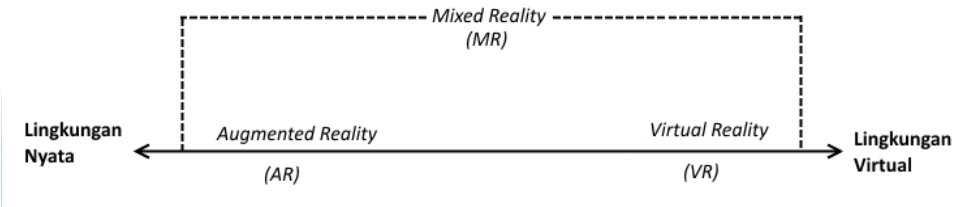
2.2 *Augmented Reality*

Augmented Reality adalah sebuah versi yang lebih kaya dari dunia nyata yang dibuat menggunakan teknologi untuk melapisi informasi digital yang dapat dilihat menggunakan alat seperti kamera *tablet* atau *smartphone* (Wang et al, 2016). Banyak orang yang kebingungan dengan perbedaan antara AR (*Augmented Reality*), VR (*Virtual Reality*) dan CG (*Computer Graphics)*. Yang pertama perlu dipertanyakan adalah *Reality* itu sendiri, apa yang sebenarnya dimaksud dari *Reality*. Penjelasan paling singkat dan mudahnya adalah *Reality* atau kenyataan adalah signal elektrik yang dibuat oleh sel otak manusia, dan ini berarti ketika otak manusia menangkap *computer generated object* sebagai bagian dari dunia nyata

Perancangan dan Pembuatan..., Rayo Roderik Mandias, FTI UMN, 2018

maka itu termasuk dalam *Reality* atau kenyataan sendiri, tetapi tidak merubah fakta bahwa objek tersebut tetap merupakan ilusi.

AR sendiri termasuk dalam bagian dari *Mixed Reality* menurut teori *Virtual Continuum.* Dimana AR lebih mengarah ke *Real Environment* (Lingkungan Nyata) sedangkan AV (*Augmented Virtuality)* mengarah ke *Virtual Environment* (Lingkungan Virtual). Dengan kata lain, AR adalah lingkungan dari sinkronisasi *mixed reality*.

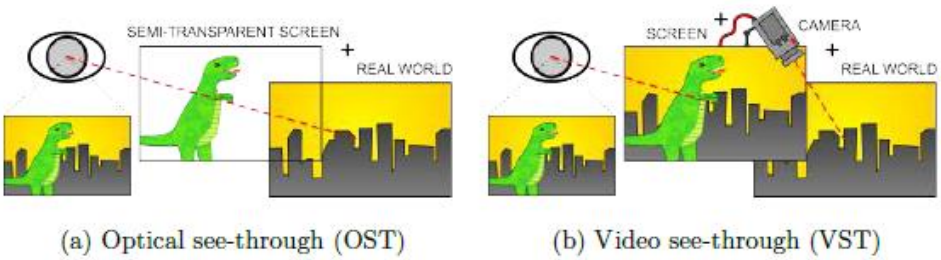


Gambar 2.1. Teori Virtual Continuum

Untuk menampilkan AR, terdapat 2 metode yang umum digunakan, yaitu *optical see-through technology (OST)* dan *video see-through technology (VST)*. Metode OST menggunakan layar semi transparan sehingga seluruh area sekitar masih terlihat langsung oleh mata. *Computer generated object* akan ditampilkan di layar semi transparan tersebut sehingga terlihat seperti AR menyatu langsung dengan dunia nyata.

Sedangkan untuk metode VST, objek yang digunakan untuk menampilkan AR adalah sesuatu yang memiliki layar dan kamera sebagai contohnya *smartphone*. Orang yang akan melihat AR tidak melihat langsung dunia nyata, tetapi dari video yang diambil oleh kamera dan ditampilkan di layar. Hal ini berarti proses penggabungan antara dunia nyata dan virtual terjadi sebelum data ditampilkan di layar.

Perancangan dan Pembuatan..., Rayo Roderik Mandias, FTI UMN, 2018



Gambar 2.2. Perbandingan *Optical see-through* dengan *Video see-through*

Sumber: Žákovská (2015)

Kedua metode ini memiliki keunggulan dan kekurangannya masing-masing, seperti pada OST keunggulannya adalah orang yang melihat akan merasa lebih nyata karena orang yang melihat masih dapat melihat langsung dunia nyata, namun kekurangannya adalah untuk menampilkan AR dengan metode OST tidaklah instan. Sedangkan untuk VST memiliki keterbatasan sendiri yaitu, semua tergantung pada hardware yang digunakan untuk melihat ARnya, sehingga terdapat kemungkinan adanya *time lag, low update rate, low resolution* dan semacamnya, dan keunggulannya tentu lebih instan terutama dengan menggunakan *mobile apps* yang dapat diakses oleh semua pengguna *smartphone* dengan asumsi semua smartphone memiliki kamera. Metode yang umum digunakan untuk AR pada *mobile application* adalah metode *Video see-through (VST)*, oleh karena itu penelitian ini juga akan menggunakan metode VST.

2.3 Sistem Operasi *Android*

Google bersama dengan *Open Handset Alliance* (OHA) meluncurkan sistem operasi *mobile Android* pada akhir tahun 2007. *Android* sendiri merupakan tumpukan software yang dibuat untuk perangkat *mobile* yang terdiri dari sistem operasi, *middleware,* dan beberapa aplikasi kunci lainnya.

Gambar 2.3. *Android Architecture*

Sumber: (Chirchir, 2013)

Application framework Android bersifat *open development*, dimana para *developer* dapat merancang aplikasinya secara bebas. *Developer* bahkan dapat memanfaatkan fitur-fitur yang terdapat dalam perangkat *smartphone*, seperti mengakses gps, menjalankan program *background*, alarm, dan semacamnya yang dapat digunakan untuk merancang aplikasi sesuai dengan keinginan pembuatnya. *Android* juga memberikan akses penuh kepada *developer* untuk mengakses API (*Application program interface)*, yang akan digunakan oleh aplikasi itu sendiri.

2.4 *Unity*

Unity Editor adalah tools utama yang digunakan dalam perancangan aplikasi ini. *Unity* adalah *cross-platform game engine* yang dibuat oleh *Unity Technologies* dan diluncurkan pertama kali pada 8 Juni 2005. *Unity* terus dikembangkan hingga

sekarang, versi paling akhir yaitu 2017.3 diluncurkan pada 17 Desember 2017. Bukan hanya sebagai *engine*, *Unity* juga menyediakan *Unity Editor*, yang merupakan *all-in-one editor* yang menyediakan banyak tools didalamnya untuk melakukan peracangan sebuah aplikasi mulai dari aplikasi *game* pada PC sampai *mobile apps.*

Keunggulan dari *Unity* adalah modelnya yang *cross-platform* sehingga hanya perlu satu kali penulisan kode atau perancangan aplikasi untuk melakukan peluncurkan ke berbagai platform baik itu untuk *desktop* ataupun *mobile*. Selain itu *Unity* juga mendukung objek 2D dan 3D, serta ditambah dengan bantuan *Vuforia*, *Unity* dapat mendukung pembuatan aplikasi yang berbasis *Augmented Reality* ataupun *Virtual Reality*.

2.5 *Vuforia*

Vuforia adalah SDK (*Software Development Kit*) AR untuk *smartphone* dan *device* lainnya yang memungkinkan untuk menjalankan aplikasi AR dengan menggunakan gambar yang diambil dari perangkat tersebut. *Vuforia* menggunakan teknologi *computer vision* untuk mengenali objek yang ditangkap oleh kamera secara *real time*. Tetapi, hal ini bergantung kepada *hardware device* ini sendiri, jumlah objek yang dapat ditangkap akan menyesuaikan dengan batasan *hardware*.

Dengan menggunakan *Vuforia*, *developer* dapat menyesuaikan letak dari objek virtual yang akan dimasukan ke dalam AR sehingga perspektif dari pengguna akan menganggap bahwa objek virtual tersebut tampil seperti layaknya objek nyata di layar *device*. *Vuforia* SDK mendukung penggunaan objek 2D dan 3D, sehingga selain memunculkan objek, *Vuforia* juga dapat memanfaatkan *virtual button* dan menjadikan aplikasi menjadi lebih interaktif dan fleksibel.

Perancangan dan Pembuatan..., Rayo Roderik Mandias, FTI UMN, 2018

2.6 *Microsoft Visual Studio*

Visual Studio interactive development environment (IDE) adalah program yang diluncurkan oleh *Microsoft* untuk keperluan *coding*. *Visual Studio* hampir bisa melihat dan meng-*edit* semua jenis kode yang ada, *debug,build,* dan meluncurkan aplikasi untuk *Android, iOS, Windows, website,* dan *cloud*. *Visual Studio* sendiri memiliki koneksi dengan *Unity* karena kode yang ditulis dalam *Unity* menggunakan C#, dan *Visual Studio* juga menjadi salah satu package yang tersedia dalam *Unity* ketika melakukan pemasangannya.

2.7 *3ds Max*

3ds Max yang sebelumnya dikenal dengan nama 3D *Studio* dan 3D *Studio Max* adalah aplikasi 3d *modeling and rendering* yang digunakan untuk membuat animasi 3D, model 3D, game, dan gambar. *3ds Max* dikembangkan dan diproduksi oleh *Autodesk Media and Entertaiment*. *3ds Max* memiliki *toolset modeling* yang terkenal sangat kuat diantara aplikasi 3d *modeling* lainnya seperti *Maya* dan *Blender*, *3ds Max* juga dilengkapi dengan *library* dari berbagai macam *modifier* yang membuat proses *modeling* lebih mudah bahkan untuk pengguna pemula sekalipun.

2.8 *QR Code*

QR Code adalah singkatan dari *Quick Response Code*, yang menjadi *trademark* dari tipe *barcode* 2-dimensi yang pertama kali didesain untuk industry otomotif di Jepang. *Barcode* itu sendiri adalah label optik yang dapat dibaca oleh mesin dan menyimpan informasi mengenai barang yang berhubungan dengan *barcode* tersebut.

Perancangan dan Pembuatan..., Rayo Roderik Mandias, FTI UMN, 2018

2.9 Shneiderman 8 *Golden Rules*

8 *Golden Rules* adalah 8 aturan dasar yang harus diperhatikan ketika membuat rancangan antarmuka untuk suatu aplikasi . 8 aturan ini diusung oleh Ben Shneiderman (2016) melalui buku "*Designing the User Interface: Strategies for Effective Human-Computer Interaction*". Ke-8 aturan dasar itu adalah:

a. *Strive for consistency*

Pada poin ini yang dimaksud dengan konsistensi adalah konsistensi dalam desain di aplikasi itu sendiri, misalnya *font* yang selalu seragam, warna dasar yang tidak berubah-ubah, jenis tombol yang sama, dan lainnya agar user tidak kebingungan ketika melihat hal yang tidak konsisten.

b. *Cater to universal usability*

Dalam perancangan antarmuka, sebuah aplikasi akan lebih baik apabila dapat digunakan oleh lebih banyak orang yang bervariasi baik itu latar belakang budaya, bahasa, atau pengetahuannya. Sebagai contohnya menggunakan bahasa universal seperti bahasa Inggris sebagai penuntunnya, atau bisa juga didalamnya sudah tersedia fungsi untuk merubah ke berbagai bahasa.

c. *Offer informative feedback*

Informative feedback yang dimaksud adalah sebuah notifikasi dari aplikasi tersebut kepada *user* apabila *user* tersebut melakukan suatu *action.* Tetapi hal ini tidak harus selalu memunculkan notifikasi seperti jawaban secara explisit, bisa juga dengan perubahan antarmuka atau halaman ketika *user* melakukan *action* tersebut.

d. *Design dialogs to yield closure*

Dalam prinsip ini maksudnya adalah tentang panduan atau arahan untuk memudahkan *user* melakukan *action* yang memerlukan langkah yang

Perancangan dan Pembuatan..., Rayo Roderik Mandias, FTI UMN, 2018

banyak. Untuk melakukannya, langkah-langkah tersebut dipisah menjadi beberapa bagian seperti tahap awal dan akhir misalnya, serta adanya feedback apabila tahap tersebut sudah dilakukan.

e. *Prevent errors*

Desain sebuah aplikasi sebisa mungkin agar *user* tidak melakukan kesalahan yang fatal. Misalnya *user* ingin menghapus *file* lebih baik ada pemberitahuan untuk konfirmasi penghapusan *file* tersebut.

f. *Permit easy reversal of actions*

User harus dapat dengan mudah mengulang atau mengembalikan ke tindakan yang sebelumnya, contoh simpelnya adalah adanya tombol untuk *undo* sebuah *action*.

g. *Support internal locus of control*

Sistem harus dirancang agar user menganggap sistem dapat merespon *action* yang dilakukan *user* dan *user* menganggap bahwa ia-lah yang mengontrol sistem dan tindakannya.

h. *Reduce short-term memory load*

Prinsip yang terakhir ini artinya membuat desain antarmuka lebih simple, jangan memasukan terlalu banyak info dalam sebuah halaman dan memaksa *user* untuk mengingat hal tersebut dalam waktu yang singkat, contoh untuk mengatasi hal ini adalah dengan membaginya menjadi beberapa halaman yang berbeda.

2.10 *User Interface*

User Interface (GUI) adalah sebuah antar muka yang merupakan kumpulan teknik dan mekanisme untuk berinteraksi dengan sesuatu (Wilbert O. Galitz, 2007). UI harus berfokus ke pada *user* bukan teknologi yang digunakannya, diawali dengan mengerti pengguna itu sendiri karena tujuan UI adalah mempermudah pengguna, sehingga harus mengetahui siapakah pengguna dan bagaimana dengan karakteristiknya. Lalu selanjutnya harus mengerti aksi yang akan dilakukan pengguna, jangan sampai menyulitkan pengguna. Dengan begitu akan didapat rancangan UI yang mempercepat aksi yang akan dilakukan oleh pengguna itu sendiri. UI harus dirancang secara interaktif dan disertai dengan penjelasan fungsi-fungsi yang harus tertulis dengan jelas.

Fungsi harus diutamakan dibandingkan lapisan representasinya, karena jika fungsinya tidak berjalan dengan sesuai maka pembuatan UI akan percuma. Penggunaan bahasa juga harus dapat dimengerti oleh *user* atau bahkan secara *universal*, penggunaan istilah teknis yang hanya dapat dimengerti oleh *developer* atau beberapa orang akan membingungkan *user*.

2.11 *User Experience*

User Experience (UX) adalah sebuah pengalaman yang dibuat oleh sebuah produk kepada orang-orang sebagai *user* didunia nyata (Garret, 2011). UX bukanlah sebuah rantai pekerjaan didalam produk tersebut atau layanan yang menjadi hal utama dalam produk. UX lebih menitik-beratkan kepada fungsi diluar produk tersebut ketika bersentuhan langsung atau terdapat kontak dengan *user*. Interaksi yang dihasilkan ini merupakan pokok dari UX.

Sebuah produk yang dibuat memiliki rancangan produk-nya sendiri, yang merupakan daya tarik mata dan terlihat nyaman bila disentuh. Rancangan produk yang baik menghasilkan harapan bahwa produk tersebut akan memiliki fungsi yang baik pula apabila digunakan. Namun, terkadang hal itu tidak terjadi. Masih ada

Perancangan dan Pembuatan..., Rayo Roderik Mandias, FTI UMN, 2018

produk yang memiliki tampilan serta rancangan yang menarik tetapi mengecewakan dalam fungsinya atau tidak memberikan kenyamanan terhadap penggunanya. Hal inilah yang harus dihindari dengan merancang sebuah produk yang disertai dengan penerapan UX sebagai sebuah keluaran produk yang eksplisit.

Menurut Garret (2011), sebagai contoh, dalam rancangan sebuah mesin pembuat minuman kopi. Estetika desain produk memastikan bahwa tombol-tombol yang digunakan pada mesin pembuat minuman kopi memiliki desain yang elegan dan memiliki teksture. Fungsional desain memastikan tombol tersebut berfungsi dengan aksi yang telah ditentukan. UX desain memastikan aspek lain selain estetika dan fungsional sehingga berada dalam konteksnya. Seperti besar atau tidaknya ukuran tombol yang dipakai mempengaruhi sudut pandang pengguna, penempatan tombol-tombol atau hubungannya terhadap kenyamanan ketika beberapa tombol dipakai pada saat bersamaan.

Berikut ini adalah elemen-elemen dari UX yang menjadi dasar dalam mendesain produk yang kemudian dapat menciptakan UX yang berkesan (Garret, 2011):

a. *The Strategy Plane*

Kebutuhan *user* menjadi sebuah tujuan dari produk (*software*) yang dibuat. Hal ini harus dipahami, apa yang menjadi keinginan *user* dan bagaimana hal tersebut berdampak terhadap produk. Tetapi ada hal yang harus dipertimbangkan juga, yaitu apabila terdapat tujuan bisnis yang diusung oleh produk tersebut, harus ada keseimbangan antara keinginan pengguna dengan apa yang menjadi tujuan bisnis produk yang dikembangkan atau dibuat.

b. *The Scope Plane*

Di sisi fungsi, "T*he Strategy Plane*" tersebut diterjemahkan ke dalam lingkup melalui penciptaan spesifikasi fungsional, yaitu penjelasan rinci

Perancangan dan Pembuatan..., Rayo Roderik Mandias, FTI UMN, 2018

tentang kumpulan fitur dari produk. Di sisi informasi, ruang lingkup mengambil bentuk persyaratan konten, yaitu deskripsi dari elemen berbagai konten yang akan diperlukan.

c. *The Structure Plane*

"*The Scope Plane*" diberikan struktur di sisi fungsionalitas melalui rancangan interaksi, di mana kita mendefinisikan bagaimana sistem berperilaku dalam menanggapi pengguna. Untuk sumber daya informasi, struktur adalah arsitektur informasi, yaitu susunan elemen konten untuk memfasilitasi pemahaman manusia.

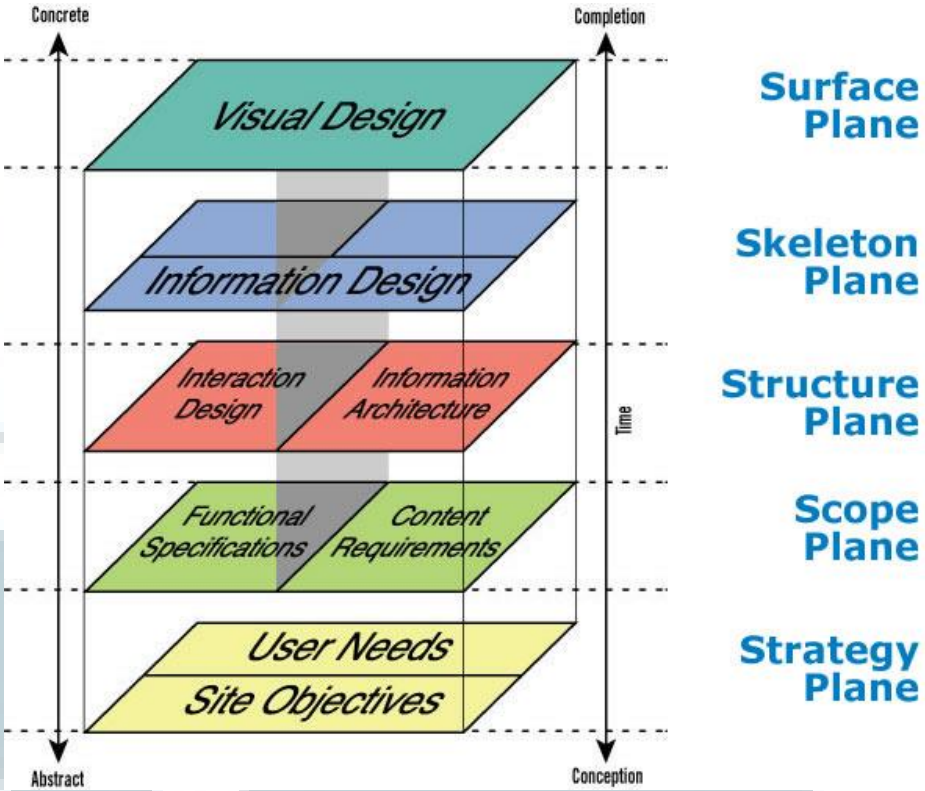
d. *The Skeleton Plane*

"*The Skeleton Plane*" terurai menjadi tiga komponen. Pada kedua sisinya, kita harus menunjuk desain informasi yaitu penyajian informasi dengan cara pemahaman terhadap fasilitas. Untuk fungsi yang berorientasi terhadap produk, "*The Skeleton Plane*" juga mencakup desain antarmuka, atau mengatur elemen antarmuka untuk memungkinkan pengguna berinteraksi dengan fungsi dari sistem. *User Interface* untuk sumber daya informasi adalah rancangan navigasi yaitu himpunan elemen layar yang memungkinkan pengguna untuk bergerak melalui arsitektur informasi.

e. *The Surface Plane*

Terlepas dari apakah kita berhadapan dengan produk yang berorientasi terhadap fungsi atau sumber informasi, perhatian tahap ini adalah sama yaitu pengalaman sensorik yang diciptakan oleh produk yang sudah selesai.

Perancangan dan Pembuatan..., Rayo Roderik Mandias, FTI UMN, 2018



Gambar 2.4. 5 ***Layer of UX Design***
Sumber: (Garrett, 2011)

2.12 *Android Design Principles*

Berikut ini adalah prinsip-prinsip desain aplikasi yang dibuat oleh *Android User Experience Team* untuk memudahkan *user* dalam penggunaan aplikasi.

a. *Delight me in surprising ways*

Beri user kejutan yang menyenangkan seperti tampilan yang cantik, animasi yang diletakan ditempat yang pas, atau suara yang terdengar pada saat yang pas adalah pengalaman yang menyenangkan.

b. ***Real objects are more fun than buttons and menus***

Buatlah *user* dapat menyentuh dan memanipulasi objek langsung pada aplikasi, hal ini mengurangi usaha kognitif yang diperlukan untuk melakukan sesuatu sekaligus membuat lebih memuaskan secara emosional.

c. ***Let me make it mine***

User sangat senang menambah sentuhan personal karena hal itu akan membuatnya merasa sedang berada di rumah dan memegang control. Sajikan tampilan *default* yang cantik tapi menyenangkan dan berikan kemungkinan untuk kostumisasi yang tidak merusak fungsi utama.

d. ***Get to know me***

Kenali kebiasaan *user* seiring waktu berjalan, buatlah pilihan yang sudah dipilih dapat dipilih dengan cepat daripada membuat user melakukan hal yang sama berulang-ulang.

e. ***Keep it brief***

Gunakan kata-kata yang singkat dan mudah dimengerti, *user* cenderung melewati kalimat yang terlalu panjang.

f. ***Pictures are faster than words***

Gambar dapat digunakan untuk menjelaskan sesuatu dengan cepat dan gambar juga menarik perhatian *user* serta lebih efisien daripada kata-kata.

g. ***Decide for me but let me have the final say***

Berikan beberapa pilihan yang mungkin akan dipilih oleh *user* daripada bertanya lebih dahulu, terlalu banyak pilihan akan membuat *user* tidak senang. Apabila salah berikan kemungkinan *user* untuk melakukan *undo*.

Perancangan dan Pembuatan..., Rayo Roderik Mandias, FTI UMN, 2018

h. ***Only show what I need when I need it***

User akan merasa kaget ketika mereka melihat sesuatu yang terlalu banyak secara bersamaan. Bagilah informasi menjadi beberapa bagian kecil yang lebih mudah dicerna. Sembuhnyikan juga opsi yang tidak terlalu penting dan ajarkan kepada *user* seiring mereka berjalan.

i. ***I should always know where I am***

Berikan *user* kepercayaan diri bahwa mereka tau jalan, buatlah lokasi-lokasi pada aplikasi terlihat agak berbeda dan gunakan transisi untuk menunjukan hubungan antar halaman, serta berikan *feedback* mengenai aksi yang sedang dilakukan.

j. ***Never lose my stuff***

Simpanlah sesuatu yang *user* buat dan buatlah mereka dapat mengaksesnya darimana saja. Ingatlah *setting* dan segala sentuhan personal yang dilakukan *user* agar dapat langsung dipakai di *device* lain.

k. ***Only interrupt me if it's important***

User ingin tetap fokus, interupsi pada waktu yang tidak tepat akan sangat menjengkelkan.

l. ***Give me tricks that work everywhere***

User akan merasa senang ketika mereka menyadari sesuatu dengan sendirinya. Buatlah aplikasi mudah digunakan dengan memanfaatkan *visual patterns* dan *muscle memory* dari aplikasi lainnya. Contohnya, gesture seperti pinch untuk zoom yang sudah umum.

m. ***It's not my fault***

Bersikap lembutlah ketika ingin meminta *user* untuk melakukan koreksi, mereka ingin merasa pintar ketika menggunakan aplikasi. Jika terjadi suatu

Perancangan dan Pembuatan..., Rayo Roderik Mandias, FTI UMN, 2018

kesalahan, berikan instruksi *recovery* yang jelas tetapi jauhkan detil teknis. Akan lebih baik jika perbaikan dapat dilakukan *behind the scenes.*

n. *Sprinkle encouragement*

Pecahkan aksi yang sulit kedalam beberapa tahapan kecil yang mudah diselesaikan, serta berikan *feedback* ketika aksi dilakukan meski hanya sedikit.

o. *Do the heavy lifting for me*

Buatlah pengguna baru merasa seperti ahli dengan memungkinkan mereka melakukan hal yang mereka piker tidak mungkin. Contohnya, *shortcut* yang menggabungkan beberapa efek foto dapat membuat fotografer amatir merasa hebat hanya dalam beberapa tahap.

p. *Make important things fast*

Tidak semua aksi sama, tentukan apa yang paling penting dalam aplikasi dan buatlah hal tersebut mudah ditemukan dan dapat digunakan dengan cepat. Contohnya, tombol *pause* pada pemutar music.

2.13 Metodologi *Prototyping*

Metode *prototyping* adalah metode yang digunakan untuk perancangan sistem menggunakan *prototype. Prototype* sendiri dapat dikatakan sebagai contoh dari sistem yang sudah selesai, tetapi masih belum berfungsi secara sempurna. *Prototype* sistem akan memberikan garis besar bagaimana sistem tersebut bekerja.

Proses pengembangan sistem menggunakan metodologi *prototyping* dimulai dengan komunikasi dan pengumpulan data, desain cepat, pembentukan *prototype*, evaluasi terhadap *prototype*, perbaikan *prototype*, dan produksi akhir. Berikut ini adalah penjelasan mengenai tahapan-tahapan dari metode *prototyping*:

Perancangan dan Pembuatan..., Rayo Roderik Mandias, FTI UMN, 2018

a. Komunikasi dan pengumpulan data

Pada tahap ini, *developer* dan *user* bertemu untuk melakukan pengumpulan data mengenai kebutuhan *user* yang diperlukan dalam pembuatan fungsi utama dari aplikasi itu.

b. Desain Cepat

Developer lalu akan membuat desain aplikasi secara umum yang akan dikembangkan kembali

c. Pembentukan *Prototype*

Pada tahap ini *developer* akan membuat *prototype* dari aplikasi yang lalu di uji dan disempurnakan kembali.

d. Evaluasi *Prototype*

Prototype yang sudah dibuat lalu akan diberikan kepada *user* untuk diuji coba dan dievaluasi, *user* lalu akan memberikan analisisnya dan kebutuhan tambahan apabila ada.

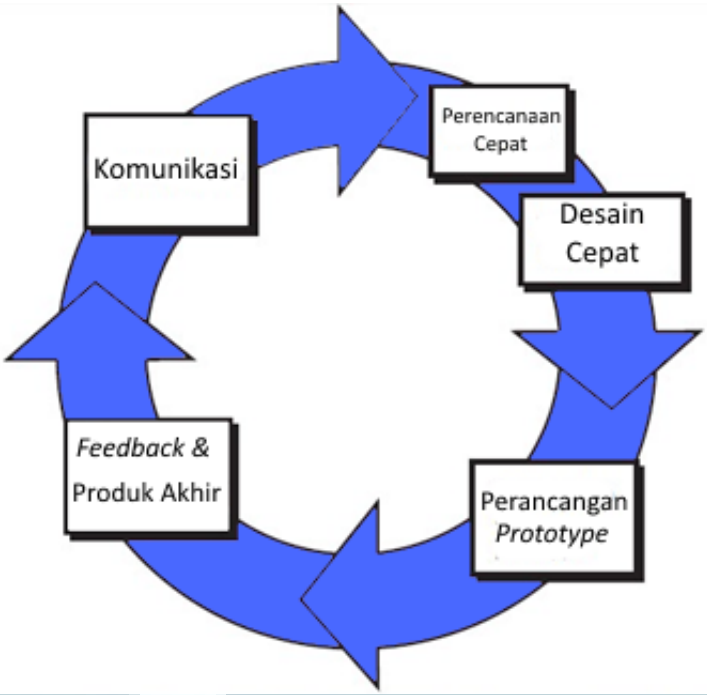
e. Perbaikan *Prototype*

Prototype akan diperbaiki dan disempurnakan sesuai dengan hasil evaluasi yang diberikan oleh *user*.

f. Produksi akhir

Prototype yang sudah disempurnakan akan diproduksi menjadi aplikasi yang utuh dan siap untuk digunakan oleh *user*.

Perancangan dan Pembuatan..., Rayo Roderik Mandias, FTI UMN, 2018



Gambar 2.4. *Evolutionary Prototyping*
Sumber: (Pressman, 2015)