



### **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.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Rencana Tata Ruang Wilayah**

Rencana Tata Ruang Wilayah (RTRW) Kota adalah rencana tata ruang yang bersifat umum dari wilayah kota, yang merupakan penjabaran dari RTRW provinsi, dan yang berisi tujuan, kebijakan, strategi penataan ruang wilayah kota, rencana struktur ruang wilayah kota, rencana pola ruang wilayah kota, penetapan kawasan strategis kota, arahan pemanfaatan ruang wilayah kota, dan ketentuan pengendalian pemanfaatan ruang wilayah. Selain itu, rencana tata ruang wilayah ini juga berfungsi sebagai acuan dalam penyusunan Rencana Pembangunan Jangka Panjang Daerah (RPJPD) dan Rencana Pembangunan Jangka Menengah Daerah (RPJMD), acuan dalam pemanfaatan ruang atau pengembangan wilayah provinsi, acuan untuk mewujudkan keseimbangan pembangunan dalam wilayah provinsi, acuan lokasi investasi dalam wilayah provinsi yang dilakukan oleh pemerintah, masyarakat, dan swasta, sebagai pedoman dalam penyusunan rencana tata ruang kawasan strategis provinsi, dasar pengendalian pemanfaatan ruang dalam penataan atau pengembangan wilayah provinsi yang meliputi indikasi arahan peraturan zonasi, arahan perizinan, arahan insentif dan disinsentif, serta arahan sanksi, dan acuan dalam administrasi pertanahan.

Manfaat yang didapatkan dengan adanya perencanaan tata ruang wilayah ini adalah terwujudnya keterpaduan pembangunan dalam suatu wilayah, keserasian pembangunan suatu wilayah dengan wilayah sekitarnya, dan menjamin terwujudnya

tata ruang wilayah yang berkualitas (penataanruang.com, Indonesian Institute for Infrastructure Studies).

## 2.2 Simulation

Menurut Diana Oblinger (2006), sebuah aplikasi *simulation game* dapat membantu pengguna untuk belajar dalam berbagai sudut pandang, sebagai berikut.

### a. Sosial

Permainan adalah suatu lingkungan yang sering kali berhubungan dengan lingkungan sosial. Terkadang dapat melibatkan masyarakat dalam jumlah yang besar.

### b. Penelitian (*Research*)

Ketika pemain atau pengguna memasuki permainan, maka ia akan mengingat apa saja yang sebelumnya telah dipelajari, dan hal atau informasi apa saja yang dibutuhkan yang dapat diimplementasikan ke dalam situasi yang ada.

### c. Penyelesaian Masalah (*Problem Solving*)

Mengetahui adanya informasi, teknik, atau sesuatu hal yang dapat digunakan untuk mendapatkan hasil yang lebih baik, spesifik, dan menyelesaikan masalah yang ada.

### d. Transfer

Menghubungkan pembelajaran yang didapatkan di berbagai lingkungan yang ada, seperti sekolah, rumah, dan lainnya. Mengetahui hubungan antara semua hal yang ada di berbagai tempat unik tersebut merupakan bagian dari sebuah *game*.

e. Pengalaman (*Experiential*)

Permainan melibatkan beberapa indera yang melibatkan setiap aksi yang menimbulkan reaksi, pengujian hipotesis, dan pembelajaran dari hasil akhir.

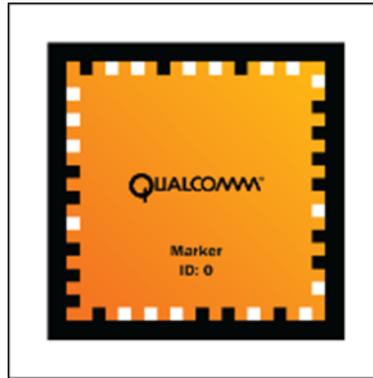
Dalam penelitian ini, aplikasi yang dibuat bersifat sebagai penelitian (*Research*) dan penyelesaian masalah, dimana *user* diharapkan dapat mendapatkan informasi mengenai apa saja hal yang perlu diperbaiki atau ditambahkan ke dalam suatu perencanaan tata ruang wilayah melalui objek yang ada.

### **2.3 Augmented Reality**

Menurut Dedi Eko Nurcahyo, Selo, dan Bimo Sunarfri Hantono (2015) *Augmented Reality* merupakan penggabungan benda-benda yang ada di dunia maya (virtual) ke dalam dunia nyata dalam bentuk dua dimensi maupun tiga dimensi yang dapat disentuh, maupun dilihat, dan juga dapat didengar. *Augmented reality* merupakan cabang dari *Virtual Reality* (VR). Dalam AR, informasi ditingkatkan atau ditambahkan dengan menggabungkan antara realitas dan objek maya yang telah didesain sebelumnya. Dengan AR pengguna dapat berinteraksi dengan dunia nyata secara bersamaan menggunakan komputer atau *gadget* untuk mengeksplorasi informasi dan berinteraksi dengan benda-benda virtual.

Salah satu teknik penggunaan dalam *Augmented Reality* yaitu dengan metode *marker-based*. Metode ini digunakan dengan menyelaraskan data virtual dan lingkungan yang ada melalui gambar-gambar yang berperan sebagai *marker* atau *tag* (Siltanen S., 2012).

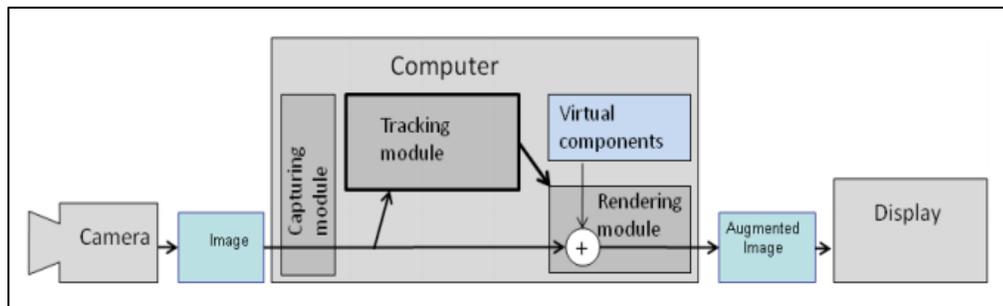
Gambar 2.1 menunjukkan salah satu contoh *marker* atau *tag* yang digunakan untuk menampilkan objek 3D. *Marker* atau *tag* tersebut bersifat unik antara satu dengan yang lainnya.



Gambar 2.1 *Marker* atau *Tag*

(Sumber : <https://developer.vuforia.com>)

Kemudian, pada Gambar 2.2 di bawah dapat dilihat contoh *flowchart* sistem *Augmented Reality*. *Capturing Module* menangkap gambar pada *marker* yang berada di dalam jangkauan kamera. *Tracking Module* mengkalkulasikan posisi penempatan objek yang akan ditampilkan dengan tepat. *Rendering Module* berfungsi untuk menggabungkan gambar asli dan komponen virtual dengan menggunakan perhitungan yang telah dilakukan oleh *Tracking Module*, kemudian melakukan proses *rendering* pada gambar objek 3D yang akan ditampilkan pada layar.



Gambar 2.2 *Flowchart Simple AR System*

(Sumber : Siltanen S., 2012)

## 2.4 Algoritma Fisher Yates

Algoritma Fisher Yates merupakan metode pengacakan yang lebih baik atau dapat dikatakan sesuai untuk pengacakan angka, dengan waktu eksekusi yang cepat serta tidak memerlukan waktu yang lama untuk melakukan proses pengacakan (*shuffle*). Algoritma Fisher Yates dibagi menjadi dua metode, yaitu *original* dan *modern*. Pada penelitian ini akan digunakan metode modern yang memang khusus untuk pengacakan dengan sistem komputerisasi. Fisher Yates *modern* sedikit berbeda dalam kecepatannya, hal ini dapat dibedakan pada sifat algoritma Fisher Yates modern yang memindahkan angka yang telah tereliminasi ke urutan *array* terakhir. Berikut langkah-langkah yang dilakukan dalam melakukan pengacakan dengan algoritma Fisher Yates *modern* (Nugraha, Ryan. Exridores, Edo., dan Hendri Sopryadi, 2012).

- a. Siapkan *array* berisi angka 1-N.
- b. Pilih sebuah angka acak K diantara 1 sampai dengan jumlah angka yang belum dieliminasi.
- c. Dihitung dari bawah, tukar angka yang akan dieliminasi dengan angka terakhir pada *array*, kemudian masukkan angka yang tereliminasi ke *array temp(Result)*.
- d. Ulangi langkah kedua dan ketiga sampai semua angka tereliminasi.
- e. Urutan angka yang terdapat pada *array temp* adalah hasil permutasi acak.

Sebagai contoh, perbedaan cara kerja algoritma Fisher Yates *original* dan *modern* dapat dilihat pada gambar-gambar berikut.

a. Algoritma Fisher Yates Original

Untuk Algoritma Fisher Yates Original, langkah pertama yang dilakukan adalah dengan menyiapkan *array* berisi angka 1 sampai dengan N (*total length*).

Tabel 2.1 Ilustrasi Algoritma Fisher Yates Original (1)

Range	Roll	Scratch	Result
		1 2 3 4 5 6 7 8	

Kemudian, ambil sebuah angka secara acak di antara 1 sampai dengan jumlah angka yang belum tereliminasi. Misalnya hasil pengambilan nilai secara acak adalah angka 3, maka angka 3 yang berada pada *array* ke-3 akan dieliminasi dan dimasukkan ke *array temp*.

Tabel 2.2 Ilustrasi Algoritma Fisher Yates Original (2)

Range	Roll	Scratch	Result
1-8	3	1 2 <del>3</del> 4 5 6 7 8	3

Selanjutnya, lakukan proses yang sama dengan mengambil angka secara acak diantara 1 sampai dengan jumlah angka yang belum tereliminasi. Misalnya hasil pengambilan nilai acak adalah 4, maka angka 5 pada posisi ke-4 akan tereliminasi dan dikirim ke *array temp*.

Tabel 2.3 Ilustrasi Algoritma Fisher Yates Original (3)

Range	Roll	Scratch	Result
1-7	4	1 2 <del>3</del> <del>4</del> <del>5</del> 6 7 8	3 5

Proses yang sama akan terus dilakukan hingga semua angka tereliminasi. Hasil pengacakan dengan Algoritma Fisher Yates Original dapat diambil dari *array temp* (kolom *result* pada gambar).

Tabel 2.4 Ilustrasi Algoritma Fisher Yates Original (4)

Range	Roll	Scratch	Result
1-6	5	1 2 <del>3</del> 4 <del>5</del> 6 7 8	3 5 7
1-5	3	1 2 <del>3</del> 4 <del>5</del> 6 7 8	3 5 7 4
1-4	4	1 2 <del>3</del> 4 <del>5</del> 6 7 8	3 5 7 4 8
1-3	1	4 2 <del>3</del> 4 <del>5</del> 6 7 8	3 5 7 4 8 1
1-2	2	4 2 <del>3</del> 4 <del>5</del> 6 7 8	3 5 7 4 8 1 6
		4 2 <del>3</del> 4 <del>5</del> 6 7 8	3 5 7 4 8 1 6 2

b. Algoritma Fisher Yates Modern

Pada Algoritma Fisher Yates Modern terdapat sedikit perbedaan pada langkah-langkah yang dilakukan. Langkah pertama, siapkan *array* berisi angka 1 sampai dengan N (*total length*).

Tabel 2.5 Ilustrasi Algoritma Fisher Yates Modern (1)

Range	Roll	Scratch	Result
		1 2 3 4 5 6 7 8	

Langkah selanjutnya, ambil sebuah angka acak di antara 1 sampai dengan jumlah angka yang belum tereliminasi. Misalnya didapatkan angka 6, maka angka 6 akan ditukar dengan angka 8 yang berada pada posisi *array* terakhir dan dieliminasi. Angka 6 akan masuk ke *array temp*.

Tabel 2.6 Ilustrasi Algoritma Fisher Yates Modern (2)

Range	Roll	Scratch	Result
1-8	6	1 2 3 4 5 <b>8</b> 7	<b>6</b>

Contoh selanjutnya, misalnya didapatkan angka 2, maka angka 2 yang berada pada *array* ke-2 akan ditukar dengan angka 7 yang berada pada posisi *array* terakhir. Kemudian, angka 2 akan tereliminasi dan dikirim ke *array temp*.

Tabel 2.7 Ilustrasi Algoritma Fisher Yates Modern (3)

Range	Roll	Scratch	Result
1-7	2	1 7 3 4 5 8	2 6

Proses dengan cara yang sama akan terus berlanjut hingga semua angka tereliminasi. Urutan angka yang terdapat pada *array temp* adalah hasil dari pengacakan yang dilakukan.

Tabel 2.8 Ilustrasi Algoritma Fisher Yates Modern (4)

Range	Roll	Scratch	Result
1-6	6	1 7 3 4 5	8 2 6
1-5	1	5 7 3 4	1 8 2 6
1-4	3	5 7 4	3 1 8 2 6
1-3	3	5 7	4 3 1 8 2 6
1-2	1	7	5 4 3 1 8 2 6

## 2.5 Unity Software

Unity merupakan suatu aplikasi yang digunakan untuk mengembangkan game *multi-platform* yang didesain untuk mudah digunakan. *Editor* pada Unity dibuat dengan *user interface* yang sederhana. Grafis pada unity dibuat dengan grafis tingkat tinggi untuk OpenGL dan DirectX. Unity mendukung semua format file terutama format umum seperti semua format dari *art applications*. Secara lebih rinci, Unity dapat digunakan untuk membuat video game 3D, *real-time 3D animation*, dan

visualisasi arsitektur (Maulana Rosikhan Y. dan Aristiawan, 2015). Selain itu Unity juga bersifat *open-ended*, yang dapat membuat berbagai jenis aplikasi game, simulasi, dan berbagai aplikasi *multi-platform* lainnya. Kekurangan yang ada terdapat pada *tools* yang ada pada Unity. *Tools* yang ada sudah mendukung untuk membuat berbagai jenis aplikasi, tetapi tidak lebih baik dari *software* lain yang bersifat lebih fokus ke dalam pembuatan suatu aplikasi (misal, khusus untuk membuat *game* FPS, khusus animasi 3D, dll). Namun, kekurangan ini dapat ditutupi dengan berbagai *plug-in* atau *tools* tambahan yang di-*share* pada komunitas Unity, contohnya adalah Vuforia (Finnegan T., 2015).

## 2.6 Vuforia

Vuforia merupakan sebuah *Software Development Kit* untuk perangkat bergerak yang memungkinkan pembuatan aplikasi dengan teknologi *Augmented Reality*. Vuforia menggunakan teknologi *Computer Vision* untuk mengenali dan melacak *marker* atau *image target* dan objek 3D sederhana secara *real-time* (Setya N.I., Satoto K.I., dan Martono K.T., 2014).

Proses penggunaan Vuforia SDK dapat dilakukan dengan mendaftar terlebih dahulu ke *website* Vuforia Developer. Selain itu, pengguna juga harus mengerti mengenai cara menggunakan Vuforia *License Manager* dan *Target Manager* sesuai dengan syarat-syarat yang dibutuhkan. Kedua poin tersebut akan berfungsi untuk mendaftarkan aplikasi yang akan dibangun dan menambahkan ataupun mengurangi jumlah *marker* atau *tag* yang akan digunakan.

## 2.7 Uji User Experience dan Penilaian Skala Likert

Menurut Andini A. (2014), kriteria penilaian dengan menggunakan *Likert Scale* adalah sebagai berikut.

Tabel 2.9 Kriteria Penilaian Likert Scale

Persentase	Keterangan
0 – 25 (%)	Tidak Baik
26 – 50 (%)	Kurang Baik
51 – 75 (%)	Baik
76 – 100 (%)	Sangat Baik

Data yang akan digunakan diambil dengan cara membagikan kuesioner pada para responden setelah melakukan uji coba pada aplikasi yang dibuat. Pertanyaan-pertanyaan yang ada dibuat dengan memperhatikan faktor-faktor yang berpengaruh terhadap *usability* aplikasi.

*Usability* adalah kemampuan sebuah aplikasi atau perangkat lunak untuk dipahami, dipelajari, digunakan, dan menarik bagi pengguna ketika digunakan pada kondisi tertentu (Hutomo A., 2014). Menurut Rahadi D.R.(2014) *usability* dapat diukur berdasarkan komponen-komponen berikut.

### 1. *Learnability*

*Learnability* didefinisikan dengan seberapa cepat pengguna mahir atau mengetahui cara menggunakan sistem dan juga kemudahan dalam menggunakan fungsi-fungsi dalam mencapai tujuan yang mereka inginkan.

### 2. *Efficiency*

*Efficiency* didefinisikan sebagai sumber daya yang dikeluarkan untuk mencapai ketepatan dan kelengkapan tujuan.

### 3. *Memorability*

*Memorability* didefinisikan dengan bagaimana kemampuan pengguna mempertahankan pengetahuannya setelah jangka waktu tertentu, kemampuan mengingat didapatkan dari peletakkan *menu* yang jelas.

### 4. *Errors*

*Errors* didefinisikan dengan seberapa banyak kesalahan-kesalahan yang dibuat pengguna dalam menggunakan aplikasi. Kesalahan yang dilakukan mencakup ketidaksesuaian apa yang pengguna pikirkan dengan apa yang sebenarnya disajikan oleh sistem.

### 5. *Satisfaction*

*Satisfaction* didefinisikan sebagai kebebasan dari rasa ketidaknyamanan (kepuasan) dan sikap positif pengguna terhadap penggunaan produk atau ukuran subjektif sebagaimana pengguna rasakan tentang penggunaan sistem.