

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Dalam penelitian ini digunakan beberapa sumber pustaka sebagai acuan dan pedoman dalam membangun aplikasi. Pustaka yang digunakan ditinjau dari segi objek, metode, dan hasil penelitian yang dilakukan, seperti yang dapat dilihat pada

Tabel 2.1

Tabel 2.1 Tinjauan Pustaka

No	Penulis	Objek	Metode	Hasil Penelitian Aplikasi
1.	Ridha Naufal (2018)	Hardware Komputer 3D	<i>Multi Maker Cub</i>	Menampilkan objek 3D hardware computer
2.	Rohmat Nianto (2018)	Anatomi Tubuh Manusia	<i>Multi Silinder</i>	Menampilkan objek 3D anatomi lengan
3.	Muhammad Widanarko (2019)	Pengenalan Buah	<i>Text Recognition and Tracking</i>	Menampilkan objek 3D karakter buah-buahan
4.	Zainudin Achmad (2015)	Tata Surya	<i>Based</i>	Menampilkan objek 3D Tata Surya
5.	Usulan Peneliti : Gusti Ichlasul Amal (2021)	Pengenalan Hewan	<i>Based Tracking (user defined target)</i>	Menampilkan objek 3D Hewan

Penelitian yang dilakukan oleh Rohmat Nianto, 2018 dengan judul skripsi “Pembelajaran Anatomi Lengan Menggunakan Multi *Marker Silinder*”, dimana hasil aplikasi yang ditampilkan adalah berupa bentuk visual 3D anatomi tubuh dengan menggunakan metode multi *marker silinder*.

Penelitian AR lainnya dibuat oleh Ridha Naufal, 2018 dengan judul skripsi “Penerapan Multi *Marker* Pada *Augmented Reality* untuk Pengenalan Komponen Hardware Komputer Berbasis Android”. Hasil akhir penelitian tersebut

menampilkan objek 3D hardware komputer dengan menggunakan metode multi *marker* cuboid.

Pada tahun 2019 Muhammad Widanarko, melakukan penelitian dengan judul skripsi “Pengenalan Buah Menggunakan Teknologi *Augmented Reality* Dengan Metode *Marker* Text Recognition and Tracking Berbasis Android”. Penelitian tersebut menghasilkan aplikasi yang dapat menampilkan karakter buah dan informasi lain dengan menggunakan salah satu metode *marker* yaitu text recognition.

Zainudin Achmad (2015). STMIK Akakom Yogyakarta, dengan judul *Augmented Reality* tata surya sebagai visualisasi berbasis Android pada penelitian ini dapat memberikan informasi tentang penggunaan *marker* untuk mendeteksi gambar.

Usulan penelitian yang membedakan dengan yang lain adalah menampilkan karakter animasi 3D berdasarkan *marker* yang digunakan berupa *marker* yang ada di sekitar kita seperti buku pelajaran poster dan *image* target yang pengguna aplikasi tentukan sendiri

2.2 Dasar Teori

2.2.1 Hewan

Hewan atau disebut juga dengan binatang adalah kelompok organisme yang diklasifikasikan dalam kerajaan Animalia atau metazoa, adalah salah satu dari berbagai makhluk hidup di bumi. Sebutan lainnya adalah fauna dan margasatwa (atau satwa saja)

Hewan dalam pengertian sistematika modern mencakup hanya kelompok bersel banyak (multiselular) dan terorganisasi dalam fungsi-fungsi yang berbeda (jaringan), sehingga kelompok ini disebut juga histozoa. Semua binatang heterotrof, artinya tidak membuat energi sendiri, tetapi harus mengambil dari lingkungan sekitarnya (Hayatun,Mursyidah,Amri, 2019).

2.2.2 Vuforia

Vuforia SDK Engine adalah *platform* atau *plugin* perangkat lunak untuk membuat aplikasi *Augmented Reality* khususnya yang digunakan pada aplikasi Unity 3D. *Developer* dapat dengan mudah menambahkan fungsionalitas visi komputer canggih ke aplikasi apa pun, memungkinkan-kannya mengenali gambar dan objek, dan berinteraksi dengan ruang di dunia nyata (Vuforia, 2019).

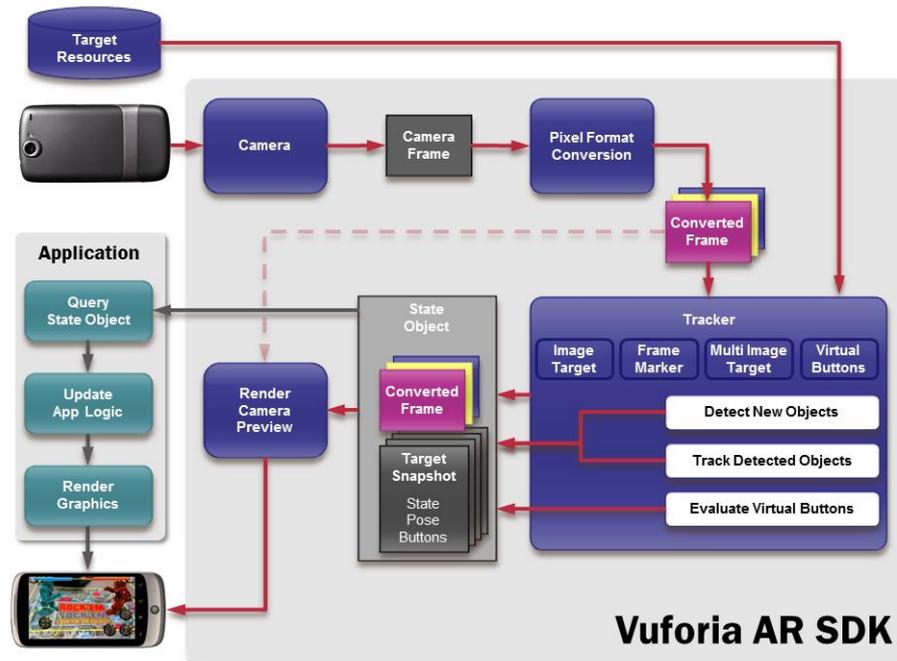
Vuforia SDK ini memiliki berbagai fitur menarik yang digunakan dalam membuat aplikasi *Augmented Reality* seperti:

- a. *Model Targets*: Memungkinkan mengenali objek berdasarkan bentuk menggunakan model 3D yang sudah ada sebelumnya.
- b. *Area Targets*: Memungkinkan untuk melacak dan menambah area dan ruang.
- c. *Image Targets*: Memungkinkan untuk mendeteksi dan mengenali gambar yang digunakan sebagai target.
- d. *Object Targets*: Dibuat dengan memindai objek. Target objek baik digunakan untuk memindai benda seperti mainan atau produk lainnya dengan detail permukaan yang kompleks dan bentuk yang konsisten.
- e. *Multi-Target*: Dibuat menggunakan lebih dari satu target gambar dan dapat disusun menjadi bentuk geometris (kotak).

- f. *Cylinder Targets*: Memungkinkan mengenali gambar yang membungkus objek berbentuk silinder (botol, cangkir, kaleng soda).
- g. *VuMarks*: Ini adalah khusus yang dapat menyandingkan berbagai format data. Mereka mendukung identifikasi dan pelacakan unik untuk aplikasi AR.
- h. *Externals Camera*: Memungkinkan mengakses data video dari kamera di luar ponsel atau tablet saat membuat pengalaman AR.
- i. *Ground Plane*: Memungkinkan untuk menempatkan konten pada permukaan horizontal seperti meja dan lantai.
- j. *User Defined Target*: Memungkinkan pengguna akhir memilih *marker* atau gambar sendiri pada waktu proses pengambilan *image* target

2.2.3 Vuforia AR SDK

Vuforia adalah *Augmented Reality* Software Development Kit (SDK) untuk perangkat mobile yang memungkinkan pembuatan aplikasi *Augmented Reality*. Dulunya lebih dikenal dengan QCAR (Qualcomm Company Augmentend Reality). Ini menggunakan teknologi Computer Vision untuk mengenali dan melacak gambar planar (*Target Image*) dan objek 3D sederhana, seperti kotak, secara realtime. SDK Vuforia mendukung berbagai jenis target 2D dan 3D termasuk target gambar 'markerles', 3D multi target konfigurasi, dan bentuk *Marker Frame*. Fitur tambahan dari SDK termasuk Deteksi Oklusi Local menggunakan 'Tombol virtual', runtime pemilihan gambar target, dan kemampuan untuk membuat dan mengkonfigurasi ulang set pemrograman pada saat *runtime*. Berikut ini adalah gambaran dari diagram aliran data Vuforia. (Vuforia, 2019)



Gambar 2.1 Alur Vuforia AR SDK

2.2.4 User Defined Target

Merujuk pada laman website resmi Vuforia, *User Defined Target* merupakan salah satu metode *marker-based* yang digunakan dalam pengembangan aplikasi *Augmented Reality*. Memungkinkan pengguna akhir memilih gambar pada waktu proses. Dengan demikian, pengguna mengalami AR "kapan saja, di mana saja" dengan memilih gambar - seperti foto, sampul buku, atau poster - dari lingkungan terdekatnya tanpa harus membawa target yang telah ditentukan.

User Defined Target merupakan sebuah metode pelacakan *Augmented Reality* menggunakan objek di dunia nyata sebagai *marker* atau tanpa menggunakan *marker* khusus. *Augmented Reality* dengan Teknik tanpa penanda ini menggunakan teknik pelacakan secara alami (natural feature) bukan

pengenalan penanda (*fiducial marker*). Teknik ini menggunakan prinsip deteksi tepi deteksi sudut dan teksture dari gambar atau objek

User defined target adalah *image target* (gambar penanda) yang terbuat pada saat *runtime* dari frame kamera yang dipilih oleh *user* atau dengan kata lain, *user defined target* merupakan *marker* yang terbentuk pada saat kamera memindai suatu target (Vuforia, 2019)

Target yang didukung

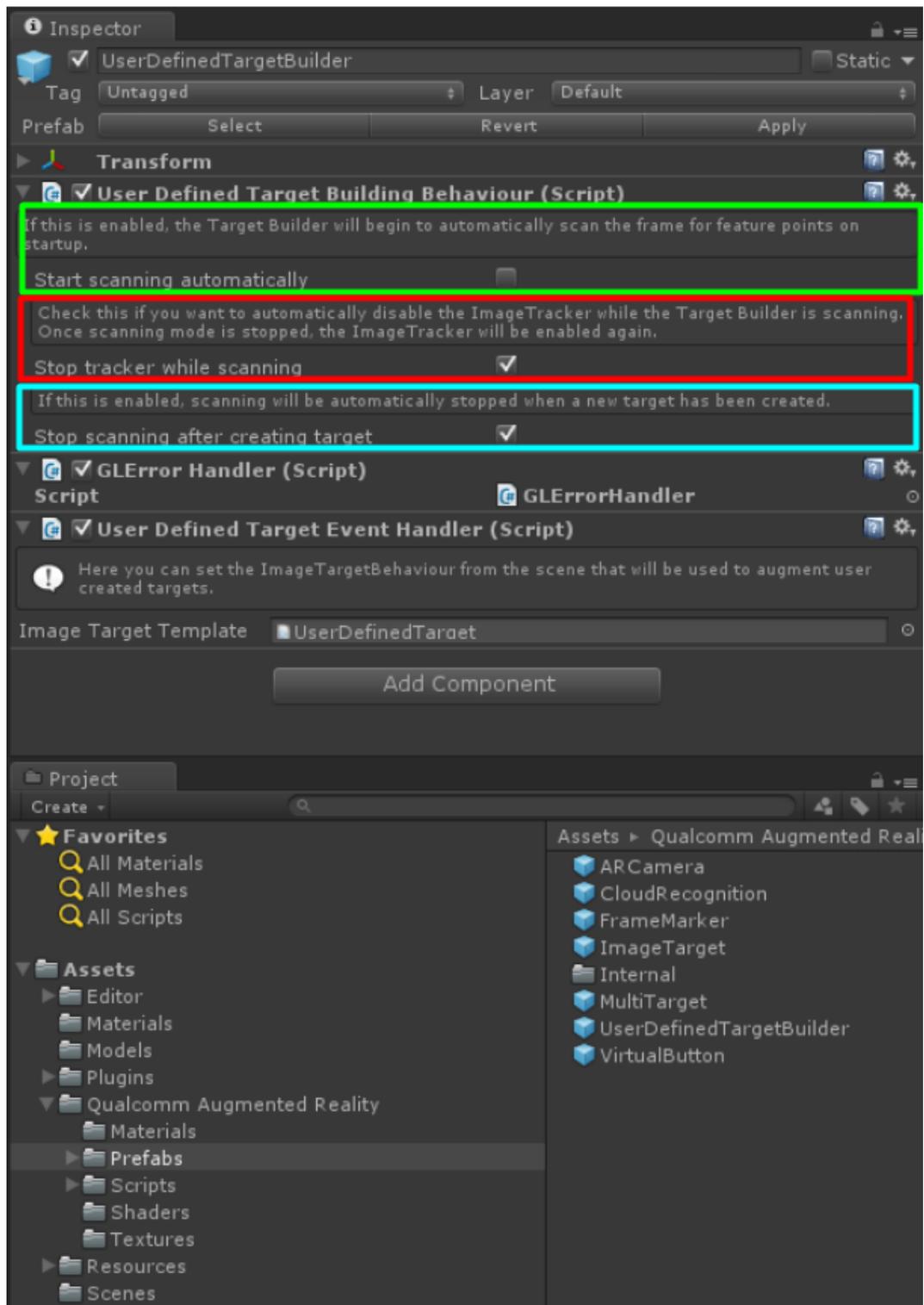
User Defined Target adalah representasi fitur dari hasil pemindaian dengan menggunakan *marker* atau *image target* yang memerlukan penggunaan gambar planar sebagai sumber utama.

Tabel 2.2 *Kriteria attribute yang cocok untuk User Defined Target*

Attribute	Contoh
Kaya akan detailnya(Unik)	Objek dengan <i>pattern</i> warna , gambar dan tulisan yang unik tidak berulang
Kontras yang bagus	Termasuk wilayah terang dan gelap, dan cukup terang
Tidak ada pola yang berulang	Model papan catur, kaca warna polos
Kemudahan Ketersediaan	Buku gambar berwarna, foto dan <i>image</i> yang bergambar dan karya warna lainnya.

User Defined Target dan cara menggunakannya di unity seperti di dokumentasi di bawah ini *UserDefinedTargetBuildingBehaviour* dari prefab *User Defined Target Builder* menyediakan metode untuk memulai dan menghentikan pemindaian target yang baik dan untuk membangun target baru. Dalam inspektornya, opsi berikut dapat dikonfigurasi melalui unity (Vuforia, 2019):

- *Start scanning automatically* (Mulai memindai secara otomatis) Jika ini diaktifkan, pembuat target akan secara otomatis mulai memindai gambar kamera untuk titik fitur saat pengaktifan.
- *Stop tracker while scanning* (Hentikan pelacak saat memindai) jika Anda ingin menghentikan *ImageTracker* secara otomatis saat Target Builder memindai. Setelah mode pemindaian dihentikan, *ImageTracker* akan dimulai kembali secara otomatis.
- *Stop scanning after creating a target* (Hentikan pemindaian setelah membuat target) Jika ini diaktifkan, pemindaian akan secara otomatis dihentikan ketika target baru telah dibuat.



Gambar 2.2 Metode Untuk Memulai Dan Menghentikan Pemindaian Target

Berikut ini beberapa komponen yang digunakan dalam proses pendeteksian fitur yang dilakukan:

a. TRACKABLE STATUS

Informasi status pelacakan yang dilakukan per-*frame* dan akan selalu di-*update* secara *real-time*. Status-status yang dapat terjadi pada saat pelacakan adalah sebagai berikut:

- 1) *DETECTED*: objek terdeteksi dalam *frame*,
- 2) *TRACKED*: objek pelacakan telah terlacak dalam *frame*.
- 3) *EXTENDED_TRACKED*: pelacakan ketika target tidak lagi berada dalam jangkauan pandang kamera (ukuran objek virtual besar),
- 4) *UNKNOWN*: objek pada pelacakan tidak diketahui biasanya dikembalikan sebelum *tracker initialization*,
- 5) *NO_POSE*: pose untuk objek tidak dapat dikirimkan,
- 6) *LIMITED*: pelacakan dari objek terbatas atau dibatasi,
- 7) *UNDEFINE*: objek tidak dapat didefinisikan,
- 8) *NOT_FOUND*: objek tidak ditemukan pada library yang dituju.

b. TRACKABLE NAME

Pelacakan menggunakan kumpulan karakter unik yang disusun sebagai target identifikasi pelacakan. Untuk penulisan nama hanya diperbolehkan maksimal 64 karakter dan hanya mengandung karakter (a-z, A-Z, 0-9, [-_.]).

Beberapa fungsi yang digunakan pada vuforia dalam proses *tracking* adalah sebagai berikut:

- a. *TrackableBehaviour*: Pelacakan pada saat *runtime*.

```
TrackableBehaviour.Trackable [get]
```

- b. *TrackableBehaviour.Status*: Status pelacakan dari *TrackableBehaviour*.

```
TrackableBehaviour.Status
```

- c. *RegisterTrackableEventHandler*: Berfungsi sebagai *register* baru *Tracker Event Handler* pada *tracker*. Penanganan ini dilakukan segera setelah semua *Trackables* telah diperbarui.

```
TrackableBehaviour.RegisterTrackableEventHandler
```

- d. *UnregisterTrackableEventHandler*: Berfungsi sebagai unregisters sebuah *Tracker Event Handler*. Returns “*false*” jika *event handler* tidak ada.

```
TrackableBehaviour.UnregisterTrackableEventHandler
```

- e. *TrackableName*: Pelacakan nama yang ada pada *TrackableBehaviour*.

```
String TrackableBehaviour.TrackableName [get]
```

- f. *OnTrackerUpdate*: Dipicu oleh *TrackerBehaviour* setelah itu diperbarui.

```
void TrackableBehaviour.OnTrackerUpdate (Status newStatus) [inline, virtual]
```

- g. *ITrackableEventHandler*: Fungsi ini berisikan antarmuka untuk menangani perubahan yang dilacak.

```
ITrackableEventHandler
```

- h. *OnTrackableStateChanged*: Sebuah handler *custom* yang mengimplementasikan *ITrackableEventHandler* antarmuka.

```
DefaultTrackableEventHandler.OnTrackableStateChanged
```

2.2.5 Unity 3D

Unity 3D adalah sebuah tool *game engine* yang digunakan untuk mempermudah proses pembuatan *game* dan aplikasi 2D, 3D, VR dan AR. *Game engine* ini dilengkapi dengan GUI (*graphic user interface*) yang memudahkan untuk membuat, mengedit serta membuat script untuk menciptakan sebuah *game* 3D. Unity 3D mendukung *multi-platform*, *game* atau aplikasi yang dibuat menggunakan *game engine* ini dapat di *build* ke banyak *platform device* diantaranya PC (windows, mac, linux), android, ios, windows phone, tizen, web, game console (PS, xbox, nitendo wii) bahkan sampai *smart tv*. *Scripting* pada Unity 3D sangat mudah dipelajari dan cukup sederhana. Unity 3d mendukung 3 bahasa pemrograman yaitu JavaScript, C# dan Boo.