



### **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 III

### METODOLOGI PENELITIAN DAN PERANCANGAN SISTEM

#### 3.1 Metode Penelitian

Metode penelitian yang digunakan dalam membuat aplikasi ini adalah sebagai berikut.

##### a. Studi Literatur

Tahap ini dilakukan dengan mempelajari teori-teori dan algoritma yang akan digunakan dalam penelitian, terutama mengenai tata ruang wilayah, *Augmented Reality*, algoritma Fisher Yates, Unity, dan Vuforia SDK. Sumber dapat berupa buku, jurnal, makalah ilmiah, artikel, dan sebagainya.

##### b. Perancangan Sistem

Pada tahap ini, perancangan dilakukan dengan pembuatan *Data Flow Diagram* dan *flowchart*. Selain itu juga dilakukan perancangan desain tampilan, struktur menu aplikasi, dan penentuan penanda yang digunakan untuk memunculkan objek-objek yang akan ditampilkan.

##### c. Implementasi

Pada tahap ini akan dilakukan proses pembuatan aplikasi secara bertahap mulai dari *coding*, *debugging*, dan *testing* akhir hasil aplikasi yang dibuat.

##### d. Uji Coba dan Evaluasi

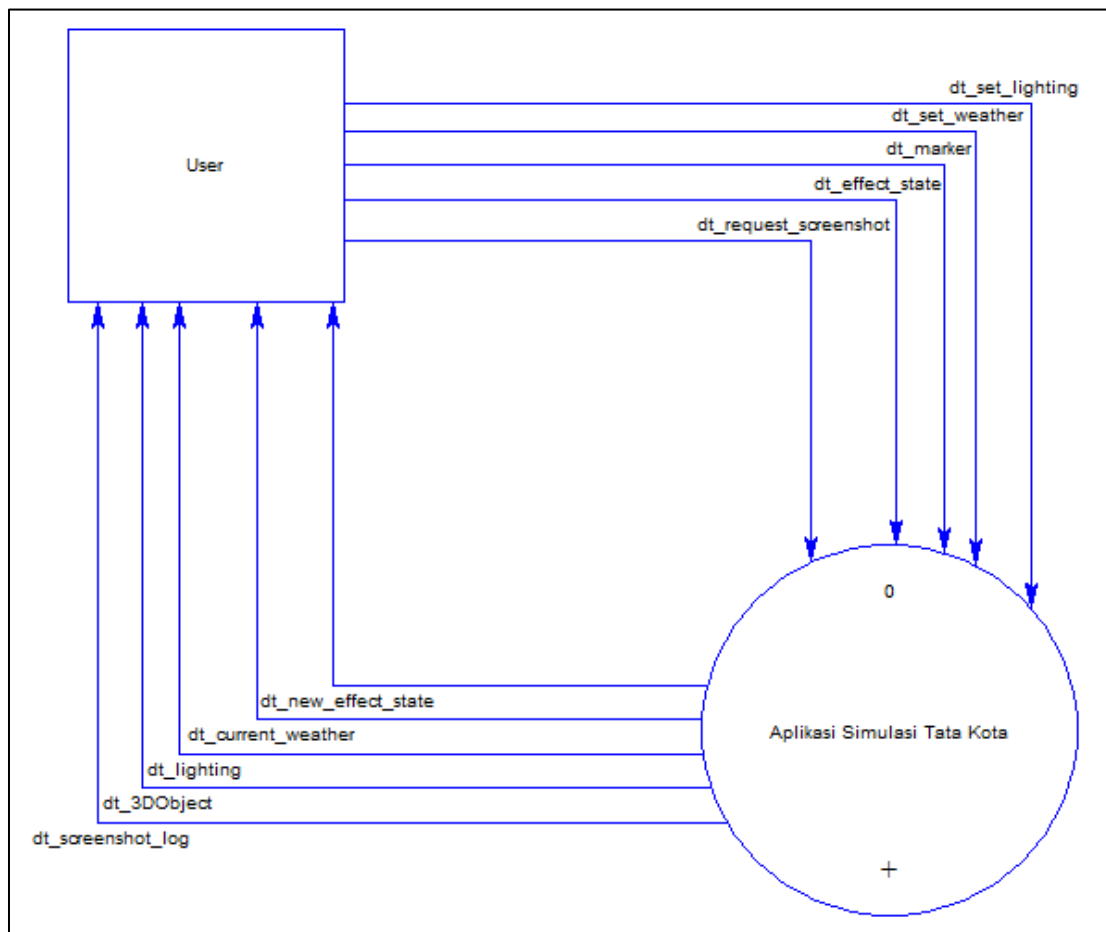
Aplikasi yang telah dibuat diuji coba dengan tujuan untuk mengetahui *user experience* saat menggunakan aplikasi, serta mencari *error* atau *bug* yang ada.

Setelah melakukan uji coba, para responden akan diminta untuk mengisi sebuah survei. Metode pengukuran hasil survei akan menggunakan *Likert Scale*.

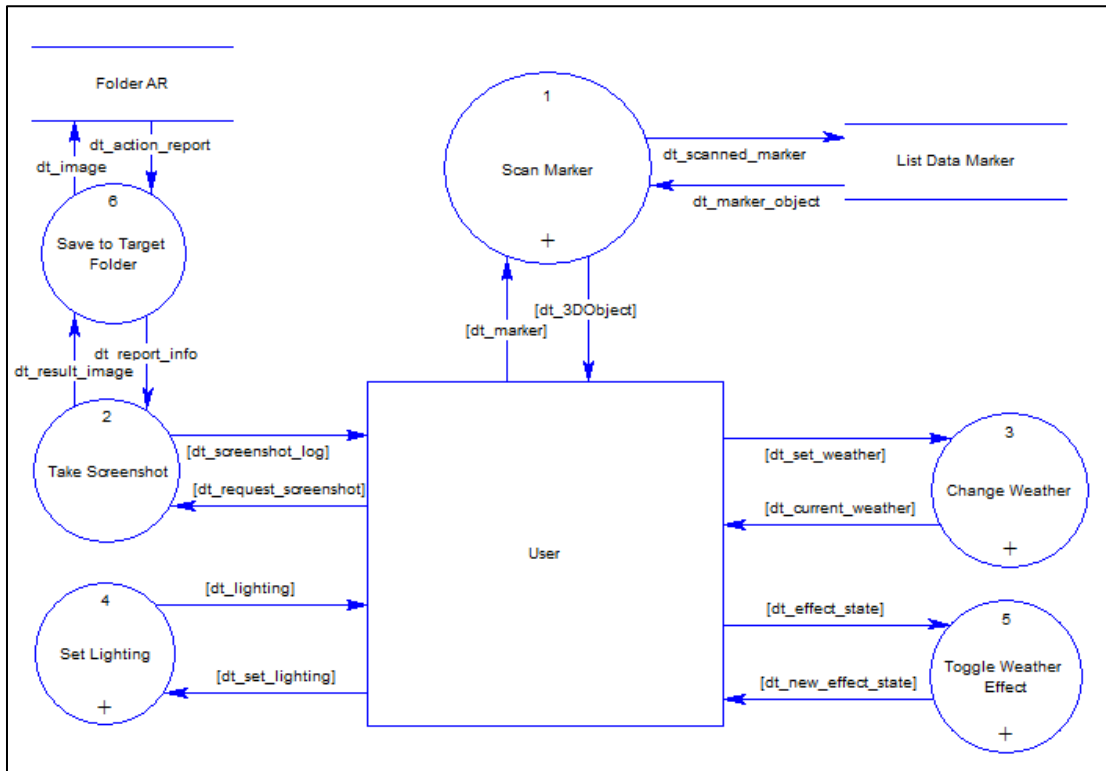
## 3.2 Perancangan Aplikasi

### 3.2.1 Data Flow Diagram

Pada aplikasi ini, *input* dari *user* berupa sebuah marker yang telah dicetak, kemudian diarahkan pada kamera *smartphone* yang digunakan.

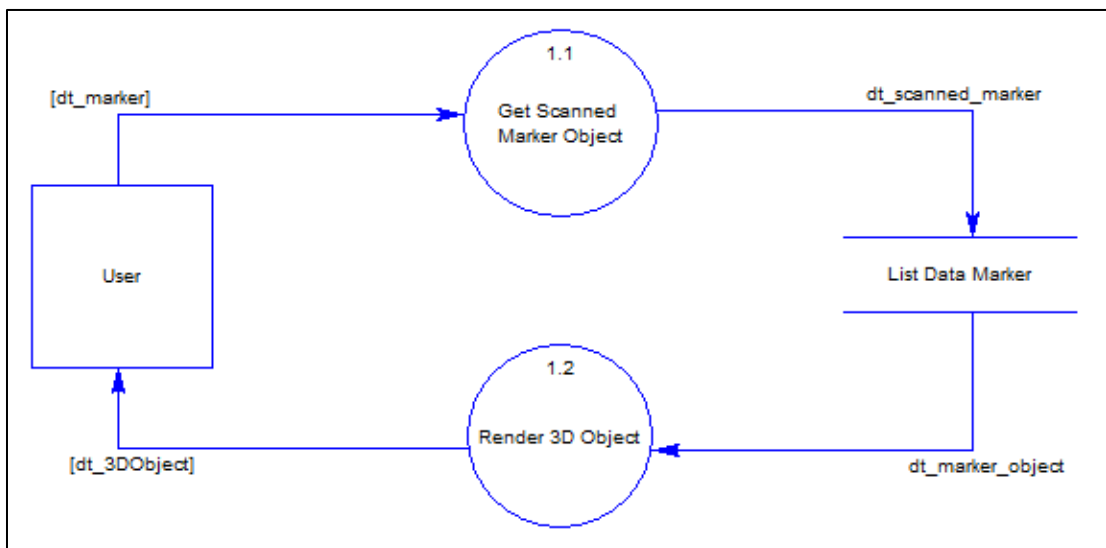


Gambar 3.1 Data Flow Diagram Level 0



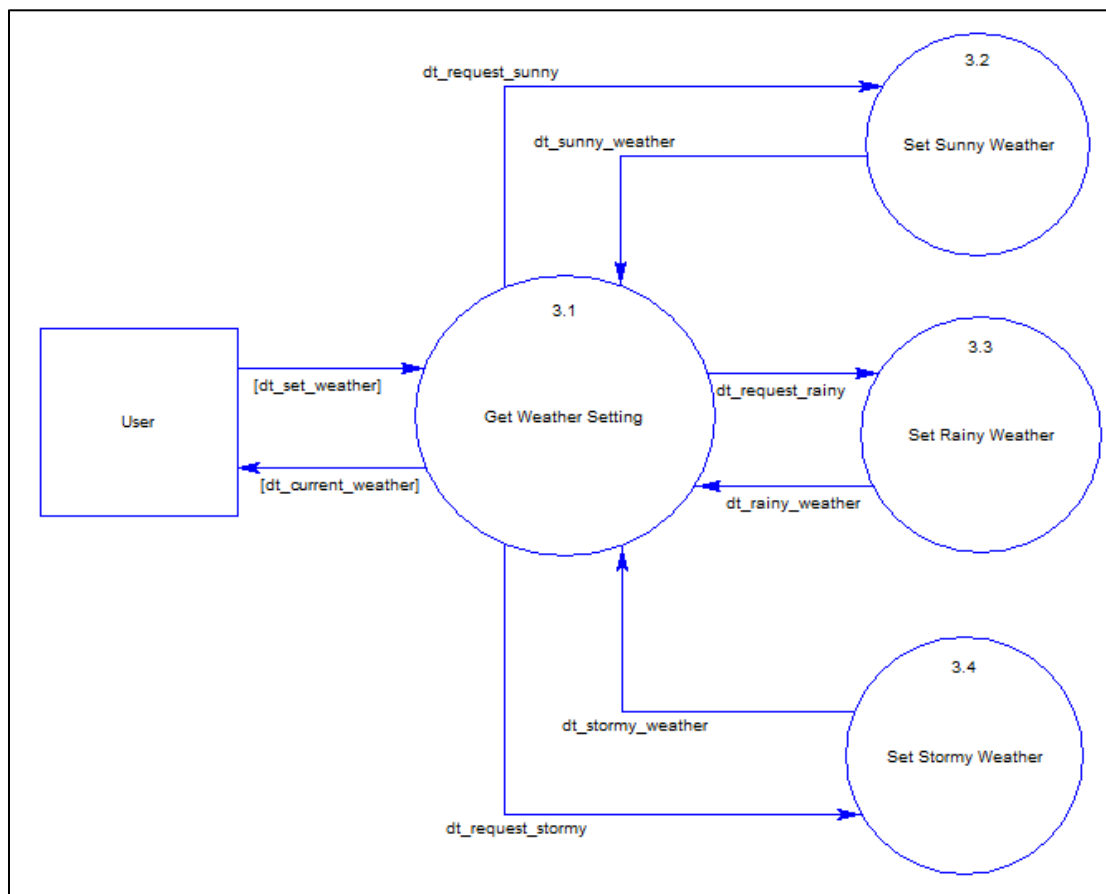
Gambar 3.2 Data Flow Diagram Level 1

Pada Gambar 3.2 di atas menunjukkan *Data Flow Diagram Level 1*. Setelah sistem menerima beberapa *input* yang dapat diberikan oleh *user*, terdapat beberapa proses yang dapat dilakukan, seperti *scan marker*, *take screenshot*, *change weather*, *set lighting*, dan *toggle weather effect*.



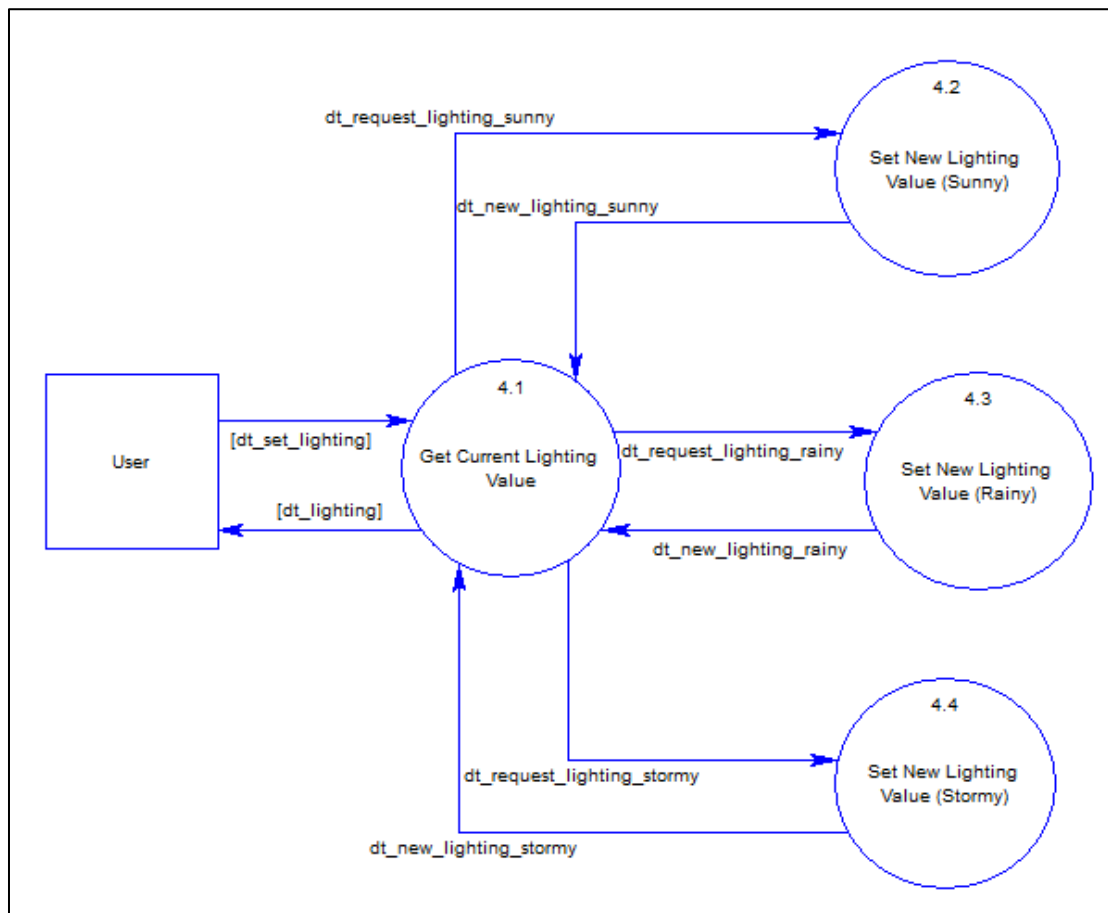
Gambar 3.3 Data Flow Diagram Level 2 Subproses *Scan Marker*

Pada Gambar 3.3 di atas dijelaskan proses *scan marker* yang dilakukan oleh sistem, dimana data gambar dari *marker* atau *tag* yang di-*scan* oleh kamera akan disesuaikan dengan data pada *database* aplikasi. Jika *marker* terdaftar pada *database*, maka objek yang akan ditampilkan pada *marker* tersebut akan di-*render* dan ditampilkan pada layar. Selain itu, pada Gambar 3.2 sebelumnya juga dijelaskan proses *take screenshot* sebagai salah satu fitur yang dapat digunakan oleh *user* pada aplikasi ini. Setelah gambar diambil, target folder akan dibuat dan diarahkan pada folder bernama “AR” untuk lokasi penyimpanan data gambar hasil proses *take screenshot*.



Gambar 3.4 Data Flow Diagram Level 2 Subproses *Change Weather*

Pada Gambar 3.4 di atas dijelaskan proses dari *change weather*. Pertama-tama sistem akan memeriksa *current weather setting* yang sedang aktif. Jika *user* ingin mengganti kondisi cuaca yang ditampilkan, maka sistem akan menjalankan proses pergantian cuaca sesuai dengan permintaan *user*. *Set sunny weather* digunakan untuk mengubah cuaca menjadi cerah. *Set rainy weather* digunakan untuk mengubah cuaca menjadi hujan. *Set stormy weather* digunakan untuk mengubah cuaca menjadi hujan berangin (badai).

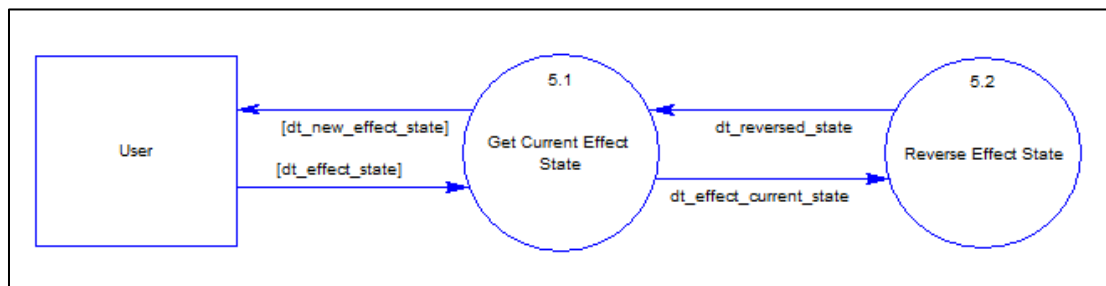


Gambar 3.5 Data Flow Diagram Level 2 Subproses *Set Lighting*

Gambar 3.5 di atas menjabarkan proses *set lighting* yang dapat dilakukan oleh *user*. Proses akan dimulai dengan mengambil nilai atau *value* dari objek *lighting* yang

sedang ditampilkan, kemudian nilai tersebut dapat diubah berdasarkan keinginan *user* yang disesuaikan dengan pencahayaan pada efek cuaca yang sedang aktif pada saat itu. Penyesuaian ini bertujuan untuk membedakan pencahayaan yang ada ketika cuaca cerah, hujan, atau hujan berangin dan juga membedakan intensitas serta warna cahaya ketika pagi, siang, sore, dan malam hari. Setelah nilai dari cahaya yang baru telah di-*set*, maka warna dan intensitas cahaya yang ditampilkan akan diubah sesuai dengan kriteria yang telah ditentukan.

Selanjutnya, pada Gambar 3.6 di bawah dijabarkan proses dari *toggle weather effect* yang berfungsi untuk menyalakan dan mematikan efek-efek yang bersangkutan dengan cuaca yang sedang aktif.

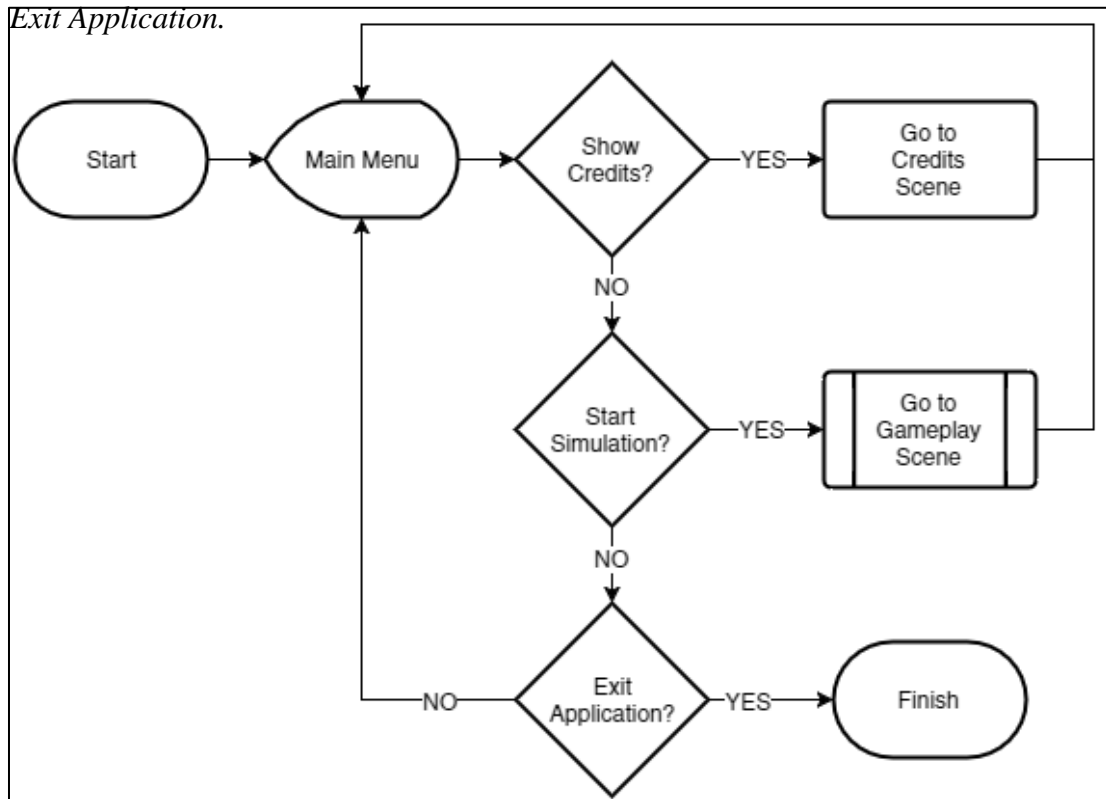


Gambar 3.6 Data Flow Diagram Level 2 Subproses *Toggle Weather Effect*

Proses dilakukan dengan menerima *state* dari *weather effect* pada aplikasi. Proses *Reverse Effect State* akan dilakukan untuk membalikkan keadaan dari *state* yang ada. Jika *state* dalam keadaan aktif, maka akan diubah menjadi nonaktif, namun apabila *state* dalam keadaan nonaktif maka akan diubah menjadi aktif.

### 3.2.2 Flowchart

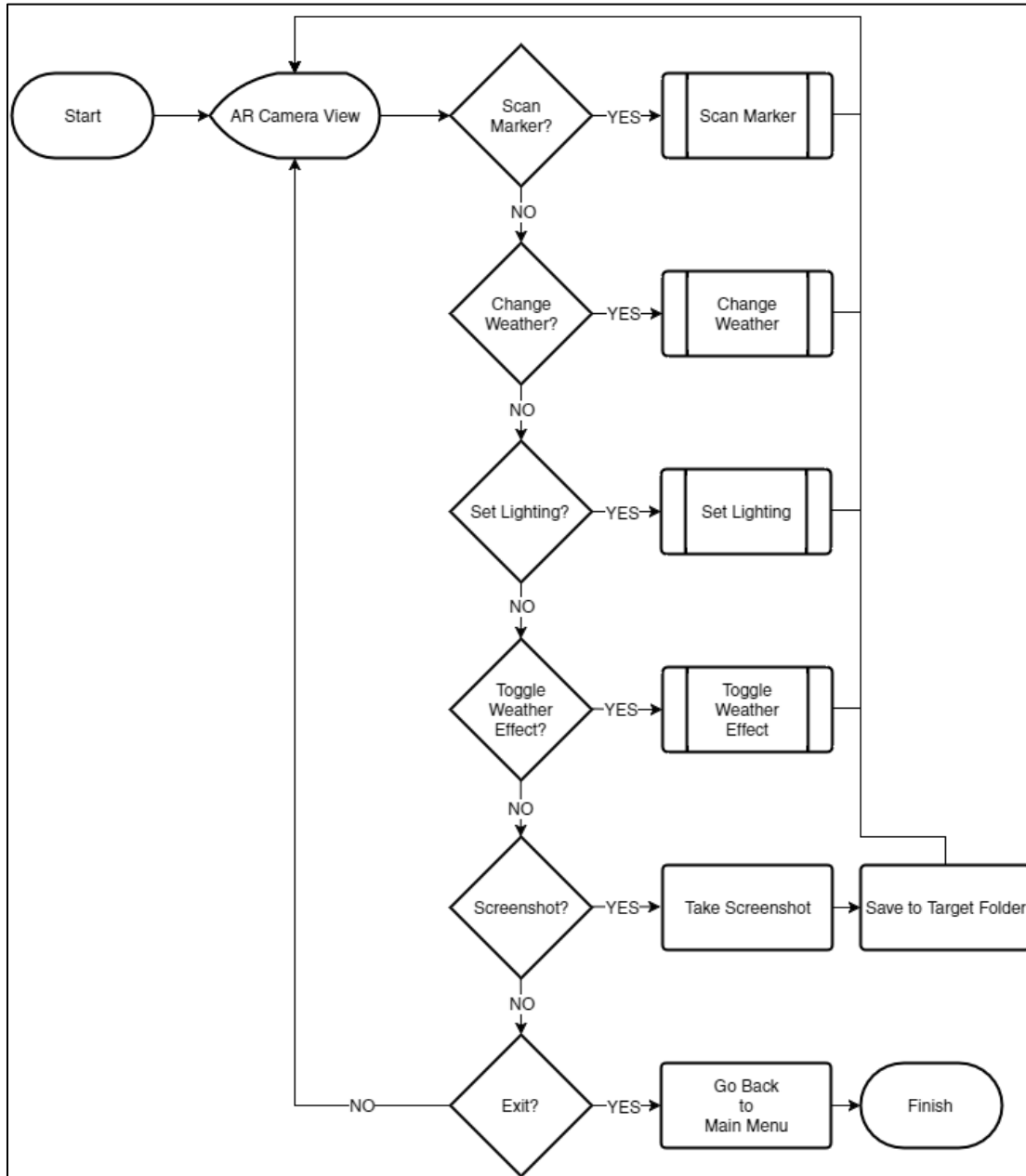
Pada dasarnya, aplikasi akan dimulai dari *Main Menu scene* dimana *user* akan diberikan tiga buah pilihan, yaitu *Go to Credits Scene*, *Go to Gameplay Scene*, atau



Gambar 3.7 Flowchart Main Menu

Gambar 3.7 di atas menjelaskan proses *Go to Credits Scene* akan membawa *user* ke halaman *Credits*, sedangkan *Go to Gameplay Scene* akan membawa *user* ke *Gameplay Scene* untuk melakukan proses simulasi. Kemudian, jika *user* memilih *Exit Application*, maka akan dilakukan konfirmasi terlebih dahulu, jika ya maka aplikasi ditutup. Jika tidak, maka *user* akan dikembalikan ke proses pemilihan *menu*.

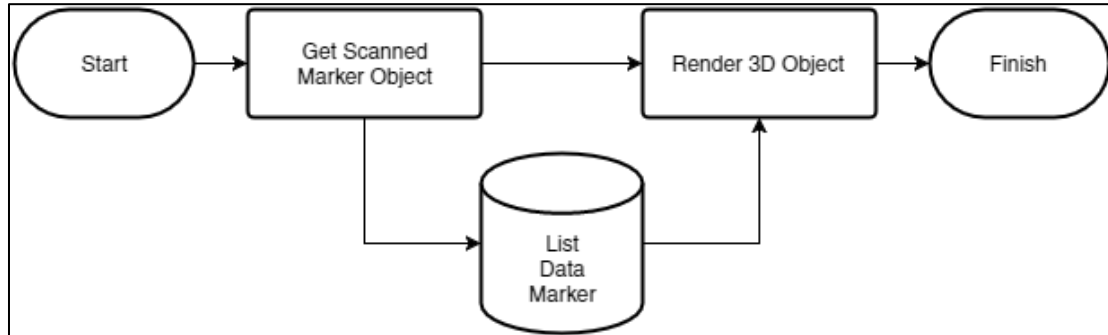




Gambar 3.8 Flowchart Go to Gameplay Scene

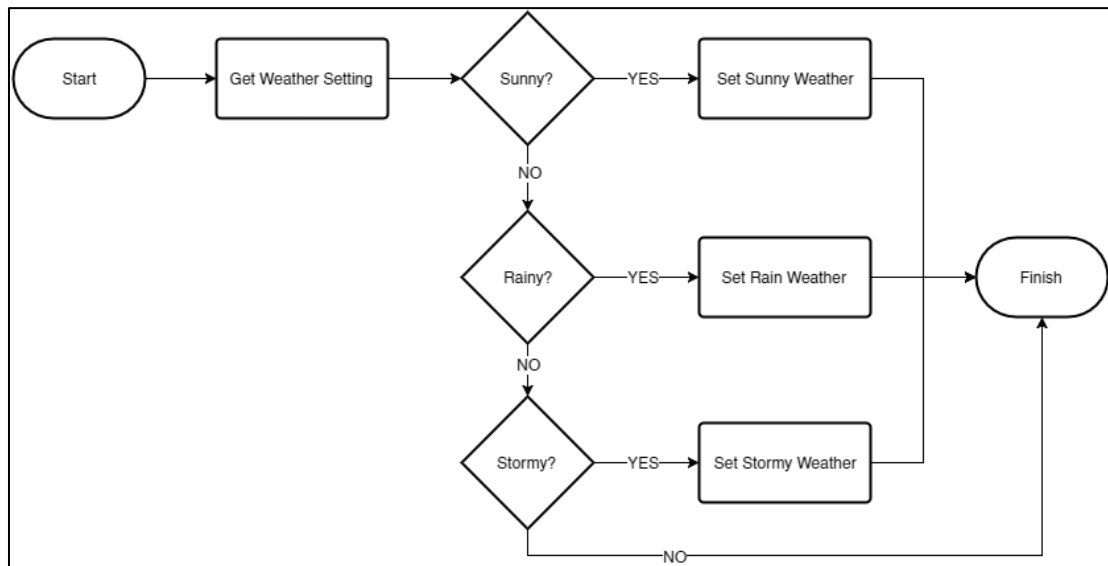
Pada Gambar 3.8 di atas, terdapat proses-proses yang dapat dilakukan pada *Gameplay Scene*. Pada tampilan utama *AR Camera View* dapat dilakukan berbagai proses seperti *scan marker*, *take screenshot*, *set lighting*, *change weather*, *toggle weather effect*, dan *go back to main menu*. Pada *scene* ini, *AR Camera* akan selalu dalam keadaan *stand-by* untuk melakukan *scan* pada *marker* atau *tag* yang berada di

dalam jangkauan kamera. Pada Gambar 3.9 di bawah dijabarkan proses dari *scan marker* tersebut.



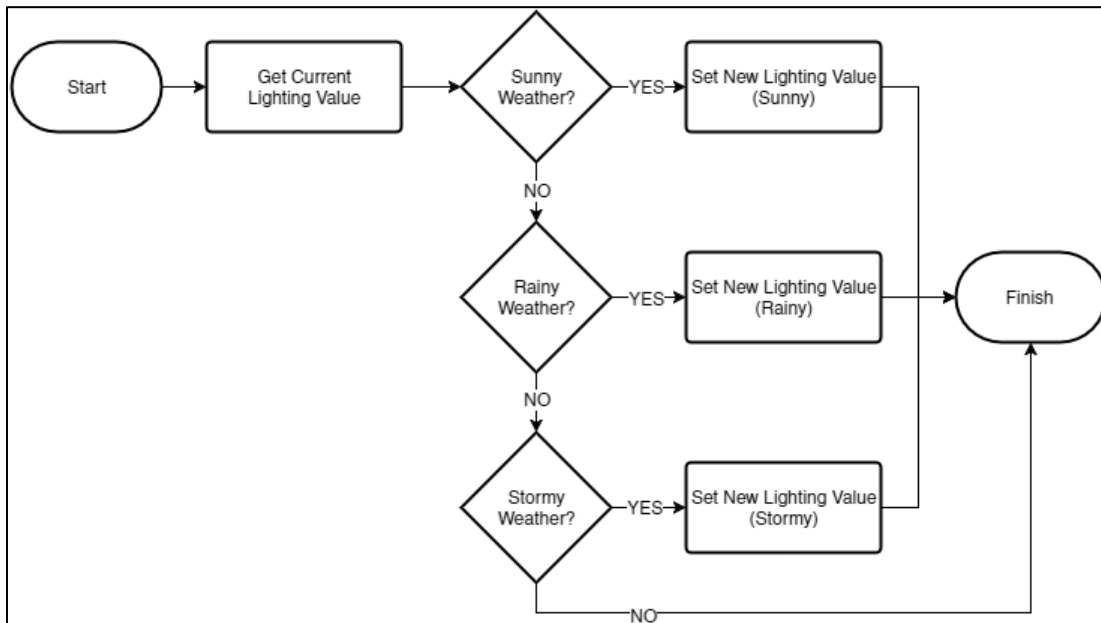
Gambar 3.9 *Flowchart* Proses *Scan Marker*

Pada Gambar 3.10 di bawah, dijelaskan proses pada fitur perubahan efek cuaca (*change weather*). Efek cuaca dapat diubah menjadi cerah, hujan, dan hujan berangin (badai).



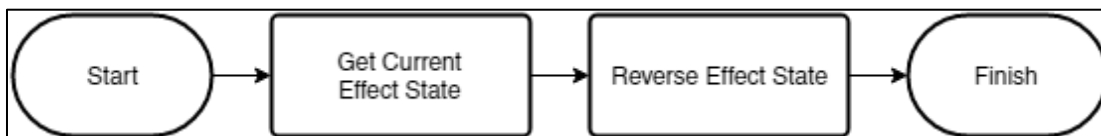
Gambar 3.10 *Flowchart* Proses *Change Weather*

Gambar 3.11 di bawah menjelaskan proses *set lighting* yang memungkinkan *user* untuk mengatur keadaan cahaya yang ditampilkan dalam proses simulasi sesuai dengan cuaca yang sedang aktif.



Gambar 3.11 *Flowchart Proses Set Lighting*

Kemudian, pada Gambar 3.12 di bawah dijelaskan fitur *toggle weather effect*, apabila *user* tidak menginginkan adanya efek cuaca *user* dapat menonaktifkan atau pun mengaktifkan efek cuaca yang ditampilkan pada layar.



Gambar 3.12 *Flowchart Proses Toggle Weather Effect*

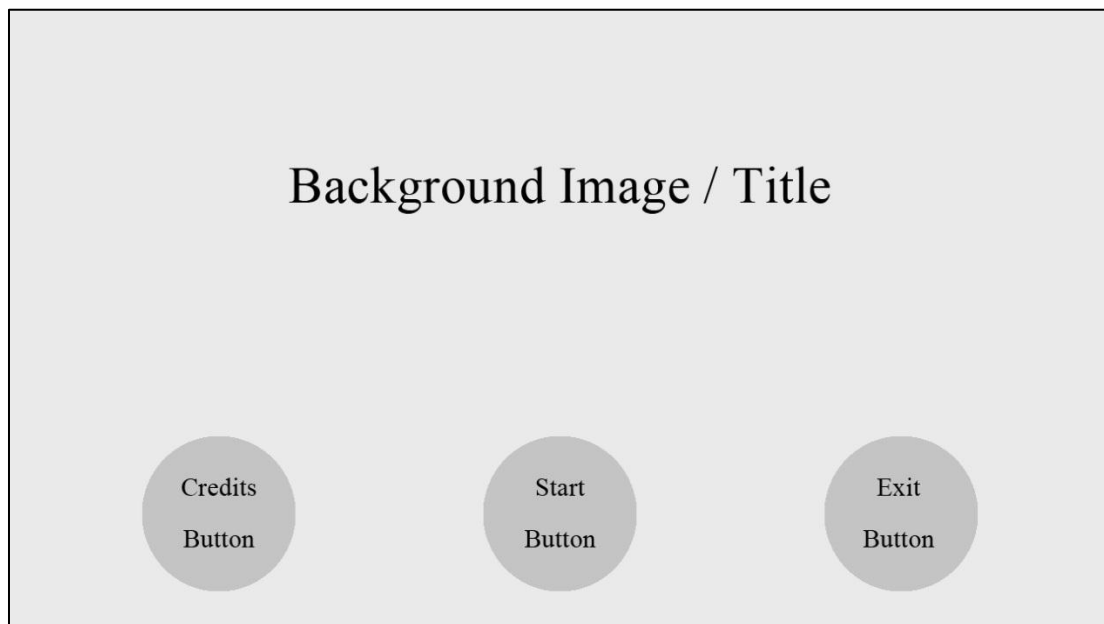
Sebelumnya, pada Gambar 3.8 juga dijelaskan proses *take screenshot* yang berfungsi untuk menyimpan tampilan yang ada pada layar. Hasil *screenshot* akan disimpan pada *gallery smartphone* dengan nama folder AR. Terakhir, jika *user* telah selesai melakukan simulasi, *user* dapat kembali ke *main menu*.

### 3.3 Perancangan Tampilan Antarmuka Aplikasi

Tampilan pada aplikasi ini terdiri dari tiga buah *scene*, yaitu *Main Menu*, *Gameplay*, dan *Credits*. Perancangan *scene* yang ada telah dibuat sedemikian rupa agar lebih mudah dalam proses penggunaannya. Untuk lebih jelasnya, maka akan dijelaskan lebih lanjut mengenai setiap *scene* sebagai berikut.

#### 1. *Main Menu Scene*

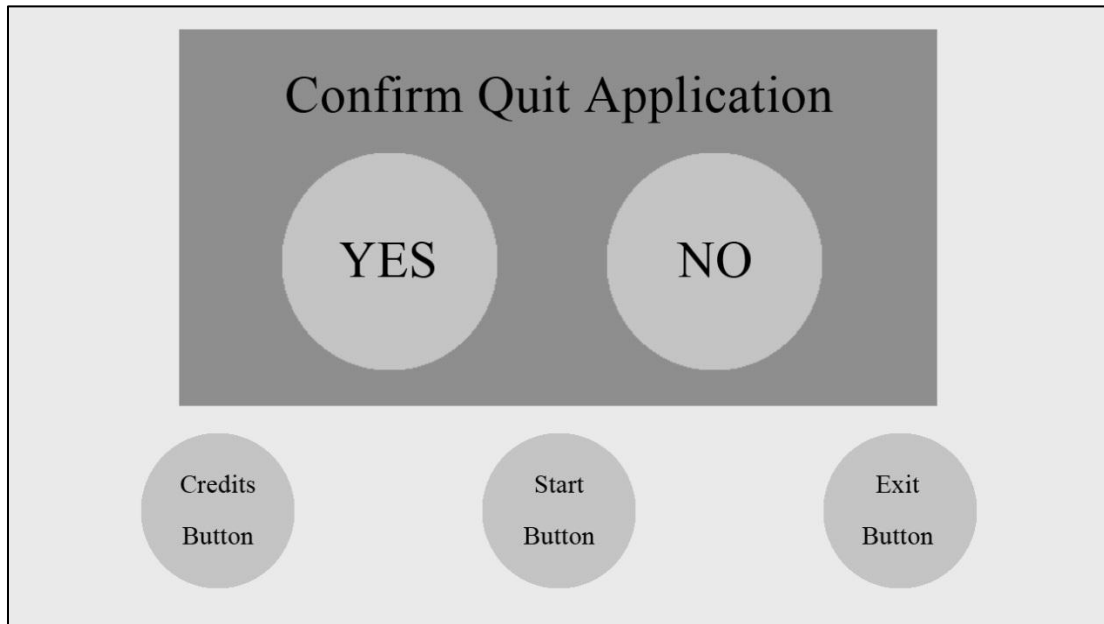
Gambar 3.13 di bawah memperlihatkan desain tampilan dari *Main Menu scene* yang terdiri dari beberapa bagian, yaitu tombol untuk menuju ke *Credits scene*, tombol untuk menuju *Gameplay scene*, dan tombol untuk keluar dari aplikasi.



Gambar 3.13 Rancangan Tampilan *Main Menu Scene*

Kemudian, pada Gambar 3.14 di bawah menjelaskan apabila *user* menekan tombol *Exit*, maka akan ditampilkan sebuah kotak konfirmasi *Exit Application* terlebih dahulu yang akan menanyakan apakah benar *user* ingin keluar dari aplikasi

atau tidak. Jika ya, maka aplikasi ditutup, tetapi jika tidak, maka kotak konfirmasi akan ditutup dan kembali ke tampilan awal *Main Menu scene*.



Gambar 3.14 Rancangan Tampilan Kolom Konfirmasi *Exit Application*

## 2. *Credits Scene*

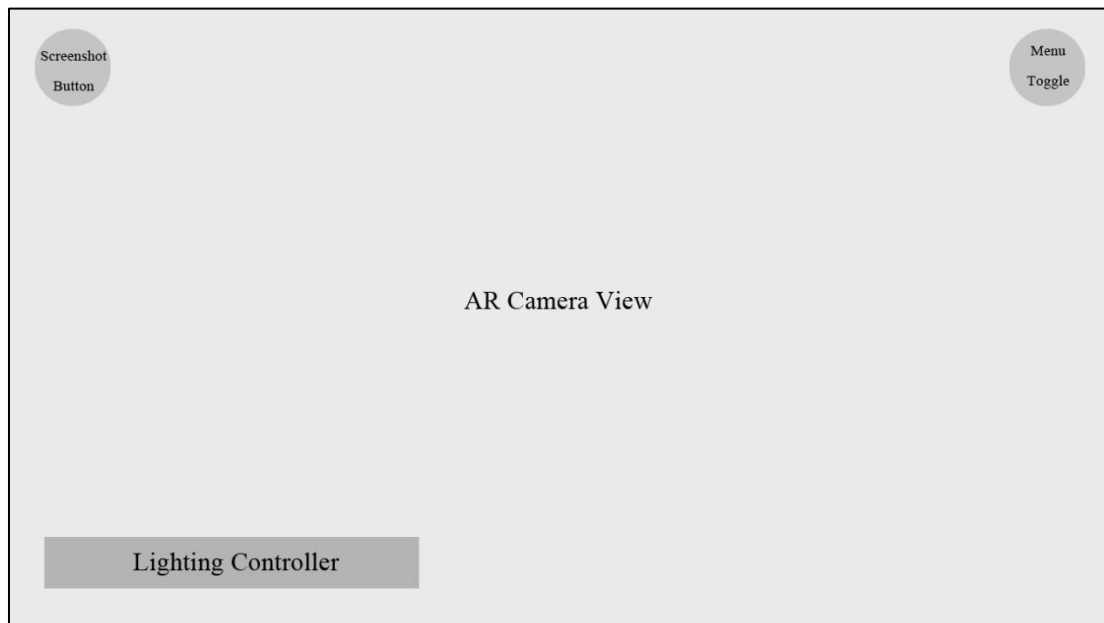


Gambar 3.15 Rancangan Tampilan *Credits Scene*

Gambar 3.15 di atas memperlihatkan desain tampilan *Credits scene* yang berisi data identitas pembuat aplikasi, nama pembimbing, logo universitas, dan program-program maupun *library* yang digunakan dalam proses pembuatan aplikasi. Untuk keluar dari *scene Credits* dapat digunakan tombol *Exit* yang akan membawa *user* kembali ke tampilan *Main Menu*.

### 3. *Gameplay Scene*

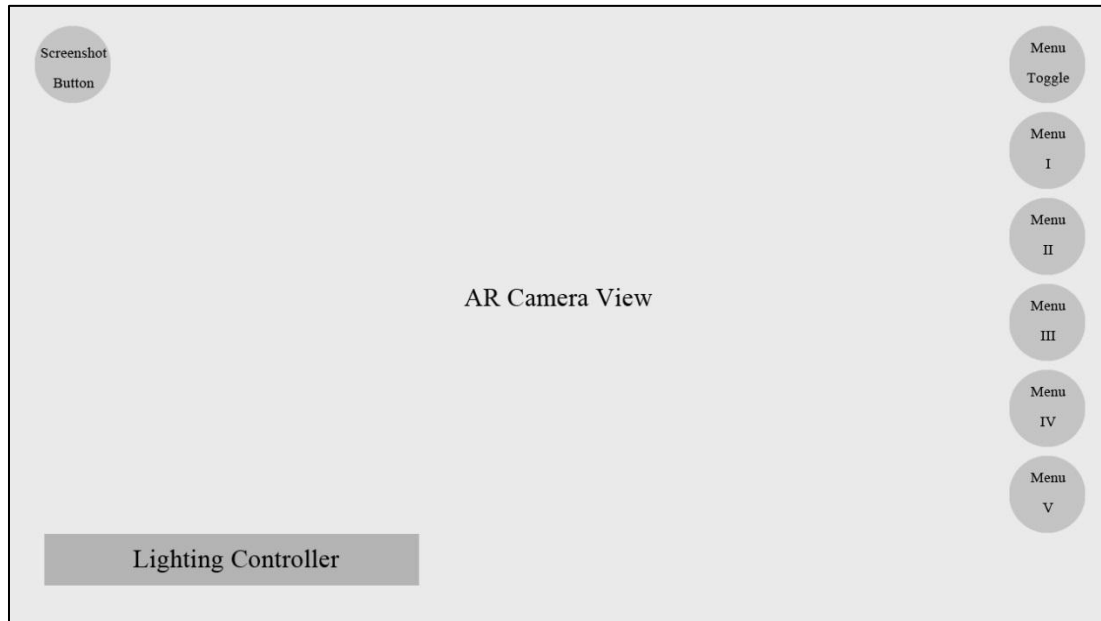
Gambar 3.16 di bawah menampilkan *Gameplay scene* yang memiliki berbagai fitur yang dapat digunakan. Pada bagian kiri atas terdapat sebuah tombol yang digunakan untuk mengambil *screenshot* dari tampilan yang ada pada layar.



Gambar 3.16 Rancangan Tampilan *Gameplay Scene* (Menu Tertutup)

Kemudian, pada Gambar 3.17 di bawah terdapat tombol *menu* yang dapat memanggil beberapa *menu* lainnya, seperti tombol *Sunny Weather* yang digunakan untuk mengubah cuaca menjadi cerah, tombol *Weather Rainy* untuk mengubah cuaca menjadi hujan, dan tombol *Weather Stormy* untuk mengubah cuaca menjadi hujan

berangin (badai). Selain itu, terdapat 2 tombol lagi yang dapat digunakan. Tombol *Weather Effect* yang digunakan untuk mengaktifkan atau menonaktifkan efek cuaca beserta suara yang ada, juga tombol *Back to Menu* atau *Exit* yang akan mengembalikan *user* ke *Main Menu*.

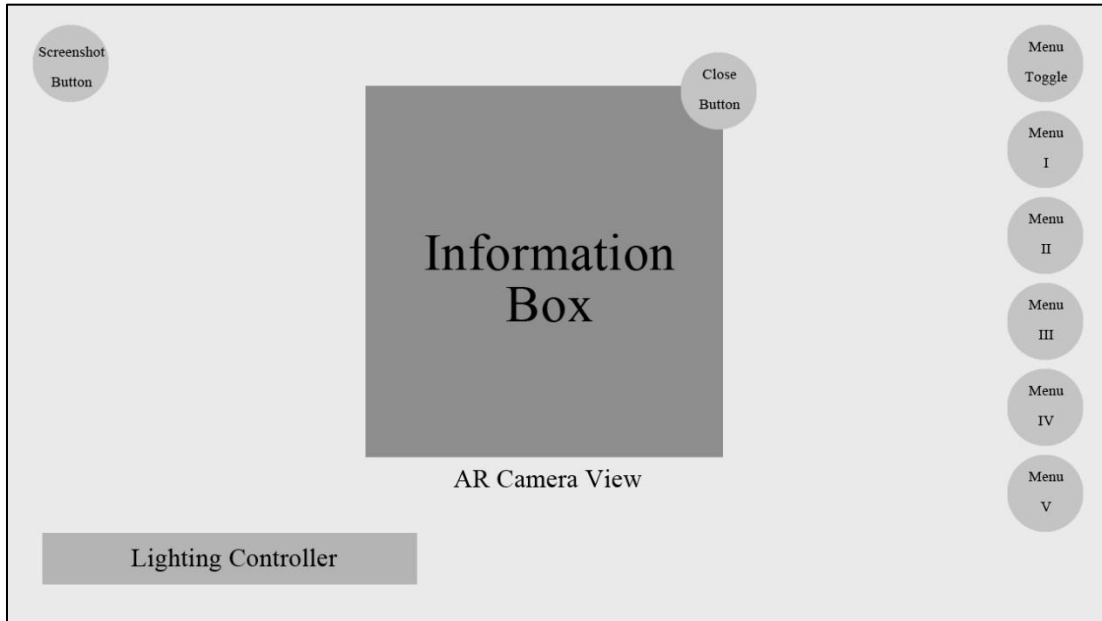


Gambar 3.17 Rancangan Tampilan *Gameplay Scene* (Menu Terbuka)

Gambar 3.17 di atas juga menampilkan *lighting controller* pada bagian kiri bawah yang berfungsi untuk mengatur pencahayaan pagi, siang, sore, dan malam sesuai dengan efek cuaca yang sedang aktif. Terakhir, pada tampilan layar utama aplikasi terdapat *AR Camera View* yang berfungsi untuk membaca *marker* atau *tag* yang sudah terdaftar pada *database* Vuforia, dan memunculkan objek 3D dari *marker* atau *tag* tersebut.

Gambar 3.18 di bawah menampilkan tampilan kotak informasi yang akan muncul ketika suatu tempat atau gedung yang muncul pada layar di-*click*. Kotak informasi berisi informasi singkat mengenai tempat atau gedung yang di-*click* dan dapat di-

*close* dengan menggunakan tombol yang tersedia di bagian kanan atas kotak informasi tersebut.



Gambar 3.18 Rancangan Tampilan *Information Box*