

BAB II

KAJIAN LITERATUR

2.1 Virtual Reality

Virtual Reality (VR) merupakan teknologi berbasis komputer yang digunakan untuk mensimulasikan objek atau lingkungan nyata ke dalam bentuk digital tiga dimensi. Melalui teknologi ini, pengguna tidak hanya melihat tampilan visual, tetapi juga memperoleh pengalaman seolah-olah hadir dan terlibat secara langsung di dalam lingkungan virtual yang disajikan. Dengan demikian, VR dapat dipahami sebagai teknologi yang mampu menghadirkan simulasi ruang secara lebih nyata dibandingkan media digital biasa [6]. Selain sebagai teknologi simulasi, *Virtual Reality* juga memiliki beberapa komponen inti yang membentuk pengalaman pengguna di dalam lingkungan virtual. Komponen tersebut meliputi *immersion*, *interaction*, dan *imagination*. *Immersion* berkaitan dengan kemampuan sistem dalam membuat pengguna merasa terlibat atau berada di dalam lingkungan virtual. *Interaction* menunjukkan kemampuan pengguna untuk mengendalikan, menavigasi, atau memengaruhi objek dan aktivitas di dalam dunia virtual. Sementara itu, *imagination* berkaitan dengan kemampuan lingkungan virtual dalam membangun pengalaman yang dapat dipahami, dibayangkan, dan dirasakan oleh pengguna. Ketiga komponen tersebut menjadi dasar penting dalam pengembangan media berbasis *Virtual Reality*, karena kualitas pengalaman pengguna tidak hanya ditentukan oleh tampilan visual, tetapi juga oleh tingkat keterlibatan, interaksi, dan rasa hadir pengguna di dalam lingkungan virtual [5]. Secara konseptual, *Virtual Reality* dibangun oleh beberapa elemen utama. Elemen pertama adalah *virtual world*, yaitu lingkungan digital tiga dimensi yang dibentuk melalui proses komputasi seperti *modeling*, *rendering*, dan teknik visual lainnya. Elemen kedua adalah *immersion*, yaitu kondisi ketika pengguna merasakan keterlibatan yang kuat di dalam dunia virtual. Elemen berikutnya adalah *sensory feedback*, yakni respons yang disampaikan sistem kepada indra pengguna, seperti visual, audio, maupun sentuhan. Adapun elemen terakhir adalah *interactivity*, yaitu kemampuan sistem untuk merespons tindakan pengguna sehingga pengguna dapat berinteraksi secara langsung dengan lingkungan virtual yang disediakan [7].

Virtual Reality dapat dibedakan berdasarkan tingkat *immersion* menjadi *immersive*, *semi-immersive*, dan *non-immersive*. Klasifikasi ini menunjukkan bahwa pengalaman virtual yang diperoleh pengguna tidak selalu berada pada tingkat keterlibatan yang sama. Pada *non-immersive virtual reality*, pengguna berinteraksi dengan lingkungan virtual yang

ditampilkan melalui layar, namun tetap dapat merasakan keberadaan dunia nyata di sekitarnya. Interaksi pada jenis ini umumnya dilakukan melalui perangkat seperti komputer, tablet, atau *joystick*. Adapun *semi-immersive virtual reality* menghadirkan pengalaman yang sebagian imersif melalui tampilan layar dan dalam beberapa kondisi memungkinkan interaksi melalui gerakan tubuh, walaupun pengguna masih rentan terhadap gangguan dari lingkungan sekitar [8]

Untuk menghadirkan pengalaman VR yang baik, teknologi ini memerlukan dukungan perangkat pendukung yang memadai. Perangkat tersebut tidak hanya mencakup perangkat keras dan perangkat lunak, tetapi juga berkaitan dengan aspek persepsi manusia. Perangkat keras berperan sebagai sarana *input* dan *output* agar pengguna dapat berinteraksi dengan sistem, sedangkan perangkat lunak mengolah data dan menghasilkan respons yang sesuai. Di sisi lain, pemahaman terhadap persepsi manusia juga penting karena kualitas pengalaman *virtual* sangat dipengaruhi oleh cara sistem menyajikan rangsangan yang dapat diterima oleh indra pengguna [9]

2.2 Virtual Tour

Virtual tour merupakan bentuk tur terpandu yang difasilitasi melalui teknologi digital untuk menampilkan suatu lokasi atau fasilitas secara virtual. Dalam konteks pendidikan, *virtual tour* digunakan sebagai media yang memungkinkan pengguna memperoleh gambaran mengenai lingkungan tertentu, seperti laboratorium, kawasan industri, maupun kampus, tanpa harus hadir secara langsung di lokasi. Dengan demikian, *virtual tour* dapat dipahami sebagai media penyajian ruang yang mendukung pengalaman eksplorasi secara lebih imersif dibandingkan penyajian informasi statis [10].

Perkembangan *virtual tour* menunjukkan bahwa media ini tidak hanya berfungsi sebagai sarana visualisasi, tetapi juga sebagai sarana penyampaian informasi yang interaktif. Pengguna tidak sekadar melihat representasi lokasi, melainkan dapat menelusuri area tertentu, mengakses informasi fasilitas, dan memahami tata letak lingkungan yang ditampilkan. Dalam pengembangan berbasis kampus, *virtual tour* juga dinilai sebagai alternatif yang menarik, ekonomis, dan berkelanjutan untuk memperkenalkan sarana serta layanan kampus kepada calon mahasiswa maupun pengunjung lain [11].



Gambar 2.1 Contoh Tampilan *Virtual Tour* 360 derajat [24]



Gambar 2.2 Contoh Tampilan *Virtual Tour* 3D interaktif

Selain itu, *virtual tour* dapat dikembangkan dalam bentuk yang lebih interaktif melalui pendekatan berbasis permainan. Pendekatan ini memungkinkan pengguna mengendalikan pergerakan dan melakukan eksplorasi lingkungan virtual secara lebih aktif, baik melalui komputer maupun perangkat bergerak. Dalam bentuk ini, *virtual tour* tidak hanya berfungsi sebagai media visualisasi, tetapi juga sebagai media interaktif yang memberikan keleluasaan kepada pengguna untuk menjelajahi area tertentu, memahami tata letak lingkungan, dan mengakses informasi yang tersedia pada lingkungan virtual tersebut. Dengan demikian, *virtual tour* berbasis permainan dapat memberikan pengalaman kunjungan yang lebih imersif dan mendukung penyampaian informasi secara lebih menarik [12].

2.3 Roblox

Roblox merupakan sebuah platform digital berbasis *co-experience* yang memungkinkan pengguna untuk bermain, belajar, berkomunikasi, mengeksplorasi, serta

membangun hubungan sosial di dalam pengalaman virtual yang bersifat imersif. Platform ini didukung oleh komunitas pengembang global yang membuat berbagai pengalaman multiplayer menggunakan *Roblox Studio* sebagai alat pengembangan utamanya. Dalam kajian akademik, *Roblox* juga dipahami sebagai *user-generated game platform*, yaitu platform yang tidak hanya menyediakan ruang untuk memainkan konten, tetapi juga memberi kesempatan kepada pengguna untuk menciptakan, memodifikasi, dan mempublikasikan konten digitalnya sendiri. Dengan demikian, *Roblox* tidak hanya berfungsi sebagai permainan daring, melainkan juga sebagai ekosistem kreatif, sosial, dan teknologis [13]. *Roblox* memiliki beberapa karakteristik utama sebagai berikut:

1. *Roblox* sebagai platform digital berbasis lingkungan virtual interaktif. *Roblox* dapat dipahami sebagai platform digital yang menghadirkan lingkungan virtual interaktif, sehingga pengguna tidak hanya berperan sebagai pemain, tetapi juga dapat berinteraksi, berkolaborasi, dan berpartisipasi aktif di dalam ruang virtual. Dalam kajian pendidikan, *Roblox* juga dipandang sebagai bagian dari ekosistem metaverse yang mendukung pembelajaran sosial, kolaboratif, dan berbasis simulasi [14].
2. Platform sebagai *user-generated content/game* salah satu karakter utama *Roblox* adalah sifatnya sebagai platform *user-generated content* atau *user-generated game*, yaitu platform yang memungkinkan pengguna ikut menciptakan, memodifikasi, dan membagikan konten digital. Literatur terbaru menjelaskan bahwa *Roblox* merupakan contoh kuat dari platform permainan berbasis kreasi pengguna, di mana jutaan pengguna berkontribusi pada pembuatan sekaligus konsumsi konten di dalam ekosistem yang sama [15].
3. *Roblox studio* sebagai alat pengembangan utama. Dalam konteks pengembangan, *Roblox* tidak dapat dipisahkan dari *Roblox Studio* sebagai sarana utama untuk merancang, membangun, dan memublikasikan dunia *virtual* atau permainan interaktif. Kajian terbaru tentang minat penggunaan *Roblox Studio* menunjukkan bahwa platform ini dipersepsikan bermanfaat untuk mendukung kreativitas dan aktivitas pengembangan *game*, sementara kemudahan penggunaannya berpengaruh positif terhadap niat untuk terus memakainya [16].

2.3.1 Non-Player Character (NPC)

Non-Player Character (NPC) merupakan karakter dalam permainan digital yang tidak dikendalikan secara langsung oleh pemain, melainkan oleh sistem di dalam lingkungan

virtual. *NPC* digunakan untuk mensimulasikan perilaku dan rasionalitas karakter lain di dalam permainan, sehingga keberadaannya tidak hanya berfungsi sebagai unsur visual, tetapi juga sebagai bagian dari mekanisme interaksi dalam dunia game. Dengan adanya *NPC*, lingkungan virtual dapat terasa lebih hidup karena pemain tidak hanya berhadapan dengan objek statis, tetapi juga dengan karakter yang dapat menjalankan peran tertentu sesuai rancangan sistem [22].

Dalam pengembangan permainan digital, perilaku *NPC* menjadi salah satu aspek penting karena berkaitan dengan kualitas pengalaman bermain. Perilaku yang kurang responsif atau terlalu artifisial dapat menurunkan kualitas permainan dan mengurangi ketertarikan pemain. Oleh sebab itu, pengembangan *NPC* tidak hanya menekankan keberadaan karakter sebagai pelengkap permainan, tetapi juga menekankan bagaimana karakter tersebut mampu menampilkan perilaku yang sesuai dengan kondisi lingkungan *virtual*. Dengan demikian, *NPC* dapat dipahami sebagai komponen penting yang mendukung interaksi, penyampaian peran, serta pembentukan pengalaman bermain yang lebih meyakinkan [22].

Dalam pengembangan permainan digital, kualitas *NPC* tidak hanya ditentukan oleh keberadaannya sebagai karakter, tetapi juga oleh kemampuannya menampilkan perilaku yang sesuai dengan konteks permainan. Tinjauan sistematis da Silva dan Ribeiro menegaskan bahwa salah satu persoalan utama dalam pengembangan *NPC* adalah bagaimana membuat perilakunya tampak lebih meyakinkan dan lebih sesuai dengan lingkungan virtual yang ditempatinya. Hal yang serupa juga terlihat pada penelitian Siswoko dkk. tentang *serious game Warik's Adventure*, yang menempatkan *NPC* sebagai elemen penting untuk membangun interaktivitas dan pengalaman yang menyenangkan. Dalam penelitian tersebut, perilaku *NPC* diatur dengan *behaviour tree* dan dievaluasi menggunakan *Game Experience Questionnaire*, dengan hasil yang menunjukkan adanya peningkatan skor pengalaman imersif dari 3,07 menjadi 3,2. Temuan tersebut memperlihatkan bahwa rancangan perilaku *NPC* yang baik dapat berkontribusi pada peningkatan kualitas pengalaman pengguna dalam permainan [28]

Contoh penerapan *NPC* dapat ditemukan baik pada permainan hiburan maupun pada lingkungan virtual yang bersifat edukatif. Dalam permainan, *NPC* dapat berperan sebagai penjaga, penjual, pemberi misi, musuh, atau pemandu yang membantu pemain memahami tujuan dan alur permainan. Dalam konteks pendidikan dan *metaverse*, *NPC* juga dapat difungsikan sebagai agen *virtual* yang berinteraksi dengan pengguna untuk mendukung proses belajar. Penelitian Huyen menunjukkan bahwa *NPC* yang digunakan dalam

lingkungan metaverse dapat memberikan dampak positif pada beberapa aspek penting, seperti *empathy*, *problem reframing*, dan keterbukaan terhadap perspektif yang berbeda. Dengan demikian, *NPC* tidak hanya relevan dipahami sebagai karakter pelengkap dalam permainan, tetapi juga sebagai sarana interaksi *virtual* yang dapat mendukung penyampaian informasi, pembentukan pengalaman belajar, dan peningkatan keterlibatan pengguna [28].

2.3.2 Teleportation

Teleportation merupakan salah satu teknik *virtual locomotion* yang digunakan untuk memindahkan pengguna dari satu titik ke titik lain secara instan di dalam lingkungan *virtual*. Teknik ini banyak diterapkan ketika ruang *virtual* yang dijelajahi lebih luas daripada ruang fisik yang tersedia bagi pengguna. Dalam penerapannya, pengguna terlebih dahulu menentukan lokasi tujuan di lingkungan *virtual*, kemudian sistem memindahkan sudut pandang pengguna secara langsung ke titik tersebut. Mekanisme ini menjadikan *Teleportation* sebagai salah satu metode navigasi yang efisien untuk menjelajahi ruang *virtual* yang luas [23].

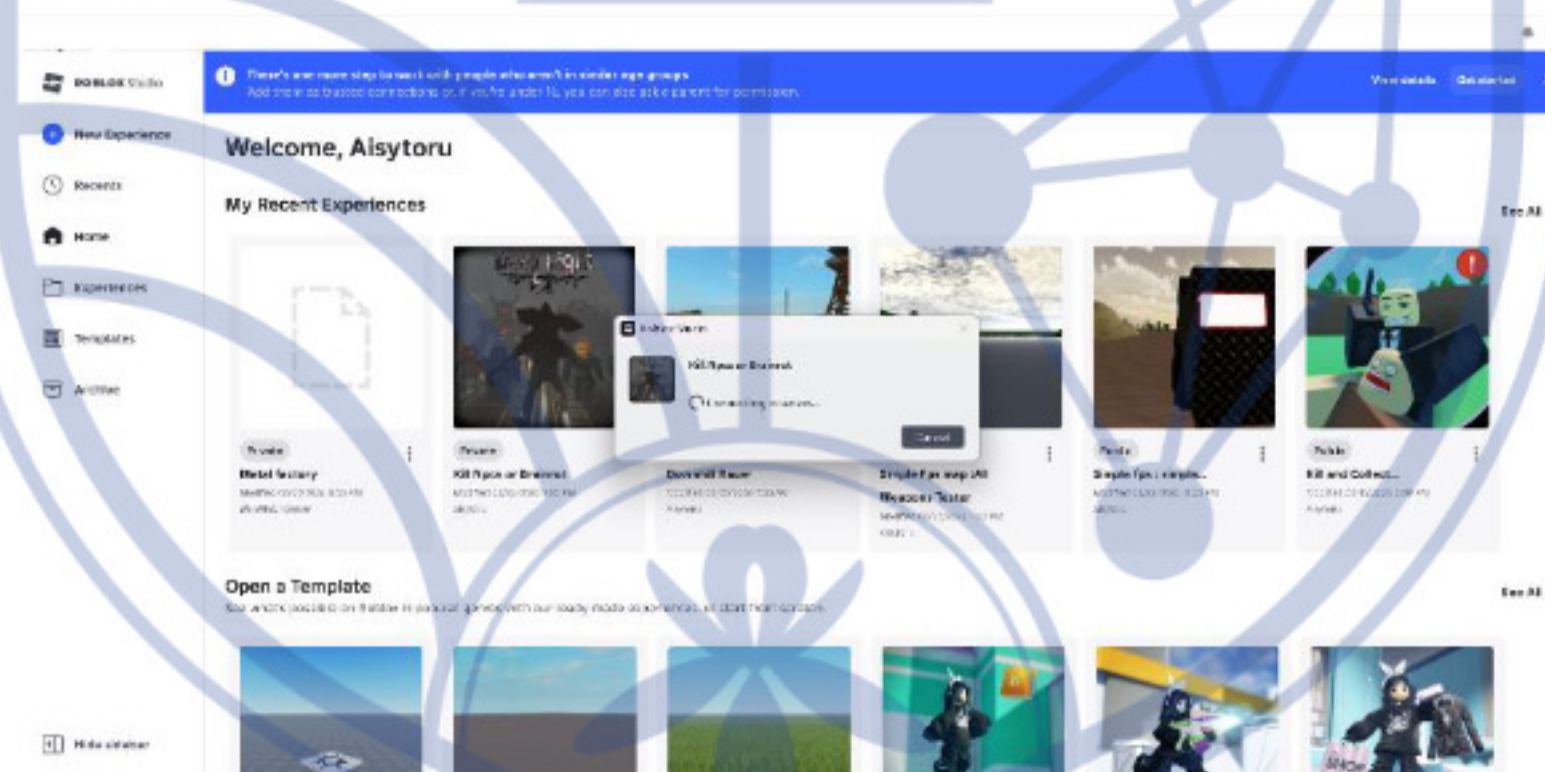
Salah satu keunggulan utama *teleportation* adalah perpindahan instan yang tidak menghasilkan *optical flow* seperti pada perpindahan kontinu. Ketiadaan *optical flow* ini penting karena ilusi gerak yang muncul dari isyarat visual tertentu dapat memicu *vection*, yang pada kondisi tertentu berkontribusi terhadap timbulnya *virtual reality sickness*. Oleh karena itu, *teleportation* banyak dipandang sebagai teknik navigasi yang relatif aman dan memiliki kemungkinan rendah dalam menimbulkan ketidaknyamanan selama penggunaan lingkungan *virtual*. Dengan karakteristik tersebut, *teleportation* relevan digunakan sebagai teknik perpindahan yang mendukung efisiensi navigasi sekaligus kenyamanan pengguna [23].

Contoh penerapan *teleportation* dalam lingkungan *virtual* dapat dilihat pada penelitian Rihs dkk. yang membandingkan teknik *teleportation* dan berjalan langsung (*walking*) dalam tugas pembelajaran deklaratif berbasis *virtual reality*. Penelitian tersebut menunjukkan bahwa hasil belajar peserta yang menggunakan *teleportation* tidak berbeda secara signifikan dibandingkan peserta yang berpindah dengan cara berjalan, baik pada pengukuran langsung maupun pada pengukuran tertunda satu hari setelah pembelajaran. Selain itu, rasa kehadiran (*sense of presence*) pada lingkungan *virtual* juga tidak menunjukkan perbedaan yang berarti antara kedua kondisi tersebut. Temuan ini menunjukkan bahwa *teleportation* dapat digunakan sebagai teknik perpindahan yang memadai dalam aplikasi pembelajaran berbasis *virtual reality*, khususnya ketika tujuan

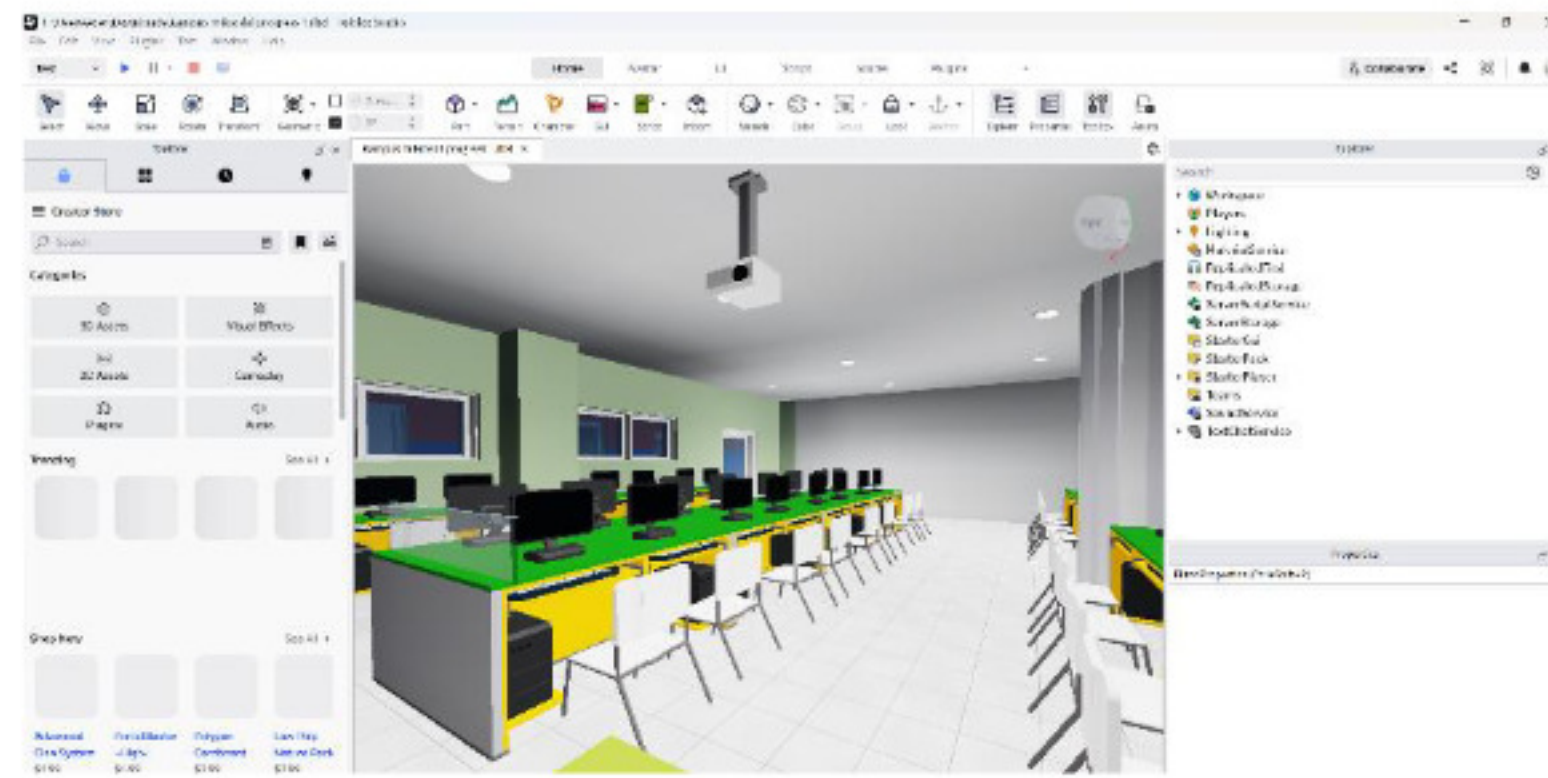
utama sistem adalah membantu pengguna menjelajahi lingkungan *virtual* dan memperoleh informasi tanpa harus bergantung pada area fisik yang luas. Dengan demikian, *teleportation* tidak hanya relevan sebagai teknik navigasi yang efisien, tetapi juga berpotensi mendukung pengalaman belajar *virtual* secara efektif [30].

2.4 Roblox Studio

Roblox Studio adalah aplikasi gratis atau perangkat lunak (*software*) yang disediakan oleh *Roblox* untuk membuat pengalaman virtual tiga dimensi (3D) dan tersedia pada sistem operasi *Windows* maupun *Mac*. *Roblox Studio* merupakan lingkungan pengembangan yang menyediakan berbagai alat untuk membangun, menyusun, menguji, dan memublikasikan pengalaman *virtual* pada platform *Roblox*. Dalam proses pengembangannya, *Roblox Studio* menggunakan bahasa pemrograman *Luau*, yaitu bahasa skrip turunan *Lua 5.1* yang dirancang untuk mendukung pembuatan perilaku interaktif di dalam lingkungan virtual [17]. Selain itu, *Roblox Studio* juga dinilai bermanfaat dalam mendukung kreativitas dan aktivitas pengembangan gim, serta kemudahan penggunaannya berpengaruh positif terhadap minat penggunaan platform tersebut. Oleh karena itu, *Roblox Studio* relevan digunakan dalam penelitian ini karena mampu mendukung perancangan kampus *virtual* yang interaktif, imersif, dan dapat dipublikasikan pada platform *Roblox*.



Gambar 2.3 Tampilan *Roblox Studio*



Gambar 2.4 Tampilan *Roblox Studio* Dalam Pengembangan Game

Dalam kajian pendidikan, *Roblox Studio* juga dipandang memiliki potensi yang mendukung aktivitas perancangan lingkungan *virtual* interaktif. Zhai menjelaskan bahwa *Roblox* menyediakan lingkungan yang dapat diprogram, sehingga pengguna dapat membangun permainan maupun dunia interaktif sesuai kebutuhan. Pada artikel yang sama juga dijelaskan bahwa *Roblox Studio* dipandang sebagai program yang sederhana dan intuitif, serta telah digunakan dalam kegiatan perancangan lingkungan virtual tiga dimensi yang memungkinkan pengguna menavigasi dunia interaktif. Temuan tersebut menunjukkan bahwa *Roblox Studio* tidak hanya berfungsi sebagai perangkat lunak pengembangan game, tetapi juga sebagai sarana yang mendukung visualisasi ruang, kreativitas, dan interaksi digital [26].

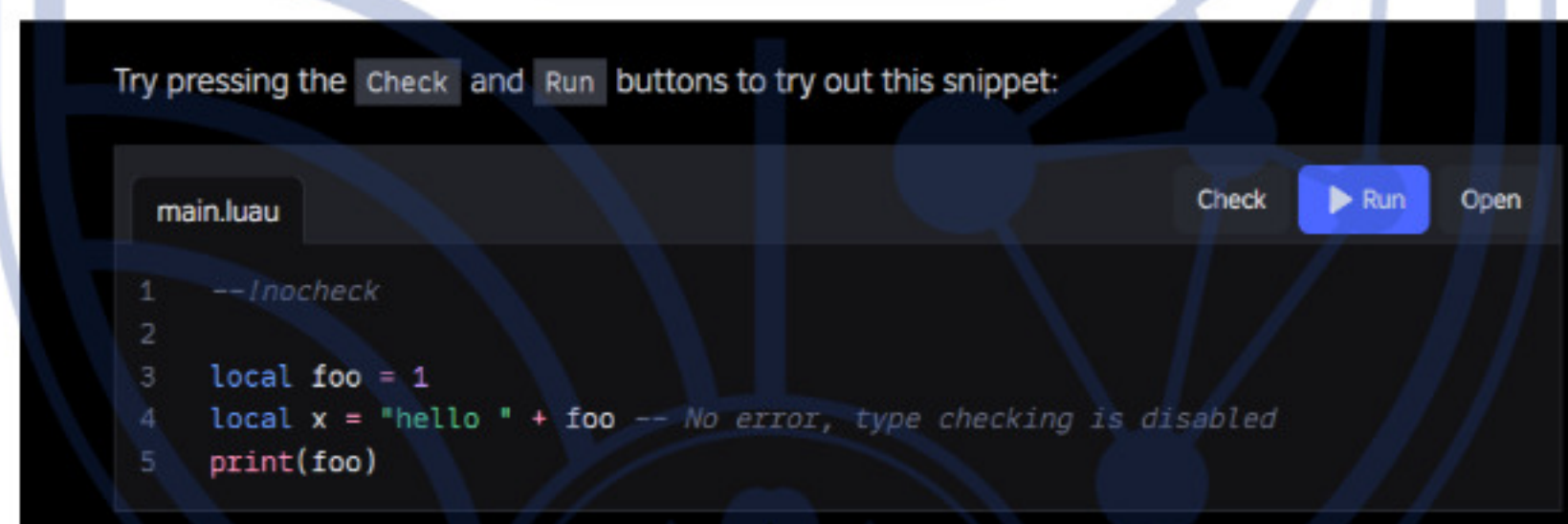
Roblox Studio merupakan lingkungan pengembangan yang dapat dimanfaatkan untuk merancang, membangun, dan menguji media digital interaktif berbasis permainan. Dalam konteks pendidikan, pemanfaatan *Roblox Studio* telah menunjukkan potensi yang cukup luas karena platform ini memungkinkan pengembang menghadirkan simulasi, interaksi, dan evaluasi dalam satu lingkungan *virtual* yang terintegrasi. Penelitian Mindarta menunjukkan bahwa penggunaan *Roblox Studio* pada pengembangan alat evaluasi pembelajaran di bidang otomotif mampu menjadi alternatif evaluasi yang efektif serta menjawab keterbatasan pendekatan evaluasi konvensional. Temuan tersebut memperlihatkan bahwa *Roblox Studio* tidak hanya relevan sebagai alat pembuatan gim, tetapi juga dapat berfungsi sebagai sarana pengembangan media interaktif yang mendukung aktivitas belajar dan evaluasi secara lebih kontekstual. Dengan karakteristik tersebut, *Roblox Studio* dapat dipahami sebagai perangkat pengembangan yang mendukung integrasi unsur visual, logika program, dan interaksi pengguna dalam satu lingkungan kerja digital [27].

2.5 Bahasa pemrograman Luau

Luau adalah bahasa pemrograman skrip yang digunakan pada *Roblox Studio* untuk membangun perilaku dan logika dalam pengalaman *virtual*. Secara teknis, *Luau* dikembangkan dari *Lua* 5.1, tetapi dirancang dengan karakteristik yang lebih sesuai untuk kebutuhan pengembangan pada platform Roblox, yaitu cepat, ringan, aman, dan mendukung sistem tipe bertahap (*gradual typing*). *Luau* tidak hanya berfungsi sebagai bahasa skrip biasa, tetapi juga sebagai sarana untuk mendukung pengembangan pengalaman interaktif secara lebih terstruktur pada *Roblox Studio* [17].



Gambar 2.5 Tampilan Penulisan Skrip *Luau* Pada *Roblox Studio*



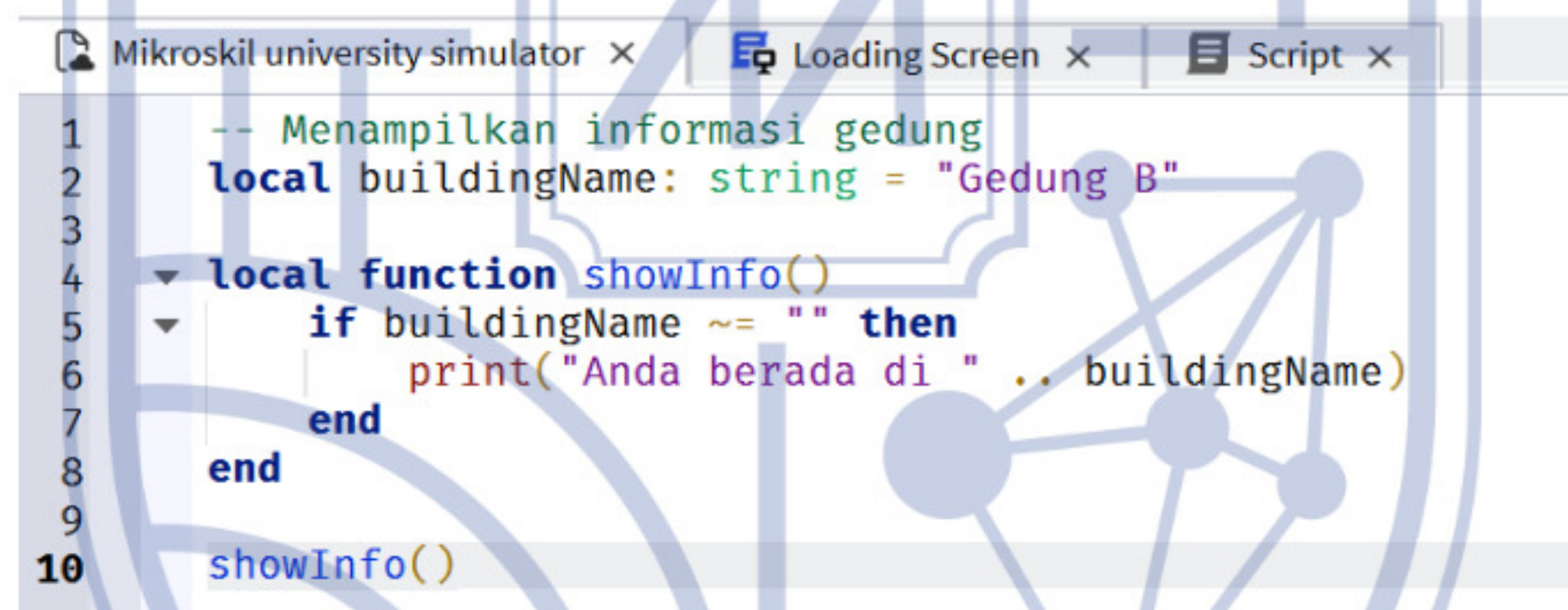
Gambar 2.6 Contoh Tampilan Dasar Penulisan Skrip *Luau* [18]

Salah satu karakter penting *Luau* adalah dukungannya terhadap *gradual typing*. Dalam sistem ini, pengembang dapat tetap menulis kode secara fleksibel seperti pada bahasa dinamis, tetapi juga dapat menambahkan anotasi tipe dan memanfaatkan *type inference* untuk membantu pemeriksaan kesalahan. Dokumentasi resmi Luau menjelaskan bahwa sistem tipe tersebut digunakan untuk memberikan peringatan, kesalahan, dan saran selama proses penulisan kode, sehingga kesalahan dapat dideteksi lebih awal sebelum program dijalankan. Selain itu, Luau menyediakan beberapa *mode* pemeriksaan tipe, seperti *nonstrict*

dan *strict*, yang membantu pengembang menyesuaikan tingkat keketatan analisis kode sesuai kebutuhan [18].

Secara umum, struktur dasar kode program *Luau* memiliki kesamaan dengan Lua 5.1 karena *Luau* dikembangkan dari bahasa tersebut. Dalam penulisannya, kode *Luau* disusun dari beberapa unsur dasar, seperti komentar, deklarasi variabel, fungsi, struktur kontrol, tabel, serta anotasi tipe. Dokumentasi resmi *Luau* menjelaskan bahwa bahasa ini tetap menggunakan sintaks dasar Lua 5.1, namun menambahkan beberapa ekstensi sintaks yang mendukung pengembangan program secara lebih terstruktur [17][18].

Struktur dasar kode program *Luau* pada *Roblox Studio* umumnya disusun dari beberapa komponen utama, yaitu komentar, deklarasi variabel, fungsi, struktur kontrol, tabel, dan anotasi tipe. Pemahaman terhadap struktur tersebut penting agar pembaca dapat memahami bagaimana skrip ditulis dan dijalankan dalam pengembangan sistem pada *Roblox Studio*. Oleh karena itu, contoh struktur dasar kode program *Luau* ditampilkan beserta penjelasan tiap bagiannya.



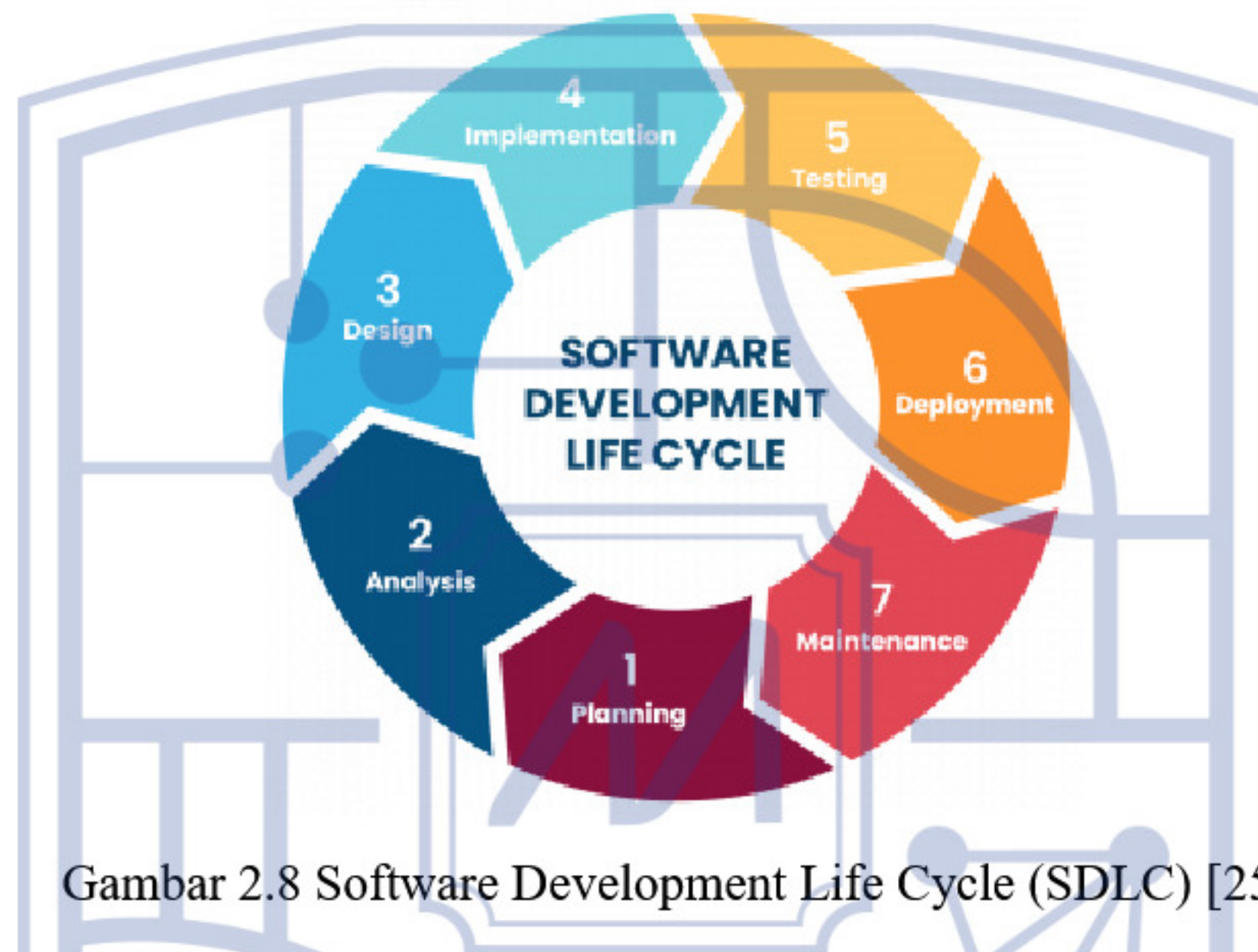
```
1  -- Menampilkan informasi gedung
2  local buildingName: string = "Gedung B"
3
4  local function showInfo()
5      if buildingName ~= "" then
6          print("Anda berada di " .. buildingName)
7      end
8  end
9
10 showInfo()
```

Gambar 2.7 Contoh skrip *luau*

Berdasarkan contoh skrip pada Gambar 2.7 tersebut, baris pertama merupakan komentar yang digunakan untuk memberikan keterangan mengenai fungsi skrip. Baris kedua menunjukkan deklarasi variabel *local buildingName* dengan anotasi tipe *string*, yang menyatakan bahwa variabel tersebut menyimpan data teks. Baris berikutnya merupakan deklarasi fungsi *showInfo()*, yaitu sekumpulan instruksi yang dapat dipanggil ketika dibutuhkan. Di dalam fungsi tersebut terdapat struktur kontrol *if*, yang digunakan untuk memeriksa apakah variabel *buildingName* memiliki isi. Jika kondisi terpenuhi, sistem menjalankan perintah *print()* untuk menampilkan informasi ke output. Adapun baris terakhir merupakan pemanggilan fungsi *showInfo()* agar instruksi di dalam fungsi dapat dijalankan.

2.6 Metode Software Development Life Cycle (SDLC)

Software Development Life Cycle (SDLC) merupakan proses pengembangan perangkat lunak yang dilakukan secara terstruktur melalui tahapan-tahapan yang saling berkaitan. SDLC digunakan untuk membantu pengembang dalam membangun, menguji, menerapkan, dan memelihara perangkat lunak agar sesuai dengan kebutuhan pengguna. Dengan adanya tahapan yang jelas, proses pengembangan sistem dapat dilakukan secara lebih terarah, sistematis, dan mudah dievaluasi [19].



Gambar 2.8 Software Development Life Cycle (SDLC) [25]

Berdasarkan Gambar 2.8, tahapan utama dalam *Software Development Life Cycle* (SDLC) terdiri atas *planning*, *analysis*, *design*, *implementation*, *testing*, *deployment*, dan *maintenance* [25]. Penjelasan masing-masing tahapan adalah sebagai berikut:

1. *Planning* / Perencanaan

Pada tahapan ini, pengembang melakukan perencanaan awal terhadap sistem yang akan dibangun. Kegiatan yang dilakukan meliputi penentuan tujuan, identifikasi permasalahan, penetapan ruang lingkup, serta penentuan kebutuhan awal proyek. Tahap ini penting karena menjadi dasar bagi keseluruhan proses pengembangan. Dalam pengembangan game maupun kampus *virtual*, tahap perencanaan membantu menentukan konsep utama, sasaran pengguna, serta fitur-fitur yang akan dikembangkan [19].

2. *Analysis* / Analisis Kebutuhan

Pada tahap analisis, pengembang menelaah kebutuhan sistem secara lebih rinci. Proses ini mencakup identifikasi kebutuhan fungsional dan nonfungsional, analisis karakteristik pengguna, serta penentuan spesifikasi sistem yang akan dibangun. Tahap

analisis berfungsi untuk memastikan bahwa perangkat lunak yang dikembangkan benar-benar sesuai dengan kebutuhan pengguna. Dalam pengembangan *game* dan kampus *virtual*, analisis dapat meliputi kebutuhan navigasi, interaksi pengguna, tampilan antarmuka, penyajian informasi, serta performa sistem.

3. *Design* / Perancangan

Pada tahap perancangan, pengembang menyusun rancangan sistem berdasarkan hasil analisis kebutuhan. Perancangan ini dapat meliputi rancangan alur permainan, struktur antarmuka, desain lingkungan *virtual*, pembagian area permainan, serta rancangan interaksi antarobjek di dalam *game*. Untuk pengembangan *game* di *Roblox Studio*, tahap desain juga mencakup penyusunan konsep dunia *virtual*, peletakan objek, serta perencanaan hubungan antara elemen visual dan logika permainan. Dengan adanya tahap desain, proses implementasi dapat dilakukan secara lebih terarah karena pengembang telah memiliki gambaran yang jelas mengenai bentuk sistem yang akan diwujudkan.

4. *Coding* / Implementasi

Tahap implementasi merupakan tahap penerapan rancangan ke dalam bentuk sistem yang dapat dijalankan. Pada tahap ini, pengembang mulai membangun lingkungan permainan di *Roblox Studio*, menulis skrip untuk logika permainan, serta menghubungkan elemen-elemen visual dengan perilaku sistem. *Roblox Studio* secara resmi menyediakan alat untuk *build*, *script*, *test*, dan *publish*, sehingga tahap implementasi dalam pengembangan *game* dapat dilakukan langsung pada lingkungan pengembangan yang sama. Tahap ini menghasilkan prototipe atau versi awal permainan yang sudah dapat diuji fungsinya.

5. *Testing* / Pengujian

Tahap pengujian dilakukan untuk meninjau hasil implementasi dan menemukan kesalahan yang masih terdapat dalam sistem. *IBM* menjelaskan bahwa pengujian bertujuan untuk meninjau kode dan menghilangkan *bug*, sedangkan *Roblox* menyediakan *playtest* dan panduan *test and save* untuk memverifikasi bahwa kode atau fitur berjalan dengan baik sebelum dipublikasikan. Dalam pengembangan *game* di *Roblox Studio*, tahap ini dapat mencakup pengujian fungsi tombol, interaksi objek, alur permainan, respons antarmuka, serta kestabilan sistem. Dengan pengujian yang memadai, kualitas permainan dapat ditingkatkan sebelum dirilis kepada pengguna.

6. *Deployment* / Penerapan

Tahap penerapan merupakan tahap ketika perangkat lunak yang telah diuji dipublikasikan atau digunakan oleh pengguna. Pada tahap ini, sistem dipindahkan ke lingkungan operasional agar dapat diakses sesuai tujuan pengembangannya. Dalam konteks *game* dan kampus *virtual*, tahap ini berarti sistem mulai digunakan oleh target pengguna, baik untuk tujuan hiburan, simulasi, maupun penyampaian informasi.

7. *Maintenance* / Pemeliharaan

Tahap pemeliharaan dilakukan setelah sistem diterapkan. Kegiatan pada tahap ini meliputi perbaikan kesalahan, penyesuaian fungsi, optimasi performa, dan pengembangan lanjutan sesuai kebutuhan baru. Tahap ini penting karena perangkat lunak tidak berhenti berkembang setelah dirilis, melainkan perlu terus diperbaiki agar tetap relevan dan berjalan dengan baik. Dalam pengembangan *game* dan kampus *virtual*, pemeliharaan diperlukan untuk memperbaiki *bug*, menyesuaikan sistem dengan kebutuhan pengguna, dan menambah fitur baru bila diperlukan.

2.7 Black Box testing

Black Box testing merupakan salah satu metode pengujian perangkat lunak yang dilakukan berdasarkan fungsionalitas program atau sistem. Dalam metode ini, pengujian dilakukan dengan memperhatikan masukan yang diberikan dan keluaran yang dihasilkan, tanpa perlu mengetahui bagaimana struktur internal program bekerja. Dengan demikian, *black-box testing* berfokus pada kesesuaian fungsi sistem terhadap spesifikasi yang telah ditetapkan [20].

Dalam penerapannya, *black-box testing* dilakukan melalui beberapa tahapan utama, yaitu *test analysis*, *test design*, *test implementation*, dan *test execution*. Pada tahap *test analysis*, penguji mengidentifikasi fitur, fungsi, atau kebutuhan sistem yang perlu diuji berdasarkan spesifikasi yang telah ditetapkan. Selanjutnya, pada tahap *test design*, penguji menyusun kondisi pengujian, menentukan data masukan, serta menetapkan hasil yang diharapkan. Tahap *test implementation* dilakukan dengan menyiapkan *test case*, data uji, dan prosedur pengujian yang akan digunakan. Setelah itu, pada tahap *test execution*, penguji menjalankan skenario pengujian, mengamati keluaran sistem, dan membandingkannya dengan hasil yang diharapkan. Melalui tahapan tersebut, *black-box testing* dapat digunakan untuk menilai kesesuaian fungsi sistem tanpa meninjau struktur internal program [21].

Pada *black-box testing*, program dipandang sebagai sebuah kotak hitam, sehingga perhatian utama penguji terletak pada respons sistem terhadap skenario pengujian yang

diberikan. Pendekatan ini digunakan untuk memastikan bahwa fitur atau fungsi yang tersedia dapat berjalan sesuai dengan kebutuhan pengguna. Oleh karena itu, *black box testing* sesuai digunakan untuk menilai apakah suatu sistem telah berfungsi dengan baik tanpa meninjau logika program di dalamnya [20]. Beberapa karakteristik *black-box testing* meliputi:

1. Berorientasi pada fungsi sistem, yaitu pengujian dilakukan terhadap fitur atau fungsi yang tersedia pada perangkat lunak sesuai dengan spesifikasi.
2. Menggunakan data masukan dan keluaran yang diamati, yaitu penguji menilai hasil sistem berdasarkan respons yang muncul setelah masukan diberikan,
3. Tidak memerhatikan struktur internal program, yaitu penguji tidak perlu mengetahui logika atau kode program untuk melaksanakan pengujian.
4. Berfokus pada kesesuaian sistem terhadap spesifikasi, yaitu pengujian diarahkan untuk memastikan bahwa sistem bekerja sesuai kebutuhan yang telah ditentukan.

Berdasarkan uraian tersebut, *black-box testing* merupakan metode pengujian yang berorientasi pada fungsi sistem, sehingga dapat digunakan untuk memastikan bahwa setiap fitur yang dikembangkan telah berjalan sesuai dengan kebutuhan dan spesifikasi yang telah ditentukan. Dengan pendekatan ini, pengujian dapat dilakukan secara sistematis melalui skenario masukan dan keluaran, sehingga hasil evaluasi sistem menjadi lebih terarah dan mudah dianalisis.

2.8 Gambaran Umum Objek Penelitian

Universitas Mikroskil merupakan objek penelitian yang digunakan dalam pengembangan *game* kampus virtual berbasis *Roblox Studio*. Lingkungan kampus yang menjadi fokus pengembangan meliputi beberapa area utama, yaitu Gedung A, Gedung B, dan Gedung C. Pemilihan area tersebut dilakukan karena ketiga gedung tersebut memiliki fungsi penting dalam aktivitas akademik, seperti ruang kelas, laboratorium komputer, perpustakaan, dan fasilitas pendukung lainnya.

Gambaran umum objek penelitian diperlukan untuk memperjelas kondisi nyata lingkungan kampus yang akan direpresentasikan ke dalam bentuk lingkungan virtual tiga dimensi (3D). Informasi mengenai letak gedung, ruangan, dan fasilitas digunakan sebagai acuan dalam proses perancangan model 3D, penempatan objek, serta penyusunan fitur interaktif di dalam *game*. Dengan adanya gambaran objek penelitian, hasil pengembangan dapat dibandingkan dengan kondisi nyata lingkungan kampus sehingga visualisasi yang dibuat lebih sesuai dengan objek aslinya.