

BAB 2

LANDASAN TEORI

2.1 Unity 3D

Unity 3D merupakan salah satu *game engine* yang banyak digunakan dalam pengembangan aplikasi interaktif. Unity menyediakan berbagai fitur yang mendukung pembuatan lingkungan tiga dimensi, seperti sistem *rendering* grafis, fisika, animasi, serta dukungan terhadap berbagai *platform* perangkat keras [29].

2.2 Gamifikasi

Gamifikasi merupakan pendekatan yang menerapkan elemen-elemen permainan (*game elements*) ke dalam konteks non-permainan dengan tujuan meningkatkan keterlibatan dan motivasi pengguna [21]. Elemen tersebut dapat berupa poin, level, tantangan, penghargaan, maupun sistem umpan balik yang mendorong pengguna untuk terus berpartisipasi [10]. Elemen-elemen gamifikasi tersebut secara umum dapat dikelompokkan ke dalam tiga lapisan, yaitu *game mechanics* (mekanisme aturan dan sistem, seperti skor dan validasi aksi), *game dynamics* (interaksi yang muncul akibat mekanisme tersebut, seperti siklus umpan balik), serta *game aesthetics* (respons emosional pengguna, seperti rasa capaian dan kepuasan yang terkontrol) [20].

Penerapan gamifikasi perlu dirancang dengan baik dan matang untuk mencapai dampak positif yang diinginkan sehingga penting untuk memilih *framework* rancangan yang tepat. Terdapat lebih dari 40 *framework* rancangan gamifikasi yang tercatat dalam studi literatur, sebagian besar bersifat konseptual dan belum diuji secara langsung pada skenario nyata [23]. Salah satu *framework* yang paling populer dan paling banyak diacu dalam penelitian adalah *6D Framework* yang dicetuskan oleh Werbach dan Hunter [23, 18], sebagaimana dijabarkan lebih lanjut pada Subbab 2.4.

Dalam penelitian ini, pendekatan gamifikasi diterapkan melalui beberapa elemen konkret pada aplikasi simulasi evakuasi gempa bumi, yaitu: (1) sistem skor yang bertambah atau berkurang berdasarkan ketepatan tindakan pengguna selama simulasi; (2) mekanisme validasi dan umpan balik (*feedback*) berupa peringatan (*warning*) ketika prosedur keselamatan tidak diikuti dengan benar; (3) *minigame* pada tahap *Hold* yang menuntut pengguna mempertahankan posisi aman

selama guncangan berlangsung; serta (4) siklus aksi-validasi-umpan balik (*activity loop*) yang berulang pada setiap tahapan evakuasi untuk mendorong pengguna memperbaiki tindakannya hingga berhasil. Keempat elemen ini dirancang secara terstruktur melalui tahapan *6D Framework* yang dijelaskan pada Subbab 2.4, dan diimplementasikan lebih lanjut pada Bab 3 dan Bab 4.

2.3 Simulasi Evakuasi Bencana

Simulasi evakuasi bencana merupakan metode pelatihan yang digunakan untuk mempersiapkan individu dalam menghadapi situasi darurat dengan mensimulasikan kondisi bencana secara terkontrol [11, 17]. Tujuan utama dari simulasi ini adalah untuk meningkatkan kesiapsiagaan, pemahaman terhadap prosedur keselamatan, serta kemampuan pengambilan keputusan dalam kondisi darurat.

Dalam konteks gempa bumi, simulasi evakuasi difokuskan pada respons awal saat terjadi guncangan, seperti mencari perlindungan di bawah meja (*drop, cover, and hold on*), serta langkah-langkah evakuasi setelah guncangan berhenti. Simulasi juga dapat mencakup kondisi lingkungan yang realistis, seperti runtuhnya bangunan, jalur evakuasi yang terhalang, serta dinamika pergerakan individu dalam situasi darurat [13, 24]. Selain itu, studi terkait perilaku manusia dalam kondisi ekstrem juga menunjukkan bahwa faktor lingkungan sangat memengaruhi kemampuan individu dalam melakukan evakuasi [31], sehingga simulasi yang realistis menjadi penting dalam proses pelatihan.

2.4 6D Framework

6D Framework merupakan suatu *framework* rancangan gamifikasi yang dicetuskan oleh Werbach dan Hunter, digunakan dalam perancangan pengalaman pembelajaran berbasis gamifikasi [22, 18]. *6D Framework* terdiri dari enam tahap berikut [18, 22]. Pertama, *Define business objectives*, yaitu tahap penentuan tujuan bisnis atau pembelajaran yang ingin dicapai, termasuk identifikasi target pengguna dan kebutuhan utama dari sistem yang akan dikembangkan. Kedua, *Delineate target behavior*, yaitu tahap pengidentifikasian perilaku spesifik yang diharapkan dari pengguna sebagai hasil dari penggunaan sistem, sehingga perancangan dapat diarahkan pada pencapaian perilaku tersebut. Ketiga, *Describe the players*, yaitu tahap penyusunan profil dan karakteristik pengguna yang menjadi

sasaran sistem. Keempat, *Devise activity loop*, yaitu tahap perancangan siklus aktivitas yang memotivasi pengguna melalui siklus keterlibatan dan kemajuan, cara menarik pengguna baru, serta cara mendorong tindakan lebih lanjut dari pengguna. Kelima, *Don't forget the fun*, yaitu tahap peninjauan kembali rancangan untuk mengidentifikasi dan memastikan aspek yang memotivasi pengguna secara intrinsik tetap terjaga. Keenam, *Deploy appropriate tools*, yaitu tahap penentuan mekanisme, metrik, dan perangkat yang sesuai, termasuk elemen gim yang relevan, untuk diimplementasikan kepada pengguna serta dievaluasi.

Pemilihan *6D Framework* dalam penelitian ini didasarkan pada perbandingannya dengan metodologi pengembangan lain yang umum digunakan, seperti *System Development Life Cycle (SDLC)*, *Design Thinking*, *Game Development Life Cycle (GDLC)*, maupun *Mechanic, Dynamic, Aesthetic (MDA) Framework* yang banyak digunakan pada penelitian media edukasi mitigasi bencana di Indonesia [19]. *SDLC* dan *GDLC* secara umum berfokus pada tahapan teknis pembangunan perangkat lunak (analisis, desain, implementasi, dan pengujian), sedangkan *Design Thinking* berfokus pada eksplorasi kebutuhan dan empati terhadap pengguna secara umum. Ketiga pendekatan tersebut tidak menyediakan tahapan eksplisit untuk merancang elemen motivasi, perilaku target (*target behavior*), maupun siklus keterlibatan (*activity loop*) yang menjadi inti dari pendekatan gamifikasi. Sementara itu, *MDA Framework* berfokus pada analisis mekanik, dinamika, dan estetika permainan, namun tidak secara eksplisit mengaitkan setiap tahapannya dengan tujuan pembelajaran maupun karakteristik pengguna sejak awal perancangan. *6D Framework* dipilih karena keenam tahapannya secara eksplisit menempatkan penyelarasan antara tujuan edukasi, perilaku target pengguna, dan elemen gamifikasi sejak tahap awal (*Define business objectives*) hingga tahap evaluasi (*Deploy appropriate tools*), sehingga lebih sesuai dengan tujuan penelitian ini yang tidak hanya membangun aplikasi secara fungsional tetapi juga memastikan aplikasi tersebut terarah sebagai media pembelajaran.

Perlu dicatat bahwa *6D Framework* pada dasarnya merupakan *framework* yang bersifat generik dan tidak dirancang khusus untuk domain atau skenario tertentu, sehingga fleksibel untuk diadaptasi pada berbagai konteks [23]. Sebuah studi kuasi-eksperimental yang membandingkan *6D Framework* dengan *framework* gamifikasi yang dirancang khusus untuk konteks edukasi menemukan bahwa sifat generik *6D Framework* memberikan keleluasaan dan kemudahan dalam perancangan, tetapi berisiko membuat elemen gamifikasi yang dirancang menjadi

kurang terhubung secara langsung dengan tujuan pedagogis apabila tidak dikelola dengan cermat [23]. Oleh karena itu, dalam penelitian ini, penerapan *6D Framework* tidak berdiri sendiri, melainkan dilengkapi dengan landasan teori pembelajaran berbasis permainan (Subbab 2.5) sejak tahap perancangan, sebagai mekanisme tambahan untuk memastikan elemen gamifikasi yang dirancang tetap terarah pada tujuan edukasi mitigasi bencana, bukan hanya pada aspek keterlibatan dan motivasi semata.

Penerapan *6D Framework* dalam pengembangan simulasi memungkinkan perancangan sistem yang lebih terstruktur dan berorientasi pada pengalaman pengguna, sehingga aplikasi yang dikembangkan dapat memberikan pengalaman belajar yang menarik dan sesuai dengan tujuan pelatihan. Penerapan keenam tahap tersebut secara spesifik pada aplikasi simulasi evakuasi gempa bumi dijabarkan pada Subbab 3.3.

Evaluasi pada tahap *Deploy appropriate tools* dalam penelitian ini dilakukan melalui dua jenis pengujian yang saling melengkapi, yaitu *Unit Testing* dan *User Acceptance Testing* (UAT). *Unit Testing* digunakan terlebih dahulu untuk memastikan bahwa setiap unit fungsi program telah berjalan sesuai dengan rancangan teknis yang disusun pada tahap *Devise activity loop* dan *Don't forget the fun*, sedangkan UAT digunakan untuk mengevaluasi apakah hasil implementasi tersebut telah memenuhi perilaku target pengguna yang telah ditetapkan sejak tahap *Define business objectives* dan *Delineate target behavior*. Dengan demikian, kedua pengujian tersebut bukan merupakan tahapan yang berdiri sendiri, melainkan instrumen evaluasi yang secara langsung diturunkan dari tujuan tahap *Deploy appropriate tools* pada *6D Framework*, sesuai dengan alat dan mekanisme evaluasi yang ditentukan pada tahap tersebut. Penjelasan lebih lanjut mengenai *Unit Testing* dan UAT dijabarkan pada Subbab 2.10 dan Subbab 2.7.

2.5 Pembelajaran Berbasis Permainan (Game-Based Learning)

Game-Based Learning (GBL) merupakan pendekatan pembelajaran yang memanfaatkan elemen dan mekanisme permainan untuk mencapai tujuan pembelajaran tertentu, dengan menempatkan pengguna pada pengalaman yang interaktif, adaptif, dan berbasis simulasi situasi nyata [9]. Pendekatan ini berbeda dengan gamifikasi yang hanya menambahkan elemen permainan ke dalam konteks non-permainan; pada GBL, keseluruhan aktivitas belajar dikemas dalam bentuk permainan itu sendiri, sehingga pemahaman dan keterampilan pengguna dibentuk

melalui pengalaman langsung (*experiential learning*) selama berinteraksi dengan simulasi [9]. Dalam konteks penelitian ini, GBL menjadi landasan bahwa aplikasi simulasi evakuasi gempa bumi tidak hanya berfungsi sebagai media hiburan interaktif, tetapi juga sebagai sarana untuk meningkatkan pemahaman pengguna terhadap prosedur keselamatan dan evakuasi gempa bumi.

Pada penelitian ini, pengukuran sisi edukasi dilakukan melalui dua pendekatan yang saling melengkapi: (1) desain *pre-test* dan *post-test* menggunakan instrumen *Knowledge Testing* yang dijabarkan pada Subbab 2.6, untuk mengukur besaran peningkatan pemahaman (*knowledge gain*) secara kuantitatif; serta (2) butir pernyataan persepsi mengenai kesiapsiagaan yang disisipkan ke dalam kuesioner UAT, untuk memperoleh indikasi persepsi pengguna terhadap materi setelah menggunakan aplikasi.

2.6 Knowledge Testing

Instrumen pengetahuan (*Knowledge Testing*) disusun berdasarkan konten perilaku saat gempa bumi dan evakuasi pasca-gempa di dalam bangunan, yang diadaptasi dari penelitian Feng et al. [3]. Setiap butir soal diklasifikasikan menurut level kognitif Taksonomi Bloom Revisi yang dikembangkan oleh Anderson dan Krathwohl [32], yang terdiri atas enam tingkatan, yaitu mengingat, memahami, menerapkan, menganalisis, mengevaluasi, dan mencipta. Klasifikasi tersebut bertujuan untuk memastikan bahwa instrumen mampu mengukur pemahaman responden secara bertingkat, mulai dari pengetahuan dasar hingga kemampuan evaluasi dan sintesis terhadap prosedur evakuasi. Daftar lengkap pertanyaan beserta klasifikasinya disajikan pada Tabel 3.3 di Bab 3.

Instrumen ini digunakan dalam desain *pretest-posttest*, yaitu diberikan kepada responden sebelum dan sesudah intervensi (edukasi/pelatihan kesiapsiagaan gempa). Desain ini memungkinkan penghitungan *normalized gain* sebagaimana diusulkan oleh Hake [15], untuk mengukur efektivitas peningkatan pengetahuan responden setelah intervensi. Adapun jumlah sampel minimal penelitian mengacu pada pedoman ukuran sampel yang direkomendasikan oleh Roscoe [33].

Efektivitas peningkatan pengetahuan responden dihitung menggunakan rumus *normalized gain* $\langle g \rangle$ yang dikembangkan oleh Hake [15], sebagaimana dirumuskan pada Persamaan (2.1):

$$\langle g \rangle = \frac{S_{post} - S_{pre}}{S_{maks} - S_{pre}} \quad (2.1)$$

dengan S_{post} adalah skor rata-rata posttest, S_{pre} adalah skor rata-rata pretest, dan S_{maks} adalah skor maksimum yang mungkin dicapai pada instrumen tersebut. Nilai $\langle g \rangle$ yang dihasilkan kemudian diinterpretasikan ke dalam tiga kategori menurut Hake [15], yaitu *gain* tinggi ($\langle g \rangle \geq 0,7$), *gain* sedang ($0,3 \leq \langle g \rangle < 0,7$), dan *gain* rendah ($\langle g \rangle < 0,3$).

2.7 User Acceptance Test (UAT)

User Acceptance Testing (UAT) merupakan metode pengujian yang dilakukan untuk mengetahui apakah sistem yang dikembangkan telah sesuai dengan kebutuhan dan harapan pengguna [34]. Pengujian ini dilakukan secara langsung oleh pengguna akhir (*end user*) dengan mencoba fitur-fitur yang tersedia pada aplikasi serta memberikan penilaian berdasarkan pengalaman mereka saat menggunakan sistem. UAT menjadi salah satu tahap penting dalam pengembangan perangkat lunak karena digunakan untuk memastikan bahwa sistem dapat berfungsi dengan baik dan memenuhi tujuan yang telah dirancang.

Dalam penelitian ini, UAT digunakan untuk mengevaluasi aplikasi simulasi evakuasi gempa bumi yang telah dikembangkan [35]. Pengujian dilakukan dengan melibatkan pengguna untuk mencoba berbagai skenario yang tersedia dalam aplikasi, seperti proses penyelamatan diri saat gempa terjadi, pencarian jalur evakuasi, dan interaksi dengan lingkungan simulasi. Melalui pengujian ini, peneliti dapat mengetahui tingkat penerimaan pengguna terhadap aplikasi dari aspek fungsionalitas, kemudahan penggunaan, interaktivitas, serta pengalaman pengguna selama menjalankan simulasi. Hasil pengujian diharapkan dapat menjadi dasar evaluasi dalam melakukan perbaikan dan pengembangan aplikasi agar lebih efektif sebagai media pembelajaran dan pelatihan kesiapsiagaan bencana gempa bumi.

Persentase tingkat penerimaan pengguna dihitung menggunakan Persamaan 2.2 [36].

$$UAT (\%) = \frac{\sum \text{Skor Responden}}{\text{Jumlah Responden} \times \text{Jumlah Pertanyaan} \times \text{Skor Maksimum}} \times 100\% \quad (2.2)$$

dengan:

1. Σ Skor Responden adalah jumlah seluruh skor jawaban responden setelah dilakukan *reverse scoring* pada pertanyaan negatif.
2. Jumlah Responden adalah banyaknya responden yang mengikuti pengujian.
3. Jumlah Pertanyaan adalah jumlah pertanyaan pada kuesioner UAT.
4. Skor Maksimum adalah skor tertinggi pada skala Likert, yaitu 5.

Hasil persentase UAT kemudian diinterpretasikan berdasarkan kategori tingkat penerimaan sistem yang ditunjukkan menggunakan Skala Likert.

Terkait jumlah responden yang dilibatkan dalam pengujian, ukuran sampel dikatakan layak apabila berada pada rentang 30 hingga 500 responden untuk sebagian besar jenis penelitian [36]. Oleh karena itu, jumlah responden pada penelitian ini yang berjumlah lebih dari 30 orang telah memenuhi syarat kelayakan minimum tersebut sehingga hasil pengujian UAT dapat dipertanggungjawabkan secara statistik.

2.8 Skala Likert

Skala Likert merupakan skala pengukuran yang digunakan untuk menilai sikap atau persepsi responden terhadap suatu pernyataan [37, 36]. Dalam penelitian ini, penilaian pada kuesioner UAT menggunakan skala Likert lima poin, yaitu nilai 1 untuk *Sangat Tidak Setuju*, nilai 2 untuk *Tidak Setuju*, nilai 3 untuk *Netral*, nilai 4 untuk *Setuju*, dan nilai 5 untuk *Sangat Setuju*.

Tabel 2.1. Kategori Persentase Hasil

Persentase	Kategori
0% – 20%	Sangat Tidak Baik
21% – 40%	Tidak Baik
41% – 60%	Cukup
61% – 80%	Baik
81% – 100%	Sangat Baik

Pada kuesioner penelitian ini terdapat dua pernyataan negatif (*reverse question*), yaitu pertanyaan nomor 3 dan nomor 8. Oleh karena itu, sebelum dilakukan perhitungan, jawaban pada kedua pertanyaan tersebut diubah menggunakan metode *reverse scoring* sehingga skor yang lebih tinggi tetap menunjukkan tingkat penerimaan pengguna yang lebih baik.

2.9 NavMesh Agent

Navigation Mesh Agent atau yang lebih dikenal sebagai NavMesh Agent merupakan komponen bawaan (*built-in component*) dalam Unity 3D yang digunakan untuk menciptakan karakter berbasis *Artificial Intelligence* (AI) dengan kemampuan navigasi dan pencarian rute cerdas (*pathfinding*). Komponen ini bekerja dengan cara memetakan area permukaan yang dapat dilalui oleh karakter di dalam lingkungan virtual tiga dimensi, yang disebut sebagai struktur NavMesh. Berdasarkan peta geometri tersebut, NavMesh Agent menggunakan variasi algoritma A* (A-Star) untuk menghitung rute terpendek dan paling efisien dari titik koordinat awal menuju titik target yang telah ditentukan [30]. Selain kemampuan navigasi dasar, komponen ini juga dilengkapi dengan fitur penghindaran rintangan secara otomatis (*obstacle avoidance*) yang memungkinkan agen virtual untuk mendeteksi serta memutar objek statis maupun dinamis yang menghalangi jalur tanpa merusak kelancaran pergerakan.

Dalam konteks simulasi evakuasi bencana gempa bumi, implementasi NavMesh Agent memegang peranan krusial untuk mengatasi keterbatasan lingkungan permainan yang statis. Komponen ini diaplikasikan untuk mengontrol perilaku *Non-Player Character* (NPC), yang berfungsi representasi kerumunan masyarakat (*crowd simulation*) [24]. Dengan memanfaatkan NavMesh Agent, karakter-karakter tersebut dapat mensimulasikan pergerakan manusia yang dinamis saat kondisi darurat terjadi, seperti kemampuan merespons reruntuhan bangunan yang tiba-tiba menutupi jalur evakuasi utama dan secara mandiri mencari rute penyelamatan alternatif yang lebih aman. Pendekatan teknis ini penting untuk menghadirkan atmosfer kepanikan dan kepadatan koridor pada *platform desktop*, sehingga mampu melatih kesadaran situasional (*situational awareness*) [16] dan ketajaman pengambilan keputusan pengguna selama proses simulasi berlangsung.

NavMesh Agent dipilih sebagai komponen navigasi NPC pada penelitian ini karena merupakan komponen bawaan Unity yang efisien secara komputasi untuk skala NPC kecil hingga menengah, tidak memerlukan implementasi algoritma *pathfinding* maupun *crowd simulation* dari awal (seperti *Reciprocal Velocity Obstacles* atau *flow field pathfinding*), serta terintegrasi langsung dengan sistem deteksi tabrakan dan navigasi Unity 3D. Pemilihan ini disesuaikan dengan cakupan penelitian yang berfokus pada perancangan pengalaman gamifikasi dan evaluasi penerimaan pengguna, bukan pada pengembangan model *crowd dynamics* yang kompleks.

Perlu ditekankan bahwa perilaku NPC yang dihasilkan oleh NavMesh Agent pada penelitian ini merupakan representasi sederhana dari pergerakan kerumunan, dan belum divalidasi secara kuantitatif terhadap data maupun tolok ukur evakuasi di dunia nyata, seperti waktu evakuasi aktual, pola pembentukan *bottleneck*, maupun *herding behavior* yang dikaji pada studi evakuasi sesungguhnya [4]. Dengan kata lain, perilaku NPC dalam simulasi ini bertujuan untuk menghadirkan atmosfer kepanikan yang mendukung pengalaman belajar pengguna, bukan untuk memprediksi secara akurat dinamika evakuasi pada kondisi nyata. Keterbatasan ini sejalan dengan batasan penelitian yang telah dijelaskan pada Bab 1.

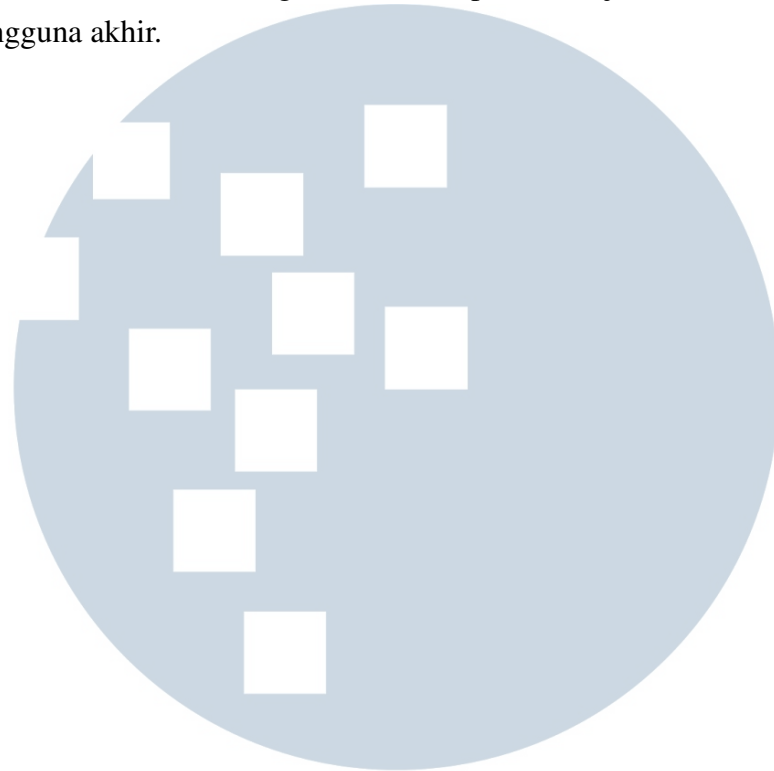
2.10 Unit Testing

Unit Testing (pengujian unit) merupakan metode pengujian perangkat lunak yang menguji unit terkecil dari kode program, seperti fungsi, metode, atau kelas, secara terisolasi untuk memastikan bahwa setiap unit tersebut bekerja sesuai dengan rancangan dan tidak menghasilkan perilaku yang tidak diinginkan [38]. Pengujian ini umumnya ditulis dan dijalankan langsung oleh pengembang selama proses implementasi berlangsung, sehingga kesalahan pada level fungsi dapat dideteksi dan diperbaiki lebih awal sebelum unit-unit tersebut diintegrasikan menjadi satu sistem yang utuh.

Unit Testing berbeda dengan *Black Box Testing*. *Black Box Testing* menguji aplikasi dari sisi luar, yaitu tampilan, aksi input, dan keluaran yang dihasilkan, tanpa memeriksa struktur kode maupun logika internal program [39], sedangkan *Unit Testing* justru menguji langsung logika internal setiap fungsi berdasarkan struktur kode program itu sendiri. Karena pengujian pada penelitian ini dilakukan dengan menyusun *test case* terhadap fungsi-fungsi individual dalam kode program seperti fungsi validasi tindakan pengguna, fungsi perhitungan skor, dan fungsi pergerakan NPC dan bukan terhadap aplikasi secara keseluruhan dari sisi pengguna, maka metode yang sesuai untuk digunakan adalah *Unit Testing*, bukan *Black Box Testing*.

Dalam penelitian ini, *Unit Testing* digunakan untuk menguji fungsionalitas setiap unit fungsi pada aplikasi simulasi evakuasi gempa bumi sebelum dilakukan pengujian penerimaan pengguna (UAT). Pengujian dilakukan dengan menyusun skenario uji (*test case*) terhadap fungsi-fungsi utama aplikasi, seperti fungsi navigasi karakter, fungsi interaksi dengan objek lingkungan, serta fungsi perpindahan antar skenario simulasi, kemudian membandingkan hasil keluaran aktual dari setiap fungsi dengan hasil yang diharapkan. Melalui pengujian ini,

peneliti dapat memastikan bahwa setiap unit fungsi aplikasi telah berjalan dengan baik dan bebas dari kesalahan logika sebelum aplikasi diujikan secara menyeluruh kepada pengguna akhir.



UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA