

BAB 3

METODE PENELITIAN

Penelitian ini menggunakan metode penelitian *research and development* (R&D), tepatnya dalam bentuk *design and development* untuk merancang dan mengembangkan sebuah simulator berbasis *virtual reality* berupa media pembelajaran tanggap darurat pada K3 kebakaran. Tahap pengembangan dilakukan dengan mengikuti model pengembangan ADDIE untuk menciptakan sebuah simulator yang efisien, informatif, mudah dipahami, dan realistik.

3.1. Subjek dan Objek Penelitian

Penelitian yang dilakukan adalah untuk menilai performa dan usability VR terhadap pembelajaran K3. Subjek dari penelitian ini adalah performa akhir dari aplikasi simulasi kebakaran berbasis VR yang dibuat.

3.2. Metode Penelitian

Metode penelitian yang digunakan dalam pengembangan pelatihan K3 kebakaran berbasis Virtual Reality (VR) adalah ***Research and Development (R&D)***. Metode ini melibatkan pengembangan produk hingga tahap pengujian dan evaluasi. Model yang digunakan dalam R&D ini adalah model ADDIE (*Analysis, Design, Development, Implementation, Evaluation*).

3.3. Langkah-Langkah Perancangan

3.3.1. Analisis (*Analysis*)

1. Mengidentifikasi kebutuhan pelatihan K3 kebakaran, termasuk aspek teknis, kompetensi yang perlu dicapai, dan potensi penerapan VR.
2. Melakukan studi literatur terkait metode pelatihan kebakaran berbasis teknologi dan standar K3 yang relevan.
3. Mengumpulkan data melalui studi kasus pada sumber-sumber terkait.

3.3.2. Perancangan (*Design*)

1. Merancang skenario simulasi kebakaran yang realistik, seperti simulasi evakuasi, penggunaan alat pemadam api ringan (APAR), atau penanganan api besar.

2. Menentukan perangkat keras (headset VR, controller) dan perangkat lunak (platform VR) yang akan digunakan.
3. Menyusun desain alur pelatihan yang mencakup informasi awal dan proses simulasi

3.3.3. Pengembangan (*Development*)

1. Mengembangkan prototipe pelatihan berbasis VR dengan bantuan aplikasi seperti Unity.
2. Melakukan uji coba awal terhadap prototipe untuk memastikan fungsi teknis dan kesesuaian dengan kebutuhan pelatihan.
3. Mendokumentasikan data performa dari aplikasi simulasi (pemakaian RAM dan GPU, kecepatan FPS, dll)

3.3.4. Implementasi (*Implementation*)

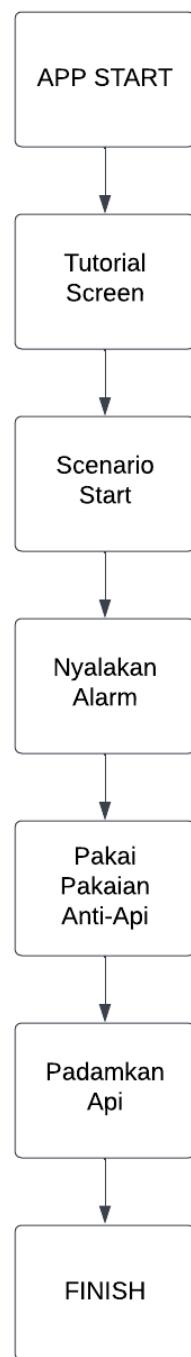
1. Pembuatan hasil akhir aplikasi simulasi

3.3.5. Evaluasi (*Evaluation*)

1. Menganalisis performa dari aplikasi akhir untuk mengetahui kualitas teknis dari simulasi yang dibuat
2. Mengumpulkan data dengan metode *benchmarking* dari *OVR Metric Tool*
3. Melakukan revisi berdasarkan hasil evaluasi untuk menyempurnakan produk pelatihan.

3.4. Perancangan Aplikasi VR

Dikembangkan tiga versi dari aplikasi yang sama, masing-masing dengan skenario dan area pengujian yang berbeda (Kantor, rumah sakit, dan sekolah). Aplikasi VR yang dikembangkan dirancang untuk memberi pelatihan kepada subjek penelitian tentang simulasi K3 kebakaran dasar berupa identifikasi tipe APAR, pemakaian pakaian darurat kebakaran, dan metode pemadaman api yang benar dan aman, seperti yang bisa dilihat di diagram dibawah ini:



Gambar 3.1. Diagram Simulasi

3.5. Spesifikasi dan Alat Pengembangan

Aplikasi VR untuk penelitian ini dikembangkan di dua komputer, yaitu komputer laptop pribadi milik peneliti dan komputer milik FPTI UPI yang dipinjamkan kepada peneliti. Spesifikasi dari kedua komputer tersebut adalah sebagai berikut:

3.5.1. HP ENVY TouchSmart 15-j173cl Laptop PC

1. *Product Number: E7Z16UA*
2. *OS: Windows 8.1 (64-bit)*
3. *Processor: 2.5GHz up to 3.5GHz AMD Quad-Core A10-5750M Accelerated Processor*
4. *Ports: Multi-format digital media card reader for Secure Digital cards. 4 SuperSpeed USB 3.0, 1 supporting USB Boost, 1 HDMI, 1 RJ-45 (LAN), 1 Headphone-out/microphone-in combo jack*
5. *Graphics: AMD Radeon HD 8650G graphics with up to 6252MB total graphics memory*
6. *Memory: 12GB DDR3L SDRAM (2 DIMM)*
7. *Storage: 125 GB SSD Storage*

3.5.2. HP Desktop PC Type 24-cb1021d

1. *Product Number: 7J057PA*
2. *OS: Windows 11 Home Single Language*
3. *Processor: Intel® Core™ i7-1255U (up to 4.7 GHz with Intel® Turbo Boost Technology, 12 MB L3 cache, 10 cores, 12 threads)*
4. *Ports: 1 USB Type-A 5Gbps signaling rate (Battery Charging 1.2); 1 headphone/microphone combo*
5. *Graphics: Intel® Iris® Xe Graphics*
6. *Memory: 16 GB memory*
7. *Storage: 512 GB SSD storage*
8. *Video Connectors: 1 HDMI-out 1.4*

Selain itu, adapun sistem VR *Headset* yang dijadikan basis untuk pengembangan aplikasi, yaitu sistem VR Oculus Meta Quest 2 berbasis Android, dengan spesifikasi sebagai berikut:

3.5.3. Oculus Meta Quest 2

1. *CPU: Snapdragon XR2, octa-core, Kryo 585 (1 x 2.84 GHz, 3 x 2.42 GHz, 4 x 1.8 GHz), 7 nm process technology*
2. *GPU: Adreno 650 RAM: 6 GB. 72 Hz refresh rate.*
3. *OS: Android 10*

3.6. Metode Pengujian

3.6.1. Pendekatan Pengujian

Pengujian dilakukan untuk mengevaluasi fungsionalitas teknis aplikasi pelatihan K3 berbasis *Virtual Reality* (VR). Ada beberapa metode yang digunakan untuk mengevaluasi aplikasi simulasi akhir.

Pengujian teknis aplikasi dilakukan dengan cara mengambil metrik performa aplikasi saat sedang digunakan (**benchmarking**). Data untuk pengujian ini diambil dengan menggunakan aplikasi OVR *Metrics Tool* yang tersedia di *Meta Horizon Store*. Metrik utama yang diambil adalah penggunaan memori RAM, Level CPU, dan konsumsi energi. Selain itu, beberapa metrik lain juga diukur. Metrik-metrik ini diambil karena secara fundamental kualitas teknis sebuah aplikasi selalu dinilai

Rezandra Rizky Irianto, 2025

PERANCANGAN SIMULATOR VIRTUAL REALITY SEBAGAI MEDIA PEMBELAJARAN TANGGAP DARURAT TERHADAP K3 KEBAKARAN

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

dari faktor-faktor ini. Contohnya pada sebuah studi oleh Rua dan Saraiva (2024) yang menilai performa berbagai aplikasi *mobile* berbasis *Android (Platform* yang juga digunakan oleh *Oculus Meta Quest 2*) secara skala besar. (Rua dan Saraiva, 2024)

Selain itu juga, metrik ini diambil sesuai dengan rekomendasi oleh *developer OVR Metric Tool* itu sendiri, disusun oleh Trevor Dasch. (Meta, 2025)

3.6.2. Langkah-Langkah Pengujian Teknis

3.6.2.1. Persiapan Pengujian:

1. Menginstall aplikasi *OVR Metrics Tool* di headset *Oculus Meta Quest 2*
2. Menentukan parameter pengujian dan metrik yang akan diambil dari simulasi menggunakan *OVR Metrics Tool*.
3. Menyalakan opsi *persistent overlay* untuk memberikan data secara *real-time* dan opsi *export data to CSV file* untuk mendapatkan informasi dan *datalog* mengenai aplikasi yang dijalankan.

3.6.2.2. Prosedur Pengujian:

1. Aplikasi VR dijalankan secara normal dan simulasi dilakukan hingga akhir.
2. Setelah penggunaan, *datalog* mengenai penggunaan sumber daya *device* oleh aplikasi akan secara otomatis dikembalikan ke aplikasi *OVR Metrics Tool* dalam bentuk *file CSV*.
3. Data yang tersedia dikirimkan dari *headset VR* kembali ke PC.
4. Menggunakan aplikasi seperti Microsoft Excel untuk mengolah data CSV menjadi grafik yang menunjukkan perubahan metrik sesuai dengan jalannya waktu.

3.6.2.3. Metrik yang Diuji:

1. FPS (*Frames per Second*):

FPS adalah satuan yang digunakan untuk mengukur jumlah frame atau gambar yang ditampilkan pada layar setiap detiknya. Satu frame adalah satu gambar statis, dan serangkaian frame yang diputar cepat akan menciptakan

ilusi gerakan. Menurut *Google Android Developer Documentation*, FPS ideal (sempurna) untuk aplikasi Android adalah 60 FPS karena sebagian besar layar smartphone memiliki *refresh rate* 60 Hz. Maka dalam kasus *Oculus Meta Quest 2* yang memiliki *refresh rate* 72 Hz, 72 FPS merupakan FPS paling ideal. Secara umum di berbagai *device*, FPS dikatakan “baik” apabila rata-rata keseluruhan berada diatas 30 FPS.

2. *Stale Frame*:

Stale frame dalam *Virtual Reality (VR)* adalah kondisi ketika gambar atau *frame* yang ditampilkan ke pengguna bukanlah hasil *render* terbaru, melainkan *frame* lama yang sudah “basi (*stale*)” karena sistem tidak sempat menghasilkan *frame* baru tepat waktu. *Stale frame* biasanya terjadi saat perangkat VR tidak mampu mempertahankan *frame rate* target (72 FPS dalam kasus ini). Akibatnya, *rendering pipeline* menampilkan ulang *frame* sebelumnya agar *display* tetap sinkron dengan *refresh rate* perangkat. Karena CPU dan GPU bekerja secara paralel, proses *rendering* satu *frame* bisa saja memakan waktu lebih lama dari durasi satu *frame* secara total, meskipun secara individual baik CPU maupun GPU tidak melebihi waktu satu *frame*.

Oleh karena itu, sebuah aplikasi bisa saja berjalan pada 72 *FPS*, namun memiliki 72 *stale frame* per detik. Masalah muncul ketika jumlah *stale frame* lebih dari nol, tetapi kurang dari 72 *frame*. Jika ini terjadi, beberapa frame akan terulang sementara frame lain akan terlewati (*Irregular Frame Pacing*). Fenomena ini akan menghasilkan sebuah *stutter* atau *lag*, yang berpotensi membuat peserta pusing akibat visual yang tidak stabil. Maka daripada itu, jumlah *stale frame* sangat terikat dengan FPS. Jumlah *stale frame* yang baik adalah 0, 72, dan atau menyamai FPS saat itu juga. Jumlah *stale frame* paling buruk adalah 36 (nilai tengah antara 0 dan 72, dimana *stutter* dan *lag* sangat terlihat). Maka semakin jauh jumlah *stale frame* dari 36, semakin baik pula kualitas gambar.

3. *App GPU Time*:

Rezandra Rizky Irianto, 2025

PERANCANGAN SIMULATOR VIRTUAL REALITY SEBAGAI MEDIA PEMBELAJARAN TANGGAP DARURAT TERHADAP K3 KEBAKARAN

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

App GPU Time menunjukkan berapa sebuah aplikasi me-*render* satu *frame*. Jika waktunya lebih lama dari durasi satu *frame* (13,88 ms untuk 72 FPS), berarti aplikasi lebih bergantung kepada GPU. Jika waktunya lebih singkat dari itu, kemungkinan besar aplikasi lebih bergantung pada CPU.

4. Level CPU dan GPU:

Metrik ini menentukan keseimbangan antara *workload* CPU dan GPU, dan apakah aplikasi lebih bergantung kepada CPU atau GPU. Metrik yang baik adalah jika jumlah level CPU dan GPU setara. Jika nilai ini timpang (Semisal level GPU adalah 3 dan level CPU adalah 4), artinya harus dilakukan optimisasi.

5. Persentase pemakaian (utilisasi) CPU dan GPU:

Metrik ini menampilkan jumlah kerja yang diberikan kepada CPU dan GPU. Persentase yang baik adalah dibawah 90 persen, untuk memastikan aplikasi tidak mengalami *freeze*.

6. *Proportional Set Size (PSS)* RAM:

Proportional Set Size (PSS) RAM adalah suatu metrik pengukuran penggunaan memori pada sistem operasi berbasis Linux, termasuk Android, yang merepresentasikan jumlah memori fisik (RAM) yang dialokasikan untuk suatu proses, dengan memperhitungkan proporsi dari segmen memori yang digunakan bersama (*shared memory*) oleh beberapa proses.

Berbeda dengan *Resident Set Size (RSS)* yang menghitung keseluruhan memori fisik yang ditempati proses (termasuk *shared memory* secara penuh), PSS membagi *shared memory* secara proporsional di antara seluruh proses yang menggunakannya. Maka daripada itu, PSS dianggap sebagai indikator yang lebih akurat terhadap beban memori yang secara efektif dikonsumsi oleh sebuah proses. Tidak ada ketentuan khusus untuk metrik ini, tetapi semakin kecil semakin baik.

7. *Screen Tear*:

Screen tearing adalah artefak visual yang terjadi ketika frame buffer yang sedang dikirim ke display device berisi bagian dari dua atau lebih

frame berbeda secara bersamaan. *Screen tear* terjadi saat GPU me-render *frame* secara tidak sinkron dengan *display refresh rate* (72 Hz di *Oculus Meta Quest*). Fenomena ini biasanya muncul sebagai garis horizontal di layar, di mana bagian atas dan bawah gambar menampilkan informasi visual dari frame yang berbeda. Dalam aplikasi yang ideal, *screen tear* seharusnya tidak terjadi.

3.6.3. Instrumen yang Digunakan

1. *OVR Metrics Tool*: Alat pemantau kinerja untuk *headset Meta Quest 2* yang menyediakan berbagai metrik kinerja, termasuk *frame rate* (FPS), suhu, nilai *throttling* GPU dan CPU, serta jumlah *screen tear* dan *stale frame* per detik. *OVR Metrics Tool* menyajikan data ini melalui grafik di layar yang ditampilkan di atas aplikasi yang sedang berjalan atau melalui laporan yang dapat diambil dari perangkat.
2. Perangkat VR: Headset VR yang digunakan untuk menjalankan aplikasi selama pengujian.

3.6.4. Analisis Data

3.6.4.1. Analisis Kualitatif:

Feedback langsung dari hasil tes terkait masalah teknis atau pengalaman penggunaan aplikasi.

1.6.4.2. Analisis Kuantitatif:

Grafik dan tabel penyajian data hasil *benchmarking* performa aplikasi.

3.6.5. Hasil yang Diharapkan

1. Aplikasi Simulasi Tanggap Darurat Kebakaran Berbasis VR.
2. Data hasil *benchmarking* yang menunjukkan performa dan efisiensi aplikasi
3. Rekomendasi untuk meningkatkan kualitas aplikasi, jika diperlukan.