

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Relevan

Bagian ini merupakan bagian untuk meninjau literatur yang relevan dengan topik yang berkaitan dengan pengimplementasian algoritma SLAM. Analisis dari literatur ini bertujuan untuk memberikan pemahaman yang lebih dalam mengenai perkembangan terkini dalam bidang pembuatan dunia virtual, terutama dibagian *Augmented Reality*. Tinjauan ini juga akan menjadi landasan kuat bagi penelitian yang akan dilakukan. Berikut ini merupakan beberapa penelitian yang dianggap relevan dengan fokus penelitian ini.

Penelitian oleh Hořejší et al (2024) Dalam penelitiannya yang berjudul “*Reliability and Accuracy of Indoor Warehouse Navigation Using Augmented Reality*”, merupakan sebuah aplikasi navigasi *indoor* yang dikembangkan untuk lingkungan gudang (*warehouse*) dengan menggunakan Unity3D, AR Foundation, dan ARCore. Penelitian ini bertujuan untuk menguji keandalan sensor inersia pada *smartphone* untuk navigasi di dalam gudang. Hasil pengujian ini menunjukkan tingkat keberhasilan mencapai 83% dalam mencapai target dengan rata-rata akurasi (deviasi) sebesar 0.48 meter. Perbedaan dari penelitian ini dengan penelitian yang saat ini sedang dilakukan adalah letak lokasi (gudang vs kampus) serta penelitian yang sedang dikerjakan saat ini berfokus pada perbandingan akurasi antara model SLAM (*Cloud Anchor*) dengan *Non-SLAM*.

Penelitian oleh Alkady et al. (2024) Penelitian dengan judul “*SINS_AR: An Efficient Smart Indoor Navigation System Based on Augmented Reality*” mengusulkan sistem navigasi menggunakan penanda visual (QR Code) serta algoritma Theta. Sistem ini nantinya akan menempatkan QR Code di berbagai titik strategis untuk melakukan lokalisasi. Hasilnya menunjukkan sebuah sistem yang akurat dan fleksibel. Untuk perbedaan dasar dengan penelitian ini terletak di bagian metode pelacakannya. Alkady menggunakan metode *Marked-based* (bergantung dengan QR Code), sedangkan penelitian ini menggunakan metode *Markerless* berbasis SLAM yang lebih minim infrastruktur.

Penelitian oleh Ress et al. (2024) Di dalam jurnalnya yang berjudul “*SLAM FOR INDOOR MAPPING OF WIDE ARE CONSTRUCTION ENVIRONMENTS*”, dilakukan perbandingan antara LiDAR SLAM dan Visual SLAM di area konstruksi yang luas. Hasilnya menunjukkan bahwa meskipun LiDAR lebih akurat, Visual SLAM menawarkan solusi biaya

yang rendah. Penelitian ini menjadi landasan bahwa metode Visual SLAM memiliki potensi besar untuk navigasi di area yang luas, meskipun memiliki tantangan akurasi yang akan diuji lebih lanjut dalam penelitian ii menggunakan parameter RMSE.

Penelitian oleh Anggaresta et al (2024) Penelitian dengan judul “*Implementasi Teknologi Augmented Reality Untuk Navigasi Pada Area Kantor Pemerintahan Kabupaten Mojokerto*” mengembangkan aplikasi navigasi dengan menggunakan metode *markerless* dengan bantuan plugin AR ditambah GPS. Hasil pengujian fungsionalitas mencapai 100% dengan akurasi visual 80%. Perbedaan penelitian ini dengan penelitian yang sedang dilaksanakan adalah penggunaan GPS, yang dimana penelitian Anggaresta menggabungkan GPS (yang dimana kurang efektif di *indoor*). Sedangkan penelitian ini murni menggunakan Visual SLAM dan *Cloud Anchors* untuk lingkungan *indoor* tanpa sinyal GPS.

Penelitian oleh Maran et al (2023) Pada penelitian “*Augmented reality-based indoor navigation using unity engine*”, dikembangkan sistem navigasi AR dengan menggunakan Unity Engine dengan memanfaatkan *NavMesh Agent* untuk pencarian jalur navigasi. Penelitian ini membuktikan bahwa Unity adalah *platform* yang handal untuk mengintegrasikan logika navigasi AI dengan visualisasi AR. Penelitian ini mengadopsi penggunaan Unity, namun berfokus pada pengukuran kuantitatif *drift* posisi pengguna.

Penelitian oleh Jiang et al(2023) Penelitian dengan judul “*An Indoor Location-Based Augmented Reality Framework*” mengusulkan kerangka kerja (framework) navigasi AR dengan mengintegrasikan sensor yang terdapat pada *smartphone* dan tanda (*marker*) visual. Penelitian ini menekankan pentingnya arsitektur sistem yang efisien. Relevansi dengan penelitian yang sedang dilaksanakan terletak pada arsitektur sistem navigasi *mobile*, namun penelitian ini menggunakan *Cloud Anchors Google* sebagai basis data spasial utama, dan bukan kerangka kerja kustom.

Penelitian oleh Nukarinen (2023) Di dalam tesis yang berjudul “*Indoor Navigation Using Unity Augmented Reality*”, dikembangkan sebuah aplikasi navigasi *indoor* berbasis Unity dengan metode lokalisasi berbasis gambar. Pengukuran error jarak menghasilkan tingkat akurasi antara 10 hingga 140 cm. Penelitian ini juga menjadi acuan dalam metodologi pengujian, di mana penelitian ini juga akan menggunakan deviasi jarak (*End Position Error*) untuk mengevaluasi performa dari metode SLAM.

Penelitian oleh Zhang et al. (2022) Penelitian “*Visual indoor navigation using mobile augmented reality*” berfokus pada penggunaan kamera *smartphone* untuk navigasi visual. Hasilnya menunjukkan bahwa navigasi visual bisa lebih intuitif bagi penggunanya dibandingkan peta 2D. Penelitian ini mendukung pemilihan antarmuka AR, namun penelitian ini akan menambahkan analisis secara komparatif teknik antara mode SLAM dan VO murni yang belum dibahas secara mendalam pada penelitian tersebut.

Penelitian oleh Chidsin et al (2021) Penelitian berjudul “*AR-Based Navigation Using RGB-D Camera and Hybrid Map*” mengusulkan peta hibrida yang menggabungkan point cloud SLAM dengan peta lantai 2D. Hasilnya menunjukkan peningkatan dari orientasi pengguna. Penelitian ini relevan dalam hal konsep pemetaan (*mapping*), namun penelitian ini menggunakan ARCore *Persistent Cloud Anchor* yang tidak memerlukan kamera RGB-D khusus (seperti Kinect), melainkan cukup kamera *smartphone* standar.

Penelitian oleh Manuaba (2021) Pada penelitian “*Mobile based Augmented Reality Application Prototype for Remote Collaboration Scenario Using ARCore Cloud Anchor*”, diuji performa *hosting* dan *resolving Cloud Anchor*. Hasilnya menunjukkan sinkronisasi posisi yang stabil dengan *delay* yang rendah. Penelitian ini menjadi landasan teknis bahwa teknologi *Cloud Anchor* layak digunakan sebagai titik referensi persisten untuk mengkoreksi posisi dalam aplikasi navigasi.

2.1.1 Matriks Penelitian

Berikut ini merupakan tabel matriks penelitian terkait.

No	Peneliti (Tahun)	Judul Penelitian	Sumber Jurnal	Data	Metodologi	Kesimpulan
1	Hořejší, Macháč, dan Šimon (2024)	Reliability and Accuracy of Indoor Warehouse Navigation Using Augmented Reality	IEEE Access	Data sensor inersia <i>smartphone</i> (Akselerometer, Giroskop) dan Kamera	Markerless AR menggunakan Unity, AR Foundation, dan ARCore	Tingkat keberhasilan navigasi mencapai 83% dengan rata-rata akurasi (deviasi) 0.48 meter.

No	Peneliti (Tahun)	Judul Penelitian	Sumber Jurnal	Data	Metodologi	Kesimpulan
2	Alkady et al. (2024)	SINS_AR: An Efficient Smart Indoor Navigation System Based on Augmented Reality .	IEEE Access	Penanda visual (Visual Markers/ QR Codes) dan data kamera smartph one .	Marker-based tracking dikombinasikan dengan algoritma pencarian jalur Theta*.	Sistem menawarkan solusi navigasi yang fleksibel, dapat disesuaikan (scalable), dan hemat biaya dibandingkan sistem tradisional .
3	Ress et al (2024)	SLAM FOR INDOOR MAPPING OF WIDE AREA CONSTRUCTION ENVIRONMENTS	arXiv / ISPRS Annals	Data LiDAR 3D, kamera stereo, dan IMU.	Perbandingan antara LiDAR SLAM (DMSA) dan Visual SLAM (DROID-SLAM).	LiDAR memberikan akurasi geometris tinggi, sedangkan Visual SLAM efisien namun rentan terhadap gangguan pencahayaan.
4	Anggar esta, Wibowo, & Khairy (2024)	Implementasi Teknologi Augmented Reality Untuk Navigasi Pada Area Kantor Pemerintahan Kabupaten Mojokerto	Journal of Animation and Games Studies	Data lokasi (GPS) dan kamera smartph one.	Markerless AR dengan plugin AR+GPS.	Akurasi visualisasi mencapai 80% dan tingkat user friendly sebesar 91,6% .
5	Maran, Giridharan, & Krishna	Augmented Reality-based Indoor Navigation	Proc. Of ICSCSS (IEEE)	Model lingkungan 3D (<i>Environ</i>	Navigasi menggunakan	Sistem tidak memerlukan perangkat keras

No	Peneliti (Tahun)	Judul Penelitian	Sumber Jurnal	Data	Metodologi	Kesimpulan
	veni (2023)	using Unity Engine		<i>tment Model)</i> dan pelacak virtual	Kan Unity NavMesh Agent	eksternal mahal dan memberikan visualisasi <i>real-time</i> yang efisien
6	Jiang & Subakti (2023)	An Indoor Localization – Based Augmented Reality Framework	Sensors (MDPI)	<i>Marker</i> terlihat (QR/ Gambar) Marker tak terlihat (Wi-Fi/ BLE), dan sensor perangkat	<i>Framework</i> ILARF yang mengintegrasikan lokalisasi dan AR	<i>Framework</i> ILARF memfasilitasi pengembangan sistem AR yang mendukung mode layar perangkat dan <i>Head Mounted Display</i>
7	Nukari nen (2023)	Indoor Navigation Using Unity Augmented Reality	Tesis Master, Aalto University	Gambar kamera dan model bangunan yang dibuat sebelumnya	Lokalisasi berbasis gambar (<i>Image-based localization</i>) menggunakan Kan Unity	Aplikasi dinilai andal dengan akurasi lokalisasi antara 10 hingga 140 cm dan durasi 2-48 detik

No	Peneliti (Tahun)	Judul Penelitian	Sumber Jurnal	Data	Metodologi	Kesimpulan
8	Zhang et al (2022)	Visual Indoor Navigation Using Mobile Augmented Reality	Advances in Computer Graphics (Springer)	Citra visual dari kamera <i>smartphone</i>	Visual SLAM untuk mendeteksi fitur visual lingkungan	Aplikasi berbasis SLAM terbukti jauh lebih akurat dibandingkan aplikasi berbasis GPS untuk di dalam ruangan
9	Chidsin , Gu, & Goncharenko (2021)	AR-Based Navigation Using RGB-D Camera and Hybrid Map	Sustainability (MDPI)	Citra RGB dan Kedalaman (<i>Depth</i>) dari kamera RGB-D	SLAM membuat peta <i>point cloud</i> yang diintegrasikan dengan peta lantai (<i>Hybrid Map</i>)	Metode koreksi error orientasi yang diusulkan secara signifikan meningkatkan kinerja navigasi disbanding <i>baseline</i>
10	Manuaba (2021)	Mobile based Augmented Reality Application Prototype for Remote Collaboration	Procedia Computer Science (Elsevier)	Input kamera <i>host</i> dan <i>client</i>	ARCore Cloud Anchors untuk sinkronisasi posisi antar perangkat	<i>Cloud Anchor</i> berhasil <i>hosting</i> dengan rata-rata delay 700ms,

No	Peneliti (Tahun)	Judul Penelitian	Sumber Jurnal	Data	Metodologi	Kesimpulan
						menunjukkan potensi

Tabel 2. 1 Matriks Penelitian

Tabel 2.1 menyajikan matriks penelitian yang merangkum kajian terhadap penelitian-penelitian terdahulu yang relevan dengan sistem navigasi berbasis *Augmented Reality*. Matriks ini memuat informasi mengenai peneliti, judul, sumber publikasi, jenis data, metodologi, serta kesimpulan utama dari masing-masing penelitian. Melalui matriks ini, dapat diketahui bahwa penelitian sebelumnya umumnya memanfaatkan data sensor inersia dan visual dengan pendekatan seperti *visual odometry*, *markerless tracking*, dan ARCore. Selain itu, sebagian besar penelitian menekankan aspek pengembangan sistem tanpa disertai evaluasi kuantitatif yang komprehensif terhadap akurasi lintasan.

Oleh karena itu, matriks penelitian ini digunakan sebagai dasar untuk mengidentifikasi celah penelitian (*research gap*) serta memperkuat landasan metodologis dalam penelitian ini, khususnya dalam melakukan evaluasi akurasi metode SLAM dan *Non-SLAM* secara sistematis dan terukur.

2.2 Landasan Teori

2.2.1 *Augmented Reality* (AR)

Augmented Reality (AR) Merupakan sebuah teknologi yang menggabungkan benda maya (*virtual*) dua dimensi atau tiga dimensi ke dalam lingkungan nyata, lalu memproyeksikan benda-benda maya tersebut secara *real-time* (waktu nyata). Berbeda dengan *Virtual Reality* (VR) yang membawa pengguna keluar dari dunia nyata ke dalam lingkungan buatan, AR mempertahankan lingkungan dunia nyata sebagai latar belakang utama serta memperkayanya dengan informasi digital.

Menurut Ronald T.Azuma (1997) dalam penelitiannya yang menjadi rujukan standar, sebuah sistem dapat didefinisikan sebagai *Augmented Reality* jika memenuhi tiga karakteristik utama, yaitu:

1. **Menggabungkan Nyata dan Virtual (*Combines real dan virtual*):** Sistem harus bisa menampilkan objek virtual yang berdampingan dengan objek nyata dalam satu tampilan.

2. **Interaktif dalam Waktu Nyata (*Interactive in real time*):** Sistem harus merespons perubahan lingkungan atau input pengguna secara langsung tanpa *delay* yang signifikan.
3. **Terdaftar dalam Tiga Dimensi (*Registered in 3-D*):** Objek virtual harus tampak “menempel” atau terikat pada koordinat tiga dimensi dunia nyata, sehingga perspektifnya tetap sesuai meskipun pengguna bergerak.

Dalam konteks navigasi di dalam ruangan, AR memiliki peran vital sebagai antarmuka visual. Penelitian oleh Hořejší et al. (2024) menyatakan bahwa sistem AR dapat mengintegrasikan instruksi navigasi secara langsung ke lingkungan fisik, yang dapat memungkinkan navigasi yang lebih lancar dan meningkatkan kewaspadaan pengguna dibandingkan menggunakan peta 2D abstrak.

2.2.2 Navigasi Di Dalam Ruangan (*Indoor Navigation*)

Navigasi merupakan suatu proses memantau dan mengendalikan pergerakan pengguna dari satu tempat ke tempat lain. Di lingkungan luar ruangan (*outdoor*), teknologi *Global Positioning System* (GPS) telah menjadi solusi yang dominan. Namun, efektivitas GPS menurun drastis di dalam ruangan.

Menurut Alkady et al. (2024), sinyal GPS bekerja pada rentang frekuensi radio ultra-tinggi yang sulit menembus material padat seperti atap beton dan dinding tebal. Ketidadaan garis pandang (*line of sight*) secara langsung dari satelit menyebabkan fenomena *multipath* dan atenuasi sinyal, sehingga akurasi posisi menjadi sangat buruk dan tidak dapat diandalkan untuk navigasi presisi di dalam Gedung kompleks. Oleh karena itu, diperlukan teknologi alternatif seperti *Inertial Navigation System* (INS), *Visual Positioning*, atau SLAM untuk menggantikan peran GPS di dalam ruangan.

2.2.3 Simultaneous Localization and Mapping (SLAM)

Simultaneous Localization and Mapping (SLAM) merupakan sebuah metode komputasi fundamental di dalam bidang robotika dan visi komputer yang memungkinkan sebuah perangkat atau *hardware* mampu untuk membangun atau memperbarui peta dari lingkungan yang tidak diketahui (*mapping*), sekaligus melacak posisi perangkat itu sendiri di dalam peta tersebut (*localization*) secara bersamaan. Masalah ini biasanya digambarkan seperti berikut: untuk mengetahui posisi yang akurat diperlukan peta yang akurat, tetapi untuk membangun peta yang akurat diperlukan pengetahuan posisi yang tepat. Dalam hal tersebut SLAM telah

memecahkan paradoks tersebut dengan melakukan kedua proses itu secara iterative dan simultan.

Dalam konteks *Augmented Reality* (AR) pada perangkat *mobile* (Seperti yang digunakan didalam penelitian ini dengan ARCore), metode ini secara spesifik disebut sebagai Visual SLAM. Berbeda dengan SLAM yang berbasis LiDAR yang menggunakan sinar laser untuk mengukur jarak, Visual SLAM mengandalkan data visual dari kamera untuk memahami geometri ruang.

Secara arsitektur, sistem SLAM modern umumnya terdiri dari tiga komponen utama yang bekerja secara parallel, sebagaimana dijelaskan dalam penelitian Chidsin et al. (2021) mengenai ORB-SLAM:

1. **Tracking (Pelacakan)** Komponen ini bertugas untuk mengolah input visual dari kamera secara *real-time* (setiap frame). Sistem akan mengekstraksi “titik fitur” (*feature points*) yang unik dari gambar, seperti sudut meja, tepi pintu, atau pola lantai. Dengan mencocokkan titik-titik fitur ini antara frame sebelumnya dengan frame saat ini, sistem dapat mengestimasi perubahan orientasi perangkat (*pose*). Proses ini mirip dengan Visual Odometry, yang memberikan estimasi posisi cepat namun rentan terhadap akumulasi kesalahan (*drift*).
2. **Local Mapping(Pemetaan)** Komponen ini bekerja untuk membangun representasi 3D dari lingkungan di sekitar. Sistem memilih frame kunci atau *keyframes* tertentu dan menyimpannya untuk membangun peta local yang terdiri atas titik awan (*point cloud*). Proses ini melibatkan optimasi matematis (seperti *Bundle Adjustment*) untuk meminimalkan kesalahan antara posisi titik fitur yang diprediksi dengan posisi yang sebenarnya terlihat di kamera, sehingga akan menghasilkan peta local yang konsisten dan akurat.
3. **Loop Closing** Ini adalah komponen paling kritis dalam SLAM untuk mengatasi *drift*. *Loop Closing* merupakan kemampuan sistem untuk mengenali Kembali lokasi yang sudah pernah dikunjungi sebelumnya. Jika perangkat mendeteksi kesamaan fitur visual antara lokasi saat ini dengan lokasi yang tersimpan di peta global, sistem akan menyadari bahwa “dia pernah disini sebelumnya”. Sistem kemudian akan menghitung selisih kesalahan

(*accumulated drift*) yang terjadi sepanjang perjalanan dan mengoreksi seluruh jalur navigasi agar peta menjadi konsisten secara global.

Dalam penelitian ini, implementasi SLAM diwujudkan melalui teknologi ARCore *Cloud Anchors*. Fitur ini memanfaatkan prinsip *loop closure* dan *relocalization* yang kuat. Perangkat ini akan memetakan fitur visual lingkungan dan menggongahnya ke layanan cloud untuk mendapatkan ID unik. Ketika perangkat kembali ke area tersebut, maka sistem akan melakukan pencocokan fitur visual *real-time* dengan data peta yang tersimpan, sehingga memungkinkan perangkat untuk “mengunci” kembali posisinya di dunia nyata dengan presisi tinggi, dan meniadakan efek *drift* yang biasanya terjadi pada metode navigasi visual biasa.

Keunggulan utama SLAM dibandingkan metode lain adalah kemampuannya untuk beroperasi tanpa infrastruktur eksternal (seperti sinyal GPS atau marker fisik di setiap sudut) dan kemampuannya untuk memperbaiki kesalahan posisi secara otomatis, menjadikannya solusi ideal untuk bernavigasi di dalam ruangan.

2.2.4 Visual-Odometry (Metode *Non-SLAM*)

Visual-Odometry (VO) merupakan sebuah proses estimasi posisi dan orientasi (pose) dari sebuah agen, seperti robot atau perangkat seluler, dengan cara menganalisis perubahan visual yang terjadi pada urutan gambar kamera secara *-real-time*. Istilah ini diadopsi dari konsep “odometri” pada kendaraan beroda yang menghitung jarak tempuh berdasarkan putaran roda, namun di dalam VO sensor utamanya digantikan oleh kamera. Pada konteks penelitian ini, Visual-Odometry merepresentasikan metode navigasi *Non-SLAM*. Perbedaan yang sangat fundamental antara SLAM dan VO terletak pada ketiadaan peta global yang persisten. Jika SLAM dapat membangun peta dan melacak posisi sekaligus (*Mapping & Localization*), VO hanya berfokus pada pelacakan pergerakan relative dari satu titik ke titik berikutnya secara incremental tanpa menyimpan memori spasial jangka panjang atau peta global.

Pada implementasi modern yang menggunakan perangkat *mobile* dan framework ARCore, metode ini bekerja dengan menggabungkan data visual dengan sensor inersia, yang dikenal sebagai Visual-Inertial Odometry (VIO). Hořejší, Macháč, dan Šimon (2024) menjelaskan bahwa integrasi ini sangat krusial untuk menjaga stabilitas pelacakan pada lingkungan yang dinamis. Sistem VIO dengan mensinergikan komponen-komponen sensor utama. Pertama, sistem visual (kamera)

mendeteksi titik-titik fitur (*feature points*) yang kontras pada lingkungan, seperti sudut lantai atau tekstur dinding, serta melacak pergeseran optic titik-titik tersebut antar frame untuk memberikan data presisi pergerakan. Kedua, sistem inersia menggunakan akselerometer dan giroskop untuk mengukur percepatan linier dan rotasi sudut yang sangat efektif menangani pergerakan cepat (*high-frequency motion*) saat kamera mengalami *motion blur* atau kehilangan fokus visual sesaat.

Meskipun VIO mampu melacak pergerakan dengan cukup baik dalam jangka pendek, metode ini memiliki kelemahan fundamental yang menjadikan variabel control di dalam penelitian ini, yaitu fenomena *drift* atau penyimpangan akumulatif. Berbeda dengan SLAM yang memiliki mekanisme *Loop Closing* untuk mengenali lokasi lama serta memperbaiki kesalahan., VO bekerja dengan menggunakan prinsip *Dead Reckoning*. Posisi saat ini dihitung secara matematis berdasarkan posisi sebelumnya ditambah dengan estimasi pergerakan saat ini. Karena setiap pengukuran sensor pasti memiliki nilai kesalahan atau error sekecil apapun, kesalahan ini tidak pernah dihilangkan tetapi terus dijumlahkan (*accumulated*) seiring berjalannya waktu dan bertambahnya jarak tempuh.

Dampak dari akumulasi kesalahan ini dijelaskan lebih lanjut oleh Chidsin, Gu, dan Goncharenko (2021), yang menyoroti bahwa kesalahan orientasi (*orientation error*) adalah penyumbang terbesar terhadap ketidakakuratan navigasi visual. Sedikit saja kesalahan pada perhitungan sudut putar (*yam*), maka perangkat akan menyebabkan garis lajur navigasi virtual melenceng jauh dari posisi sebenarnya setelah pengguna berjalan beberapa meter. Dalam scenario penelitian ini, mode *Non-SLAM* akan merepresentasikan kondisi dimana sistem berjalan secara *Open Loop* atau tanpa umpan balik peta global. Tanpa adanya *Cloud Anchor* untuk melakukan relokalisasi, sistem tidak memiliki cara untuk memverifikasi posisi aktualnya di dunia nyata, yang mengakibatkan penurunan akurasi yang signifikan pada saat melakukan navigasi dengan jarak tempuh yang jauh.

2.2.5 ARCore dan Cloud Anchors

A. Arsitektur ARCore

ARCore merupakan sebuah *software Development Kit* (SDK) yang dikembangkan oleh *Google* untuk menghadirkan pengalaman *Augmented Reality* di dalam perangkat Android tanpa memerlukan *hardware* tambahan khusus. Hořejší,

Macháč, dan Šimon (2024) menjelaskan bahwa ARCore bekerja dengan mengintegrasikan data dari kamera, akselerometer, dan giroskop untuk memahami lingkungan nyata di sekitar pengguna.

Secara fundamental, ARCore dibangun di atas tiga teknologi utama yaitu:

1. **Motion Tracking (Pelacakan Gerakan):** Memungkinkan *smartphone* untuk memahami dan melacak posisinya terhadap lingkungan nyata menggunakan metode *Concurrent Odometry and Mapping* (COM).
2. **Environmental Understanding (Pemahaman Lingkungan):** Memungkinkan perangkat untuk mendeteksi ukuran dan lokasi permukaan vertical atau horizontal, seperti lantai atau meja.
3. **Light Estimation (Estimasi Cahaya):** Memungkinkan perangkat untuk memperkirakan kondisi pencahayaan lingkungan agar objek virtual tampak realistis.

Dalam konteks navigasi AR pada penelitian ini, beberapa komponen seperti *Motion Tracking* dan *Environmental Understanding* berperan penting sebagai dasar perhitungan pose pengguna serta mendeteksi bidang lantai yang menjadi media penempatan jalur navigasi virtual.

B. ARCore Cloud Anchors

ARCore *Cloud Anchors* merupakan fitur yang memungkinkan terbentuknya titik referensi spasial yang diproses melalui layanan cloud *Google*. Manuaba (2011) mendefinisikan *Cloud Anchor* sebagai titik referensi yang dapat digunakan berbagai perangkat untuk berbagi kerangka koordinat spasial yang sama.

Namun perlu ditegaskan bahwa didalam penelitian ini, *Cloud Anchor* tidak digunakan untuk kolaborasi multi-perangkat (*multi-user sharing*), melainkan hanya akan dimanfaatkan sebagai mekanisme relokalisasi tunggal (*single-user*). Tujuannya adalah untuk mengoreksi kesalahan pelacakan posisi (*drift*) yang terakumulasi selama navigasi jarak jauh. Meskipun pemrosesan peta visual dilakukan di sisi cloud *Google*, aplikasi hanya akan menyimpan ID *Cloud Anchor* secara local pada perangkat dalam bentuk format CSV. ID ini kemudian dapat digunakan kembali untuk proses *resolving* sehingga posisi virtual dapat dikunci ulang secara konsisten tanpa memerlukan server atau basis data eksternal mandiri.

C. Mekanisme *Hosting* dan *Resolving*

Implementasi teknis *Cloud Anchor* melibatkan dua proses utama yang bekerja secara bergantian dan berulang:

1. Proses *Hosting* (Pemetaan)

Perangkat akan memindai lingkungan untuk mengumpulkan data titik fitur visual. Data ini dikirim ke layanan ARCore untuk diproses menjadi ID unik. Hořejší et al. (2024) mencatat bahwa kualitas *hosting* sangat bergantung pada kondisi pencahayaan dan kekayaan tekstur visual di lingkungan tersebut.

2. Proses *Resolving* (Relokalisasi)

Perangkat akan mengirimkan ID yang tersimpan beserta data visual kamera saat ini ke layanan ARCore. Sistem ini kemudian membandingkan fitur visual *real-time* dengan peta yang tersimpan. Jika cocok, maka sistem akan mengembalikan transformasi koordinat yang presisi.

Pada konteks penelitian ini, proses *hosting* dan *resolving* dilakukan pada perangkat yang sama, menegaskan bahwa fungsi *Cloud Anchor* sebagai memori spasial untuk koreksi *drift* individu, bukan untuk sinkronisasi antar perangkat yang berbeda.

2.2.6 Unity Game Engine dan AR Foundation

Unity merupakan platform pengembangan yang menyediakan lingkungan pengembangan yang terintegrasi (*Integrated Development Environment/ IDE*) untuk membuat aplikasi interaktif 2D, 3D, *Virtual Reality* (VR), *Augmented Reality* (AR), dan *Mixed Reality* (MR). Meskipun awalnya dirancang untuk industri permainan (*game*), Unity kini telah banyak dimanfaatkan tidak hanya di dalam industri *game*, tetapi juga pada pengembangan simulasi teknis, visualisasi arsitektural, hingga pembuatan prototipe sistem navigasi digital. Maran et al. (2023) dalam penelitiannya menyebutkan bahwa Unity menyediakan ekosistem yang handal untuk menangani kalkulasi fisika, rendering grafis, dan juga logika navigasi spasial dalam satu wadah yang efisien. Keunggulan utama Unity terletak pada arsitektur berbasis komponen, di mana setiap objek dalam aplikasi (*GameObject*) dibentuk oleh berbagai komponen skrip (C#) yang mendefinisikan perilaku dan fungsinya, memungkinkan pengembangan sistem navigasi yang modular dan mudah dikelola.

Dalam konteks pengembangan aplikasi AR lintas platform, Unity menyediakan framework khusus yang disebut AR Foundation. Framework ini bertindak sebagai lapisan abstraksi atau antarmuka yang menjembatani kode program pengembang dengan platform AR asli dari perangkat keras, yaitu ARCore (untuk Android) dan ARKit (untuk iOS). Dengan menggunakan AR Foundation, pengembang hanya perlu menulis satu basis kode logika navigasi, dan kemudian sistem akan secara otomatis menerjemahkannya ke dalam instruksi *native* yang sesuai dengan perangkat pengguna. Hořejší et al. (2024) menekankan bahwa penggunaan alat standar seperti AR Foundation sangat penting untuk memastikan kompatibilitas aplikasi terhadap berbagai jenis sensor inersia dan kamera pada *smartphone* modern, serta mempermudah integrasi fitur-fitur kompleks seperti deteksi bidang serta pencahayaan lingkungan.

Dengan demikian, AR Foundation berfungsi sebagai kerangka interoperabilitas yang menyatukan logika pemetaan, deteksi lingkungan, dan pelacakan pose ke dalam satu mode pengembangan berbasis komponen di Unity. Secara teknis, AR Foundation bekerja dengan mengelola berbagai subsistem AR yang berjalan secara independent namun terintegrasi. Subsistem utama yang dimanfaatkan dalam penelitian ini meliputi:

1. **XRCameraSubsystem**

Mengelola pelacakan pose dengan menyelaraskan kamera virtual di Unity dengan pergerakan kamera fisik perangkat.

2. **XRPlaneSubsystem**

Mendeteksi permukaan datar (terutama lantai) sebagai basis penempatan jalur navigasi

3. **ARCore Extensions**

Berfungsi sebagai paket tambahan untuk mengakses fitur eksklusif ARCore, seperti *Cloud Anchors* API. Pada penelitian ini, fitur tersebut nantinya akan digunakan untuk melakukan *hosting* dan *resolving* guna mengurangi *drift* selama proses navigasi di dalam ruangan.

2.2.7 Perbandingan Metode Penentuan Posisi

Dalam pengembangan sistem navigasi dalam ruangan (*indoor navigation*), pemilihan metode penentuan posisi atau *positioning* menjadi faktor paling krusial

yang menentukan keberhasilan sistem. Berbeda dengan lingkungan di luar ruangan yang memiliki standar tunggal berupa GPS, lingkungan dalam ruangan memiliki tantangan kompleks seperti oklusi jaringan, gangguan magnetic, serta kondisi pencahayaan yang dinamis.

Untuk dapat memposisikan kontribusi dari penelitian ini, maka dilakukan perbandingan antara empat metode utama, yaitu GPS (sebagai standar *outdoor*), *Marker-based AR* (teknologi AR generasi awal), visual odometry (teknologi dasar ARCore), dan SLAM berbasis *Cloud Anchor* (solusi yang diusulkan). Perbandingan-perbandingan tersebut dirangkum di dalam tabel 2.2 berikut:

Kriteria	Lingkungan Operasional	Akurasi di dalam ruangan	Masalah Utama	Infrastruktur	Mekanisme Koreksi	Peran dalam penelitian
GPS (Global Positioning System)	Outdoor (Luar Ruangan)	Sangat Buruk (Tidak ada sinyal)	Signal Loss & Multipath Effect	Satelit	Tidak Bisa	Tidak Relevan
Marker-based AR (QR Code/Image)	Indoor & Outdoor	Sangat Tinggi (Hanya pada titik marker)	Jangkauan terbatas (kamera harus melihat marker)	Fisik (QR Code / Gambar)	Ya (Saat memindai ulang)	Pembandingan (Studi Terdahulu)
Visual Odometry (Non-SLAM)	Indoor & Outdoor	Sedang (Akurasi menurun seiring waktu)	Drift (Akumulasi Error)	Tidak Ada (Markerless)	Tidak Bisa (Open Loop)	Variabel Kontrol (Mode 1)
SLAM (Cloud Anchor)	Indoor & Outdoor	Tinggi (Konsisten &)	Mebutuhkan koneksi untuk	Tidak Ada (Peta	Bisa (Loop Closure)	Variabel Eksperimen

Kriteria	Lingkungan Operasional	Akurasi di dalam ruangan	Masalah Utama	Infrastruktur	Mekanisme Koreksi	Peran dalam penelitian
		Persisten)	inisialisasi peta	Digital / Cloud)		n (Mode 2)

Tabel 2. 2 Komparasi Karakteristik Metode Pelacakan

Berdasarkan tabel 2.2 diatas, analisis lebih mendalam mengenai relevansi setiap metode terhadap penelitian ini adalah sebagai berikut:

1. Keterbatasan GPS di dalam ruangan

Meskipun navigasi dengan menggunakan GPS merupakan standar navigasi global, teknologi ini tidak dapat diterapkan di dalam penelitian ini. Alkady et al. (2024) menjelaskan bahwa sinyal satelit frekuensi tinggi mengalami atenuasi (pelemahan) yang besar saat menembus atap beton dan dinding Gedung. Selain itu, fenomena seperti *multipath* yang dimana sinyal memantul pada permukaan Gedung sehingga menyebabkan kekacauan pembacaan posisi yang membuat GPS seringkali memiliki error hingga puluhan meter didalam ruangan.

2. Kendala Metode *Marker-based*

Metode berbasis penanda (*marker*) seperti QR Code dapat menawarkan akurasi tinggi, namun memiliki kelemahan dalam hal skalabilitas dan estetika. Pengguna hanya dapat mengetahui posisinya ketika kamera diarahkan langsung ke penanda fisiknya. Jika pengguna sudah mencapai tujuan, maka pengguna harus melakukan scan ulang agar dapat melakukan navigasi ke tempat lain. Oleh karena itu penelitian ini beralih ke metode *markerless* yang lebih fleksibel.

3. Visual Odometry sebagai Variabel Kontrol (Mode *Non-SLAM*)

Metode ini merupakan cara kerja standar dari sensor *smartphone* (ARCore) tanpa bantuan peta eksternal. Dalam tabel perbandingan, VO dikategorikan memiliki masalah “*Drift*” dan bersifat “*Open Loop*”. Artinya, sistem tidak memiliki mekanisme umpan balik untuk mengoreksi kesalahan. Pada penelitian ini, mode VO akan dijadikan Variabel Kontrol atau *baseline*. Tujuannya adalah untuk mengukur seberapa besar penyimpangan (error)

murni yang dihasilkan oleh sensor perangkat jika tidak dibantu oleh teknologi SLAM.

4. SLAM sebagai Solusi Utama

Metode ini dipilih sebagai variabel eksperimen atau solusi yang diusulkan. Keunggulan fundamental SLAM dibandingkan VO terletak pada kemampuan *Loop Closure* nya. Dengan menyimpan fitur visual lingkungan ke dalam *Cloud Anchor*, sistem akan memiliki ingatan spasial, yang dimana jika terjadi *drift* pada sensor VO, sistem SLAM akan dapat membandingkan visual saat ini dengan ingatan tersebut, lalu akan melakukan koreksi posisi (*snap*) secara otomatis. Inilah hipotesis utama dari penelitian, bahwa integrasi SLAM (*Cloud Anchor*) akan secara signifikan menurunkan nilai error (RMSE dan EPE) dibandingkan metode VO murni.

2.2.8 Parameter Evaluasi Akurasi

Agar dapat mengevaluasi performa sistem navigasi secara ilmiah dan objektif, penelitian ini akan menggunakan dua parameter statistik utama, yaitu *Root Mean Square Error* (RMSE) dan *End Position Error* (EPE). Penggunaan kedua parameter ini bertujuan untuk memberikan gambaran yang komprehensif mengenai stabilitas pelacakan (*Tracking Stability*) dan tingkat penyimpangan (*drift*) yang terjadi selama proses navigasi.

1. Root Mean Square Error (RMSE)

RMSE merupakan sebuah metode standar statistik yang digunakan untuk mengukur rata-rata besarnya kesalahan (error) antara nilai prediksi model dengan nilai observasi sebenarnya. Pada konteks navigasi, RMSE digunakan untuk menghitung rata-rata deviasi jalur virtual yang dihasilkan oleh aplikasi AR terhadap jalur fisik yang sebenarnya ditempuh pengguna sepanjang lintasan.

Rumus RMSE dapat didefinisikan sebagai berikut:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n ((X_{uji,i} - X_{gt,i})^2 + (Y_{uji,i} - Y_{gt,i})^2)} \quad (2.1)$$

Dimana:

- n : Jumlah total sampel data yang dicatat (waypoints).

- X_{uji}, Y_{uji} : Koordinat posisi yang dicatat oleh aplikasi navigasi AR pada titik ke-i.
- X_{gt}, Y_{gt} : Koordinat posisi referensi (*Ground Truth*) yang diukur secara manual pada titik ke-i.

Nilai RMSE yang rendah mengindikasikan bahwa jalur virtual menempel dengan konsisten pada jalur fisik, yang berarti akurasi pelacakan tinggi.

2. *End Position Error (EPE)*

Selain rata-rata kesalahan, penelitian ini juga secara spesifik mengukur *End Position Error (EPE)*. EPE didefinisikan sebagai jarak *Euclidean* antara titik akhir estimasi aplikasi dengan titik akhir sebenarnya di dunia nyata. Parameter ini sangat krusial pada penelitian komparasi SLAM vs *Non-SLAM* karena metode *Non-SLAM* (Visual Odometry) memiliki karakteristik kesalahan akumulatif (*accumulated drift*). Artinya, semakin jauh jarak tempuh, semakin besar penyimpangan yang terjadi. EPE menjadi indikator utama untuk membuktikan seberapa efektif metode SLAM dalam mengeleminasi akumulasi kesalahan tersebut di akhir perjalanan.

Rumus EPE didefinisikan sebagai berikut:

$$EPE = \sqrt{(X_{akhir_uji} - X_{akhir_gt})^2 + (Y_{akhir_uji} - Y_{akhir_gt})^2} \quad (2.2)$$

Dimana:

- $X_{akhir_uji}, Y_{akhir_uji}$: Koordinat titik terakhir yang dicatat aplikasi.
- $X_{akhir_gt}, Y_{akhir_gt}$: Koordinat titik terakhir yang sebenarnya (tujuan).

2.2.9 Metodologi Penelitian dan Pengembangan

Menurut Creswell & Creswell (2017), metodologi penelitian adalah pendekatan sistematis dan terstruktur yang digunakan untuk mengumpulkan, menganalisis, dan menginterpretasikan data guna menjawab pertanyaan penelitian dan mencapai tujuan penelitian. Pada konteks penelitian ilmu komputer, khususnya pada pengembangan sistem yang melibatkan integrasi perangkat keras dan lunak, metode yang diterapkan adalah *Research and Development (R&D)*.

Menurut Sugiyono (2019), metode R&D digunakan untuk menghasilkan produk tertentu dan menguji keefektifan produk tersebut. Pada penelitian ini, produk yang dihasilkan bukanlah perangkat lunak komersial, melainkan sebuah instrument

alat ukur berupa aplikasi navigasi *Augmented Reality* (AR) berbasis Android. Aplikasi ini berfungsi sebagai media utama dalam melaksanakan eksperimen komparasi akurasi antara metode SLAM (*cloud anchor*) dan *Non-SLAM* (Visual Odometry).

Untuk model pengembangan perangkat lunak, penelitian ini mengadopsi model Prototyping. Menurut Pressman (2015), model prototyping adalah sebuah pendekatan pengembangan perangkat lunak di mana purwarupa (prototype) dibangun, diuji, dan diperbaiki secara cepat dan berulang hingga memenuhi spesifikasi sistem. Model ini dipilih karena perilaku sensor AR sangat dipengaruhi oleh kondisi lingkungan fisik, sehingga memerlukan iterasi berulang untuk memastikan stabilitas sistem sebelum pengambilan data dilakukan.

Berikut ini merupakan tahapan-tahapan model prototyping yang diterapkan dalam penelitian ini:

1. Pengumpulan Kebutuhan (Communication)

Tahap ini mendefinisikan spesifikasi teknis agar aplikasi dapat berfungsi sebagai alat navigasi dan pencatat data yang valid. Kebutuhan yang diidentifikasi meliputi:

- **Fungsional:** Sistem harus mampu memindai lingkungan (*mapping*), menyimpan ID *Cloud Anchor*, melakukan relokalisasi posisi (*resolving*), serta pencatat log koordinat (x,y,z) ke dalam format CSV.
- **Perangkat Keras:** Spesifikasi perangkat menggunakan Xiaomi Redmi Note 11 Pro 5G yang mendukung ARCore Depth API.
- **Perangkat Lunak:** Penggunaan Unity Engine dan ARCore Extensions untuk akses API *Google Cloud*.

2. Perancangan Cepat (Quick Plan & Modeling)

Merancang arsitektur sistem dan antarmuka pengguna (UI) secara garis besar. Fokus utama di tahap ini adalah merancang skrip *ARNavigationManager.cs* yang menangani logika pertukaran data antara penyimpanan local dan layanan cloud ARCore, serta merancang struktur data CSV untuk keperluan analisis RMSE.

3. Pembentukan Prototipe (Construction of Prototype)

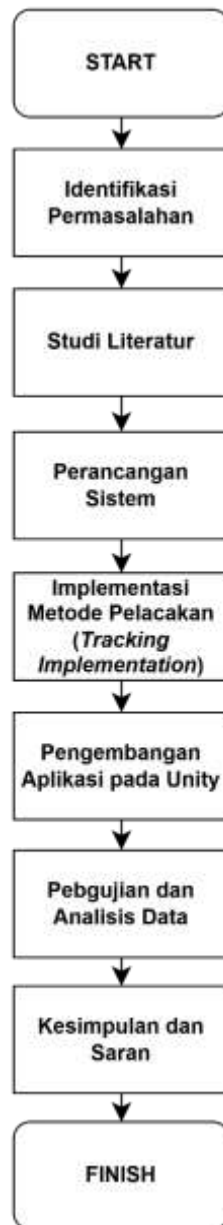
Tahap implementasi teknis di mana kode program ditulis serta dikompilasi ke dalam Unity Game Engine. Prototipe awal akan di-*build* menjadi APK untuk dipasang pada perangkat uji. Pada tahap ini, fitur *Hosting* dan *Resolving* diintegrasikan ke dalam tombol antarmuka aplikasi.

4. Evaluasi dan Penyerahan (Deployment, Delivery, & Feedback)

Prototipe akan diuji coba secara langsung di lokasi penelitian. Evaluasi difokuskan pada dua aspek: stabilitas teknis (apakah aplikasi crash atau tidak) dan validitas fungsi sensor (apakah drift terdeteksi). Jika ditemukan bug atau kegagalan fungsi sensor, maka aplikasi harus kembali ke tahap perancangan untuk melakukan perbaikan kode (debugging). Setelah prototipe dinyatakan stabil, aplikasi digunakan sebagai instrumen final untuk pengambilan data eksperimen lapangan.

2.2.10 Kerangka Penelitian

Kerangka penelitian memiliki fungsi sebagai bahan acuan konseptual dalam menggambarkan alur kegiatan penelitian yang dilakukan secara sistematis. Penelitian ini diawali dengan identifikasi masalah, yaitu ketidakakuratan sistem navigasi di dalam ruangan yang hanya mengandalkan sensor inersia (*Visual Odometry*) akibat fenomena drift. Sebagai solusi, dikembangkan sistem navigasi *Augmented Reality* (AR) yang menerapkan metode SLAM (*Simultaneous Localization and Mapping*) dengan menggunakan teknologi ARCore *Cloud Anchors* untuk dibandingkan akurasinya dengan metode *Non-SLAM*.



Gambar 2. 1 Kerangka Penelitian

Berikut ini merupakan rincian penjelasan pada gambar 2.1 mengenai kerangka penelitian:

1. Identifikasi Permasalahan

Tahapan ini dilakukan untuk menggali isu-isu teknis yang relevan dengan kondisi nyata, khususnya mengenai kegagalan sinyal GPS di dalam gedung bertingkat dan kelemahan metode pelacakan visual standar (Visual Odometry) yang rentan mengalami penyimpangan posisi (drift) seiring bertambahnya jarak tempuh pengguna.

2. Studi Literatur

Studi literatur dilakukan untuk memahami teori-teori fundamental yang mendasari penelitian, meliputi konsep *Augmented Reality*, prinsip kerja algoritma SLAM dan *Non-SLAM*, arsitektur ARCore *Cloud Anchors*, serta metode statistik *Root Mean Square Error* (RMSE) dan *End Position Error* (EPE) yang digunakan sebagai parameter ukur akurasi.

3. Perancangan Sistem

Pada tahap ini dilakukan desain arsitektur sistem navigasi secara menyeluruh. Hal ini mencakup perancangan diagram UML (Use Case dan Activity Diagram), desain antarmuka pengguna (UI), serta perancangan skenario eksperimen lapangan untuk membandingkan dua mode navigasi (SLAM dan *Non-SLAM*) pada rute yang sama.

4. Implementasi Metode Pelacakan (Tracking Implementation)

Tahap ini berfokus pada implementasi logika algoritma pelacakan ke dalam kode program. Implementasi mencakup integrasi ARCore *Cloud Anchors* API untuk melakukan fungsi *hosting* (pemetaan) dan *resolving* (relokalisasi/koreksi posisi) sebagai representasi metode SLAM. Selain itu, dibuat pula modul pencatatan data (data logging) untuk merekam koordinat posisi pengguna secara *real-time* ke dalam format CSV.

5. Pengembangan Aplikasi pada Unity

Proses ini mencakup pembangunan teknis (development) aplikasi menggunakan Unity Game Engine 2022. Pada tahap ini, seluruh komponen logika, aset visual jalur navigasi, dan antarmuka pengguna diintegrasikan menjadi satu kesatuan aplikasi (Build APK) yang siap dijalankan pada perangkat mobile Android.

6. Pengujian dan Analisis Data

Setelah aplikasi dikembangkan, dilakukan eksperimen lapangan untuk mengambil data koordinat pergerakan pengguna pada dua kondisi uji (SLAM vs *Non-SLAM*). Data yang terekam kemudian diolah menggunakan rumus RMSE dan EPE untuk mengukur tingkat akurasi dan besarnya penyimpangan drift yang terjadi, guna membuktikan hipotesis penelitian secara kuantitatif.

7. Kesimpulan dan Saran

Penelitian ditutup dengan penarikan kesimpulan berdasarkan hasil analisis data yang diperoleh, apakah metode SLAM terbukti efektif mengurangi error

navigasi, serta pemberian saran teknis untuk pengembangan sistem navigasi AR selanjutnya.