

Pengembangan Aplikasi *Monitoring* Kondisi Tanaman Berbasis *Markerless Augmented Reality* dengan Metode MDLC

Muhammad Gemilang Ramadhan^{1*}, Lita Karlitasari², Mohamad Iqbal Suriansyah³

Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Pakuan Bogor
Jl. Pakuan Tegallega, Kecamatan Bogor Tengah, Lota Bogor, Indonesia
edtgemilang@gmail.com¹, lita.karlitasari@unpak.ac.id², mohamad.iqbal@unpak.ac.id³

Submitted : 16/09/2025; Reviewed : 17/09/2025; Accepted : 31/10/2025; Published : 31/10/2025

Abstract

Augmented Reality (AR) has been increasingly applied in the agricultural sector to support monitoring and visualization of environmental data. However, most existing AR applications still rely on marker-based systems and have not yet integrated real-time sensor data. This study develops a markerless AR application for monitoring plant conditions using the Multimedia Development Life Cycle (MDLC) method. The development process includes conceptualization, design, material collection, assembly, testing, and distribution. The application was built using Unity and AR Foundation with a plane tracking approach to display plant condition information visually through text and digital indicators. Environmental sensor data were integrated via an ESP32 microcontroller; however, the main focus of this research is on data visualization rather than IoT control. The test results show that the system can display real-time data with good visual stability under optimal lighting conditions. This study demonstrates that the application of the MDLC method effectively supports the structured and efficient development of markerless AR applications for agricultural monitoring systems.

Keywords : application, augmented reality, IoT, markerless tracking, mdlc, monitoring,

Abstrak

Augmented Reality (AR) semakin banyak diterapkan dalam bidang pertanian untuk membantu proses monitoring dan visualisasi data lingkungan. Namun, sebagian besar aplikasi AR yang ada masih bergantung pada penggunaan marker dan belum banyak mengintegrasikan data sensor secara real-time. Penelitian ini mengembangkan aplikasi markerless AR untuk memantau kondisi tanaman dengan menerapkan metode Multimedia Development Life Cycle (MDLC). Tahapan pengembangan meliputi konseptualisasi, perancangan, pengumpulan material, perakitan, pengujian, dan distribusi. Aplikasi dibangun menggunakan Unity dan AR Foundation dengan pendekatan plane tracking untuk menampilkan informasi kondisi tanaman secara visual melalui teks dan angka digital. Data dari sensor lingkungan diintegrasikan melalui mikrokontroler ESP32, namun fokus penelitian ini adalah penyajian visual, bukan kontrol IoT. Hasil pengujian menunjukkan bahwa aplikasi mampu menampilkan data secara real-time dengan stabilitas visual yang baik pada kondisi pencahayaan optimal. Penelitian ini menunjukkan bahwa penerapan MDLC dapat mendukung pengembangan aplikasi markerless AR yang terstruktur, efisien, dan relevan untuk sistem monitoring pertanian.

Kata kunci : aplikasi, augmented reality, IoT, markerless tracking, mdlc, monitoring

1. Pendahuluan

Pertanian modern saat ini menghadapi tantangan besar dalam hal pemantauan kondisi tanaman secara akurat dan efisien. Faktor lingkungan seperti suhu dan kelembapan tanah memiliki pengaruh signifikan terhadap pertumbuhan tanaman, sehingga diperlukan sistem pemantauan yang mampu menyajikan informasi kondisi tanaman secara *real-time* dan mudah dipahami oleh pengguna [1]. Selama ini, sebagian besar petani masih mengandalkan metode manual dalam observasi kondisi tanaman, yang tidak hanya memakan waktu tetapi juga rentan terhadap kesalahan manusia [2].

Perkembangan teknologi *Augmented Reality* (AR) telah membuka peluang baru dalam bidang pertanian presisi (*precision farming*). AR memungkinkan visualisasi data sensor dalam bentuk digital yang terintegrasi langsung dengan lingkungan nyata, sehingga memudahkan pengguna dalam memahami kondisi tanaman tanpa harus melalui proses interpretasi data yang rumit [3], [4]. Lebih lanjut, penerapan *markerless AR*, khususnya dengan dukungan fitur *plane tracking*, memungkinkan penempatan objek

digital pada bidang nyata tanpa harus menggunakan penanda fisik, menjadikan sistem lebih fleksibel dan praktis dibandingkan pendekatan berbasis *marker* [5].

Sejumlah penelitian menunjukkan bahwa integrasi antara AR dan *Internet of Things* (IoT) mampu meningkatkan efektivitas dalam pemantauan lingkungan pertanian [6], [7]. Slimani *et al.* [1] mengembangkan sistem manajemen rumah kaca berbasis IoT, digital *twin*, dan AR untuk pengambilan keputusan secara optimal, sementara Ponnusamy *et al.* [4] merancang kerangka kerja AR–IoT untuk deteksi penyakit tanaman. Penelitian lain oleh Huong *et al.* [5] serta Pramudito *et al.* [6] menunjukkan bahwa integrasi *LoRaWAN*, pembelajaran mesin, dan AR dapat meningkatkan efisiensi sistem *monitoring* pertanian. Namun, sebagian besar penelitian tersebut masih berfokus pada aspek IoT dan integrasi data sensor, sedangkan aspek pengembangan perangkat lunak AR secara sistematis dan metodologis belum banyak dieksplorasi [8], [9], [11].

Dalam konteks pengembangan aplikasi AR, metode *Multimedia Development Life Cycle* (MDLC) terbukti memberikan alur kerja yang terstruktur mulai dari tahap konseptualisasi, perancangan, pengumpulan material, perakitan, hingga pengujian [13], [14]. Pendekatan ini tidak hanya memastikan dokumentasi yang sistematis, tetapi juga menjamin bahwa proses pengembangan dapat di replikasi dan dievaluasi secara ilmiah.

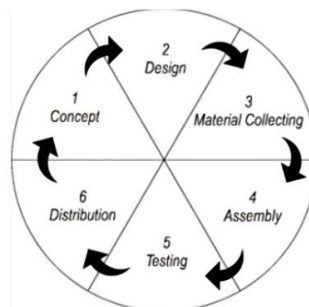
Berdasarkan identifikasi tersebut, penelitian ini berupaya mengisi celah penelitian pada pengembangan aplikasi *markerless* AR berbasis MDLC untuk sistem *monitoring* kondisi tanaman. Fokus utama penelitian ini bukan pada kontrol IoT, tetapi pada penyajian visual data sensor lingkungan secara *real-time* melalui tampilan teks dan angka digital yang sederhana, informatif, serta mudah dipahami. Dengan demikian, penelitian ini diharapkan memberikan kontribusi metodologis dalam pengembangan aplikasi AR pertanian yang tidak hanya interaktif, tetapi juga dikembangkan melalui pendekatan yang sistematis dan terukur.

2. Metodologi

Metode penelitian ini menggunakan pendekatan *Multimedia Development Life Cycle* (MDLC) yang terdiri atas enam tahap, yaitu *concept*, *design*, *material collecting*, *assembly*, *testing*, dan *distribution*. Setiap tahap saling berkaitan dalam proses pembangunan aplikasi berbasis *Augmented Reality* (AR).

Metode MDLC dipilih karena sesuai dengan karakteristik penelitian yang berfokus pada pengembangan aplikasi multimedia interaktif, di mana integrasi elemen visual, teks, dan objek digital menjadi komponen utama. Dibandingkan metode *Waterfall*, yang bersifat kaku dan linear, MDLC memungkinkan revisi pada setiap tahap tanpa harus mengulang keseluruhan proses [13]. Sementara itu, berbeda dari metode *Prototype* yang menitikberatkan pada iterasi desain, MDLC memberikan perhatian khusus pada pengumpulan dan perakitan konten multimedia, yang penting dalam pengembangan aplikasi AR [14].

Seperti dijelaskan oleh Kurniasari *et al.* [13] dan Roedavan *et al.* [14], MDLC dirancang khusus untuk pengembangan sistem multimedia yang menuntut konsistensi desain, efisiensi proses, serta dokumentasi yang sistematis. Oleh karena itu, metode ini dipandang paling tepat untuk mendukung pengembangan aplikasi *markerless* AR secara terstruktur dan efisien dalam konteks penelitian ini.



Gambar 1. Metode *Multimedia Development Life Cycle*

- A. *Concept*. Pada tahap ini, tujuan dan manfaat aplikasi multimedia dirumuskan. Proses ini mencakup identifikasi kebutuhan pengguna, aspek, penempatan tujuan pengembangan, serta pemilihan jenis perangkat lunak yang digunakan.
- B. *Design*. Dalam tahap ini, spesifikasi aplikasi mulai dirancang termasuk arsitektur program, gaya, tampilan dan kebutuhan perangkat lunak maupun keras.
- C. *Material Collecting*. Dalam tahap ini pengumpulan bahan yang sesuai dengan kebutuhan yang akan dilakukan.
- D. *Assembly*. Pada tahap adalah tahap di mana semua objek atau bahan *multimedia* dibuat berdasarkan desain yang telah direncanakan. Semua elemen dirakit menjadi satu kesatuan produk yang siap diuji.
- E. *Testing*. Tahap ini dilakukan untuk memastikan bahwa aplikasi yang telah dibuat sesuai dengan rancangan dan berfungsi dengan baik.
- F. *Distribution*. Pada titik ini, aplikasi di-deploy dan dieksekusi pada *smartphone* Android. Fase ini juga bisa disebut sebagai fase evaluasi pengembangan produk yang telah selesai untuk memperbaikinya.

3. Hasil dan Pembahasan

3.1 Hasil

Hasil penelitian ini berupa aplikasi *Augmented Reality* yang berfungsi untuk menampilkan data sensor IoT dalam bentuk panel teks digital secara *real-time*. Aplikasi dikembangkan menggunakan pendekatan *Multimedia Development Life Cycle* (MDLC) yang terdiri dari tahap berikut:

3.1.1. Concept

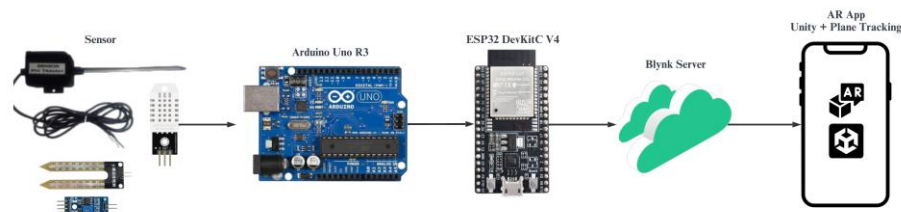
Pada tahap konsep, penelitian ini berfokus pada pengembangan aplikasi *Augmented Reality* sebagai media visualisasi data sensor IoT dalam bentuk panel teks digital. Ide utama yang melatarbelakangi penelitian adalah kebutuhan akan sistem *monitoring* lingkungan sederhana yang mampu menyajikan informasi secara *real-time*, sekaligus dapat divisualisasikan langsung dalam konteks dunia nyata melalui AR. Teknologi *markerless* AR dengan metode *plane tracking* dipilih agar panel digital dapat ditampilkan pada permukaan datar yang dikenali kamera perangkat, sehingga lebih praktis dibandingkan metode berbasis *marker*.

Tujuan dari aplikasi ini adalah menampilkan data suhu, kelembaban tanah, dan pH tanah yang diperoleh dari sensor IoT dalam bentuk teks angka digital pada panel AR. Data dari sensor DHT22, YL-69, dan sensor pH tanah pertama-tama diproses oleh *mikrokontroler Arduino Uno R3*, kemudian dikirim ke ESP32 untuk ditransmisikan ke *Blynk server*. Selanjutnya, aplikasi AR mengambil data dari server tersebut untuk ditampilkan secara *real-time*. Dengan pendekatan ini, fokus penelitian tidak pada kompleksitas grafis atau animasi, melainkan pada pengembangan media multimedia sederhana yang fungsional, interaktif, dan mampu memvisualisasikan data dengan jelas.

Kebutuhan utama dalam tahap konsep meliputi: perangkat keras (*Arduino Uno R3*, ESP32, sensor DHT22, sensor YL-69, dan sensor pH tanah), platform IoT (*Blynk server*), serta perangkat Android yang mendukung *ARCore* untuk menjalankan aplikasi *markerless* AR. Batasan penelitian ditetapkan agar lebih fokus, yaitu aplikasi hanya menampilkan tiga parameter (suhu, kelembaban tanah, dan pH tanah), visualisasi berupa panel teks digital tanpa animasi atau indikator warna, serta penggunaan metode *plane tracking* untuk menampilkan panel AR. Dengan batasan tersebut, penelitian ini diarahkan pada pengembangan aplikasi multimedia yang sederhana namun efektif untuk *monitoring* data lingkungan berbasis AR.

3.1.2. Design

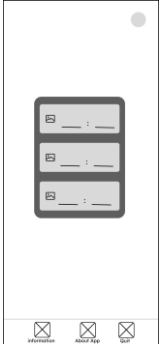
Tahap desain dilakukan untuk merancang alur sistem dan tampilan aplikasi. Alur sistem menggambarkan komunikasi data dari perangkat sensor hingga divisualisasikan melalui aplikasi AR. Sensor DHT22 (suhu udara), YL-69 (kelembaban tanah), dan sensor pH tanah dihubungkan ke mikrokontroler *Arduino Uno R3* untuk membaca dan mengolah data. Data selanjutnya dikirim melalui modul ESP32 menuju *Blynk server*. Aplikasi AR mengambil data tersebut secara *real-time* dan menampilkannya dalam bentuk panel teks digital menggunakan metode *markerless* AR (*plane tracking*). Alur sistem ditunjukkan pada Gambar 2.



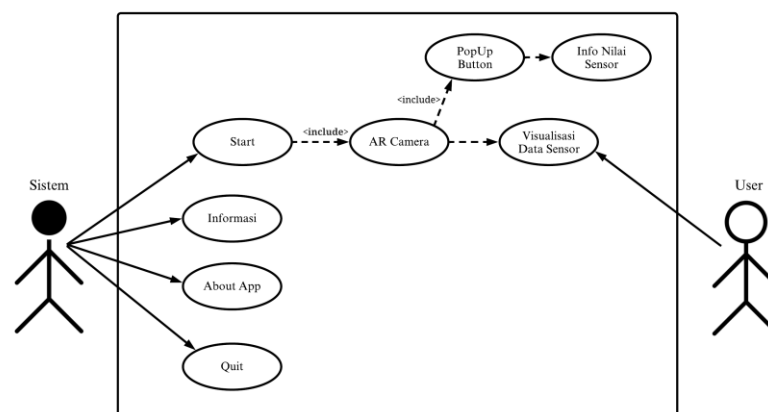
Gambar 2. Alur Sistem

Dari sisi visualisasi, aplikasi AR dirancang menampilkan panel digital sederhana berisi nilai suhu, kelembaban tanah, dan pH tanah. Panel muncul ketika kamera perangkat mendeteksi permukaan datar, tanpa memerlukan *marker* khusus. Tampilan panel difokuskan pada keterbacaan angka digital agar informasi dapat dipahami secara cepat. *Storyboard* interaksi pengguna ditunjukkan pada Tabel 1.

Tabel 1. Storyboard Aplikasi AR Monitoring Data Sensor

Desain	Keterangan
	Pada halaman ini <i>AR Camera</i> akan menangkap gambar dari lingkungan dengan cara melacak permukaan datar.
	Tampilan ini merupakan tampilan visual AR dengan elemen teks dan angka digital, visual ini akan muncul ketika <i>plane tracking</i> berhasil melacak permukaan datar pada area yang terdeteksi.

Selain *storyboard*, perancangan aplikasi juga digambarkan dalam bentuk *Use Case Diagram* untuk memperjelas interaksi antara pengguna dan sistem. Diagram ini menunjukkan bahwa pengguna dapat melakukan beberapa fungsi utama, seperti memulai aplikasi, melihat informasi, mengetahui tentang aplikasi (*about app*), dan keluar (*quit*). Pada saat pengguna memilih menu Start, sistem akan mengaktifkan kamera AR yang selanjutnya menampilkan panel digital berisi data sensor. Panel ini dapat menampilkan nilai suhu, kelembaban tanah, dan pH tanah melalui fitur visualisasi data sensor. Selain itu, terdapat pula tombol *pop-up* yang menyediakan informasi nilai sensor secara ringkas. *Use case diagram* ditunjukkan pada Gambar 3.



Gambar 3. Use Case Diagram

3.1.3. Material Collecting

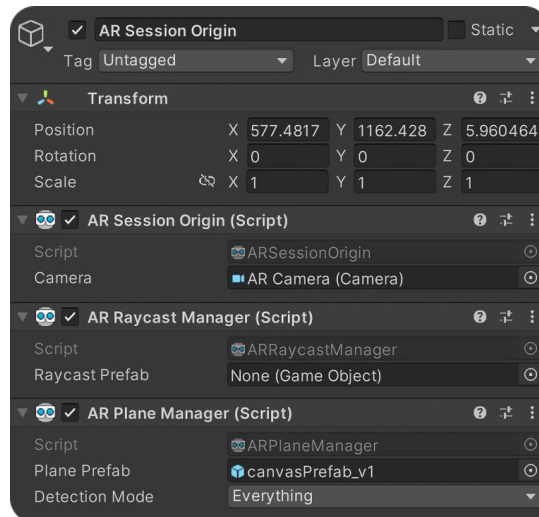
Pada tahap ini, bahan yang diperlukan untuk pengembangan aplikasi dikumpulkan, baik berupa perangkat keras maupun perangkat lunak. Perangkat keras terdiri dari sensor DHT22 untuk suhu udara, sensor YL-69 untuk kelembaban tanah, sensor pH tanah, mikrokontroler Arduino Uno R3 untuk pengolahan data, serta modul ESP32 untuk komunikasi dengan Blynk server. Dari sisi perangkat lunak, material yang digunakan meliputi Arduino IDE untuk kompilasi program pembacaan dan pengiriman data sensor, platform Blynk untuk penyimpanan serta pengiriman data secara *real-time*, *Unity Engine* dengan *AR Foundation* untuk pengembangan aplikasi AR berbasis markerless (plane tracking), serta *Visual Studio Code* sebagai lingkungan pengembangan aplikasi. Selain itu, material grafis berupa elemen panel teks digital sederhana juga dipersiapkan sebagai media visualisasi data sensor dalam aplikasi.

3.2 Pembahasan

3.2.1 Assembly

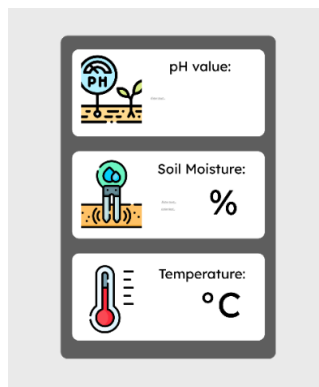
Tahap *assembly* dalam metode *Multimedia Development Life Cycle* (MDLC) merupakan proses penggabungan seluruh komponen yang telah dirancang dan dikumpulkan pada tahap sebelumnya. Pada tahap ini, desain antarmuka, elemen multimedia, serta modul pendukung mulai diintegrasikan dengan perangkat lunak pengembang agar membentuk sebuah aplikasi yang utuh.

Proses implementasi dimulai dengan pembuatan *AR Camera* pada *Unity Engine* menggunakan *AR Foundation*. *AR Camera* dikonfigurasi melalui komponen *AR Session Origin* yang terhubung dengan kamera perangkat Android. Untuk mendukung deteksi permukaan datar, ditambahkan *AR Raycast Manager* yang berfungsi mengarahkan *raycast* ke area lingkungan nyata, serta *AR Plane Manager* yang digunakan untuk mengatur *prefab* panel agar dapat muncul pada bidang datar yang terdeteksi. Pada tahap ini, *Unity* diatur agar panel digital hanya muncul ketika permukaan valid berhasil dikenali kamera, sehingga interaksi AR dapat berjalan secara stabil. Konfigurasi *AR camera* pada *Unity Engine* ditunjukkan pada Gambar 4.



Gambar 4. Konfigurasi AR Camera pada Unity Engine

Selanjutnya dibuat panel AR berupa UI *Canvas* yang berisi teks digital untuk menampilkan informasi suhu, kelembapan tanah, dan pH tanah. Panel dirancang sederhana tanpa animasi maupun indikator warna agar tetap fokus pada penyajian data sensor secara jelas dan efisien. *Design* panel AR untuk menampilkan data sensor ditunjukkan pada Gambar 5.



Gambar 5. Design Panel AR untuk Data Sensor

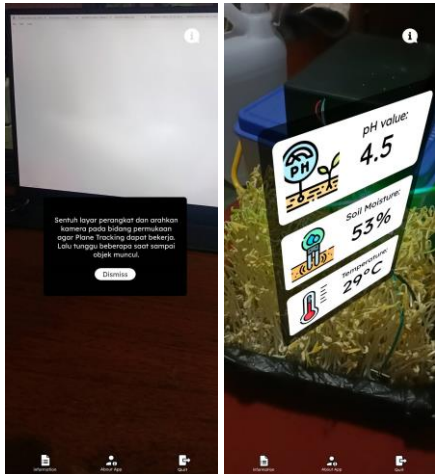
Tahap berikutnya adalah integrasi AR-IoT, yaitu menghubungkan aplikasi dengan perangkat keras sensor. Data dari sensor DHT22, YL-69, dan sensor pH tanah diolah oleh Arduino Uno R3 dan dikirimkan melalui ESP32 menuju *Blynk server*. Dari sisi aplikasi, *Unity* dihubungkan dengan *API Blynk* untuk menarik data tersebut dan menampilkannya secara *real-time* pada panel AR. Implementasi pengambilan data ditunjukkan pada Gambar 6.

```
IEnumerator GetDataFromBlynk()
{
    // URL untuk mendapatkan data dari Blynk
    string phUrl = "https://sgpl.blynk.cloud/external/api/get?token=G1AMr2uQhVJkvmH4Z407ppInqs8CkMrx&v1";
    UnityWebRequest phRequest = UnityWebRequest.Get(phUrl);
    yield return phRequest.SendWebRequest();
    if (phRequest.result == UnityWebRequest.Result.ConnectionError || phRequest.result == UnityWebRequest.Result.ProtocolError)
    {
        Debug.LogError(phRequest.error);
    }
    else
    {
        string phData = phRequest.downloadHandler.text.Trim(new char[] { '[', ']', ' ', '\n' });
        pH.text = phData;
    }
}
```

Gambar 6. Implementasi Pengambilan Data

Hasil akhir dari tahap *assembly* ditunjukkan pada tampilan aplikasi *Augmented Reality* yang telah berhasil dikembangkan. Aplikasi menampilkan panel digital berisi data sensor berupa suhu, kelembapan tanah, dan pH tanah secara *real-time*. Panel tersebut muncul secara otomatis pada bidang datar yang terdeteksi kamera perangkat melalui metode *markerless AR (plane tracking)*.

Panel dirancang sederhana dengan elemen teks digital yang bersih tanpa tambahan animasi maupun indikator warna, sehingga pengguna dapat langsung membaca informasi lingkungan secara jelas. Tampilan AR tetap stabil selama kamera mengunci permukaan datar, dan data diperbarui secara dinamis seiring perubahan nilai sensor. Dengan demikian, implementasi aplikasi telah menghasilkan visualisasi yang sesuai dengan tujuan penelitian, yaitu menghadirkan media *monitoring* berbasis multimedia yang informatif dan interaktif. Hasil implementasi aplikasi AR ditampilkan pada Gambar 7.



Gambar 7. Hasil Implementasi Aplikasi AR

3.2.2 Testing

Pada tahap testing, pengujian dilakukan untuk memastikan aplikasi berjalan sesuai dengan rancangan serta dapat digunakan secara optimal oleh pengguna. Beberapa metode pengujian yang digunakan meliputi *black box testing*, pengujian akurasi data, pengujian stabilitas AR (*plane tracking*), pengujian sudut pandang, dan uji kompatibilitas perangkat.

Pengujian *Black Box* dilakukan untuk memverifikasi setiap fungsi utama aplikasi berdasarkan interaksi pengguna. Hasil pengujian menunjukkan bahwa semua menu utama seperti *Start*, *Informasi*, *About App*, dan *Quit* dapat dijalankan sesuai dengan skenario. Selain itu, pada menu *Start*, kamera AR dapat aktif, menampilkan panel teks *digital*, serta menampilkan data sensor yang diambil dari *Blynk server* secara *real-time* tanpa mengalami *error*. Pengujian *black box* ditunjukkan pada Tabel 2.

Tabel 2. Pengujian Black Box

No	Test Case	Input/Action	Hasil yang Diharapkan	Status
1	Membuka aplikasi	Ketuk ikon aplikasi	Aplikasi terbuka dan menampilkan layar utama	Lulus
2	Mulai kamera AR	Ketuk tombol “START”	Kamera terbuka dan mulai melacak permukaan	Lulus
3	Melacak permukaan	Arahkan kamera ke permukaan datar	Permukaan terdeteksi; siap untuk penempatan objek AR	Lulus
4	Visualisasi data pada AR	Permukaan terdeteksi	Nilai sensor ditampilkan dalam <i>overlay</i> AR	Lulus
5	Buka menu informasi	Ketuk menu informasi	Membuka dan menampilkan halaman informasi	Lulus
6	Buka menu aplikasi	Ketuk menu <i>About the App</i>	Membuka dan menampilkan halaman tentang aplikasi	Lulus
7	Keluar aplikasi	Ketuk menu <i>Quit</i>	Keluar dari aplikasi	Lulus

Pengujian akurasi data dilakukan untuk mengevaluasi data yang dikirim oleh sensor melalui aplikasi AR, dengan cara membandingkan dengan hasil pembacaan sensor dengan alat ukur standar. Pengujian ditunjukkan pada tabel 3.

Tabel 3. *Pengujian Akurasi Data*

Parameter	Alat Pembanding	Sensor IoT	Selisih	Akurasi (%)
Suhu (°C)	29.5	29.2	0.3	98.9
Kelembapan (%)	55	53	2	96.4
pH Tanah	5.0	4.9	0.1	98.5

Berdasarkan Tabel 3, hasil pengujian mendapatkan nilai akurasi di atas 95% yang menunjukkan sistem dapat diandalkan untuk monitoring sederhana.

Pengujian stabilitas AR (*Plane Tracking*) dilakukan untuk mengevaluasi kemampuan *plane tracking* dalam mendeteksi permukaan dan mempertahankan panel AR pada kondisi pencahayaan berbeda. Untuk setiap kondisi dilakukan 10 percobaan. Pengujian ditunjukkan pada Tabel 4.

Tabel 4. *Pengujian Stabilitas AR (Plane Tracking)*

Kondisi Pencahayaan	Jumlah Percobaan	Keberhasilan Deteksi (hasil)	Persentase Keberhasilan (%)	Rentang Waktu Deteksi (s)	Stabilitas Pelacakan
Terang (<i>Indoor</i> lampu/ siang hari)	10	10	100%	1-3	Stabil
Redup (cahaya minim)	10	7	70%	3-8	Cukup stabil
Outdoor (terik matahari langsung)	10	9	90%	2-5	stabil

Hasil pengujian pada Tabel 4, menunjukkan bahwa kondisi pencahayaan berpengaruh signifikan terhadap stabilitas AR. Pada pencahayaan terang, sistem mampu mendeteksi permukaan dengan 100% keberhasilan dan waktu deteksi yang sangat cepat (1–3 detik). Pada kondisi redup, tingkat keberhasilan menurun menjadi 70% dengan waktu deteksi lebih lama, yaitu 3–8 detik. Sementara itu, pada kondisi luar ruangan dengan cahaya matahari langsung, sistem tetap stabil dengan keberhasilan 90% dan rentang deteksi 2–5 detik. Dengan demikian, penggunaan aplikasi paling optimal dilakukan pada kondisi pencahayaan yang cukup terang, baik di dalam maupun luar ruangan.

Pengujian sudut pandang dilakukan untuk mengetahui sejauh mana visualisasi panel AR tetap terbaca dari berbagai arah pandang. Pengujian ditunjukkan pada Tabel 5.

Tabel 5. *Pengujian Sudut Pandang dan Jarak Ideal*

Jarak (cm)	Jumlah Percobaan	Terdeteksi (kali)	Persentase Keberhasilan (%)	Rata-rata Delay (detik)
0 – 49	10	0	0 %	–
50 – 100	10	9	90 %	5.1
101 – 150	10	7	70 %	14.8
151 – 200	10	0	0 %	–
≥ 200	10	0	0 %	–

Berdasarkan Tabel 5, Hasil pengujian menunjukkan bahwa jarak ideal untuk deteksi visual AR dengan metode *markerless (plane tracking)* berada pada rentang 50–100 cm dengan tingkat keberhasilan deteksi mencapai 90% dan rata-rata *delay* sekitar 5 detik. Pada jarak 101–150 cm, keberhasilan menurun menjadi 70% dengan *delay* lebih tinggi, yaitu sekitar 15 detik. Sementara itu, pada jarak di bawah 50 cm maupun di atas 150 cm, sistem gagal mendeteksi permukaan. Temuan ini menegaskan bahwa kinerja aplikasi

paling optimal digunakan pada jarak sedang, khususnya sekitar satu meter dari permukaan datar yang terdeteksi.

Terakhir, uji kompatibilitas perangkat dilakukan untuk memastikan aplikasi dapat berjalan pada berbagai jenis perangkat Android yang mendukung *ARCore*. Aplikasi diuji pada beberapa *smartphone* dengan spesifikasi berbeda. Pengujian ditunjukkan pada Tabel 6.

Tabel 6. Pengujian Kompabilitas

Perangkat	Versi Android	FPS Rata-rata	RAM	Hasil
High-end (8 GB)	Android 12	30 FPS	8 GB	Sangat Lancar
Mid-end (4 GB)	Android 11	24 FPS	4 GB	Sebagian kompatibel (lag, loading lama)
Low-end (2 GB)	Android 10	12 FPS	2 GB	Tidak kompatibel

Berdasarkan Tabel 6, pengujian memperlihatkan bahwa aplikasi dapat berjalan lancar pada perangkat dengan RAM minimal 3 GB dan prosesor menengah ke atas, sedangkan pada perangkat dengan spesifikasi rendah aplikasi mengalami jeda (*lag*) saat memuat data AR.

4. Kesimpulan

Penelitian ini berhasil mengembangkan aplikasi *monitoring* berbasis *Augmented Reality* (AR) menggunakan metode *Multimedia Development Life Cycle* (MDLC) yang terintegrasi dengan data sensor IoT melalui *Blynk server*. Aplikasi mampu menampilkan informasi suhu, kelembapan tanah, dan pH tanah secara *real-time* dalam bentuk panel digital sederhana menggunakan *markerless AR* (*plane tracking*) pada perangkat Android. Hasil pengujian menunjukkan seluruh fungsi utama berjalan sesuai harapan melalui uji *black box*, dengan visualisasi data yang akurat dan tampilan AR yang stabil pada kondisi pencahayaan optimal.

Secara akademis, penelitian ini memberikan kontribusi metodologis melalui penerapan MDLC dalam pengembangan aplikasi *markerless AR* secara terstruktur, serta kontribusi praktis berupa media interaktif untuk mendukung sistem *monitoring* pertanian berbasis IoT. Namun, penelitian ini masih memiliki keterbatasan pada aspek kompatibilitas perangkat *ARCore*, variasi kondisi lingkungan uji, serta bentuk visualisasi data yang masih sederhana.

Ke depan, penelitian dapat dikembangkan dengan menambahkan fitur visualisasi dinamis seperti grafik perubahan data, indikator warna berbasis kondisi lingkungan, serta uji kelayakan pengguna agar sistem dapat diterapkan secara lebih luas pada konsep pertanian cerdas (*smart farming*).

Daftar Pustaka

- [1] H. Slimani, J. El Mhamdi, and A. Jilbab, "Real-time greenhouse management using IoT, digital twin, and augmented reality for optimal control and decision making," *Acta IMEKO*, vol. 14, no. 2, pp. 1–8, Jun. 2024.
- [2] W. Hurst, F. R. Mendoza, and B. Tekinerdogan, "Augmented Reality in Precision Farming: Concepts and Applications," *Smart Cities*, vol. 4, no. 4, pp. 1454–1468, Dec. 2021.
- [3] P. Phupattanasilp and S. R. Tong, "Augmented Reality in the Integrative Internet of Things (AR IoT): Application for Precision Farming," *Sustainability*, vol. 11, no. 9, art. 2658, May 2020.
- [4] P. Ponnusamy, S. Natarajan, N. Ramasamy, C. Clement, P. Rajalingam, and M. Mitsunori, "An IoT-enabled augmented reality framework for plant disease detection," *Revista Investigación Operacional Electrónica y de Ingeniería*, vol. 13, no. 3, pp. 185–192, 2021.
- [5] D. T. Huong, N. T. H. Duy, P. V. M. Tu, H. H. Hanh, and K. Yamada, "Enhancing precision agriculture: An IoT-based smart monitoring system integrated with LoRaWAN, ML, and AR," *EAI Endorsed Transactions on Smart Cities*, vol. 4, no. 24, pp. 1–10, 2024.

- [6] A. Pramudito, S. K. Dewi, A. Rachmad, and H. Prasetyo, "Real-time temperature and humidity monitoring with augmented reality interface," *Technologies*, vol. 12, no. 1, art. 14, Jan. 2024.
- [7] L. G. Chavan, R. S. Shaikh, S. S. Chaudhari, V. R. Irani, and S. B. Shinde, "Flora monitoring and visualization using augmented reality (AR) and Internet of Things (IoT)," *International Journal for Scientific Research & Development (IJSRD)*, vol. 10, no. 3, pp. 1–5, 2022.
- [8] R. Awang, M. Nur, and S. Basri, "Monitoring panel listrik berbasis IoT dan augmented reality," *Prosiding Seminar Nasional Penelitian dan Pengabdian Masyarakat (SNP2M) Politeknik Negeri Ujung Pandang*, pp. 12–18, 2023.
- [9] R. Winata, T. Saputra, and D. R. Putra, "Sistem monitoring limbah septic tank berbasis augmented reality dan IoT," *Jurnal Teknologi dan Sistem Komputer*, vol. 11, no. 3, pp. 210–218, 2023.
- [10] C. Mucchiani, D. Chatziparaschis, and K. Karydis, "Augmented reality enabled crop monitoring with robot assistance," *arXiv preprint*, arXiv:2411.03483, Nov. 2024. [Online]. Available: <https://arxiv.org/abs/2411.03483>
- [11] R. Yauri and G. Mallqui, "IoT control and visualization system with digital twins and augmented reality in a digital transformation space," *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 20, no. 4, pp. 1–14, 2024.
- [12] U. Eswaran and V. Eswaran, "Augmented reality and Internet of Things: A review of convergence, applications and challenges," *i-manager's Journal on Augmented & Virtual Reality*, vol. 1, no. 2, pp. 41–52, 2023.
- [13] A. A. Kurniasari, T. D. Puspitasari, and A. D. S. Mutiara, "Application of the Multimedia Development Life Cycle (MDLC) method on a magical augmented reality book based on Android," *Antivirus: Jurnal Ilmiah Teknik Informatika*, vol. 17, no. 1, pp. 19–31, May 2023. [Online]. Available: <https://doi.org/10.35457/antivirus.v17i1.2801>
- [14] R. Roedavan, A. P. Leman, and S. G. Putri, "A framework for developing augmented reality applications based on the Multimedia Development Life Cycle (MDLC)," *Jurnal Ilmiah Informatika Global*, vol. 16, no. 1, pp. 20–27, Apr. 2025. [Online]. Available: <https://doi.org/10.36982/jiig.v16i1.5183>