

**PEMBUATAN APLIKASI *MOBILE* UNTUK PENGAMANAN  
DAN VERIFIKASI DOKUMEN DIGITAL DAN FISIK  
BERBASIS *WATERMARK*, *METADATA* DAN  
STEGANOGRAFI**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh kelulusan  
Jenjang Strata Satu (S1)  
Pada program Studi Sistem Informasi**

**Oleh  
Ujang Maryanto  
351941004**



**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
INDONESIA MANDIRI  
2025**

**PEMBUATAN APLIKASI *MOBILE* UNTUK PENGAMANAN  
DAN VERIFIKASI DOKUMEN DIGITAL DAN FISIK  
BERBASIS *WATERMARK*, *METADATA* DAN  
STEGANOGRAFI**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh kelulusan  
Jenjang Strata Satu (S1)  
Pada program Studi Sistem Informasi**

**Oleh  
Ujang Maryanto  
351941004**



**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
INDONESIA MANDIRI**

**2025**

**LEMBAR PENGESAHAN**

**PEMBUATAN APLIKASI *MOBILE* UNTUK PENGAMANAN  
DAN VERIFIKASI DOKUMEN DIGITAL DAN FISIK  
BERBASIS *WATERMARK, METADATA* DAN  
STEGANOGRAFI**

***DEVELOPMENT OF A MOBILE APPLICATION FOR SECURING AND  
VERIFYING DIGITAL AND PHYSICAL DOCUMENTS BASED ON  
WATERMARKING, METADATA, AND STEGANOGRAPHY***

Oleh  
Ujang Maryanto  
351941004

Tugas Akhir ini telah diterima dan disahkan untuk  
memenuhi persyaratan mencapai gelar

**SARJANA SISTEM INFORMASI**

Pada  
**PROGRAM STUDI SISTEM INFORMASI  
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
INDONESIA MANDIRI**

Bandung, Juni 2025  
Disahkan oleh



Ketua Program Studi,

Harvoso Wicaksono, S.Si., M.M., M.Kom.  
NIDN. 437300020

Dosen Pembimbing,

Patah Herwanto, S.T., M.Kom.  
NIDN. 002710701

## LEMBAR PERSETUJUAN REVISI

### PEMBUATAN APLIKASI *MOBILE* UNTUK PENGAMANAN DAN VERIFIKASI DOKUMEN DIGITAL DAN FISIK BERBASIS *WATERMARK*, *METADATA* DAN STEGANOGRAFI

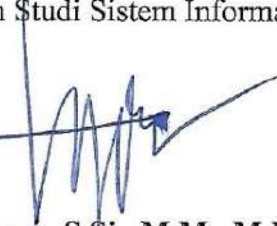
Oleh  
Ujang Maryanto  
351941004

Telah melakukan sidang tugas akhir dan telah melakukan revisi sesuai dengan perubahan dan perbaikan yang diminta pada saat sidang tugas akhir.

Bandung, Juni 2025  
Menyetujui

| No | Nama Dosen                        | Keterangan | Tanda Tangan                                                                          |
|----|-----------------------------------|------------|---------------------------------------------------------------------------------------|
| 1. | Patah Herwanto, S.T., M.Kom.      | Pembimbing |  |
| 2. | Yudhi W. Arthana R., S.T., M.Kom. | Penguji 1  |  |
| 3. | Dr. Chairuddin, M.M., M.T.        | Penguji 2  |  |

Mengetahui  
Ketua Program Studi Sistem Informasi



Haryoso Wicaksono, S.Si., M.M., M.Kom.  
NIDN. 437300020

## SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa:

- (1) Naskah skripsi ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik, baik di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri maupun perguruan tinggi lainnya.
- (2) Skripsi ini murni merupakan karya penelitian saya sendiri dan tidak menjiplak karya pihak lain. Dalam hal ada bantuan atau arahan dari pihak lain maka telah saya sebutkan identitas dan jenis bantuannya di dalam lembar ucapan terima kasih.
- (3) Seandainya ada karya pihak lain yang ternyata memiliki kemiripan dengan karya saya ini, maka hal ini adalah di luar pengetahuan saya dan terjadi tanpa kesengajaan dari pihak saya.

Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terbukti adanya kebohongan dalam pernyataan ini, maka saya bersedia menerima sanksi akademik sesuai norma yang berlaku di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri.

Bandung, Juni 2025  
Yang membuat pernyataan



Ujang Maryanto  
351941004

## ABSTRAK

# PEMBUATAN APLIKASI *MOBILE* UNTUK PENGAMANAN DAN VERIFIKASI DOKUMEN DIGITAL DAN FISIK BERBASIS *WATERMARK*, *METADATA* DAN STEGANOGRAFI

Oleh  
Ujang Maryanto  
351941004

Penelitian ini bertujuan untuk merancang dan membangun aplikasi *mobile* yang dapat digunakan untuk mengamankan dokumen digital maupun fisik dari risiko pemalsuan dan penyalahgunaan. Sistem yang dikembangkan menerapkan teknologi *watermark*, *metadata* dan steganografi untuk memberikan lapisan pengamanan tambahan pada dokumen. Metode pengumpulan data dilakukan melalui observasi, wawancara, dan studi literatur guna memahami kebutuhan pengguna serta potensi penyalahgunaan dokumen. Aplikasi ini memungkinkan pengguna untuk menandai dokumen dengan informasi pemilik dan penerima, baik dalam bentuk *softcopy* maupun *hardcopy*, serta memverifikasi keasliannya melalui pemindaian atau unggahan *file*. Hasil implementasi menunjukkan bahwa sistem mampu meningkatkan keamanan dokumen secara signifikan dan memberikan kemudahan dalam proses verifikasi. Dengan pendekatan ini, diharapkan masyarakat dapat lebih terlindungi dari berbagai bentuk kejahatan dokumen yang semakin marak.

**Kata Kunci:** Pengamanan Dokumen, *Watermark*, *Metadata*, Steganografi, Aplikasi *Mobile*, Verifikasi Keaslian.

## **ABSTRACT**

# **DEVELOPMENT OF A MOBILE APPLICATION FOR SECURING AND VERIFYING DIGITAL AND PHYSICAL DOCUMENTS BASED ON WATERMARKING, METADATA, AND STEGANOGRAPHY**

By  
Ujang Maryanto  
351941004

*This research aims to design and develop a mobile application that secures both digital and physical documents from the risks of forgery and misuse. The developed system implements watermarking, metadata and steganography technologies to provide an additional layer of document protection. Data collection methods include observation, interviews, and literature studies to understand user needs and the potential for document misuse. The application enables users to embed ownership and recipient information into documents, both in softcopy and hardcopy formats, and verify their authenticity through scanning or file upload. The implementation results show that the system significantly enhances document security and simplifies the verification process. With this approach, the public is expected to be better protected against the increasing threats of document-related crimes.*

**Keywords:** Document Security, Watermark, Metadata, Steganography, Mobile Application, Authenticity Verification

## Kata Pengantar

Seiring dengan terselesaikannya penyusunan skripsi ini, saya menyampaikan rasa syukur ke hadirat Allah SWT yang telah memberikan rahmat, kekuatan, dan kelancaran sehingga penelitian ini dapat diselesaikan dengan baik.

Skripsi dengan judul **“Pembuatan Aplikasi Mobile untuk Pengamanan dan Verifikasi Dokumen Digital dan Fisik Berbasis *Watermark*, *Metadata* dan *Steganografi*”** ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana di Program Studi Sistem Informasi STMIK Indonesia Mandiri. Penelitian ini lahir dari keprihatinan atas maraknya pemalsuan dokumen dan penyalahgunaan data, sehingga diharapkan mampu memberikan solusi digital yang bermanfaat bagi masyarakat.

Saya mengucapkan terima kasih yang sebesar-besarnya kepada:

- Bapak Patah Herwanto, S.T., M.Kom, selaku dosen pembimbing yang telah membimbing dengan penuh kesabaran, memberikan arahan, dan mendampingi penulis selama proses penyusunan skripsi ini.
- Pihak-pihak yang telah bersedia memberikan data, wawasan, atau dukungan dalam pengujian dan validasi sistem, yang tidak dapat disebutkan satu per satu, namun memiliki peran penting dalam kelancaran penelitian ini.
- Pimpinan dan seluruh dosen Program Studi Sistem Informasi STMIK Indonesia Mandiri yang telah menjadi sumber ilmu dan inspirasi selama masa perkuliahan.

Ucapan terima kasih juga saya sampaikan kepada semua pihak yang secara langsung maupun tidak langsung telah membantu dalam proses penyusunan skripsi ini.

Saya berharap, skripsi ini dapat memberikan kontribusi nyata dalam bidang pengamanan dokumen digital dan fisik serta menjadi pijakan bagi penelitian lanjutan di masa depan.

Bandung,      Juni 2025  
Program Studi Sistem Informasi

Ujang Maryanto  
NIM. 351941004

## Daftar Isi

|                                                        |     |
|--------------------------------------------------------|-----|
| LEMBAR PENGESAHAN .....                                | i   |
| LEMBAR PERSETUJUAN REVISI.....                         | ii  |
| SURAT PERNYATAAN .....                                 | iii |
| ABSTRAK.....                                           | iv  |
| ABSTRCT.....                                           | v   |
| Kata Pengantar .....                                   | vi  |
| Daftar Isi .....                                       | vii |
| Daftar Tabel .....                                     | ix  |
| Daftar Gambar.....                                     | x   |
| PENDAHULUAN .....                                      | 1   |
| 1.1. Latar Belakang .....                              | 1   |
| 1.2. Identifikasi Masalah .....                        | 2   |
| 1.3. Tujuan Penulisan.....                             | 2   |
| 1.4. Batasan Masalah .....                             | 2   |
| 1.5. Metode Penelitian .....                           | 3   |
| 1.5.1. Analisis Kebutuhan (requirement analysis) ..... | 3   |
| 1.5.2. Perancangan Sistem (system design) .....        | 4   |
| 1.5.3. Implementasi (implementation) .....             | 4   |
| 1.5.4. Pengujian (testing) .....                       | 5   |
| 1.5.5. Pemeliharaan (maintenance) .....                | 5   |
| 1.6. Sistematika Penulisan .....                       | 5   |
| LANDASAN TEORI.....                                    | 8   |
| 2.1. Pengertian Sistem Informasi .....                 | 8   |
| 2.2. Pengertian Aplikasi .....                         | 9   |
| 2.4.1. Aplikasi Mobile.....                            | 9   |
| 2.4.2. Aplikasi Android.....                           | 10  |
| 2.4.3. Aplikasi iOS.....                               | 11  |
| 2.4.4. Aplikasi Hybrid.....                            | 12  |
| 2.4.5. Aplikasi Web.....                               | 13  |
| 2.3. Pengertian Keamanan.....                          | 14  |
| 2.3.1. Watermark.....                                  | 16  |
| 2.3.2. Metadata.....                                   | 16  |
| 2.3.3. Steganografi .....                              | 17  |
| 2.4. Bahasa Pemrograman .....                          | 18  |
| 2.4.1. Flutter .....                                   | 18  |
| 2.4.2. Dart.....                                       | 20  |
| 2.4.3. Javascript.....                                 | 21  |
| 2.4.4. Typescript.....                                 | 22  |
| 2.4.5. NodeJS .....                                    | 23  |
| 2.4.6. ReactJS .....                                   | 24  |
| 2.4.7. AdonisJS .....                                  | 25  |

|                                                |                                                |     |
|------------------------------------------------|------------------------------------------------|-----|
| 2.5.                                           | Pengertian MinIO .....                         | 28  |
| 2.6.                                           | Pengertian Server .....                        | 28  |
| 2.7.                                           | Pengertian Microservices .....                 | 30  |
| 2.8.                                           | Pengertian Docker .....                        | 31  |
| 2.9.                                           | Pengertian Database .....                      | 32  |
| 2.10.                                          | Pengertian UML .....                           | 33  |
| 2.11.                                          | Diagram-Diagram UML .....                      | 33  |
| 2.11.1.                                        | Use Case Diagram .....                         | 34  |
| 2.11.2.                                        | Class Diagram .....                            | 34  |
| 2.11.3.                                        | Sequence Diagram .....                         | 34  |
| 2.11.4.                                        | Activity Diagram .....                         | 34  |
| 2.12.                                          | Penelitian .....                               | 35  |
| ANALISIS MASALAH DAN PERANCANGAN PROGRAM ..... |                                                | 37  |
| 3.1.                                           | Pengumpulan Kebutuhan (requirements) .....     | 37  |
| 3.1.1.                                         | Metode Pengumpulan Data .....                  | 37  |
| 3.1.2.                                         | Analisis Kebutuhan Sistem .....                | 44  |
| 3.2.                                           | Usulan Program .....                           | 45  |
| 3.2.1.                                         | Tujuan Program .....                           | 46  |
| 3.2.2.                                         | Fitur yang Diusulkan .....                     | 46  |
| 3.2.3.                                         | Flowchart Usulan Program .....                 | 48  |
| 3.2.4.                                         | Deskripsi Singkat Aplikasi .....               | 49  |
| 3.3.                                           | Desain Sistem perangkat lunak .....            | 49  |
| 3.3.1.                                         | Desain Use Case Diagram .....                  | 50  |
| 3.3.2.                                         | Desain Activity Diagram .....                  | 53  |
| 3.3.3.                                         | Desain Sequence Diagram .....                  | 56  |
| 3.3.4.                                         | Desain Class Diagram .....                     | 59  |
| 3.3.5.                                         | Desain Entity Relationship Diagram (ERD) ..... | 60  |
| 3.3.6.                                         | Struktur Basis Data .....                      | 60  |
| 3.3.7.                                         | Desain Interface yang Dibangun .....           | 80  |
| IMPLEMENTASI DAN UJI COBA .....                |                                                | 86  |
| 4.1.                                           | Implementasi (implementation) .....            | 86  |
| 4.1.1.                                         | Arsitektur Sistem .....                        | 86  |
| 4.1.2.                                         | Implementasi Tampilan Antarmuka .....          | 96  |
| 4.2.                                           | Pengujian Sistem (verification) .....          | 107 |
| 4.2.1.                                         | Fokus Pengujian .....                          | 108 |
| 4.2.2.                                         | Proses Pengujian .....                         | 109 |
| PENUTUP .....                                  |                                                | 120 |
| 5.1.                                           | Penutup .....                                  | 120 |
| 5.2.                                           | Saran .....                                    | 121 |
| DAFTAR PUSTAKA .....                           |                                                | 123 |
| LAMPIRAN .....                                 |                                                | 129 |

## Daftar Tabel

|                                                             |     |
|-------------------------------------------------------------|-----|
| Table 3.1. Wawancara.....                                   | 38  |
| Table 3.2. Study Literatur .....                            | 39  |
| Table 3.3. Struktur Field Tabel Basis Data .....            | 62  |
| Table 3.4. Tabel applications .....                         | 62  |
| Table 3.5. Tabel client_activities .....                    | 63  |
| Table 3.6. Tabel client_tokens .....                        | 63  |
| Table 3.7. Tabel clients .....                              | 64  |
| Table 3.8. Tabel services.....                              | 65  |
| Table 3.9. Tabel user_aliases .....                         | 65  |
| Table 3.10. Tabel user_attemtps .....                       | 66  |
| Table 3.11. Tabel user_passwords .....                      | 66  |
| Table 3.12. Tabel user_recoveries .....                     | 67  |
| Table 3.13. Tabel users .....                               | 67  |
| Table 3.14. Tabel language_contents.....                    | 68  |
| Table 3.15. Tabel languages.....                            | 69  |
| Table 3.16. Tabel masters .....                             | 69  |
| Table 3.17. Tabel configs.....                              | 70  |
| Table 3.18. Tabel galleries .....                           | 71  |
| Table 3.19. Tabel gallery_datas .....                       | 71  |
| Table 3.20. Tabel logs.....                                 | 72  |
| Table 3.21. Tabel profile_hobbies .....                     | 72  |
| Table 3.22. Tabel profile_interests.....                    | 73  |
| Table 3.23. Tabel profile_socials .....                     | 73  |
| Table 3.24. Tabel profiles .....                            | 74  |
| Table 3.25. Tabel apps .....                                | 75  |
| Table 3.26. Tabel onboardings.....                          | 75  |
| Table 3.27. Tabel versions .....                            | 76  |
| Table 3.28. Tabel doc_files.....                            | 77  |
| Table 3.29. Tabel doc_histories .....                       | 77  |
| Table 3.30. Tabel doc_originals.....                        | 78  |
| Table 3.31. Tabel doc_pages.....                            | 79  |
| Table 3.32. Tabel docs .....                                | 79  |
| Table 4.31. Tabel Pengujian Fungsional Mobile .....         | 110 |
| Table 4.32. Kesimpulan Pengujian Endpoint Login .....       | 114 |
| Table 4.33. Kesimpulan Umum Pengujian Endpoint Proses ..... | 117 |

## Daftar Gambar

|                                                                           |     |
|---------------------------------------------------------------------------|-----|
| Gambar 2.1. Aplikasi Hybrid (linkedin.com, 2023) .....                    | 12  |
| Gambar 2.2. Statistik pertumbuhan Flutter (GoodFirms, 2025) .....         | 20  |
| Gambar 2.3. Struktur folder AdonisJS .....                                | 26  |
| Gambar 2.4. Komparasi kecepatan respons (Judd, 2018) .....                | 27  |
| Gambar 3.1. Flowchart Usulan Program .....                                | 48  |
| Gambar 3.2. Use Case Diagram .....                                        | 50  |
| Gambar 3.3. Activity Diagram Proses Watermark .....                       | 53  |
| Gambar 3.4. Activity Diagram Proses Verifikasi .....                      | 54  |
| Gambar 3.5. Sequence Diagram Proses Watermark .....                       | 56  |
| Gambar 3.6. Sequence Diagram Proses Verifikasi dengan Gambar .....        | 57  |
| Gambar 3.7. Sequence Diagram Proses Verifikasi dengan File PDF .....      | 58  |
| Gambar 3.8. Sequence Diagram Proses Verifikasi dengan Scan QR Code .....  | 58  |
| Gambar 3.9. Desain Class Diagram .....                                    | 59  |
| Gambar 3.10. Desain Entity Relationship Diagram .....                     | 60  |
| Gambar 3.11. Perancangan Interface Halaman Login .....                    | 81  |
| Gambar 3.12. Perancangan Interface Halaman Registrasi .....               | 81  |
| Gambar 3.13. Perancangan Interface Halaman OTP Verifikasi .....           | 82  |
| Gambar 3.14. Perancangan Interface halaman Pengaturan Password Baru ..... | 82  |
| Gambar 3.15. Perancangan Interface halaman Forgot Password .....          | 83  |
| Gambar 3.16. Perancangan Interface halaman Utama 1 .....                  | 83  |
| Gambar 3.17. Perancangan Interface Halaman Utama 2 .....                  | 84  |
| Gambar 3.18. Perancangan Interface Halaman Utama 3 .....                  | 84  |
| Gambar 3.19. Perancangan Interface Halaman Profile .....                  | 85  |
| Gambar 3.20. Perancangan Interface Halaman Proses .....                   | 85  |
| Gambar 4.1. Implementasi kode pengambilan file .....                      | 87  |
| Gambar 4.2. Implementasi kode untuk tata letak watermark .....            | 88  |
| Gambar 4.3. Implementasi kode untuk membaca QR code dan barcode .....     | 89  |
| Gambar 4.4. Implementasi kode untuk unggah ke Object Storage .....        | 89  |
| Gambar 4.5. Implementasi kode untuk interaksi dengan Webservice .....     | 90  |
| Gambar 4.6. Implementasi kode untuk menambah metadata .....               | 92  |
| Gambar 4.7. Implementasi kode untuk menambah steganografi .....           | 93  |
| Gambar 4.8. Implementasi kode untuk mengambil metadata .....              | 94  |
| Gambar 4.9. Implementasi kode untuk mengambil steganografi .....          | 95  |
| Gambar 4.10. Splashscreen .....                                           | 97  |
| Gambar 4.11. Onboarding .....                                             | 98  |
| Gambar 4.12. Pendaftaran .....                                            | 98  |
| Gambar 4.13. Verifikasi kode OTP .....                                    | 99  |
| Gambar 4.14. Lupa Sandi .....                                             | 99  |
| Gambar 4.15. Lupa Sandi .....                                             | 100 |
| Gambar 4.16. Halaman Utama Kosong .....                                   | 100 |
| Gambar 4.17. Halaman Utama Isi .....                                      | 101 |

|                                                           |     |
|-----------------------------------------------------------|-----|
| Gambar 4.18. Halaman Utama Riwayat .....                  | 101 |
| Gambar 4.19. Pilih Sumber dari File .....                 | 102 |
| Gambar 4.20. Pilih Sumber dari Kamera .....               | 102 |
| Gambar 4.21. Pratinjau PDF .....                          | 103 |
| Gambar 4.22. Pratinjau Kirim .....                        | 103 |
| Gambar 4.23. Proses Watermark .....                       | 104 |
| Gambar 4.24. Dokumen Rilis .....                          | 104 |
| Gambar 4.25. Verifikasi Gagal .....                       | 105 |
| Gambar 4.26. Verifikasi Berhasil .....                    | 105 |
| Gambar 4.27. Detil Informasi Dokumen .....                | 106 |
| Gambar 4.28. Pengaturan .....                             | 106 |
| Gambar 4.29. Tampilan Web .....                           | 107 |
| Gambar 4.30. Payload Request Login .....                  | 113 |
| Gambar 4.31. Payload Response Login .....                 | 113 |
| Gambar 4.32. Hasil Pengujian Login .....                  | 114 |
| Gambar 4.33. Payload Request Proses .....                 | 115 |
| Gambar 4.34. Payload Response Proses .....                | 115 |
| Gambar 4.35. Hasil Pengujian Proses (1 detik) .....       | 116 |
| Gambar 4.36. Hasil Pengujian Proses (60 detik) .....      | 116 |
| Gambar 4.37. Response SQL Injection Login (Postman) ..... | 118 |
| Gambar 4.38. Response SQL Injection Login (Mobile) .....  | 119 |

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Rilis kasus mafia tanah di tahun 2021 tercatat ada 2.991 kasus dan 1.908 perkara yang tercatat (Pusiknas, 2021). Ditahun 2022 rata rata Polri menindak Tujuh kasus pemalsuan surat dan dokumen setiap harinya, dan tercatat juga periode 1-12 januari 2022 terdapat 95 kasus yang sama. (Pusiknas, 2021)

Kasus penyalahgunaan dokumen pentingpun terjadi di tahun 2024, dimana banyak dari warga Indonesia tiba-tiba harus membayar pinjaman online yang nominalnya cukup besar tanpa tahu dari mana pinjaman itu berasal. Kepala Bidang Pengaduan dan Hukum Yayasan Lembaga Konsumen Indonesia (YLKI) Rio Priambodo mengatakan, pihaknya banyak menerima aduan tidak meminjam tetapi mendapat tagihan. "Tidak meminjam namun ditagih memang banyak pengaduannya ke YLKI sejak munculnya pinjaman online," ujarnya, saat dihubungi *Kompas.com*, Kamis (8/2/2024).

Dikutip lagi dari halaman website *Kompas.id* tertulis bahwa Direktur Eksekutif Lembaga Studi dan Advokasi Masyarakat (Elsam) Wahyudi Djafar menyebutkan, pihaknya baru menemukan tiga kasus pelanggaran perlindungan data pribadi yang inkrah dikenakan sanksi pidana sejak UU Nomor 27 Tahun 2022 diundangkan. Kasus pertama menyangkut pengumpulan data pribadi tidak sah,

kasus kedua mengenai pemakaian data pribadi orang lain, dan kasus ketiga tentang pemalsuan data pribadi orang lain.

Dari pemaparan kasus-kasus di atas yang tentunya belum di tulis semuanya, penulis bermaksud untuk mengangkat permasalahan ini menjadi topik skripsi yang diberi judul "Pembuatan Aplikasi Mobile untuk Pengamanan dan Verifikasi Dokumen Digital dan Fisik Berbasis *Watermark*, *Metadata* dan *Steganografi*".

## **1.2. Identifikasi Masalah**

Berdasarkan dari latar belakang yang sudah dijelaskan sebelumnya, maka identifikasi masalahnya adalah:

1. Banyaknya kasus pemalsuan dokumen penting untuk mengambil keuntungan.
2. Penyalahgunaan dokumen seseorang untuk keuntungan pribadinya.

## **1.3. Tujuan Penulisan**

Tujuan penulisannya ialah sebagai berikut:

1. Membuat sistem yang membantu masyarakat untuk menjaga dokumen dokumen digital dari orang yang tidak bertanggung jawab.
2. Meminimalisir penyalahgunaan dokumen dari orang-orang yang tidak bertanggung jawab.

## **1.4. Batasan Masalah**

Batasan masalah dalam penelitian ini ialah sebagai berikut:

1. Sistem ini dibuat hanya untuk mengetahui apakah dokumen yang diberikan itu asli punya orang yang bersangkutan atau sedang ada penyalahgunaan dokumen digital.
2. Pelacakan keaslian dokumen hanya dari Watermark yang disisipkan di lembar dokumen.

### **1.5. Metode Penelitian**

Penelitian ini menggunakan metode rekayasa perangkat lunak dengan pendekatan model *Waterfall*. Model ini dipilih karena memberikan alur kerja yang sistematis, terstruktur, dan cocok untuk pengembangan sistem yang kebutuhan dan ruang lingkupnya telah ditentukan di awal. Tahapan dalam metode *Waterfall* dilakukan secara berurutan mulai dari analisis kebutuhan hingga pemeliharaan sistem.

Berikut adalah tahapan-tahapan dalam model *Waterfall* yang digunakan dalam pengembangan aplikasi ini:

#### **1.5.1. Analisis Kebutuhan (*requirement analysis*)**

Pada tahap ini dilakukan pengumpulan kebutuhan sistem melalui:

- Observasi terhadap kasus-kasus pemalsuan dan penyalahgunaan dokumen.
- Wawancara dengan pengguna yang berkepentingan terhadap keamanan dokumen.
- Studi literatur terhadap teknik watermark, metadata, dan steganografi.

Hasil dari tahap ini adalah dokumen spesifikasi kebutuhan sistem, termasuk kebutuhan fungsional seperti penambahan watermark, metadata tersembunyi, serta

penyisipan steganografi, dan kebutuhan non-fungsional seperti keamanan data, kemudahan penggunaan, dan kompatibilitas perangkat.

### **1.5.2. Perancangan Sistem (*system design*)**

Pada tahap ini dilakukan desain sistem dan antarmuka pengguna berdasarkan kebutuhan yang telah dikumpulkan. Rancangan meliputi:

- Arsitektur aplikasi *mobile*.
- Desain database untuk penyimpanan metadata dan *log* verifikasi.
- Proses penambahan dan pembacaan *watermark*/steganografi.
- *Wireframe* dan atau tampilan antarmuka pengguna.

Dokumen desain ini menjadi acuan dalam proses implementasi aplikasi.

### **1.5.3. Implementasi (*implementation*)**

Tahap ini mencakup pembuatan kode program menggunakan:

- *Flutter* untuk aplikasi *mobile* (Android/iOS).
- Bahasa pemrograman *Dart*.
- *Library* tambahan untuk pengolahan gambar, PDF, serta penerapan algoritma *watermark* dan steganografi.

Fungsi-fungsi yang dikembangkan antara lain:

- Penambahan *watermark* teks, *qr code* dan *barcode*.
- Penyisipan *metadata* tersembunyi.
- Penyisipan informasi melalui steganografi.
- Pemindaian dokumen dan validasi tanda pengaman.

#### **1.5.4. Pengujian (*testing*)**

Setelah sistem selesai diimplementasikan, dilakukan pengujian terhadap seluruh fungsi aplikasi menggunakan metode *black-box testing* untuk memastikan bahwa setiap fitur berjalan sesuai dengan yang diharapkan. Fokus pengujian meliputi:

- Keakuratan *watermark* dan pembacaan *metadata* dan atau steganografi.
- Respons antarmuka pengguna.
- Ketahanan aplikasi terhadap *input* tidak *valid*.

Pengujian dilakukan dalam beberapa skenario kasus nyata berdasarkan contoh dokumen digital dan fisik.

#### **1.5.5. Pemeliharaan (*maintenance*)**

Tahap akhir dari proses adalah pemeliharaan aplikasi yang meliputi:

- Perbaikan *bug*.
- Penyempurnaan fitur berdasarkan umpan balik pengguna.
- Penyesuaian terhadap perubahan kebutuhan di masa mendatang.

Tahap ini juga mencakup dokumentasi kode dan cara penggunaan aplikasi sebagai panduan pengguna dan pengembang selanjutnya.

### **1.6. Sistematika Penulisan**

Penulisan dalam penelitian ini dibagi menjadi beberapa bab, yaitu sebagai berikut:

## BAB I PENDAHULUAN

Bab ini menjelaskan tentang latar belakang, identifikasi masalah, tujuan penulisan, definisi atau batasan masalah, metode penelitian yang terdiri dari proses pengumpulan data dan proses pengembangan proses, dan yang terakhir adalah perancangan sistem.

## BAB II LANDASAN TEORI

Dalam bab ini menjelaskan tentang berbagai teori penunjang sistem yang akan dibuat, Seperti teori sistem, website, android, UML dan lain sebagainya.

## BAB III ANALISIS MASALAH DAN PERANCANGAN PROGRAM

Bab ini berisi tentang analisis sistem yang dilakukan dengan menerapkan metode *waterfall*, yaitu *requirements*, *design*, *implementation*, *verification* dan *maintenance*. Diantaranya adalah perancangan sistem informasi berdasarkan analisa permasalahan dan perancangan *interface* program.

## BAB IV IMPLEMENTASI DAN UJI COBA

Bab ini berisi tentang pembahasan dalam pengoperasian dari rancangan database dan implementasi dari sistem informasi yang sudah dibuat, dan juga pengujian sistem menggunakan *black box testing*.

## BAB V PENUTUP

Bab ini berisi kesimpulan dan saran yang diambil dari hasil penelitian skripsi yang telah dibuat. Kesimpulan didapatkan dari penelitian yang

sudah dilakukan serta saran dibuat agar pengembangan selanjutnya menjadi lebih baik lagi.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Pengertian Sistem Informasi**

Sistem adalah suatu jaringan kerja dari prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan sasaran tertentu. Sedangkan informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya. Sehingga dapat disimpulkan bahwa sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi tertentu dengan laporan-laporan yang dibutuhkan. (Anggraini, Pasha, Damayanti, & Setiawan, 2020)

Sistem informasi adalah kegiatan dari prosedur yang di organisasikan yang digunakan untuk menyediakan informasi pengambilan keputusan dan pengendalian pada sebuah organisasi. Dalam istilah bahasa, sistem informasi terdiri dari sistem yang dapat diartikan sebagai kumpulan orang atau beberapa orang yang saling bekerja sama dan secara terstruktur untuk memenuhi tujuan-tujuan tertentu (Kurniawan, Apriliah, Kurniawan, & Firmasyah, 2020)

Sistem informasi merupakan sebuah sistem di dalam suatu organisasi mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, sifatnya manajerial, serta aktivitas strategis dari suatu (Nurlaela, Dharmalau, & Parida, 2020)

## **2.2. Pengertian Aplikasi**

Aplikasi adalah seperangkat instruksi khusus dalam computer agar kita menyelesaikan tugas-tugas tertentu. Berdasarkan pendapat diatas, dapat disimpulkan bahwa aplikasi adalah perangkat lunak yang siap pakai dengan menjalankan instruksi-instruksi dari pengguna untuk menyelesaikan tugas-tugas seperti membuat laoran, percetakan dan lain-lain. (Mokodompit, Mustain, & Feybelia, 2020)

Aplikasi adalah sebuah program atau perangkat lunak yang dirancang untuk menjalankan fungsi tertentu pada perangkat elektronik, seperti komputer, ponsel pintar, atau tablet. Istilah ini berasal dari bahasa Inggris “*application software*” yang berarti perangkat lunak aplikasi. Dalam konteks teknologi saat ini, aplikasi sangat beragam jenisnya, mencakup berbagai sektor dan kebutuhan. (Al Fatih, 2024)

### **2.4.1. Aplikasi *Mobile***

Aplikasi *mobile* berasal dari kata *application* dan *mobile*. *Application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju, sedangkan *mobile* dapat diartikan sebagai perpindahan dari suatu tempat ke tempat yang lain. (Mokodompit, Mustain, & Feybelia, 2020)

Aplikasi *Mobile* berasal dari dua kata, yaitu aplikasi dan *mobile*. Secara istilah, aplikasi adalah program siap pakai yang dibuat untuk melaksanakan suatu

fungsi untuk pengguna atau aplikasi yang lain sedangkan *mobile* adalah perpindahan dari suatu tempat ke tempat yang lain. Secara lebih lengkap, aplikasi *mobile* adalah program siap pakai yang melaksanakan fungsi tertentu yang dipasang pada perangkat *mobile*. (Imron, Azizah, Nurhayati, & Wijonarko, 2021)

Aplikasi *mobile* atau *mobile application* merupakan proses pengembangan aplikasi yang dibuat untuk perangkat genggam, *handphone*, *PDA*, atau yang lebih dikenal dengan *smartphone*. Aplikasi tersebut sudah ada di telepon ketika proses manufaktur, disebut sebagai aplikasi bawaan. Aplikasi juga bisa didownload oleh pengguna dari toko aplikasi dan juga dari distribusi perangkat lunak mobil platform lainnya. Aplikasi *mobile* memiliki banyak keunggulan jika dibandingkan dengan aplikasi *website* dan juga *desktop*. (Annisa & Baihaqi, 2021)

#### **2.4.2. Aplikasi *Android***

*Android* adalah *software platform* yang *open source* untuk *mobile device*. *Android* berisi sistem operasi, *middleware* dan aplikasi-aplikasi dasar. Basis OS *android* adalah *kernel linux 2.6* yang telah dimodifikasi untuk *mobile device*. (Imron, Azizah, Nurhayati, & Wijonarko, 2021)

*Android* adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *Linux* yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. *Android* merupakan generasi baru *platform mobile*, *platform* yang memberikan pengembang untuk melakukan pengembangan sesuai dengan yang diharapkannya. (Wahyuni, Sopiandi, & Raharjo, 2020)

*Android* adalah *platform* terbuka pertama untuk perangkat *mobile*, semua perangkat lunak yang ada memiliki fungsi menjalankan sebuah perangkat *mobile* tanpa perlu memikirkan kendala kepemilikan yang menghambat inovasi pada teknologi *mobile*. (Suharyanto, 2022)

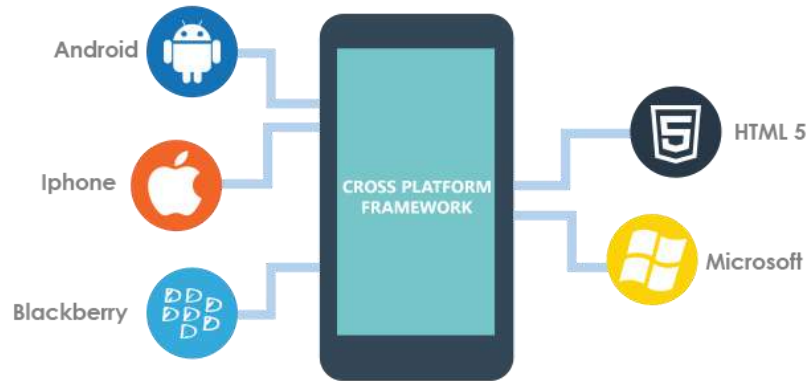
### **2.4.3. Aplikasi *iOS***

*iOS* adalah sistem operasi *mobile* yang dikembangkan oleh *Apple Inc.* Sistem ini memiliki arsitektur berlapis yang mencakup *Core OS*, *Core Services*, *Media*, dan *Cocoa Touch*. Setiap lapisan dirancang untuk memberikan keamanan dan kinerja optimal pada perangkat *Apple*. *Apple* mengintegrasikan keamanan ke dalam inti *platformnya*, menggabungkan perangkat keras, perangkat lunak, dan layanan yang dirancang untuk bekerja sama demi keamanan maksimal serta pengalaman pengguna yang transparan. (*Apple*, 2023)

*iOS* menerapkan berbagai fitur keamanan seperti *sandboxing*, enkripsi data, dan manajemen izin aplikasi. Pendekatan ini memberikan kontrol ketat terhadap aplikasi yang diizinkan masuk ke *App Store*, sehingga meningkatkan keamanan pengguna. Namun, pendekatan ekosistem tertutup ini juga membatasi fleksibilitas pengembang dalam menyesuaikan sistem operasi. (Mulyani, 2022)

Dalam pengembangan aplikasi, *iOS* menyediakan *platform* yang stabil dan seragam, memudahkan pengembang untuk menciptakan aplikasi berkualitas tinggi. Meskipun demikian, pengembang harus mematuhi pedoman ketat dari *Apple* dan proses peninjauan aplikasi yang ketat sebelum aplikasi mereka dapat tersedia di *App Store*. (Arnomo & H, 2019)

#### 2.4.4. Aplikasi *Hybrid*



Gambar 2.1. Aplikasi Hybrid ([linkedin.com](https://www.linkedin.com), 2023)

Aplikasi *hybrid* adalah aplikasi *mobile* yang pada awalnya dikembangkan menggunakan teknologi *web* seperti *HTML*, *CSS*, dan *JavaScript*, kemudian dikemas dalam wadah *native* sehingga dapat diinstal dan dijalankan di berbagai sistem operasi seperti *Android* dan *iOS*. Pendekatan ini memungkinkan pengembang menulis satu basis kode yang dapat digunakan lintas *platform*, sehingga efisien dalam hal waktu dan biaya pengembangan. (Barros Assunção, Rodrigues Pantoja, Pereira Gonçalves, Barbosa de Alencar, & Lobo de Oliveira, 2019)

Dalam penerapannya, aplikasi *hybrid* memanfaatkan *framework* seperti *Apache Cordova*, *React Native*, atau *Flutter* yang memungkinkan integrasi dengan fitur perangkat seperti kamera, *GPS*, dan penyimpanan lokal. Pendekatan ini mempermudah pengembangan meskipun tidak semua fitur perangkat dapat dimanfaatkan secara optimal sebagaimana pada aplikasi *native*. (Khoirudin, Daru, & Nugroho, 2019)

Selain itu, aplikasi *hybrid* juga dapat mendukung konsep arsitektur layanan atau *Service Oriented Architecture (SOA)* yang memungkinkan aplikasi terintegrasi dengan sistem *backend* secara fleksibel. Pendekatan ini sangat berguna dalam pengembangan aplikasi yang membutuhkan komunikasi data dengan *server* secara dinamis dan modular. (Khoirudin, Daru, & Nugroho, 2019)

Dalam penelitian lainnya, aplikasi *hybrid* juga dinilai mampu meningkatkan kemudahan akses dan penggunaan sistem informasi, seperti pada sistem penyewaan buku berbasis *mobile* yang dikembangkan dengan konsep *hybrid* untuk memberikan pengalaman pengguna yang baik serta kemudahan transaksi daring. (Pradana & & Waspada, 2019)

#### **2.4.5. Aplikasi Web**

Aplikasi *web* berbasis *HTML*, *CSS*, dan *JavaScript* merupakan jenis aplikasi yang dijalankan melalui peramban web tanpa memerlukan instalasi khusus di perangkat pengguna. *HTML* digunakan untuk membangun struktur dan konten dasar halaman *web*, *CSS* untuk mengatur tampilan dan gaya visual, serta *JavaScript* untuk menambahkan interaktivitas dan logika dinamis pada aplikasi. Kombinasi ketiga teknologi ini memungkinkan pengembangan aplikasi yang responsif dan dapat diakses lintas *platform*, seperti *desktop*, *tablet*, dan perangkat *mobile*. Sebagai contoh, dalam pengembangan aplikasi kalkulator sederhana, *HTML* digunakan untuk membuat kerangka tampilan, *CSS* untuk mengatur *layout* dan gaya, serta *JavaScript* untuk menangani interaksi pengguna dan perhitungan aritmatika. (Sinlae, 2024)

Dalam studi lain, pengembangan aplikasi web seperti "*Random Team Generator*" menggunakan *HTML*, *CSS*, dan *JavaScript* untuk memungkinkan pengguna membagi anggota tim secara acak melalui antarmuka yang intuitif. Proses pengembangan dimulai dengan perencanaan, analisis, desain, hingga implementasi menggunakan metode *Waterfall*, yang menekankan pada tahapan yang terstruktur dan sistematis. (Pratama & Siregar, Pengembangan Aplikasi Random Team Generator Berbasis Web Menggunakan HTML, CSS, dan JavaScript, 2024)

*ReactJS* adalah salah satu pustaka *JavaScript* yang digunakan secara luas dalam pengembangan antarmuka pengguna (*UI*) aplikasi web. *ReactJS* memungkinkan pengembang membangun antarmuka yang dinamis dan responsif dengan cara memecah *UI* menjadi komponen-komponen kecil yang dapat digunakan kembali. *ReactJS* juga mendukung *Virtual DOM*, yang membuat pembaruan *UI* menjadi lebih efisien dan cepat. Teknologi ini banyak digunakan dalam pengembangan aplikasi web modern karena skalabilitas dan fleksibilitasnya dalam mengelola status dan alur data dalam aplikasi. (Fauzan & Ramadhani, 2023)

### **2.3. Pengertian Keamanan**

Keamanan adalah serangkaian upaya dan mekanisme yang diterapkan untuk melindungi informasi atau data dari akses, perubahan, pengungkapan, atau kerusakan yang tidak sah. Tujuan utama keamanan adalah menjaga kerahasiaan, integritas, dan ketersediaan informasi sehingga hanya pihak yang berwenang yang dapat mengakses dan memanfaatkan data tersebut secara benar dan aman. (Putra & Santoso, 2022) (Rahman & Lestari, 2023).

Implementasi keamanan aplikasi merupakan aspek krusial dalam pengembangan sistem informasi, terutama dalam menghadapi ancaman siber yang semakin kompleks dan beragam. Terdapat berbagai pendekatan dan teknik yang dapat diterapkan untuk memastikan keamanan aplikasi, baik dari sisi *server*, klien, maupun dalam proses transmisi data.

Salah satu teknik dasar yang umum digunakan adalah enkripsi, yaitu proses mengubah data menjadi format yang tidak dapat dibaca oleh pihak yang tidak memiliki otorisasi. Sebagai contoh, penggunaan algoritma enkripsi seperti *MD5* (*Message-Digest Algorithm 5*) pada aplikasi berbasis *web* dapat berperan dalam mengurangi risiko serangan seperti *SQL Injection* (Zebua, Herwanto, & Rosida, 2022). Meskipun *MD5* tidak lagi direkomendasikan sebagai metode enkripsi utama karena kelemahan kriptografinya, dalam beberapa konteks non-kritis, algoritma ini masih digunakan untuk mengubah *input* menjadi representasi *hash* yang unik. Dengan mengenkripsi data sebelum dikirimkan ke *server*, aplikasi dapat memperkecil kemungkinan data tersebut dimanipulasi atau disisipkan kode berbahaya oleh penyerang. Ini memberi lapisan perlindungan tambahan terhadap integritas dan keaslian data yang diproses.

Selain enkripsi, berbagai teknik keamanan lainnya dapat diterapkan, seperti *input validation* (validasi masukan), *parameterized queries* untuk mencegah injeksi *SQL*, serta penggunaan *firewall* aplikasi *web* (*Web Application Firewall/WAF*) yang dapat menyaring dan memblokir permintaan mencurigakan. Penggunaan tokenisasi, autentikasi multi-faktor (*MFA*), dan manajemen sesi yang aman juga

merupakan praktik yang direkomendasikan untuk menjaga keamanan aplikasi secara menyeluruh.

### **2.3.1. *Watermark***

*Watermark* dalam konteks digital atau seni visual adalah cap atau penanda yang disematkan pada gambar, video, atau dokumen lainnya untuk menandai kepemilikan atau keaslian karya tersebut. *Watermark* bisa berupa teks, logo, atau simbol lain yang dimasukkan ke dalam atau di atas gambar untuk memberikan informasi tentang pencipta atau pemilik hak cipta serta mencegah penggunaan ilegal atau pencurian karya. (Ardian, Haniyah, & Saktiawan, 2025)

*Watermark* adalah tanda yang diletakkan pada suatu karya untuk menandakan kepemilikan atau hak cipta dari karya tersebut. *Watermark* dapat berupa tulisan, bit, gambar, logo, dan lain lain. (Gultoma & Suhartanaa, 2023)

### **2.3.2. *Metadata***

*Metadata* adalah data yang memberikan informasi tentang data lainnya, seperti struktur, isi, konteks, dan administrasi sebuah dokumen digital. *Metadata* berfungsi untuk mengidentifikasi, mendeskripsikan, dan mengelola informasi digital agar dapat diakses, ditemukan, dan digunakan kembali secara efisien dalam berbagai sistem informasi. Dalam konteks ini, metadata tidak hanya bertindak sebagai penanda informasi, tetapi juga sebagai alat kontrol yang mendukung validasi, keamanan, dan pelacakan riwayat dokumen digital. (Lubis & Nasution, 2024)

Selain itu, metadata juga berperan penting dalam ekosistem repositori penelitian, di mana ia digunakan untuk mengorganisasi, mengklasifikasikan, serta memastikan keterhubungan dan interoperabilitas antar data penelitian. Penggunaan standar metadata seperti *Dublin Core* dan *DataCite* memperkuat prinsip *FAIR* (*Findable, Accessible, Interoperable, Reusable*), yang menjadi landasan utama dalam manajemen data ilmiah modern (Yudhanto & Mayesti, 2021)

### 2.3.3. Steganografi

Steganografi adalah ilmu yang mempelajari tentang cara untuk menyembunyikan pesan atau informasi. Steganografi biasanya digunakan untuk menyembunyikan pesan pada sampul media, dimana orang tidak akan curiga, karena letak keberadaan pesan tersebut biasanya sulit diketahui. (Gultoma & Suhartanaa, 2023)

Dalam Steganografi (*steganography*) adalah ilmu dan seni menyembunyikan pesan rahasia (*hiding message*) sedemikian sehingga keberadaan (eksistensi) pesan tidak terdeteksi oleh indera manusia. Kata "steganografi" berasal dari bahasa Yunani "*steganos*", yang artinya "tersembunyi atau terselubung", dan "*graphein*", "menulis". (Putri, Kartikadewi, & Abdul Rosyid, 2020)

Sebuah pesan steganografi (*plaintext*), dienkripsikan dengan beberapa arti tradisional, yang menghasilkan *ciphertext*. Kemudian, *coverttext* dimodifikasi dalam beberapa cara sehingga berisi *ciphertext*, yang menghasilkan *stegotext*. Contohnya, ukuran huruf, ukuran spasi, jenis huruf, atau karakteristik *coverttext* lainnya dapat dimanipulasi untuk membawa pesan tersembunyi. Hanya penerima

(yang harus mengetahui teknik yang digunakan) dapat membuka pesan dan mendekripsikannya.

## **2.4. Bahasa Pemrograman**

Adalah sekumpulan instruksi yang digunakan untuk membuat program komputer. Bahasa ini memungkinkan manusia memberikan perintah kepada komputer dengan cara yang bisa dimengerti oleh mesin, sehingga komputer dapat menjalankan tugas tertentu sesuai dengan instruksi tersebut. (Sebastian & Rahman, 2020)

Bahasa pemrograman memiliki sintaks dan aturan yang harus diikuti agar instruksi dapat diproses dengan benar. Ada banyak jenis bahasa pemrograman yang dikembangkan untuk berbagai tujuan, mulai dari pengembangan aplikasi web, perangkat lunak desktop, hingga aplikasi mobile dan sistem *embedded*.

Selain itu, bahasa pemrograman juga dibedakan menjadi bahasa tingkat tinggi dan bahasa tingkat rendah. Bahasa tingkat tinggi lebih mudah dipahami manusia karena mirip dengan bahasa alami, sedangkan bahasa tingkat rendah lebih dekat dengan bahasa mesin dan lebih efisien dalam pemrosesan. (Sari & Nugroho, 2022)

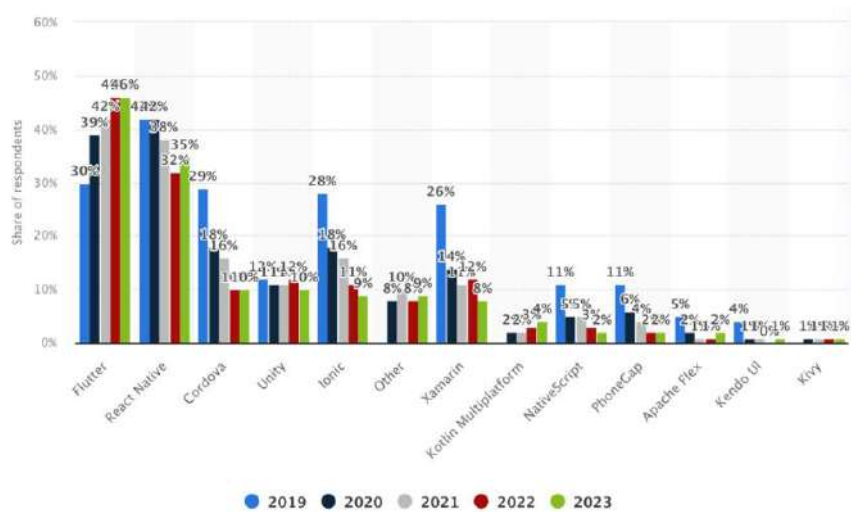
### **2.4.1. Flutter**

*Flutter* adalah sebuah *framework open-source* yang dikembangkan oleh *Google* untuk membangun aplikasi *mobile*, *web*, dan *desktop* dengan basis kode tunggal. *Flutter* menggunakan bahasa pemrograman *Dart* dan menyediakan kumpulan *widget* yang lengkap untuk mendesain antarmuka pengguna secara

deklaratif dan responsif. Keunggulan *Flutter* terletak pada performa yang tinggi karena menggunakan mesin rendering sendiri tanpa bergantung pada komponen *native* dari *platform*, sehingga menghasilkan tampilan dan animasi yang halus serta konsisten di berbagai perangkat. (Pratama & Hidayat, Pengembangan Aplikasi Mobile Cross-Platform Menggunakan Flutter, 2021)

Selain itu, *Flutter* mendukung pengembangan aplikasi lintas *platform* dengan produktivitas tinggi melalui fitur hot reload yang memungkinkan pengembang melihat perubahan kode secara langsung tanpa kehilangan status aplikasi. Hal ini mempercepat siklus pengembangan dan *debugging*. *Flutter* juga memiliki ekosistem yang terus berkembang dengan berbagai paket dan plugin yang memudahkan integrasi fitur seperti kamera, lokasi, dan akses database. (Santoso, Putra, & Utomo, 2022)

*Flutter* tidak hanya populer dalam pengembangan aplikasi *mobile*, tetapi juga mulai banyak digunakan untuk pengembangan aplikasi *web* dan *desktop*, yang menjadikannya solusi serbaguna bagi pengembang yang ingin mengoptimalkan waktu dan sumber daya dalam membuat aplikasi multi-*platform*. (Wibowo, 2023)



Gambar 2.2. Statistik pertumbuhan Flutter (GoodFirms, 2025)

*Flutter* telah mengalami pertumbuhan signifikan dalam adopsi dan penggunaan sejak diperkenalkan. Sekitar 46% pengembang perangkat lunak menggunakan *Flutter* pada tahun 2023, menjadikannya *framework* lintas *platform* paling populer di kalangan pengembang. Keunggulan *Flutter*, seperti performa tinggi, kemampuan lintas *platform*, dan komunitas yang berkembang pesat. (GoodFirms, 2025)

#### 2.4.2. *Dart*

*Dart* adalah bahasa pemrograman yang dikembangkan oleh *Google* sebagai bahasa pemrograman utama untuk membangun aplikasi *Flutter*. *Dart* dirancang untuk memberikan performa tinggi dan produktivitas bagi pengembang dengan sintaks yang mudah dipahami serta dukungan fitur modern seperti *just-in-time (JIT)* dan *ahead-of-time (AOT) compilation*. Hal ini memungkinkan pengembang untuk

melakukan pengembangan cepat dengan hot reload serta menghasilkan aplikasi yang efisien dan responsif pada platform mobile maupun *web*.

Bahasa ini juga memberikan fleksibilitas dalam pengembangan *UI* yang deklaratif dan reaktif di *Flutter*, yang memungkinkan pembuatan aplikasi lintas platform secara konsisten dengan kode yang dapat dipelihara dan skalabel. Fitur bahasa *Dart* yang kuat seperti *null safety*, *async-await*, dan *generics* juga sangat mendukung pengembangan aplikasi modern. (Nugroho & & Putra, 2023)

#### **2.4.3. Javascript**

*JavaScript* adalah bahasa pemrograman tingkat tinggi yang telah menjadi tulang punggung pengembangan *web* modern. Awalnya dikembangkan sebagai bahasa scripting untuk memperkaya halaman web dengan elemen interaktif di sisi klien (*client-side*), *JavaScript* kini telah berkembang menjadi bahasa pemrograman serbaguna yang juga dapat dijalankan di sisi *server* (*server-side*) melalui platform seperti *Node.js*. Dengan kemampuannya yang luas, *JavaScript* memungkinkan pengembang untuk membangun aplikasi *web* yang responsif, dinamis, dan interaktif, baik untuk perangkat desktop maupun mobile. Ekosistem *JavaScript* sangat kaya dengan pustaka dan *framework* populer seperti *React*, *Angular*, dan *Vue.js* yang sangat membantu dalam mempercepat proses pengembangan aplikasi *web* yang kompleks dan skalabel. (Putra & Hartono, 2023)

Selain itu, *JavaScript* menyediakan berbagai teknik pemrograman *asynchronous* seperti *callbacks*, *promises*, dan *async-await* yang memungkinkan pengelolaan proses yang berjalan secara paralel tanpa menghambat kinerja aplikasi.

Teknik-teknik ini sangat penting untuk menangani operasi yang memerlukan waktu lama, seperti pengambilan data dari *server* atau interaksi dengan *database*, sehingga pengalaman pengguna tetap lancar dan responsif. Penelitian oleh Sari dan Widodo (2022) menyoroti peran krusial JavaScript dalam pengembangan aplikasi web modern, terutama dalam meningkatkan interaktivitas dan efisiensi pengelolaan sumber daya. Sementara itu, studi oleh Putra dan Hartono (2023) mengulas secara mendalam teknik pemrograman *asynchronous* di *JavaScript*, menekankan pentingnya pendekatan ini untuk memastikan performa optimal dalam aplikasi *web* skala besar. (Sari & Widodo, 2022)

#### **2.4.4. *Typescript***

*TypeScript* adalah bahasa pemrograman yang dikembangkan oleh *Microsoft* sebagai ekstensi dari *JavaScript*. Bahasa ini menambahkan fitur *static typing* yang memungkinkan pengembang untuk menangkap kesalahan pada waktu kompilasi sebelum kode dijalankan, sehingga meningkatkan kualitas dan keamanan kode. *TypeScript* banyak digunakan dalam pengembangan aplikasi *web* skala besar karena kemampuannya untuk menyediakan struktur yang lebih kuat dan skalabilitas dalam pengelolaan kode dibandingkan *JavaScript* biasa. (Wijaya & Santoso, 2021)

Selain itu, *TypeScript* mendukung fitur-fitur modern dari *ECMAScript* sekaligus kompatibel dengan semua pustaka *JavaScript* yang ada, sehingga memudahkan migrasi dan integrasi dalam proyek yang sudah berjalan. Dengan dukungan *type annotations*, *interfaces*, dan *generics*, *TypeScript* membantu pengembang menulis kode yang lebih jelas dan mudah dipelihara. Penelitian oleh

Putri dan Hidayat (2022) menunjukkan bahwa penggunaan *TypeScript* dalam pengembangan aplikasi *front-end* dapat mengurangi waktu *debugging* dan meningkatkan produktivitas tim *developer* secara signifikan. (Putri & Hidayat, 2022)

#### **2.4.5. NodeJS**

*Node.js* adalah platform *runtime JavaScript* yang berjalan di sisi *server* dan memungkinkan pengembangan aplikasi *web* yang cepat dan *scalable*. Dengan menggunakan *event-driven* dan *non-blocking I/O*, *Node.js* mampu menangani banyak koneksi secara bersamaan secara efisien, sehingga sangat cocok untuk aplikasi *real-time* seperti chat dan layanan API. Keunggulan ini membuat *Node.js* populer dalam pengembangan sistem yang membutuhkan performa tinggi dan responsivitas cepat.

Dalam pengembangan aplikasi berbasis *REST API*, *Node.js* kerap dipadukan dengan *framework Express.js* untuk membangun *backend* yang modular dan ringan. (Kurniawan, Humaira, & Rozi, 2020) melakukan penelitian dalam pembuatan sistem transaksi jasa elektronik berbasis Android dengan menggunakan *Node.js* sebagai *backend* yang menyediakan layanan *REST API*. Penelitian tersebut menunjukkan bahwa *Node.js* mampu mendukung pengembangan sistem yang responsif dan mudah diakses oleh pengguna, terutama untuk aplikasi *mobile* yang memerlukan komunikasi *real-time*.

Selain itu, integrasi *Node.js* dengan *database* relasional seperti *PostgreSQL* juga umum digunakan untuk membangun aplikasi dengan kebutuhan penyimpanan

data yang kompleks. Penelitian oleh sumber, mengembangkan aplikasi e-commerce "Localla" yang memanfaatkan *Node.js* sebagai *backend* dan *PostgreSQL* untuk penyimpanan data. Sistem ini berhasil membangun *REST API* yang mampu mengelola data produk dan transaksi dengan baik, serta memberikan kemudahan akses bagi *frontend* aplikasi. (Barokah & Sauda, 2022)

#### **2.4.6. ReactJS**

*ReactJS* adalah pustaka *JavaScript open-source* yang digunakan untuk membangun antarmuka pengguna (*user interface*) yang dinamis dan responsif, terutama untuk aplikasi *web*. *ReactJS* memanfaatkan konsep *component-based architecture*, di mana *UI* dibagi menjadi bagian-bagian kecil yang dapat digunakan kembali (*reusable components*). Hal ini memungkinkan pengembang untuk membuat aplikasi yang lebih mudah dipelihara dan dikembangkan secara modular. Selain itu, *React* menggunakan *virtual DOM* yang meningkatkan performa dengan meminimalkan manipulasi langsung ke *DOM* asli *browser*. (Saputra & Wibowo, 2021)

*ReactJS* juga mendukung pengembangan aplikasi satu halaman (*single-page applications/SPA*) yang memberikan pengalaman pengguna lebih lancar dan interaktif tanpa perlu memuat ulang halaman secara keseluruhan. Dalam penelitian oleh Hartono dan Wijaya (2022), *ReactJS* digunakan untuk mengembangkan aplikasi *e-commerce* yang mampu menampilkan data produk secara *real-time* dan responsif. Studi tersebut menunjukkan bahwa *ReactJS* memudahkan proses pengembangan frontend sekaligus meningkatkan kecepatan render halaman,

sehingga aplikasi menjadi lebih efisien dan *user-friendly*. (Hartono & Wijaya, 2022)

#### **2.4.7. *AdonisJS***

*AdonisJS* adalah *framework backend* berbasis *Node.js* yang dirancang untuk membangun aplikasi *web* yang stabil, terstruktur, dan produktif dengan pendekatan *MVC (Model-View-Controller)*. *Framework* ini menyediakan *toolset* yang lengkap seperti *ORM (Object Relational Mapping)*, sistem *routing*, validasi, autentikasi, serta manajemen *session* dan *file upload*, menjadikannya cocok untuk pengembangan *web* modern berbasis *JavaScript*. (AdonisJS, 2025)

```

default-adonis-app/
├── app/                                # Tempat utama untuk kode aplikasi
│   ├── Controllers/Http/              # Controller HTTP (route handler)
│   ├── Middleware/                    # Middleware kustom
│   ├── Models/                        # Model (ORM)
│   ├── Validators/                    # Validator request (optional)
│   └── Services/                      # (Optional) untuk layanan bisnis
│
├── config/                            # Konfigurasi aplikasi
│   ├── app.ts
│   ├── database.ts
│   ├── drive.ts
│   └── (lainnya, sesuai package)
│
├── contracts/                         # Interface & typing untuk ekstensi
│   ├── auth.ts
│   ├── drive.ts
│   └── (lainnya)
│
├── database/
│   ├── migrations/                   # File migrasi database
│   ├── seeders/                     # Seeder data awal
│   └── factories/                    # Factory model (untuk testing/seeding)
│
├── public/                            # Public assets (JS, CSS, gambar)
│   └── index.html                    # Untuk aplikasi SPA atau statis
│
├── resources/
│   ├── views/                        # View template (Edge)
│   └── js/ css/ images/              # (Optional) aset untuk build frontend
│
├── routes/
│   └── routes.ts                     # Semua definisi route HTTP
│
├── start/
│   ├── kernel.ts                     # Registrasi global middleware
│   ├── routes.ts                     # Import file route utama
│   └── server.ts                     # Bootstrap HTTP server
│
├── ace-manifest.json                 # Manifest untuk CLI 'ace'
├── .adonisrc.json                   # Konfigurasi utama proyek AdonisJS
├── .env                             # File environment (konfigurasi lokal)
├── package.json                     # Dependency dan script
├── tsconfig.json                    # Konfigurasi TypeScript
└── README.md

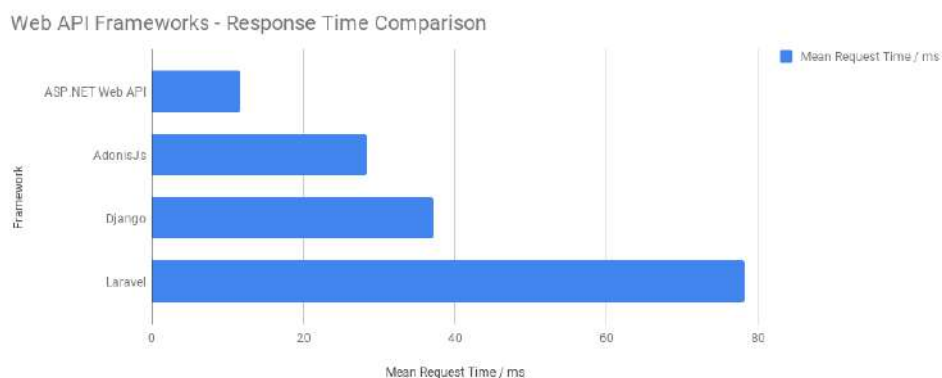
```

Gambar 2.3. Struktur folder AdonisJS

*AdonisJS* mempermudah pengembangan aplikasi *server-side* dengan struktur yang menyerupai *Laravel* pada *PHP*, sehingga membantu pengembang yang berasal dari latar belakang *Laravel* untuk beradaptasi lebih cepat. Selain itu,

*AdonisJS* didesain dengan prinsip “*developer-first*” memberikan pengalaman pemrograman yang konsisten dan menyenangkan. (Pavithran & Ganapathy, 2022)

Ketika performa *AdonisJS* dibandingkan dengan framework *Node.js* populer seperti *Express*, dan *Koa*. Hasilnya menunjukkan bahwa *AdonisJS* unggul dalam hal kestabilan kode dan pengelolaan dependensi karena sifatnya yang opiniatif. *Framework* ini juga memiliki dokumentasi yang kuat dan ekosistem plugin yang mendukung skalabilitas aplikasi enterprise. (Behrang & Ali, 2021)



Gambar 2.4. Komparasi kecepatan respons (Judd, 2018)

Seorang pengembang aplikasi menguji performa beberapa *framework web* (*Laravel*, *ASP.NET Core*, *Django*) menggunakan *Apache Benchmark*, dan menambahkan *AdonisJS* untuk melihat bagaimana *Node.js* yang *asynchronous* bersaing dalam menangani permintaan bersamaan. Hasil awal menunjukkan *C#* paling cepat, namun *AdonisJS* dipilih karena kemudahan penggunaannya mirip *Laravel* dan cocok untuk aplikasi *API* serta mudah di-Dockerisasi. (Judd, 2018)

## 2.5. Pengertian *MinIO*

*MinIO* adalah sistem penyimpanan objek sumber terbuka yang kompatibel dengan *Amazon S3*, dirancang untuk menangani data tidak terstruktur dalam lingkungan *cloud-native*. *MinIO* menawarkan performa tinggi, skalabilitas, dan kesederhanaan, menjadikannya pilihan ideal untuk berbagai kebutuhan penyimpanan data, termasuk analitik *big data*, pembelajaran mesin, pencadangan, dan pengarsipan. Kemampuannya untuk berjalan di lingkungan cloud pribadi, publik, dan hibrida memberikan fleksibilitas bagi organisasi dalam mengelola data mereka. (Magazine, 2023)

Dalam studi yang membandingkan *MinIO* dengan sistem *file* terdistribusi lainnya seperti *Google File System (GFS)* dan *Hadoop Distributed File System (HDFS)*, *MinIO* menunjukkan keunggulan dalam toleransi kesalahan dan skalabilitas. *MinIO* mampu menangani redundansi data, kegagalan server, dan akses klien secara efisien, menjadikannya solusi yang andal untuk lingkungan komputasi terdistribusi dan *cloud*. Desain sistemnya memungkinkan performa tinggi dalam arsitektur *cloud* dan komputasi terdistribusi, memberikan wawasan praktis bagi organisasi dalam memilih sistem penyimpanan yang sesuai dengan kebutuhan mereka. (Malhotra, et al., 2025)

## 2.6. Pengertian *Server*

*Server* adalah sistem komputer yang didesain untuk menyediakan layanan tertentu kepada perangkat lain yang disebut klien dalam sebuah jaringan. Fungsi utama *server* adalah menerima permintaan dari klien dan merespons dengan

menyediakan layanan seperti pengelolaan data, penyimpanan *file*, atau menjalankan aplikasi. *Server* dapat berupa perangkat keras fisik atau mesin *virtual* dalam lingkungan *cloud* yang memungkinkan skalabilitas dan efisiensi sumber daya. (Aljawarneh, 2021)

Terdapat berbagai jenis *server* sesuai fungsi dan layanan yang disediakan. *Web server* bertugas menyajikan konten halaman *web* kepada pengguna melalui protokol *HTTP* atau *HTTPS*, dengan contoh populer seperti *Apache* dan *Nginx*. *File server* menyediakan tempat penyimpanan terpusat sehingga pengguna dapat berbagi dan mengakses *file* dalam jaringan secara mudah dan aman. *Database server* mengelola dan menyediakan akses ke basis data, melayani permintaan *query* dari aplikasi atau pengguna. *Mail server* berfungsi sebagai pengatur pengiriman dan penerimaan *email* menggunakan protokol *SMTP*, *POP3*, dan *IMAP*. Selain itu, *application server* menyediakan lingkungan untuk menjalankan aplikasi bisnis atau perangkat lunak lainnya secara terpusat. (Aljawarneh, 2021)

Dalam operasionalnya, *server* memiliki karakteristik utama yaitu kemampuan bekerja nonstop selama 24/7 dengan tingkat keandalan dan performa tinggi. *Server* dilengkapi dengan perangkat keras yang kuat, kapasitas penyimpanan besar, serta mekanisme keamanan yang canggih untuk melindungi data dari serangan dan kehilangan. Sistem operasi khusus pada *server* memungkinkan pengelolaan jaringan secara efisien, termasuk monitoring aktivitas pengguna dan manajemen pemulihan sistem bila terjadi kegagalan. (Khan, Uddin, & Gani, 2022)

Perkembangan teknologi virtualisasi dan cloud computing memperkenalkan konsep *server virtual* yang memungkinkan sumber daya komputasi seperti *CPU*,

*RAM*, dan penyimpanan untuk dibagi secara dinamis di antara beberapa *server* logis. Hal ini meningkatkan efisiensi penggunaan perangkat keras dan memungkinkan organisasi menyesuaikan kapasitas server sesuai kebutuhan secara *real-time*, mengurangi biaya operasional dan meningkatkan fleksibilitas. (Khan, Uddin, & Gani, 2022)

Secara keseluruhan, *server* merupakan fondasi utama infrastruktur jaringan dan teknologi informasi modern. *Server* menjamin ketersediaan layanan, keamanan data, dan kinerja sistem yang optimal, sehingga menjadi elemen vital dalam pengembangan layanan digital dan aplikasi berbasis jaringan di berbagai sektor industri. (Aljawarneh, 2021) (Khan, Uddin, & Gani, 2022)

## **2.7. Pengertian *Microservices***

*Microservices* merupakan arsitektur perangkat lunak yang memecah aplikasi besar menjadi sekumpulan layanan kecil yang berdiri sendiri, yang dapat dikembangkan, diuji, dan di-deploy secara independen. Setiap layanan berfokus pada fungsi bisnis tertentu dan berkomunikasi satu sama lain menggunakan protokol ringan seperti *HTTP/REST* atau *message broker*. Pendekatan ini memungkinkan tim pengembang bekerja secara paralel pada layanan berbeda, meningkatkan fleksibilitas dan mempercepat siklus rilis perangkat lunak. Model ini juga mempermudah skalabilitas karena layanan dapat diskalakan sesuai kebutuhan masing-masing tanpa mempengaruhi keseluruhan aplikasi. (Dragoni, et al., 2020)

Dalam konteks interaksi antar layanan *microservices*, terdapat dua pola utama yaitu orkestrasi dan koreografi. Orkestrasi adalah pendekatan terpusat di

mana sebuah komponen orkestrator mengatur alur kerja dan komunikasi antar layanan. Orkestrator bertugas mengendalikan eksekusi layanan secara berurutan sesuai proses bisnis yang diinginkan, memudahkan monitoring dan kontrol aliran proses. Contohnya adalah penggunaan *workflow engine* atau *container orchestration* seperti *Kubernetes* untuk mengelola siklus hidup layanan secara terpusat. (Richardson, 2021)

Sebaliknya, koreografi menerapkan model desentralisasi berbasis *event-driven*, di mana setiap layanan bertindak secara otonom dan saling berinteraksi dengan merespons *event* dari layanan lain tanpa ada pengendali pusat. Pendekatan ini menciptakan *loose coupling* antar layanan, meningkatkan skalabilitas dan ketahanan sistem karena kegagalan satu layanan tidak langsung memengaruhi layanan lain. Namun, koreografi membutuhkan desain *event* dan komunikasi yang lebih kompleks, serta pemantauan yang lebih sulit karena tidak ada pusat pengendal. (Dragoni, et al., 2020)

Dalam praktiknya, kedua pendekatan tersebut sering digunakan secara bersamaan dalam arsitektur *microservices* untuk mengoptimalkan performa sistem. Orkestrasi digunakan pada bagian proses bisnis yang membutuhkan kontrol ketat dan urutan eksekusi yang jelas, sementara koreografi diterapkan pada interaksi *asynchronous* yang bersifat *loosely coupled* dan *event-driven*. (Richardson, 2021)

## **2.8. Pengertian Docker**

*Docker* adalah *platform open-source* yang memungkinkan pengembang untuk membangun, mengirim, dan menjalankan aplikasi di dalam *container* yang

ringan dan portabel. *Container* ini berfungsi sebagai lingkungan *virtual* terisolasi yang mengemas aplikasi beserta semua dependensinya termasuk *library*, *framework*, dan konfigurasi sehingga aplikasi dapat berjalan dengan konsisten di berbagai lingkungan tanpa perlu konfigurasi ulang. Hal ini sangat membantu dalam mengatasi masalah "*works on my machine*" yang sering terjadi saat aplikasi dipindahkan dari pengembangan ke produksi. (Setiawan & Hartono, 2022)

Selain itu, *Docker* menawarkan efisiensi sumber daya yang lebih baik dibandingkan dengan mesin virtual tradisional karena container berbagi kernel sistem operasi dengan host, sehingga ukurannya jauh lebih kecil dan proses startup-nya jauh lebih cepat. Keunggulan ini memungkinkan skalabilitas yang tinggi dan kemudahan integrasi dengan platform orkestrasi container seperti *Kubernetes*, yang membantu mengelola *deployment* aplikasi secara otomatis di lingkungan cloud. *Docker* juga mempercepat siklus pengembangan dan *deployment* dengan *pipeline* yang lebih otomatis dan konsisten, sehingga mendukung praktik *DevOps* secara efektif. (Lubis & Santoso, 2021)

## **2.9. Pengertian Database**

*Database* adalah kumpulan data yang disimpan secara sistematis dan terorganisir dalam sebuah sistem komputer, sehingga memudahkan proses penyimpanan, pengelolaan, dan pengambilan data secara efisien. *Database* dirancang untuk mendukung operasi *CRUD* (*Create, Read, Update, Delete*) dan digunakan dalam berbagai aplikasi untuk menyimpan informasi penting dengan struktur yang jelas dan konsisten. (Prasetya & Lestari, 2021)

*PostgreSQL* merupakan salah satu *RDBMS (Relational Database Management System)* yang cukup populer dan banyak digunakan pada beberapa instansi atau perusahaan dan bersifat open source. *PostgreSQL* menggunakan arsitektur *client-server* sehingga dapat dijalankan pada *server* atau komputer yang berbeda. (Juniarti, Pratama, Utami, Merlina, & Mayangky, 2023)

#### **2.10. Pengertian UML**

*UML (Unified Modeling Language)* adalah bahasa untuk visualisasi, spesifikasi, membangun sistem perangkat lunak, serta dokumentasi. *UML* menyediakan model- model yang tepat, tidak ambigu, dan lengkap. Secara khusus *UML* menspesifikasi langkahlangkah penting dalam pengembangan keputusan analisis, perancangan, serta implementasi dalam sistem perangkat lunak. (Nugroho & Rohimi, 2020)

#### **2.11. Diagram-Diagram UML**

*UML (Unified Modeling Language)* adalah bahasa untuk visualisasi, spesifikasi, membangun sistem perangkat lunak, serta dokumentasi. *UML* menyediakan model- model yang tepat, tidak ambigu, dan lengkap. Secara khusus *UML* menspesifikasi langkahlangkah penting dalam pengembangan keputusan analisis, perancangan, serta implementasi dalam sistem perangkat lunak. (Nugroho & Rohimi, 2020)

#### **2.11.1. Use Case Diagram**

*Use Case* pada dasarnya merupakan gambaran dari proses sistem secara keseluruhan yang melibatkan *actor* dalam hal pengguna. *Use case* adalah cara untuk menunjukan *stakeholder* sistem akan berinteraksi dengan sistem. Mengebangkan *use case* membantu memahami persyaratan sistem secara detail.

#### **2.11.2. Class Diagram**

Ini adalah diagram struktur statis yang menggambarkan struktur sistem dengan menunjukkan kelas sistem, atributnya, operasi (atau metode), dan hubungan antar kelas.

#### **2.11.3. Sequence Diagram**

*Sequence* diagram atau diagram urutan menunjukan interaksi objek yang diatur dalam urutan waktu. Ini menggambarkan objek dan kelas yang terlibat dalam skenario dan ukuran pesan yang dipertukarkan antara objek yang diperlukan untuk melaksanakan fungsi skenario.

#### **2.11.4. Activity Diagram**

*Activity* Diagram adalah sebuah cara untuk memodelkan aliran kerja (*workflow*) dari *use case* dalam bentuk grafik. Diagram ini menunjukan langkah-langkah di dalam aliran kerja, titik-titik keputusan didalam aliran kerja, siapa yang bertanggung jawab menyelesaikan masing-masing aktivitas, dan objek-objek yang digunakan dalam aliran kerja.

## 2.12. Penelitian

Berikut adalah beberapa referensi penelitian dari peneliti sebelumnya yang memiliki kaitannya dengan judul skripsi yang ditulis.

1. Berdasarkan proses dan hasil penelitian yang telah kami lakukan, yaitu pembuatan aplikasi menggunakan Steganografi dengan metode *LSB (least significant bit)* dan *visible watermark* untuk melindungi hak cipta berhasil. Aplikasi yang kami buat berhasil menyisipkan baik *visible watermark* dan pesan rahasia ke dalam sebuah gambar digital berformat *\*PNG* yang dimana hampir tidak ada perubahan yang terlihat dari sampul gambar selain penambahan *watermark*. Pesan rahasia yang disisipkan juga berhasil ditampilkan seutuhnya. Dengan hal ini, pengguna yang menyisipkan *steganomark* pada gambar yang mereka miliki dapat memiliki bukti hak cipta dari gambar tersebut.
2. Dari hasil penelitian yang telah dilakukan penulis, dapat ditarik kesimpulan bahwa sistem pengamanan dokumen dengan steganografi *Line Shifting* yang dibangun menggunakan bahasa *python* dapat di implementasikan untuk melakukan proses penyisipan pesan berupa teks ke dalam pdf. Dalam penelitian ada 1 kasus uji yang mengalami kegagalan saat proses *decode* pesan yaitu jika sebuah dokumen *pdf* yang hanya memiliki 1 halaman kemudian disisipi pesan sebanyak 35 karakter. Masih adanya karakter yang bukan merupakan karakter pesan yang ikut tercetak pada saat proses *decode*. (Saifurrohman, Poetro, & Taufik, 2023)
3. Aplikasi Pengamanan Dokumen *PDF* ini berhasil dirancang sehingga dapat membantu user dalam mengamankan data berbentuk dokumen *PDF* yang tidak

ingin disalahgunakan oleh pihak yang tidak bertanggungjawab. Teknik *watermarking* yang digunakan cukup membantu menunjukkan suatu hak cipta bahwa dokumen yang diupload adalah milik yang bersangkutan. Metode yang digunakan cukup membantu dalam mengenkripsi dan mendekripsi dokumen yang *diupload*. (Loekito, Indriyani, & Rozi, 2020)

4. Proses penyisipan menggunakan steganografi *EOF* mengharuskan teridentifikasinya byte terakhir pada file. Tujuan identifikasi tersebut memberikan posisi yang tepat dimana ciphertext akan disisipkan. Sedangkan metode enkripsi menggunakan algoritma *AES* menjadikan informasi tidak bisa dikenali atau dibaca. Dimana *AES-128* bekerja dengan panjang kunci 128 bit. (Pramudia, Fitriansyah, & Ramliana, 2021)
5. Waktu yang digunakan untuk melakukan proses *embed* dan *extract* berbanding lurus dengan ukuran *file* yang diproses. Semakin besar ukuran file yang diproses maka semakin lama proses *embed* dan *extract*, semakin kecil ukuran file yang diproses, semakin cepat proses *embed* dan *extract* dilakukan. Selanjutnya dengan adanya aplikasi steganograf dan kriptografi, proses penyimpanan informasi menjadi lebih aman dan yang terakhir proses *extract* dengan password yang asli akan mengembalikan file menjadi file semula tanpa mengalami perubahan sedikitpun. (Ridwan, Pattipeilohy, & Sanwani, 2020)

## **BAB III**

### **ANALISIS MASALAH DAN PERANCANGAN PROGRAM**

Seperti yang sudah disampaikan pada bab-bab sebelumnya, pada penelitian ini metode yang digunakan dalam pengembangan sistem adalah siklus *SDLC* (*Software Development Life Cycle*) model *waterfall* yang merupakan metode pendekatan pengembangan sistem informasi yang sistematis serta berurut mulai pada level *requirements, design, implementation, verification* dan *maintenance*

#### **3.1. Pengumpulan Kebutuhan (*requirements*)**

Pada tahap ini, dilakukan proses pengumpulan kebutuhan atau informasi berupa data yang berhubungan dengan penelitian.

##### **3.1.1. Metode Pengumpulan Data**

Metode pengumpulan data merupakan langkah awal yang dilakukan penulis

##### **1. Pengamatan Observasi**

Pengamatan atau observasi penulis lakukan melalui pengumpulan data dari berbagai berita dan sekaligus langsung menemui orang yang pernah mengalami kasus tersebut.

##### **2. Wawancara**

Wawancara dilakukan dengan mengambil sampel acak beberapa orang untuk mengidentifikasi kendala atau ketakutan yang sering terjadi.

Table 3.1. Wawancara

| No | Pertanyaan                                                                                          | Jawaban                                                                                                                                                                          |
|----|-----------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | Bagaimana respon anda dengan maraknya terjadi kasus pemalsuan data pribadi?                         | Sejujurnya takut juga mendengarnya, dikhawatirkan kami juga terkena kasus tersebut.                                                                                              |
| 2  | Apa yang ditakutkan dari kasus pemalsuan data pribadi ?                                             | Saya takut jika data saya digunakan untuk hal yang merugikan saya khususnya, seperti pinjaman online, saya takut tiba tiba ada yang nagih kerumah tanpa saya tahu itu dari mana. |
| 3  | Dari banyaknya kasus tersebut, apakah anda takut saat harus upload dokumen tertentu melalui online? | Pastinya! Saya punya rasa takut kesana, bahkan selalu bertanya teman yang sedikit paham tentang dunia teknologi                                                                  |

|   |                                                     |                                                                                                                                                                                              |
|---|-----------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4 | Apa harapan anda kedepanya untuk permasalahan ini ? | Saya berharap akan ada solusi untuk menyelesaikan permasalahan pemalsuan data tersebut, agar semua orang khususnya saya tidak khawatir saat mengupload dokumen pribadi atau dokumen penting. |
|---|-----------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### 3. Study Literatur

Studi literatur didapatkan dari beberapa referensi penelitian yang didapatkan dari jurnal penelitian dan teori lain yang menunjang dan membantu dalam penyusunan.

Table 3.2. Study Literatur

| No | Keterangan       |                                                                                                                                               |
|----|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | Penulis          | Muhammad Kailani Ridwan, William Frado Pattipeilohy, Sanwani.                                                                                 |
|    | Jenis Penelitian | Jurnal                                                                                                                                        |
|    | Tahun Penelitian | 2020                                                                                                                                          |
|    | Judul            | APLIKASI KEAMANAN <i>DOCUMENT</i> DIGITAL MENGGUNAKAN ALGORITMA STEGANOGRAFI <i>DISCRETE COSINE TRANSFORM (DCT)</i> PADA PERUSAHAN ALAT BERAT |

|   |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|---|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | Pembahasan       | <p>Waktu yang digunakan untuk melakukan proses <i>embed</i> dan <i>extract</i> berbanding lurus dengan ukuran file yang diproses. Semakin besar ukuran file yang diproses maka semakin lama proses embed dan extract, semakin kecil ukuran file yang diproses, semakin cepat proses embed dan extract dilakukan. Dengan adanya aplikasi steganografi dan kriptografi, proses penyimpanan informasi menjadi lebih aman dan yang terakhir Proses extract dengan password yang asli akan mengembalikan file menjadi file semula tanpa mengalami perubahan sedikitpun.</p> |
| 2 | Penulis          | Ahmad Saifurrohman, Bagus Satrio Waluyo Poetro, Moch Taufik                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|   | Jenis Penelitian | Jurnal                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|   | Tahun Penelitian | 2023                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|   | Judul            | SISTEM PENGAMANAN DOKUMEN PENGAJUAN HAK PATEN DENGAN MENGGUNAKAN METODE STEGANOGRAFI <i>LINE SHIFTING</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

|   |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | Pembahasan       | <p>Dari hasil penelitian yang telah dilakukan penulis, dapat ditarik kesimpulan bahwa sistem pengamanan dokumen dengan steganografi <i>Line Shifting</i> yang dibangun menggunakan bahasa python dapat di implementasikan untuk melakukan proses penyisipan pesan berupa teks ke dalam <i>pdf</i>. Dalam penelitian ada 1 kasus uji yang mengalami kegagalan saat proses <i>decode</i> pesan yaitu jika sebuah dokumen <i>pdf</i> yang hanya memiliki 1 halaman kemudian disisipi pesan sebanyak 35 karakter. Masih adanya karakter yang bukan merupakan karakter pesan yang ikut tercetak pada saat proses <i>decode</i></p> |
| 3 | Penulis          | Chelsy Elisabet Gultoma, I Ketut Gede Suhartanaa                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|   | Jenis Penelitian | Jurnal                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|   | Tahun Penelitian | 2023                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|   | Judul            | Penerapan Steganografi dan <i>Visible Watermarking</i> Pada Gambar Digital Untuk Perlindungan Hak Cipta                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

|   |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|---|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | Pembahasan       | Berdasarkan proses dan hasil penelitian yang telah kami lakukan, yaitu pembuatan aplikasi menggunakan Steganografi dengan metode <i>LSB (least significant bit)</i> dan <i>visible watermark</i> untuk melindungi hak cipta berhasil. Aplikasi yang kami buat berhasil menyisipkan baik <i>visible watermark</i> dan pesan rahasia ke dalam sebuah gambar digital berformat <i>*PNG</i> yang dimana hampir tidak ada perubahan yang terlihat dari sampul gambar selain penambahan watermark |
| 4 | Penulis          | Dika Pramudia, Aswin Fitriansyah, Randi Ramliyana                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|   | Jenis Penelitian | Jurnal                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|   | Tahun Penelitian | 2021                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|   | Judul            | APLIKASI <i>HYBRID</i> STEGANOGRAFI <i>EOF</i> DAN ENKRIPSI <i>AES-128</i> UNTUK KEAMANAN <i>FILE PDF ANDROID</i>                                                                                                                                                                                                                                                                                                                                                                           |

|   |                  |                                                                                                                                                                                                                                                                                                                                                    |
|---|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | Pembahasan       | Proses penyisipan menggunakan steganografi <i>EOF</i> mengharuskan teridentifikasinya <i>byte</i> terakhir pada <i>file</i> . Tujuan identifikasi tersebut memberikan posisi yang tepat dimana <i>ciphertext</i> akan disisipkan. Sedangkan metode enkripsi menggunakan algoritma <i>AES</i> menjadikan informasi tidak bisa dikenali atau dibaca. |
| 5 | Penulis          | Chandra Himmawan Loekito, Tutuk Indriyani, Nanang Fakhur Rozi                                                                                                                                                                                                                                                                                      |
|   | Jenis Penelitian | Jurnal                                                                                                                                                                                                                                                                                                                                             |
|   | Tahun Penelitian | 2020                                                                                                                                                                                                                                                                                                                                               |
|   | Judul            | Aplikasi Pengamanan Dokumen <i>PDF</i> dengan Teknik <i>Watermarking</i> Menggunakan Metode <i>Serpent Cipher</i>                                                                                                                                                                                                                                  |

|            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Pembahasan | <p>Aplikasi Pengamanan Dokumen <i>PDF</i> ini berhasil dirancang sehingga dapat membantu user dalam mengamankan data berbentuk dokumen <i>PDF</i> yang tidak ingin disalahgunakan oleh pihak yang tidak bertanggungjawab. Teknik <i>watermarking</i> yang digunakan cukup membantu menunjukkan suatu hak cipta bahwa dokumen yang diupload adalah milik yang bersangkutan. Metode yang digunakan cukup membantu dalam mengenkripsi dan mendekripsi dokumen yang diupload.</p> |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### 3.1.2. Analisis Kebutuhan Sistem

Berdasarkan data yang dikumpulkan, kebutuhan sistem dibagi menjadi dua bagian, yaitu kebutuhan fungsional dan kebutuhan non-fungsional:

#### A. Kebutuhan Fungsional:

- Sistem harus dapat melakukan *watermark* pada dokumen digital dari sumber *file PDF* atau gambar.
- Sistem harus mampu melakukan verifikasi keaslian dokumen melalui scanning *qr code* atau pengunggahan *file*.
- Sistem harus menyediakan fitur input dan *output* dokumen dalam format *PDF*.
- Sistem harus menyimpan metadata dokumen untuk proses validasi.

#### B. Kebutuhan Non-Fungsional:

- Sistem harus memiliki antarmuka yang mudah digunakan dan responsif.
- Sistem harus aman dari akses tidak sah dan menjaga kerahasiaan data pengguna.
- Sistem harus berjalan pada *platform mobile* (*Android* dan *iOS*).
- Sistem harus memiliki performa yang cepat dan efisien dalam proses *watermark* dan verifikasi.

### 3.2. Usulan Program

Sebagai tindak lanjut dari hasil analisis kebutuhan sistem, maka dirancang sebuah aplikasi yang mampu menjawab tantangan pengamanan dokumen baik secara digital maupun fisik. Usulan program ini mencakup penerapan beberapa teknologi utama yang dapat saling mendukung dalam menjaga keaslian, kerahasiaan, dan integritas dokumen.

Aplikasi ini akan mengimplementasikan kombinasi fitur seperti penambahan *metadata*, penyisipan *watermark*, serta penerapan steganografi struktural pada dokumen digital. Pendekatan ini bertujuan untuk memberikan lapisan keamanan yang tidak hanya terlihat secara visual, tetapi juga tersimpan secara tersembunyi di dalam struktur *file*.

### 3.2.1. Tujuan Program

Program ini bertujuan untuk membantu masyarakat menjaga keamanan dokumen pribadi maupun resmi dari pemalsuan dan penyalahgunaan, dengan menggunakan *watermark*, *metadata*, dan teknik steganografi.

### 3.2.2. Fitur yang Diusulkan

Aplikasi ini diusulkan dengan fitur utama yang berfokus pada keamanan dan keaslian dokumen, baik secara digital maupun fisik. Adapun fitur-fitur tersebut adalah:

#### A. Penyisipan *Metadata* Otentik

Sistem akan menyematkan *metadata* penting ke dalam *file* digital yang tidak terlihat secara langsung, seperti:

- Pemilik dan penerima dokumen
- Tanggal pembuatan dan UUID unik
- Informasi tambahan sebagai kebutuhan verifikasi

#### B. Penerapan *Watermark*

Aplikasi menyediakan dua jenis *watermark*:

- Teks: Terlihat jelas seperti keaslian yang menegaskan kepemilikan.
- Kode *QR/Barcode*: Berisi *URL* verifikasi atau *UUID*, dapat dipindai untuk membuka halaman detail.

#### C. Steganografi Struktural Menggunakan *XObject (PDF)*

Fitur ini menggunakan teknik struktural steganografi dengan menyisipkan informasi penting dalam bagian tersembunyi struktur *PDF*, yaitu melalui objek *XObject*:

- *XObject* adalah elemen internal *PDF* (biasanya gambar atau stream) yang bisa disisipi data tambahan tanpa mengubah tampilan visual.
- Informasi disematkan dalam stream *XObject* menggunakan format *json*.
- Karena tidak disisipkan di *layer* visual, maka keberadaannya tak mengganggu tampilan dokumen sama sekali.

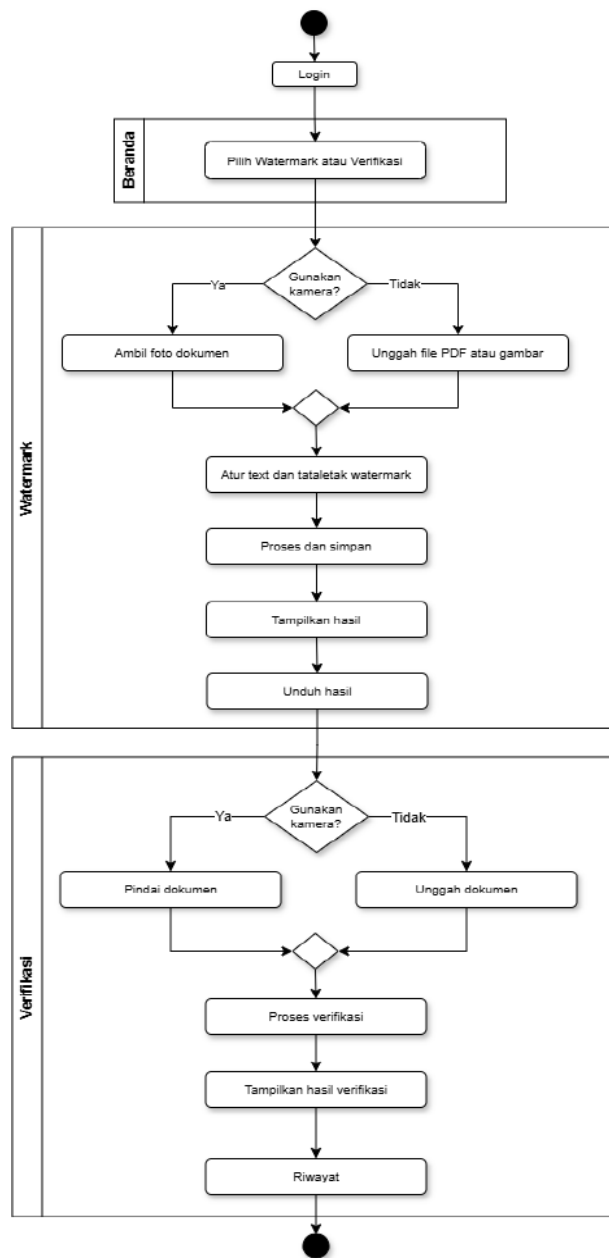
#### D. Verifikasi Digital & Fisik

- Verifikasi Digital: *File* dapat diunggah ke aplikasi untuk proses pemeriksaan *metadata*, *watermark*, dan steganografi.
- Verifikasi Fisik: Foto hasil cetak dokumen dapat dipindai untuk membaca tanda *watermark* visual atau mencocokkan *UUID*/kode.

#### E. Pelacakan Riwayat Dokumen

Sistem menyimpan setiap aktivitas penting terhadap dokumen, seperti pembuatan dan verifikasi. Ini menciptakan rekam jejak digital yang bisa ditelusuri

### 3.2.3. Flowchart Usulan Program



Gambar 3.1. Flowchart Usulan Program

Berdasarkan gambar diatas, usulan sistem dapat diuraikan sebagai berikut:

- User harus terlebih dahulu login untuk masuk ke beranda aplikasi
- Selanjutnya user memilih apakah akan melakukan *watermark* atau memverifikasi dokumen

- Jika memilih *watermak*, maka selanjutnya user mengupload dokumennya menggunakan 2 cara yakni, menggunakan camera *handphone* atau unggah menggunakan file *PDF* atau gambar
- Jika dokumen sudah berhasil diunggah, maka Langkah selanjutnya adalah atur text dan tataletak *watermak* dalam dokumen.
- Jika sudah selesai, selanjutnya sistem akan melakukan pemerosesan data, diantaranya menambahkan *metadata* dan *steganography* terlebih dahulu, jika sudah selesai sistem akan menampilkan hasilnya
- User membagikan dokumen yang sudah di *watermark*

#### **3.2.4. Deskripsi Singkat Aplikasi**

Adalah aplikasi *mobile* yang dirancang untuk membantu masyarakat dalam menjaga keamanan dokumen digital maupun fisik. Dengan menggunakan teknologi *watermark*, *metadata*, dan steganografi, aplikasi ini mampu melindungi dokumen dari pemalsuan dan penyalahgunaan. Pengguna dapat dengan mudah memberikan tanda khusus pada dokumen, kemudian melakukan verifikasi dengan cepat hanya melalui pemindaian code atau pengunggahan *file* dokumen

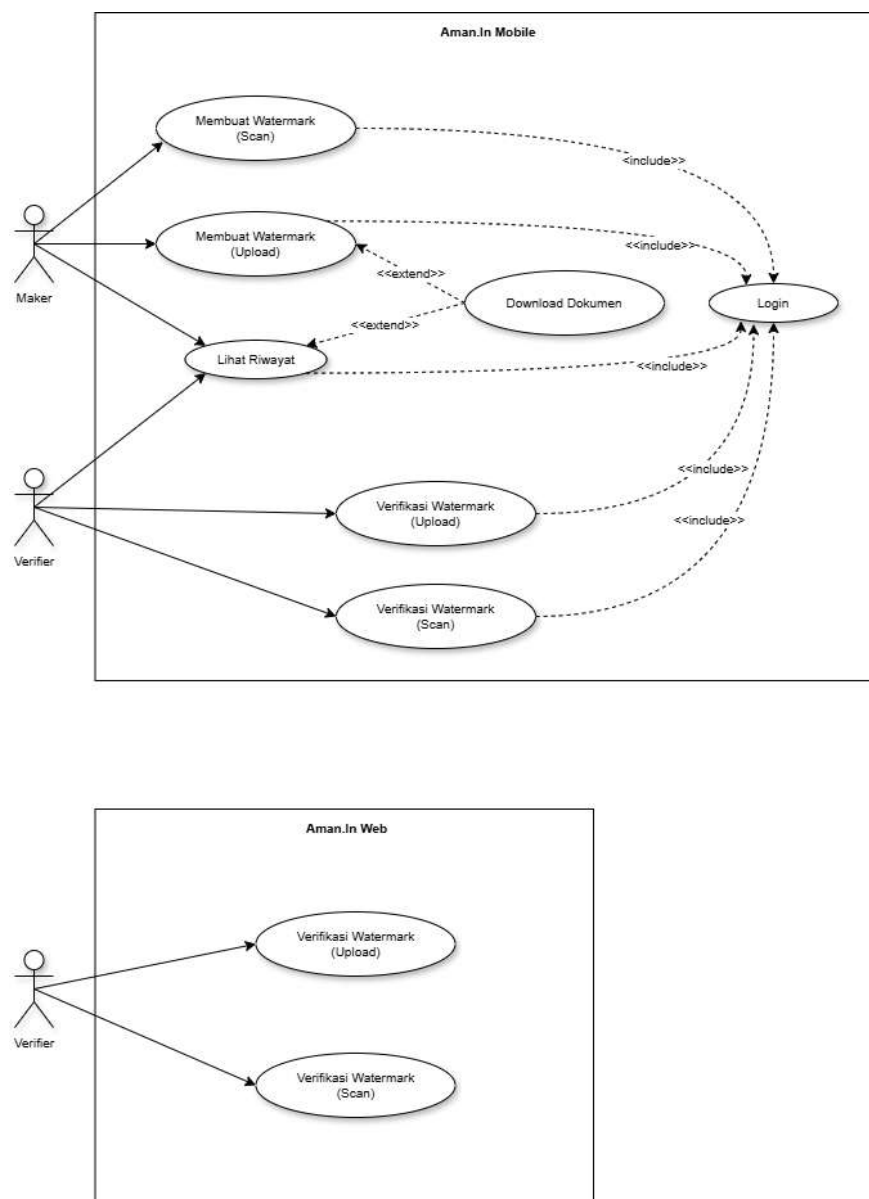
### **3.3. Desain Sistem perangkat lunak**

Pada tahap kedua dalam metode *waterfall* yaitu design. Pada tahap ini dilakukan perancangan sistem yang akan dibangun menggunakan UML (*Unified Modeling Language*), meliputi *use case diagram*, *class diagram*, *activity diagram*,

dan *sequence diagram*. Kemudian *ERD* (*Entity Relationship Diagram*) dan rancangan database sistem nya.

### 3.3.1. Desain *Use Case Diagram*

*Use case diagram* disini akan mendeskripsikan sebuah interaksi antara dua aktor yang akan berhubungan dengan sistem informasi yang akan dibuat.



Gambar 3.2. Use Case Diagram

*Use case* diagram pada diatas menjelaskan interaksi antara aktor dengan sistem yang dikembangkan yang terdiri dari dua platform, yaitu *Mobile* dan *Web*. Sistem ini memiliki dua jenis pengguna utama yaitu *Maker* dan *Verifier*, dengan fungsi yang berbeda namun saling melengkapi dalam proses pengamanan dan verifikasi dokumen.

#### A. *Mobile*

Pada platform mobile, terdapat dua aktor: *Maker* dan *Verifier*.

##### **Aktor: *Maker***

*Maker* adalah pengguna yang bertugas membuat watermark pada dokumen atau gambar. Ia memiliki beberapa fungsi utama:

- **Membuat *Watermark* (*Scan*)**

*Maker* dapat membuat watermark dengan memindai dokumen fisik menggunakan kamera.

Use case ini meng-*include* Login, karena proses ini hanya bisa dilakukan oleh pengguna yang sudah login.

- **Membuat *Watermark* (*Upload*)**

*Maker* juga dapat membuat watermark dengan mengunggah file digital (gambar atau PDF).

Use case ini memiliki relasi *extend* ke Download Dokumen, yang berarti setelah membuat watermark, pengguna dapat mengunduh file hasil *watermark*. Juga meng-*include* Login.

- **Lihat Riwayat**

*Maker* dapat melihat histori watermark yang pernah dibuat.

Fitur ini juga meng-include Login.

**Aktor: *Verifier***

*Verifier* adalah pengguna yang bertugas memverifikasi dokumen yang telah diberi watermark. Ia memiliki dua fungsi:

- **Verifikasi *Watermark* (*Upload*)**

*Verifier* mengunggah file digital untuk memverifikasi keberadaan dan keaslian watermark.

Use case ini meng-include Login.

- **Verifikasi *Watermark* (*Scan*)**

*Verifier* memindai dokumen fisik menggunakan kamera untuk memverifikasi *watermark*.

Use case ini juga meng-include Login.

**B. *Web***

Platform *web* hanya digunakan oleh *Verifier*, dengan fungsi yang sama seperti di *mobile*:

- **Verifikasi *Watermark* (*Upload*)**

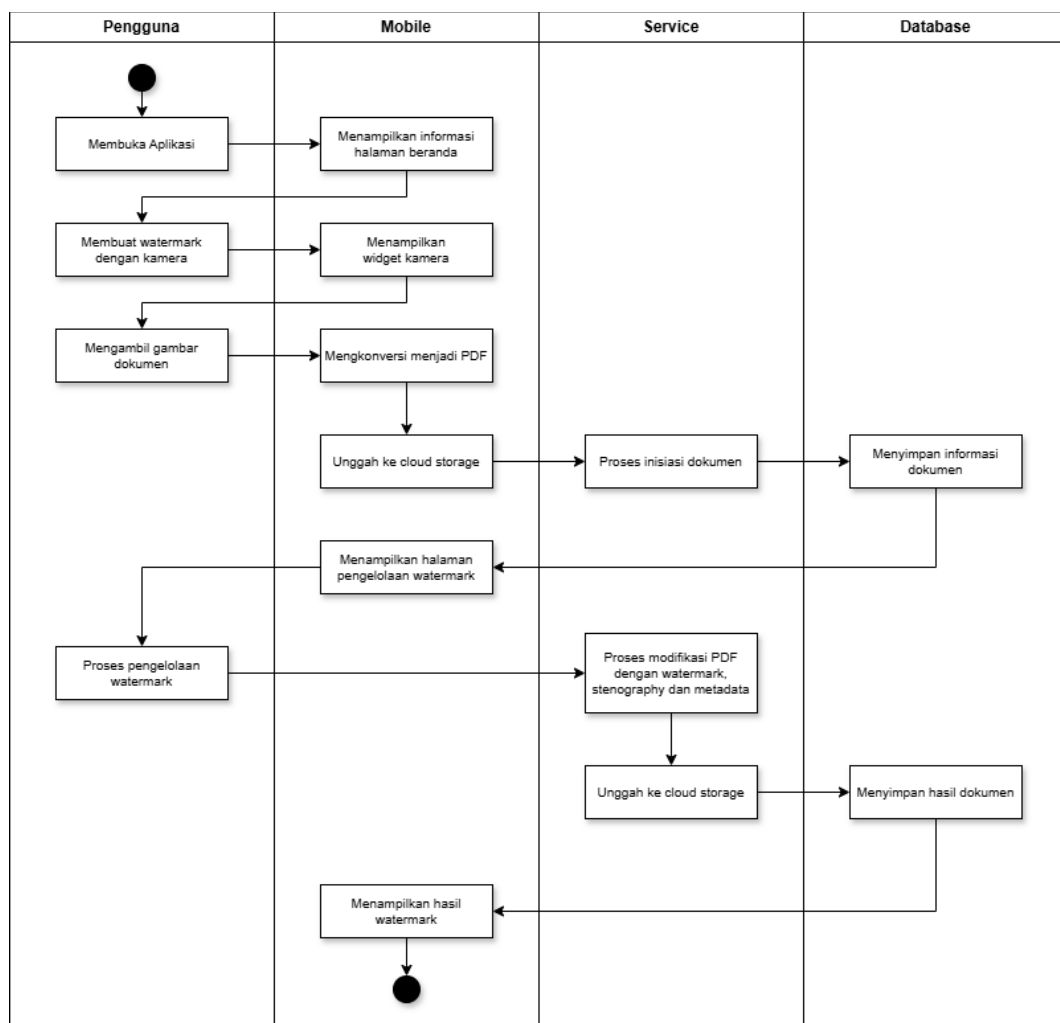
Sama seperti di *mobile*, *Verifier* dapat mengunggah file digital untuk verifikasi watermark.

- **Verifikasi *Watermark* (*Scan*)**

*Verifier* juga dapat menggunakan fitur pemindaian melalui *web*, dengan bantuan perangkat pendukung (misalnya *qr code reader*).

### 3.3.2. Desain *Activity Diagram*

#### A. *Proses Watermark*



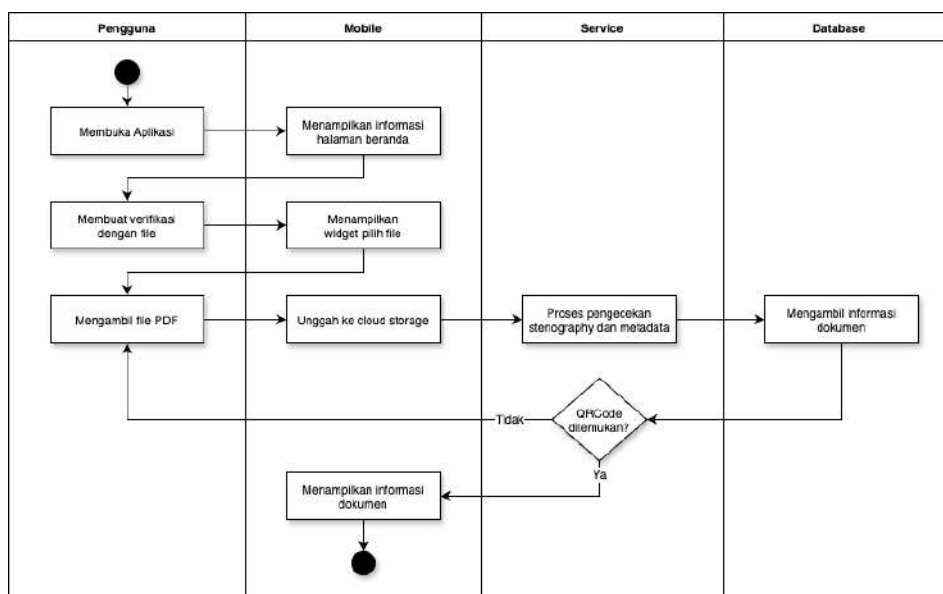
Gambar 3.3. Activity Diagram Proses Watermark

Diagram aktivitas ini menunjukkan alur kerja pembuatan *watermark* pada dokumen dengan metode pengambilan gambar melalui kamera (*scan*).

- Proses dimulai ketika pengguna membuka aplikasi dan memilih fitur *watermark* melalui kamera.
- Aplikasi mengaktifkan kamera, kemudian hasil tangkapan dikonversi ke *PDF*.
- *File* dikirim ke *server* dan diproses untuk ditambahkan *watermark*, *metadata*, dan steganografi.
- Hasil akhir disimpan di *cloud storage* dan ditampilkan kembali ke pengguna.

Diagram ini memetakan alur teknis lintas komponen (pengguna, aplikasi *mobile*, *service*, dan *database*) secara urut dari awal hingga akhir proses.

## B. Proses Verifikasi



Gambar 3.4. Activity Diagram Proses Verifikasi

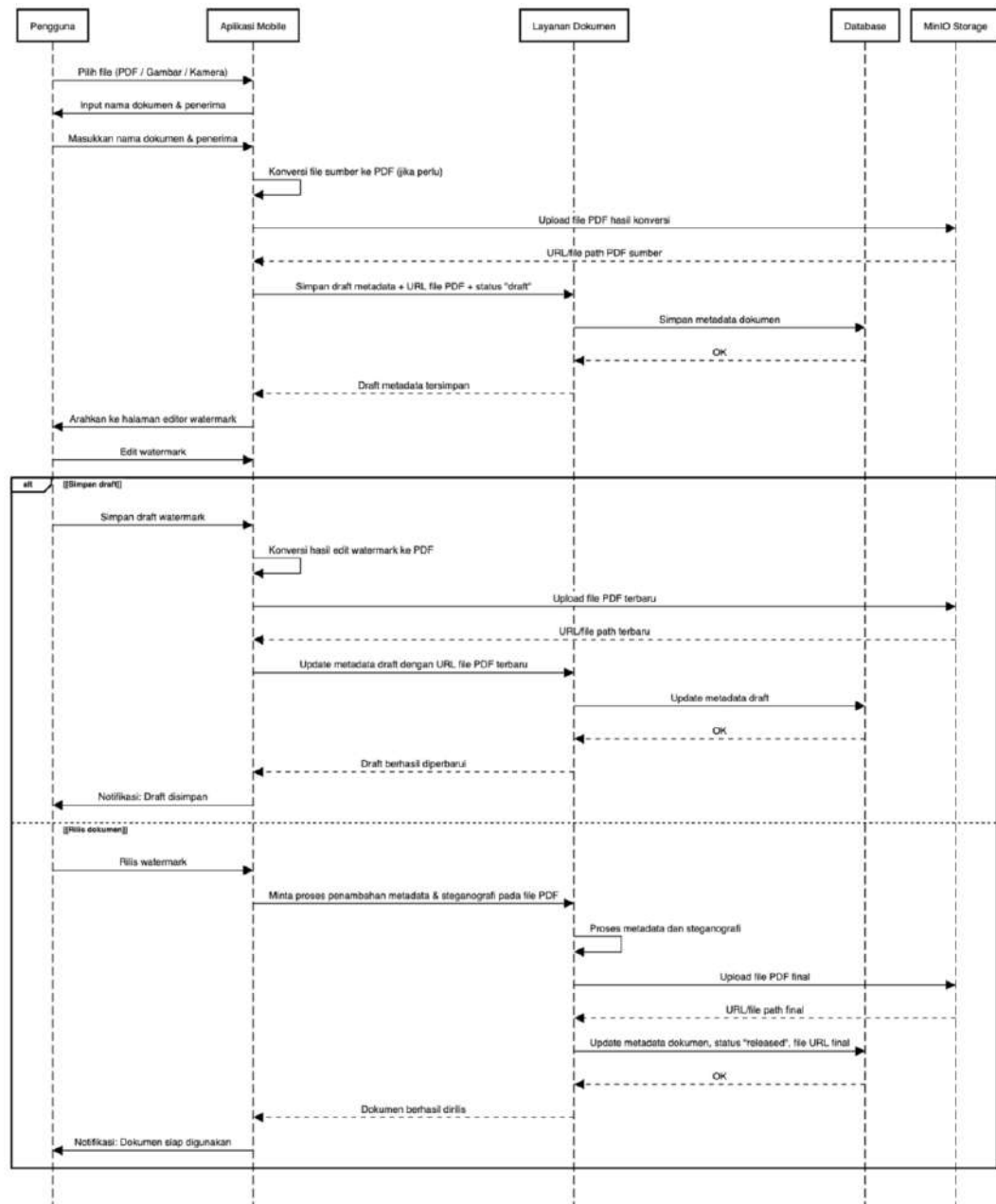
Diagram ini menjelaskan alur proses verifikasi dokumen melalui file *PDF* yang diunggah oleh pengguna menggunakan aplikasi *Mobile*.

- Proses dimulai ketika pengguna membuka aplikasi dan memilih menu verifikasi dokumen.
- Pengguna memilih file *PDF* dari perangkatnya.
- *File* tersebut diunggah ke cloud storage dan diteruskan ke *server* untuk diperiksa.
- *Server* melakukan pengecekan terhadap informasi *watermark* yang disisipkan melalui teknik steganografi dan metadata.
- Jika *QR Code* atau penanda digital ditemukan, sistem akan mengambil informasi terkait dari *database*.
- Informasi hasil verifikasi kemudian ditampilkan kepada pengguna.

Diagram ini menunjukkan bagaimana sistem mendeteksi keaslian dokumen melalui pemeriksaan tersembunyi, bukan hanya tampilan fisik, yang menjamin integritas data digital secara lebih dalam.

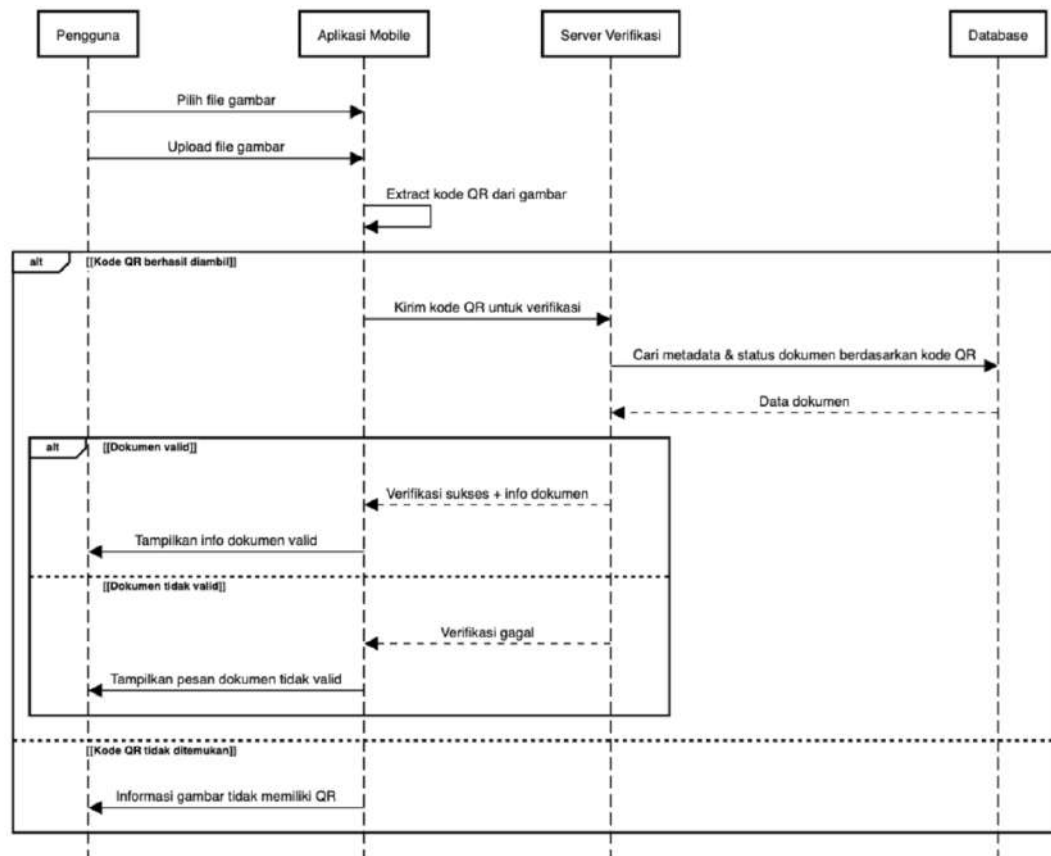
### 3.3.3. Desain *Sequence Diagram*

#### A. Proses *Watermark*

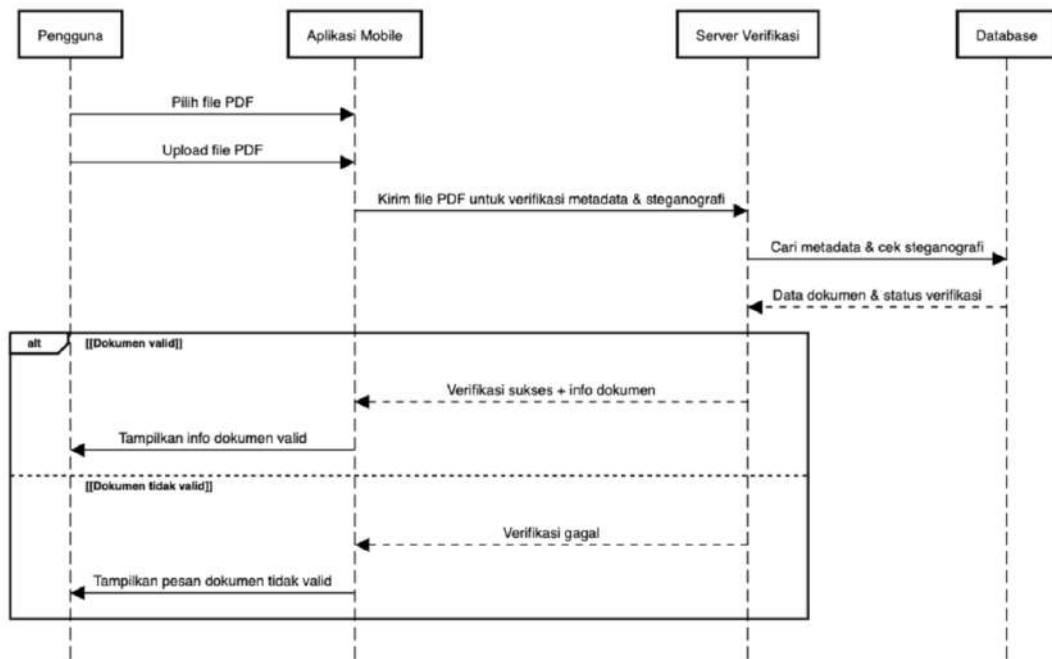


Gambar 3.5. *Sequence Diagram Proses Watermark*

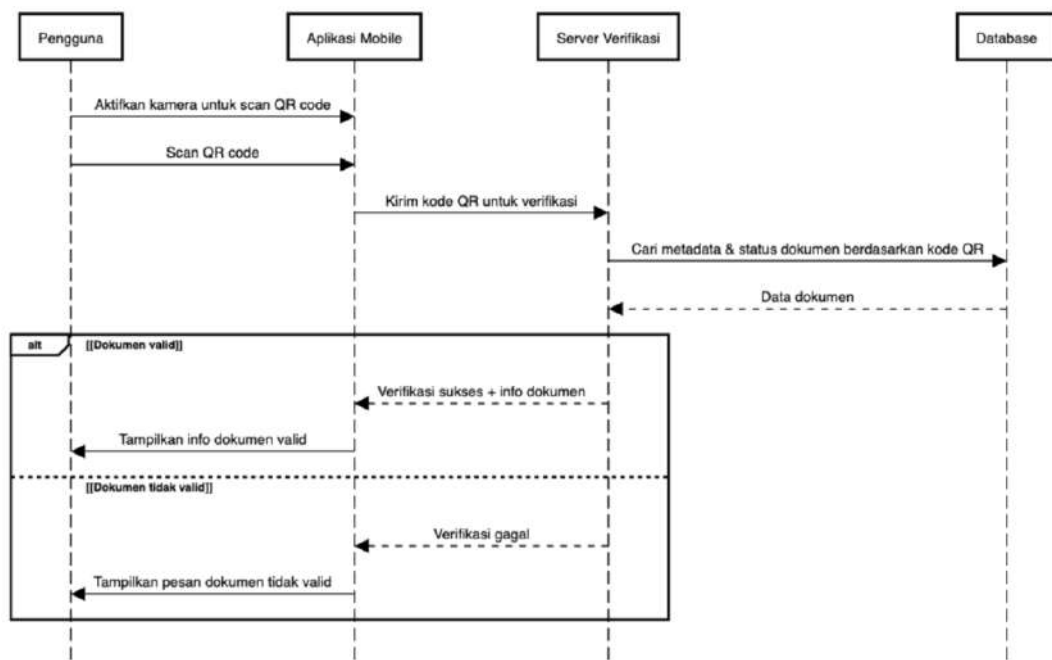
## B. Proses Verifikasi



Gambar 3.6. Sequence Diagram Proses Verifikasi dengan Gambar

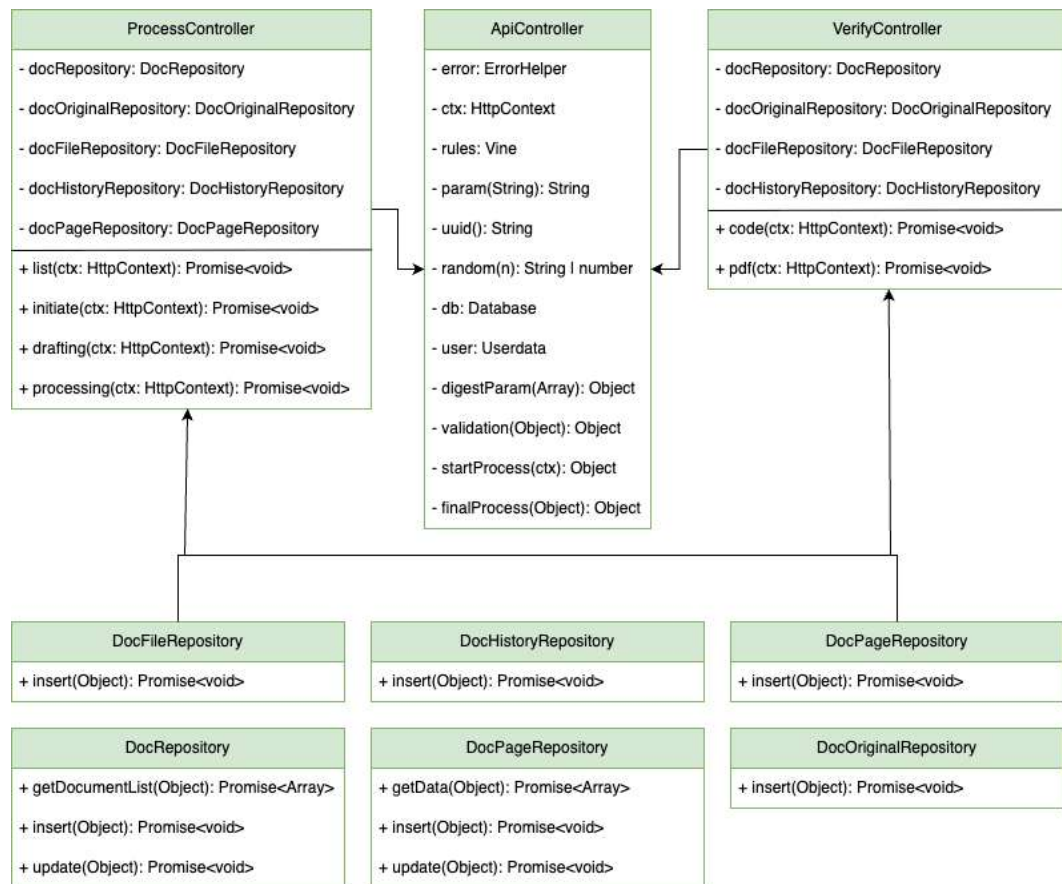


Gambar 3.7. Sequence Diagram Proses Verifikasi dengan File PDF



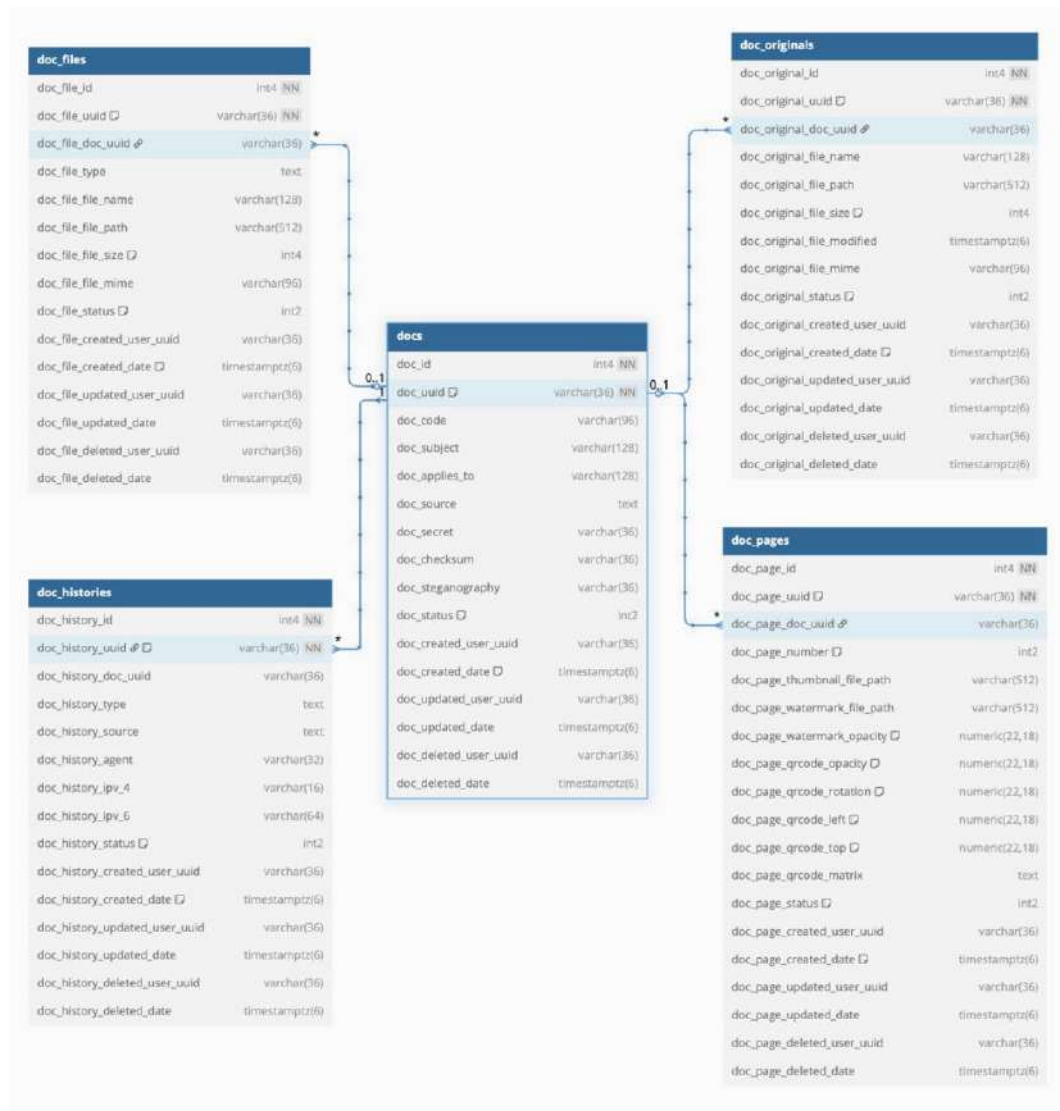
Gambar 3.8. Sequence Diagram Proses Verifikasi dengan Scan QrCode

### 3.3.4. Desain Class Diagram



Gambar 3.9. Desain Class Diagram

### 3.3.5. Desain Entity Relationship Diagram (ERD)



Gambar 3.10. Desain Entity Relationship Diagram

### 3.3.6. Struktur Basis Data

Perancangan basis data dalam sistem ini mengikuti pola penamaan yang konsisten untuk setiap tabel. Setiap nama *field* diawali dengan *prefix* nama tabel dalam bentuk tunggal, diikuti dengan nama atribut yang menggambarkan isi kolom.

Pendekatan ini bertujuan untuk menjaga keterbacaan, memudahkan pengelolaan, dan menghindari konflik nama antar tabel dalam skema basis data.

Beberapa aturan pola penamaan yang diterapkan di seluruh tabel antara lain:

- Kolom Identitas Awal

Setiap tabel selalu diawali dengan dua kolom utama:

- *prefix\_id*: Merupakan *primary key* bertipe *integer/autoincrement*.
- *prefix\_uuid*: Merupakan *unique identifier* bertipe UUID, digunakan sebagai referensi aman lintas sistem.

- Kolom Metadata Standar

Di akhir setiap tabel selalu terdapat kolom untuk mencatat informasi historis dan status data:

- *prefix\_created\_user\_uuid*: *UUID* pengguna yang membuat entri.
- *prefix\_created\_date*: Tanggal dan waktu entri dibuat.
- *prefix\_updated\_user\_uuid*: *UUID* pengguna terakhir yang mengubah data.
- *prefix\_updated\_date*: Tanggal dan waktu perubahan terakhir.
- *prefix\_deleted\_user\_uuid*: *UUID* pengguna yang menghapus (*soft delete*).
- *prefix\_deleted\_date*: Waktu penghapusan dilakukan.

Pola ini tidak hanya meningkatkan keteraturan struktur *database*, tetapi juga memudahkan pelacakan jejak data (*audit trail*) dan penerapan fitur *soft delete* tanpa kehilangan histori data.

Contoh penerapan pada tabel *documents*:

Table 3.3. Struktur Field Tabel Basis Data

|                                 |
|---------------------------------|
| <i>prefix_id</i>                |
| <i>prefix_uuid</i>              |
| ...                             |
| <i>prefix_created_user_uuid</i> |
| <i>prefix_created_date</i>      |
| <i>prefix_updated_user_uuid</i> |
| <i>prefix_updated_date</i>      |
| <i>prefix_deleted_user_uuid</i> |
| <i>prefix_deleted_date</i>      |

Berikut adalah daftar *table* basis data yang dibuat sesuai dengan rancangan untuk kebutuhan Pembangunan aplikasi ini:

## 1. Skema System

### a. Tabel *applications*

Table 3.4. Tabel *applications*

| <b>Nama</b>                    | <b>Tipe Data</b>         | <b>Ukuran</b> |
|--------------------------------|--------------------------|---------------|
| <i>application_id</i>          | <i>integer</i>           |               |
| <i>application_uuid</i>        | <i>character varying</i> | 36            |
| <i>application_code</i>        | <i>character varying</i> | 96            |
| <i>application_hash</i>        | <i>character varying</i> | 16            |
| <i>application_name</i>        | <i>character varying</i> | 256           |
| <i>application_description</i> | <i>text</i>              |               |
| <i>application_date_start</i>  | <i>date</i>              |               |
| <i>application_date_end</i>    | <i>date</i>              |               |
| <i>application_status</i>      | <i>smallint</i>          |               |

|                                      |                                 |    |
|--------------------------------------|---------------------------------|----|
| <i>application_created_user_uuid</i> | <i>character varying</i>        | 36 |
| <i>application_created_date</i>      | <i>timestamp with time zone</i> |    |
| <i>application_updated_user_uuid</i> | <i>character varying</i>        | 36 |
| <i>application_updated_date</i>      | <i>timestamp with time zone</i> |    |
| <i>application_deleted_user_uuid</i> | <i>character varying</i>        | 36 |
| <i>application_deleted_date</i>      | <i>timestamp with time zone</i> |    |

b. Tabel *client\_activities*

Table 3.5. Tabel *client\_activities*

| <b>Nama</b>                              | <b>Tipe Data</b>                | <b>Ukuran</b> |
|------------------------------------------|---------------------------------|---------------|
| <i>client_activity_id</i>                | <i>integer</i>                  |               |
| <i>client_activity_uuid</i>              | <i>character varying</i>        | 36            |
| <i>client_activity_client_token_uuid</i> | <i>character varying</i>        | 36            |
| <i>client_activity_ip_v4</i>             | <i>character varying</i>        | 16            |
| <i>client_activity_ip_v6</i>             | <i>character varying</i>        | 64            |
| <i>client_activity_port</i>              | <i>integer</i>                  |               |
| <i>client_activity_method</i>            | <i>character varying</i>        | 8             |
| <i>client_activity_uri</i>               | <i>character varying</i>        | 256           |
| <i>client_activity_user_agent</i>        | <i>character varying</i>        | 256           |
| <i>client_activity_request_time</i>      | <i>integer</i>                  |               |
| <i>client_activity_status</i>            | <i>smallint</i>                 |               |
| <i>client_activity_created_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>client_activity_created_date</i>      | <i>timestamp with time zone</i> |               |
| <i>client_activity_updated_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>client_activity_updated_date</i>      | <i>timestamp with time zone</i> |               |
| <i>client_activity_deleted_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>client_activity_deleted_date</i>      | <i>timestamp with time zone</i> |               |

c. Tabel *client\_tokens*

Table 3.6. Tabel *client\_tokens*

| <b>Nama</b> | <b>Tipe Data</b> | <b>Ukuran</b> |
|-------------|------------------|---------------|
|-------------|------------------|---------------|

|                                       |                                 |      |
|---------------------------------------|---------------------------------|------|
| <i>client_token_id</i>                | <i>integer</i>                  |      |
| <i>client_token_uuid</i>              | <i>character varying</i>        | 36   |
| <i>client_token_client_uuid</i>       | <i>character varying</i>        | 36   |
| <i>client_token_key</i>               | <i>character varying</i>        | 1024 |
| <i>client_token_expired</i>           | <i>timestamp with time zone</i> |      |
| <i>client_token_unique</i>            | <i>character varying</i>        | 256  |
| <i>client_token_status</i>            | <i>smallint</i>                 |      |
| <i>client_token_created_user_uuid</i> | <i>character varying</i>        | 36   |
| <i>client_token_created_date</i>      | <i>timestamp with time zone</i> |      |
| <i>client_token_updated_user_uuid</i> | <i>character varying</i>        | 36   |
| <i>client_token_updated_date</i>      | <i>timestamp with time zone</i> |      |
| <i>client_token_deleted_user_uuid</i> | <i>character varying</i>        | 36   |
| <i>client_token_deleted_date</i>      | <i>timestamp with time zone</i> |      |

d. Table *clients*

Table 3.7. Table *clients*

| <b>Nama</b>                     | <b>Tipe Data</b>                | <b>Ukuran</b> |
|---------------------------------|---------------------------------|---------------|
| <i>client_id</i>                | <i>integer</i>                  |               |
| <i>client_uuid</i>              | <i>character varying</i>        | 36            |
| <i>client_service_uuid</i>      | <i>character varying</i>        | 36            |
| <i>client_application_uuid</i>  | <i>character varying</i>        | 36            |
| <i>client_ip_v4</i>             | <i>character varying</i>        | 16            |
| <i>client_ip_v6</i>             | <i>character varying</i>        | 64            |
| <i>client_mac_address</i>       | <i>character varying</i>        | 32            |
| <i>client_status</i>            | <i>smallint</i>                 |               |
| <i>client_created_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>client_created_date</i>      | <i>timestamp with time zone</i> |               |
| <i>client_updated_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>client_updated_date</i>      | <i>timestamp with time zone</i> |               |
| <i>client_deleted_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>client_deleted_date</i>      | <i>timestamp with time zone</i> |               |

e. Table *services*

Table 3.8. Table *services*

| <b>Nama</b>                      | <b>Tipe Data</b>                | <b>Ukuran</b> |
|----------------------------------|---------------------------------|---------------|
| <i>service_id</i>                | <i>integer</i>                  |               |
| <i>service_uuid</i>              | <i>character varying</i>        | 36            |
| <i>service_code</i>              | <i>character varying</i>        | 96            |
| <i>service_name</i>              | <i>character varying</i>        | 128           |
| <i>service_description</i>       | <i>text</i>                     |               |
| <i>service_date_start</i>        | <i>timestamp with time zone</i> |               |
| <i>service_date_end</i>          | <i>timestamp with time zone</i> |               |
| <i>service_status</i>            | <i>smallint</i>                 |               |
| <i>service_created_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>service_created_date</i>      | <i>timestamp with time zone</i> |               |
| <i>service_updated_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>service_updated_date</i>      | <i>timestamp with time zone</i> |               |
| <i>service_deleted_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>service_deleted_date</i>      | <i>timestamp with time zone</i> |               |

f. Table *user\_aliases*

Table 3.9. Table *user\_aliases*

| <b>Nama</b>                         | <b>Tipe Data</b>                | <b>Ukuran</b> |
|-------------------------------------|---------------------------------|---------------|
| <i>user_alias_id</i>                | <i>integer</i>                  |               |
| <i>user_alias_uuid</i>              | <i>character varying</i>        | 36            |
| <i>user_alias_type</i>              | <i>smallint</i>                 |               |
| <i>user_alias_identity</i>          | <i>character varying</i>        | 64            |
| <i>user_alias_date_start</i>        | <i>timestamp with time zone</i> |               |
| <i>user_alias_date_end</i>          | <i>timestamp with time zone</i> |               |
| <i>user_alias_status</i>            | <i>smallint</i>                 |               |
| <i>user_alias_created_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>user_alias_created_date</i>      | <i>timestamp with time zone</i> |               |
| <i>user_alias_updated_user_uuid</i> | <i>character varying</i>        | 36            |

|                                     |                                 |    |
|-------------------------------------|---------------------------------|----|
| <i>user_alias_updated_date</i>      | <i>timestamp with time zone</i> |    |
| <i>user_alias_deleted_user_uuid</i> | <i>character varying</i>        | 36 |
| <i>user_alias_deleted_date</i>      | <i>timestamp with time zone</i> |    |

g. Table *user\_attempts*

Table 3.10. Table *user\_attempts*

| <b>Nama</b>                           | <b>Tipe Data</b>                | <b>Ukuran</b> |
|---------------------------------------|---------------------------------|---------------|
| <i>user_attempt_id</i>                | <i>integer</i>                  |               |
| <i>user_attempt_uuid</i>              | <i>character varying</i>        | 36            |
| <i>user_attempt_username</i>          | <i>character varying</i>        | 256           |
| <i>user_attempt_time</i>              | <i>smallint</i>                 |               |
| <i>user_attempt_date</i>              | <i>timestamp with time zone</i> |               |
| <i>user_attempt_status</i>            | <i>smallint</i>                 |               |
| <i>user_attempt_created_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>user_attempt_created_date</i>      | <i>timestamp with time zone</i> |               |
| <i>user_attempt_updated_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>user_attempt_updated_date</i>      | <i>timestamp with time zone</i> |               |
| <i>user_attempt_deleted_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>user_attempt_deleted_date</i>      | <i>timestamp with time zone</i> |               |

h. Table *user\_passwords*

Table 3.11. Table *user\_passwords*

| <b>Nama</b>                            | <b>Tipe Data</b>                | <b>Ukuran</b> |
|----------------------------------------|---------------------------------|---------------|
| <i>user_password_id</i>                | <i>integer</i>                  |               |
| <i>user_password_uuid</i>              | <i>character varying</i>        | 36            |
| <i>user_password_user_uuid</i>         | <i>character varying</i>        | 36            |
| <i>user_password_key</i>               | <i>character varying</i>        | 256           |
| <i>user_password_type</i>              | <i>smallint</i>                 |               |
| <i>user_password_status</i>            | <i>smallint</i>                 |               |
| <i>user_password_created_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>user_password_created_date</i>      | <i>timestamp with time zone</i> |               |

|                                        |                                 |    |
|----------------------------------------|---------------------------------|----|
| <i>user_password_updated_user_uuid</i> | <i>character varying</i>        | 36 |
| <i>user_password_updated_date</i>      | <i>timestamp with time zone</i> |    |
| <i>user_password_deleted_user_uuid</i> | <i>character varying</i>        | 36 |
| <i>user_password_deleted_date</i>      | <i>timestamp with time zone</i> |    |

i. Table *user\_recoveries*

Table 3.12. Table *user\_recoveries*

| <b>Nama</b>                            | <b>Tipe Data</b>                | <b>Ukuran</b> |
|----------------------------------------|---------------------------------|---------------|
| <i>user_recovery_id</i>                | <i>integer</i>                  |               |
| <i>user_recovery_uuid</i>              | <i>character varying</i>        | 36            |
| <i>user_recovery_user_uuid</i>         | <i>character varying</i>        | 36            |
| <i>user_recovery_key</i>               | <i>character varying</i>        | 256           |
| <i>user_recovery_status</i>            | <i>smallint</i>                 |               |
| <i>user_recovery_created_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>user_recovery_created_date</i>      | <i>timestamp with time zone</i> |               |
| <i>user_recovery_updated_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>user_recovery_updated_date</i>      | <i>timestamp with time zone</i> |               |
| <i>user_recovery_deleted_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>user_recovery_deleted_date</i>      | <i>timestamp with time zone</i> |               |

j. Table *users*

Table 3.13. Table *users*

| <b>Nama</b>            | <b>Tipe Data</b>                | <b>Ukuran</b> |
|------------------------|---------------------------------|---------------|
| <i>user_id</i>         | <i>integer</i>                  |               |
| <i>user_uuid</i>       | <i>character varying</i>        | 36            |
| <i>user_type</i>       | <i>smallint</i>                 |               |
| <i>user_identity</i>   | <i>character varying</i>        | 64            |
| <i>user_email</i>      | <i>character varying</i>        | 256           |
| <i>user_date_start</i> | <i>timestamp with time zone</i> |               |
| <i>user_date_end</i>   | <i>timestamp with time zone</i> |               |
| <i>user_status</i>     | <i>smallint</i>                 |               |

|                               |                                 |    |
|-------------------------------|---------------------------------|----|
| <i>user_created_user_uuid</i> | <i>character varying</i>        | 36 |
| <i>user_created_date</i>      | <i>timestamp with time zone</i> |    |
| <i>user_updated_user_uuid</i> | <i>character varying</i>        | 36 |
| <i>user_updated_date</i>      | <i>timestamp with time zone</i> |    |
| <i>user_deleted_user_uuid</i> | <i>character varying</i>        | 36 |
| <i>user_deleted_date</i>      | <i>timestamp with time zone</i> |    |

## 2. Skema Reference

### a. Table *language\_contents*

Table 3.14. Table *language\_contents*

| <b>Nama</b>                               | <b>Tipe Data</b>                | <b>Ukuran</b> |
|-------------------------------------------|---------------------------------|---------------|
| <i>language_content_id</i>                | <i>integer</i>                  |               |
| <i>language_content_uuid</i>              | <i>character varying</i>        | 36            |
| <i>language_content_language_uuid</i>     | <i>character varying</i>        | 36            |
| <i>language_content_table_name</i>        | <i>character varying</i>        | 96            |
| <i>language_content_table_uuid</i>        | <i>character varying</i>        | 36            |
| <i>language_content_code</i>              | <i>character varying</i>        | 96            |
| <i>language_content_text</i>              | <i>character varying</i>        | 256           |
| <i>language_content_alias</i>             | <i>character varying</i>        | 96            |
| <i>language_content_description</i>       | <i>text</i>                     |               |
| <i>language_content_status</i>            | <i>smallint</i>                 |               |
| <i>language_content_created_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>language_content_created_date</i>      | <i>timestamp with time zone</i> |               |
| <i>language_content_updated_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>language_content_updated_date</i>      | <i>timestamp with time zone</i> |               |
| <i>language_content_deleted_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>language_content_deleted_date</i>      | <i>timestamp with time zone</i> |               |

b. Table *languages*

Table 3.15. Table *languages*

| <b>Nama</b>                       | <b>Tipe Data</b>                | <b>Ukuran</b> |
|-----------------------------------|---------------------------------|---------------|
| <i>language_id</i>                | <i>integer</i>                  |               |
| <i>language_uuid</i>              | <i>character varying</i>        | 36            |
| <i>language_code</i>              | <i>character varying</i>        | 96            |
| <i>language_name</i>              | <i>character varying</i>        | 128           |
| <i>language_alias</i>             | <i>character varying</i>        | 96            |
| <i>language_flag</i>              | <i>character varying</i>        | 128           |
| <i>language_description</i>       | <i>text</i>                     |               |
| <i>language_status</i>            | <i>smallint</i>                 |               |
| <i>language_created_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>language_created_date</i>      | <i>timestamp with time zone</i> |               |
| <i>language_updated_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>language_updated_date</i>      | <i>timestamp with time zone</i> |               |
| <i>language_deleted_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>language_deleted_date</i>      | <i>timestamp with time zone</i> |               |

c. Table *masters*

Table 3.16. Table *masters*

| <b>Nama</b>                  | <b>Tipe Data</b>         | <b>Ukuran</b> |
|------------------------------|--------------------------|---------------|
| <i>master_id</i>             | <i>integer</i>           |               |
| <i>master_uuid</i>           | <i>character varying</i> | 36            |
| <i>master_group</i>          | <i>character varying</i> | 32            |
| <i>master_reference_uuid</i> | <i>character varying</i> | 36            |
| <i>master_parent_uuid</i>    | <i>character varying</i> | 36            |
| <i>master_code</i>           | <i>character varying</i> | 96            |
| <i>master_icon_pack</i>      | <i>character varying</i> | 32            |
| <i>master_icon_path</i>      | <i>character varying</i> | 32            |
| <i>master_name</i>           | <i>character varying</i> | 128           |
| <i>master_extra</i>          | <i>character varying</i> | 96            |

|                                 |                                 |    |
|---------------------------------|---------------------------------|----|
| <i>master_description</i>       | <i>text</i>                     |    |
| <i>master_status</i>            | <i>smallint</i>                 |    |
| <i>master_created_user_uuid</i> | <i>character varying</i>        | 36 |
| <i>master_created_date</i>      | <i>timestamp with time zone</i> |    |
| <i>master_updated_user_uuid</i> | <i>character varying</i>        | 36 |
| <i>master_updated_date</i>      | <i>timestamp with time zone</i> |    |
| <i>master_deleted_user_uuid</i> | <i>character varying</i>        | 36 |
| <i>master_deleted_date</i>      | <i>timestamp with time zone</i> |    |

### 3. Skema *Public*

#### a. Table *configs*

Table 3.17. Table *configs*

| <b>Nama</b>                     | <b>Tipe Data</b>                | <b>Ukuran</b> |
|---------------------------------|---------------------------------|---------------|
| <i>config_id</i>                | <i>integer</i>                  |               |
| <i>config_uuid</i>              | <i>character varying</i>        | 36            |
| <i>config_initial</i>           | <i>character varying</i>        | 32            |
| <i>config_name</i>              | <i>character varying</i>        | 96            |
| <i>config_description</i>       | <i>text</i>                     |               |
| <i>config_unit</i>              | <i>character varying</i>        | 64            |
| <i>config_type</i>              | <i>smallint</i>                 |               |
| <i>config_rules</i>             | <i>character varying</i>        | 512           |
| <i>config_value</i>             | <i>character varying</i>        | 512           |
| <i>config_status</i>            | <i>smallint</i>                 |               |
| <i>config_created_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>config_created_date</i>      | <i>timestamp with time zone</i> |               |
| <i>config_updated_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>config_updated_date</i>      | <i>timestamp with time zone</i> |               |
| <i>config_deleted_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>config_deleted_date</i>      | <i>timestamp with time zone</i> |               |

b. Table *galleries*

Table 3.18. Table *galleries*

| <b>Nama</b>                             | <b>Tipe Data</b>                | <b>Ukuran</b> |
|-----------------------------------------|---------------------------------|---------------|
| <i>gallery_id</i>                       | <i>integer</i>                  |               |
| <i>gallery_uuid</i>                     | <i>character varying</i>        | 36            |
| <i>gallery_reference_uuid</i>           | <i>character varying</i>        | 36            |
| <i>gallery_master_gallery_type_uuid</i> | <i>character varying</i>        | 36            |
| <i>gallery_code</i>                     | <i>character varying</i>        | 96            |
| <i>gallery_name</i>                     | <i>character varying</i>        | 128           |
| <i>gallery_description</i>              | <i>text</i>                     |               |
| <i>gallery_status</i>                   | <i>smallint</i>                 |               |
| <i>gallery_created_user_uuid</i>        | <i>character varying</i>        | 36            |
| <i>gallery_created_date</i>             | <i>timestamp with time zone</i> |               |
| <i>gallery_updated_user_uuid</i>        | <i>character varying</i>        | 36            |
| <i>gallery_updated_date</i>             | <i>timestamp with time zone</i> |               |
| <i>gallery_deleted_user_uuid</i>        | <i>character varying</i>        | 36            |
| <i>gallery_deleted_date</i>             | <i>timestamp with time zone</i> |               |

c. Table *gallery\_datas*

Table 3.19. Table *gallery\_datas*

| <b>Nama</b>                      | <b>Tipe Data</b>         | <b>Ukuran</b> |
|----------------------------------|--------------------------|---------------|
| <i>gallery_data_id</i>           | <i>integer</i>           |               |
| <i>gallery_data_uuid</i>         | <i>character varying</i> | 36            |
| <i>gallery_data_gallery_uuid</i> | <i>character varying</i> | 36            |
| <i>gallery_data_code</i>         | <i>character varying</i> | 96            |
| <i>gallery_data_title</i>        | <i>character varying</i> | 128           |
| <i>gallery_data_file_name</i>    | <i>character varying</i> | 128           |
| <i>gallery_data_file_path</i>    | <i>character varying</i> | 512           |
| <i>gallery_data_file_size</i>    | <i>integer</i>           |               |
| <i>gallery_data_file_type</i>    | <i>character varying</i> | 96            |
| <i>gallery_data_description</i>  | <i>text</i>              |               |

|                                       |                                 |     |
|---------------------------------------|---------------------------------|-----|
| <i>gallery_data_tags</i>              | <i>character varying</i>        | 512 |
| <i>gallery_data_status</i>            | <i>smallint</i>                 |     |
| <i>gallery_data_created_user_uuid</i> | <i>character varying</i>        | 36  |
| <i>gallery_data_created_date</i>      | <i>timestamp with time zone</i> |     |
| <i>gallery_data_updated_user_uuid</i> | <i>character varying</i>        | 36  |
| <i>gallery_data_updated_date</i>      | <i>timestamp with time zone</i> |     |
| <i>gallery_data_deleted_user_uuid</i> | <i>character varying</i>        | 36  |
| <i>gallery_data_deleted_date</i>      | <i>timestamp with time zone</i> |     |

#### d. Table *logs*

Table 3.20. Table *logs*

| <b>Nama</b>                  | <b>Tipe Data</b>                | <b>Ukuran</b> |
|------------------------------|---------------------------------|---------------|
| <i>log_id</i>                | <i>integer</i>                  |               |
| <i>log_uuid</i>              | <i>character varying</i>        | 36            |
| <i>log_table_name</i>        | <i>character varying</i>        | 32            |
| <i>log_table_uuid</i>        | <i>character varying</i>        | 36            |
| <i>log_row_data</i>          | <i>jsonb</i>                    |               |
| <i>log_status</i>            | <i>smallint</i>                 |               |
| <i>log_created_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>log_created_date</i>      | <i>timestamp with time zone</i> |               |
| <i>log_updated_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>log_updated_date</i>      | <i>timestamp with time zone</i> |               |
| <i>log_deleted_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>log_deleted_date</i>      | <i>timestamp with time zone</i> |               |

#### e. Table *profile\_hobbies*

Table 3.21. Table *profile\_hobbies*

| <b>Nama</b>                       | <b>Tipe Data</b>         | <b>Ukuran</b> |
|-----------------------------------|--------------------------|---------------|
| <i>profile_hobby_id</i>           | <i>integer</i>           |               |
| <i>profile_hobby_uuid</i>         | <i>character varying</i> | 36            |
| <i>profile_hobby_profile_uuid</i> | <i>character varying</i> | 36            |

|                                        |                                 |     |
|----------------------------------------|---------------------------------|-----|
| <i>profile_hobby_master_hobby_uuid</i> | <i>character varying</i>        | 36  |
| <i>profile_hobby_other_text</i>        | <i>character varying</i>        | 128 |
| <i>profile_hobby_status</i>            | <i>smallint</i>                 |     |
| <i>profile_hobby_created_user_uuid</i> | <i>character varying</i>        | 36  |
| <i>profile_hobby_created_date</i>      | <i>timestamp with time zone</i> |     |
| <i>profile_hobby_updated_user_uuid</i> | <i>character varying</i>        | 36  |
| <i>profile_hobby_updated_date</i>      | <i>timestamp with time zone</i> |     |
| <i>profile_hobby_deleted_user_uuid</i> | <i>character varying</i>        | 36  |
| <i>profile_hobby_deleted_date</i>      | <i>timestamp with time zone</i> |     |

f. Table *profile\_interests*

Table 3.22. Table *profile\_interests*

| <b>Nama</b>                                  | <b>Tipe Data</b>                | <b>Ukuran</b> |
|----------------------------------------------|---------------------------------|---------------|
| <i>profile_interest_id</i>                   | <i>integer</i>                  |               |
| <i>profile_interest_uuid</i>                 | <i>character varying</i>        | 36            |
| <i>profile_interest_profile_uuid</i>         | <i>character varying</i>        | 36            |
| <i>profile_interest_master_interest_uuid</i> | <i>character varying</i>        | 36            |
| <i>profile_interest_other_text</i>           | <i>character varying</i>        | 128           |
| <i>profile_interest_status</i>               | <i>smallint</i>                 |               |
| <i>profile_interest_created_user_uuid</i>    | <i>character varying</i>        | 36            |
| <i>profile_interest_created_date</i>         | <i>timestamp with time zone</i> |               |
| <i>profile_interest_updated_user_uuid</i>    | <i>character varying</i>        | 36            |
| <i>profile_interest_updated_date</i>         | <i>timestamp with time zone</i> |               |
| <i>profile_interest_deleted_user_uuid</i>    | <i>character varying</i>        | 36            |
| <i>profile_interest_deleted_date</i>         | <i>timestamp with time zone</i> |               |

g. Table *profile\_socials*

Table 3.23. Table *profile\_socials*

| <b>Nama</b>                | <b>Tipe Data</b>         | <b>Ukuran</b> |
|----------------------------|--------------------------|---------------|
| <i>profile_social_id</i>   | <i>integer</i>           |               |
| <i>profile_social_uuid</i> | <i>character varying</i> | 36            |

|                                          |                                 |    |
|------------------------------------------|---------------------------------|----|
| <i>profile_social_profile_uuid</i>       | <i>character varying</i>        | 36 |
| <i>profile_social_master_social_uuid</i> | <i>character varying</i>        | 36 |
| <i>profile_social_username</i>           | <i>character varying</i>        | 32 |
| <i>profile_social_email</i>              | <i>character varying</i>        | 64 |
| <i>profile_social_status</i>             | <i>smallint</i>                 |    |
| <i>profile_social_created_user_uuid</i>  | <i>character varying</i>        | 36 |
| <i>profile_social_created_date</i>       | <i>timestamp with time zone</i> |    |
| <i>profile_social_updated_user_uuid</i>  | <i>character varying</i>        | 36 |
| <i>profile_social_updated_date</i>       | <i>timestamp with time zone</i> |    |
| <i>profile_social_deleted_user_uuid</i>  | <i>character varying</i>        | 36 |
| <i>profile_social_deleted_date</i>       | <i>timestamp with time zone</i> |    |

#### h. Table *profiles*

Table 3.24. Table *profiles*

| <b>Nama</b>                         | <b>Tipe Data</b>                | <b>Ukuran</b> |
|-------------------------------------|---------------------------------|---------------|
| <i>profile_id</i>                   | <i>integer</i>                  |               |
| <i>profile_uuid</i>                 | <i>character varying</i>        | 36            |
| <i>profile_user_uuid</i>            | <i>character varying</i>        | 36            |
| <i>profile_gallery_data_uuid</i>    | <i>character varying</i>        | 36            |
| <i>profile_master_gender_uuid</i>   | <i>character varying</i>        | 36            |
| <i>profile_master_religion_uuid</i> | <i>character varying</i>        | 36            |
| <i>profile_nik</i>                  | <i>character varying</i>        | 36            |
| <i>profile_overview</i>             | <i>text</i>                     |               |
| <i>profile_fullname</i>             | <i>character varying</i>        | 96            |
| <i>profile_nickname</i>             | <i>character varying</i>        | 96            |
| <i>profile_gender</i>               | <i>smallint</i>                 |               |
| <i>profile_birth_place</i>          | <i>character varying</i>        | 64            |
| <i>profile_birth_date</i>           | <i>date</i>                     |               |
| <i>profile_status</i>               | <i>smallint</i>                 |               |
| <i>profile_created_user_uuid</i>    | <i>character varying</i>        | 36            |
| <i>profile_created_date</i>         | <i>timestamp with time zone</i> |               |
| <i>profile_updated_user_uuid</i>    | <i>character varying</i>        | 36            |

|                                  |                                 |    |
|----------------------------------|---------------------------------|----|
| <i>profile_updated_date</i>      | <i>timestamp with time zone</i> |    |
| <i>profile_deleted_user_uuid</i> | <i>character varying</i>        | 36 |
| <i>profile_deleted_date</i>      | <i>timestamp with time zone</i> |    |

#### 4. Skema *Mobile*

##### a. Tabel *apps*

Table 3.25. Tabel *apps*

| <b>Nama</b>                  | <b>Tipe Data</b>                | <b>Ukuran</b> |
|------------------------------|---------------------------------|---------------|
| <i>app_id</i>                | <i>integer</i>                  |               |
| <i>app_uuid</i>              | <i>character varying</i>        | 36            |
| <i>app_code</i>              | <i>character varying</i>        | 96            |
| <i>app_name</i>              | <i>character varying</i>        | 128           |
| <i>app_bundle</i>            | <i>character varying</i>        | 128           |
| <i>app_description</i>       | <i>character varying</i>        | 256           |
| <i>app_status</i>            | <i>smallint</i>                 |               |
| <i>app_created_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>app_created_date</i>      | <i>timestamp with time zone</i> |               |
| <i>app_updated_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>app_updated_date</i>      | <i>timestamp with time zone</i> |               |
| <i>app_deleted_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>app_deleted_date</i>      | <i>timestamp with time zone</i> |               |

##### b. Tabel *onboardings*

Table 3.26. Tabel *onboardings*

| <b>Nama</b>                         | <b>Tipe Data</b>         | <b>Ukuran</b> |
|-------------------------------------|--------------------------|---------------|
| <i>onboarding_id</i>                | <i>integer</i>           |               |
| <i>onboarding_uuid</i>              | <i>character varying</i> | 36            |
| <i>onboarding_app_uuid</i>          | <i>character varying</i> | 128           |
| <i>onboarding_gallery_data_uuid</i> | <i>character varying</i> | 36            |
| <i>onboarding_code</i>              | <i>character varying</i> | 96            |

|                                     |                                 |     |
|-------------------------------------|---------------------------------|-----|
| <i>onboarding_title</i>             | <i>character varying</i>        | 128 |
| <i>onboarding_description</i>       | <i>character varying</i>        | 256 |
| <i>onboarding_sort</i>              | <i>smallint</i>                 |     |
| <i>onboarding_date_start</i>        | <i>timestamp with time zone</i> |     |
| <i>onboarding_date_end</i>          | <i>timestamp with time zone</i> |     |
| <i>onboarding_status</i>            | <i>smallint</i>                 |     |
| <i>onboarding_created_user_uuid</i> | <i>character varying</i>        | 36  |
| <i>onboarding_created_date</i>      | <i>timestamp with time zone</i> |     |
| <i>onboarding_updated_user_uuid</i> | <i>character varying</i>        | 36  |
| <i>onboarding_updated_date</i>      | <i>timestamp with time zone</i> |     |
| <i>onboarding_deleted_user_uuid</i> | <i>character varying</i>        | 36  |
| <i>onboarding_deleted_date</i>      | <i>timestamp with time zone</i> |     |

### c. Tabel versions

Table 3.27. Tabel versions

| <b>Nama</b>                      | <b>Tipe Data</b>                | <b>Ukuran</b> |
|----------------------------------|---------------------------------|---------------|
| <i>version_id</i>                | <i>integer</i>                  |               |
| <i>version_uuid</i>              | <i>character varying</i>        | 36            |
| <i>version_app_uuid</i>          | <i>character varying</i>        | 128           |
| <i>version_type</i>              | <i>text</i>                     |               |
| <i>version_version</i>           | <i>character varying</i>        | 128           |
| <i>version_build</i>             | <i>character varying</i>        | 128           |
| <i>version_current</i>           | <i>smallint</i>                 |               |
| <i>version_relevant</i>          | <i>smallint</i>                 |               |
| <i>version_changelogs</i>        | <i>jsonb</i>                    |               |
| <i>version_description</i>       | <i>character varying</i>        | 256           |
| <i>version_status</i>            | <i>smallint</i>                 |               |
| <i>version_created_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>version_created_date</i>      | <i>timestamp with time zone</i> |               |
| <i>version_updated_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>version_updated_date</i>      | <i>timestamp with time zone</i> |               |
| <i>version_deleted_user_uuid</i> | <i>character varying</i>        | 36            |

|                             |                                 |  |
|-----------------------------|---------------------------------|--|
| <i>version_deleted_date</i> | <i>timestamp with time zone</i> |  |
|-----------------------------|---------------------------------|--|

## 5. Skema Document

### a. Table *doc\_files*

Table 3.28. Table *doc\_files*

| <b>Nama</b>                       | <b>Tipe Data</b>                | <b>Ukuran</b> |
|-----------------------------------|---------------------------------|---------------|
| <i>doc_file_id</i>                | <i>integer</i>                  |               |
| <i>doc_file_uuid</i>              | <i>character varying</i>        | 36            |
| <i>doc_file_doc_uuid</i>          | <i>character varying</i>        | 36            |
| <i>doc_file_type</i>              | <i>text</i>                     |               |
| <i>doc_file_file_name</i>         | <i>character varying</i>        | 128           |
| <i>doc_file_file_path</i>         | <i>character varying</i>        | 512           |
| <i>doc_file_file_size</i>         | <i>integer</i>                  |               |
| <i>doc_file_file_mime</i>         | <i>character varying</i>        | 96            |
| <i>doc_file_status</i>            | <i>smallint</i>                 |               |
| <i>doc_file_created_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>doc_file_created_date</i>      | <i>timestamp with time zone</i> |               |
| <i>doc_file_updated_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>doc_file_updated_date</i>      | <i>timestamp with time zone</i> |               |
| <i>doc_file_deleted_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>doc_file_deleted_date</i>      | <i>timestamp with time zone</i> |               |

### b. Table *doc\_histories*

Table 3.29. Table *doc\_histories*

| <b>Nama</b>                 | <b>Tipe Data</b>         | <b>Ukuran</b> |
|-----------------------------|--------------------------|---------------|
| <i>doc_history_id</i>       | <i>integer</i>           |               |
| <i>doc_history_uuid</i>     | <i>character varying</i> | 36            |
| <i>doc_history_doc_uuid</i> | <i>character varying</i> | 36            |
| <i>doc_history_type</i>     | <i>text</i>              |               |
| <i>doc_history_source</i>   | <i>text</i>              |               |

|                                      |                                 |    |
|--------------------------------------|---------------------------------|----|
| <i>doc_history_agent</i>             | <i>character varying</i>        | 32 |
| <i>doc_history_ipv_4</i>             | <i>character varying</i>        | 16 |
| <i>doc_history_ipv_6</i>             | <i>character varying</i>        | 64 |
| <i>doc_history_status</i>            | <i>smallint</i>                 |    |
| <i>doc_history_created_user_uuid</i> | <i>character varying</i>        | 36 |
| <i>doc_history_created_date</i>      | <i>timestamp with time zone</i> |    |
| <i>doc_history_updated_user_uuid</i> | <i>character varying</i>        | 36 |
| <i>doc_history_updated_date</i>      | <i>timestamp with time zone</i> |    |
| <i>doc_history_deleted_user_uuid</i> | <i>character varying</i>        | 36 |
| <i>doc_history_deleted_date</i>      | <i>timestamp with time zone</i> |    |

c. Table *doc\_originals*

Table 3.30. Table *doc\_originals*

| <b>Nama</b>                           | <b>Tipe Data</b>                | <b>Ukuran</b> |
|---------------------------------------|---------------------------------|---------------|
| <i>doc_original_id</i>                | <i>integer</i>                  |               |
| <i>doc_original_uuid</i>              | <i>character varying</i>        | 36            |
| <i>doc_original_doc_uuid</i>          | <i>character varying</i>        | 36            |
| <i>doc_original_file_name</i>         | <i>character varying</i>        | 128           |
| <i>doc_original_file_path</i>         | <i>character varying</i>        | 512           |
| <i>doc_original_file_size</i>         | <i>integer</i>                  |               |
| <i>doc_original_file_modified</i>     | <i>timestamp with time zone</i> |               |
| <i>doc_original_file_mime</i>         | <i>character varying</i>        | 96            |
| <i>doc_original_status</i>            | <i>smallint</i>                 |               |
| <i>doc_original_created_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>doc_original_created_date</i>      | <i>timestamp with time zone</i> |               |
| <i>doc_original_updated_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>doc_original_updated_date</i>      | <i>timestamp with time zone</i> |               |
| <i>doc_original_deleted_user_uuid</i> | <i>character varying</i>        | 36            |
| <i>doc_original_deleted_date</i>      | <i>timestamp with time zone</i> |               |

d. Table *doc\_pages*

Table 3.31. Table *doc\_pages*

| <b>Nama</b>                         | <b>Tipe Data</b>                | <b>Ukuran</b> |
|-------------------------------------|---------------------------------|---------------|
| <i>doc_page_id</i>                  | <i>integer</i>                  |               |
| <i>doc_page_uuid</i>                | <i>character varying</i>        | 36            |
| <i>doc_page_doc_uuid</i>            | <i>character varying</i>        | 36            |
| <i>doc_page_number</i>              | <i>smallint</i>                 |               |
| <i>doc_page_thumbnail_file_path</i> | <i>character varying</i>        | 512           |
| <i>doc_page_watermark_file_path</i> | <i>character varying</i>        | 512           |
| <i>doc_page_watermark_opacity</i>   | <i>numeric</i>                  |               |
| <i>doc_page_qrcode_opacity</i>      | <i>numeric</i>                  |               |
| <i>doc_page_qrcode_rotation</i>     | <i>numeric</i>                  |               |
| <i>doc_page_qrcode_left</i>         | <i>numeric</i>                  |               |
| <i>doc_page_qrcode_top</i>          | <i>numeric</i>                  |               |
| <i>doc_page_qrcode_matrix</i>       | <i>text</i>                     |               |
| <i>doc_page_status</i>              | <i>smallint</i>                 |               |
| <i>doc_page_created_user_uuid</i>   | <i>character varying</i>        | 36            |
| <i>doc_page_created_date</i>        | <i>timestamp with time zone</i> |               |
| <i>doc_page_updated_user_uuid</i>   | <i>character varying</i>        | 36            |
| <i>doc_page_updated_date</i>        | <i>timestamp with time zone</i> |               |
| <i>doc_page_deleted_user_uuid</i>   | <i>character varying</i>        | 36            |
| <i>doc_page_deleted_date</i>        | <i>timestamp with time zone</i> |               |

e. Table *docs*

Table 3.32. Table *docs*

| <b>Nama</b>           | <b>Tipe Data</b>         | <b>Ukuran</b> |
|-----------------------|--------------------------|---------------|
| <i>doc_id</i>         | <i>integer</i>           |               |
| <i>doc_uuid</i>       | <i>character varying</i> | 36            |
| <i>doc_code</i>       | <i>character varying</i> | 96            |
| <i>doc_subject</i>    | <i>character varying</i> | 128           |
| <i>doc_applies_to</i> | <i>character varying</i> | 128           |

|                              |                                 |    |
|------------------------------|---------------------------------|----|
| <i>doc_source</i>            | <i>text</i>                     |    |
| <i>doc_secret</i>            | <i>character varying</i>        | 36 |
| <i>doc_checksum</i>          | <i>character varying</i>        | 36 |
| <i>doc_steganography</i>     | <i>character varying</i>        | 36 |
| <i>doc_status</i>            | <i>smallint</i>                 |    |
| <i>doc_created_user_uuid</i> | <i>character varying</i>        | 36 |
| <i>doc_created_date</i>      | <i>timestamp with time zone</i> |    |
| <i>doc_updated_user_uuid</i> | <i>character varying</i>        | 36 |
| <i>doc_updated_date</i>      | <i>timestamp with time zone</i> |    |
| <i>doc_deleted_user_uuid</i> | <i>character varying</i>        | 36 |
| <i>doc_deleted_date</i>      | <i>timestamp with time zone</i> |    |

### 3.3.7. Desain *Interface* yang Dibangun

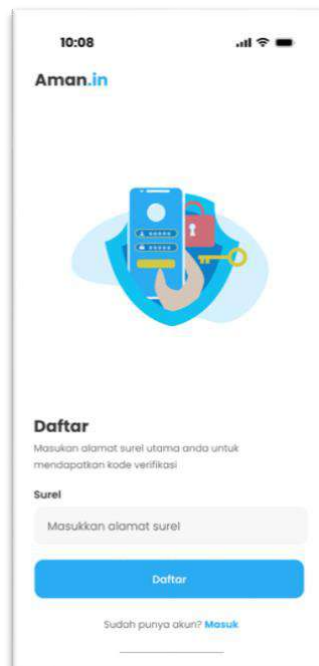
Sebelum melakukan proses pengkodean dan pembuatan aplikasi, diperlukan perancangan desain *interface* atau antarmuka untuk penataan item yang dapat memudahkan pengembangan sistem. Berikut ini adalah rancangan desain *interface* sistem :

### A. Perancangan *Interface* Halaman login



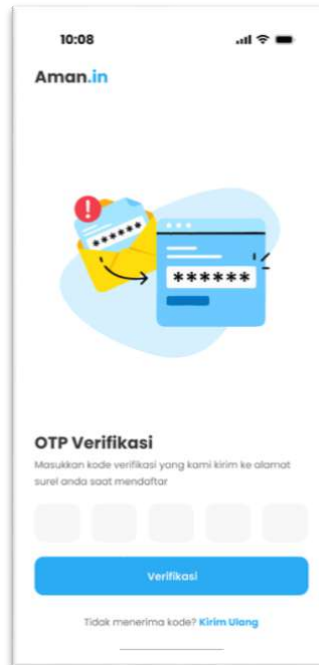
Gambar 3.11. Perancangan *Interface* Halaman Login

### B. Perancangan *Interface* Halaman Registrasi



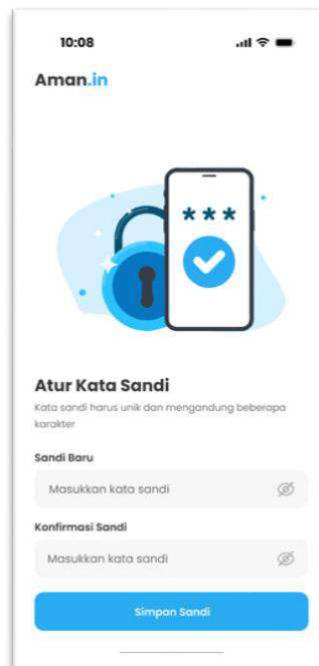
Gambar 3.12. Perancangan *Interface* Halaman Registrasi

### C. Perancangan *Interface* Halaman *OTP* Verifikasi



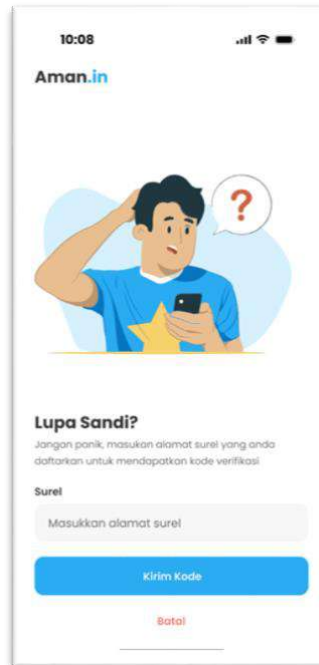
Gambar 3.13. Perancangan *Interface* Halaman *OTP* Verifikasi

### D. Perancangan *Inteface* Halaman Pengaturan Password Baru



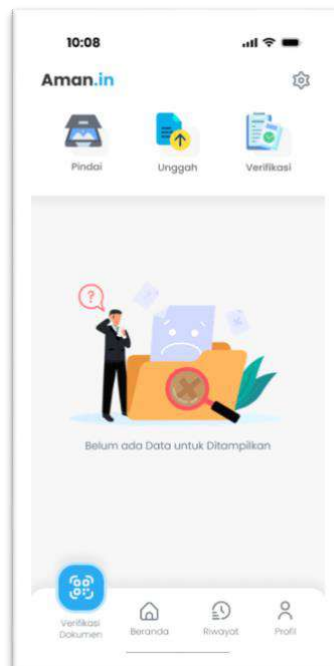
Gambar 3.14. Perancangan *Inteface* halaman Pengaturan Password Baru

E. Perancangan *Interface* Halaman *Forgot Password*



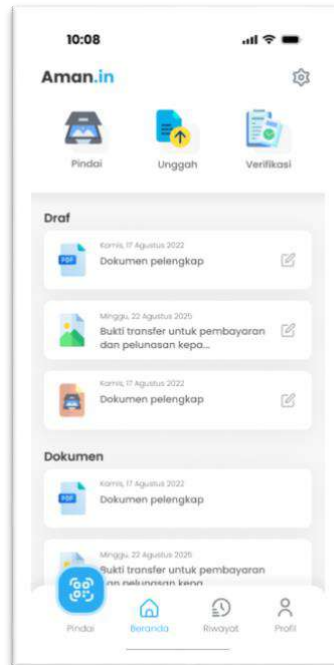
Gambar 3.15. Perancangan *Interface* halaman *Forgot Password*

F. Perancangan *Interface* Halaman Utama 1



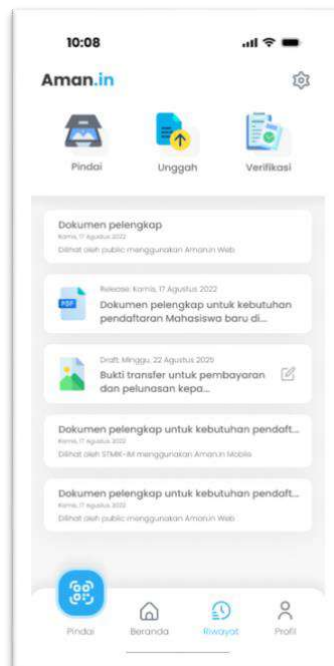
Gambar 3.16. Perancangan *Interface* halaman Utama 1

## G. Perancangan *Interface* Halaman Utama 2



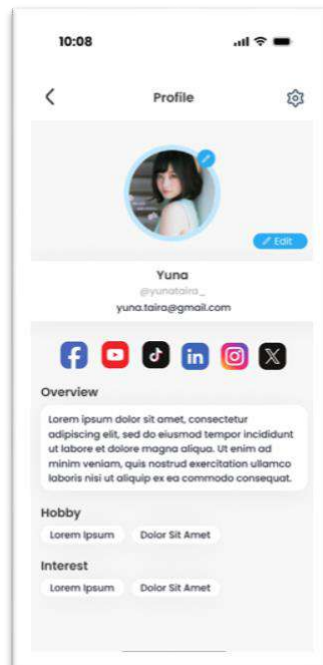
Gambar 3.17. Perancangan *Interface* Halaman Utama 2

## H. Perancangan *Interface* Halaman Utama 3



Gambar 3.18. Perancangan *Interface* Halaman Utama 3

## I. Perancangan Interface Halaman Profile



Gambar 3.19. Perancangan Interface Halaman Profile

## J. Perancangan Interface Halaman Proses



Gambar 3.20. Perancangan Interface Halaman Proses

## **BAB IV**

### **IMPLEMENTASI DAN UJI COBA**

#### **4.1. Implementasi (*implementation*)**

Tahapan ini merupakan tahapan ketika rancangan sistem yang telah disusun pada tahap sebelumnya diterjemahkan menjadi bentuk perangkat lunak melalui penulisan kode program. Dalam model *waterfall* yang digunakan pada pengembangan sistem ini, tahapan implementasi dilakukan setelah desain sistem dan perancangan *database* selesai.

##### **4.1.1. Arsitektur Sistem**

Arsitektur sistem yang dibangun dirancang dengan pendekatan terdistribusi berbasis *microservices*, di mana setiap layanan memiliki tanggung jawab spesifik. Arsitektur ini dibangun untuk memastikan sistem dapat dengan mudah di-maintain, skalabel, serta memiliki keandalan tinggi dalam pengamanan dokumen. Komponen-komponen utama dalam arsitektur sistem ini antara lain:

##### **1. Aplikasi *Mobile* (*Frontend*)**

Dibangun menggunakan Flutter, *framework UI* lintas platform yang mendukung Android dan iOS yang berdiri diatas Bahasa pemrograman *dart*. Aplikasi ini memiliki peran untuk:

- Memindai dokumen dengan kamera, mengambil *file* gambar atau file *PDF* sebagai dokumen sumber. Menggunakan *package file\_picker* versi 10.0.0.

```

FilePickerResult? result = await FilePicker.platform.pickFiles(
  type: FileType.custom, // menentukan tipe file yang diizinkan secara manual (custom)
  allowedExtensions: // ekstensi file yang diizinkan
    selectImage // jika selectImage bernilai true:
      ? ['jpg', 'jpeg', 'png', 'gif', 'bmp', 'webp', 'tga'] // izinkan hanya file gambar
      : ['pdf'], // jika false, izinkan hanya file PDF
  allowMultiple: selectImage, // jika memilih gambar, izinkan pilih banyak file sekaligus
);

```

Gambar 4.1. Implementasi code pengambilan file

- Mengatur posisi *watermark* berupa teks *qr code* dan *barcode*. Dengan memanfaatkan feature bawaan dari flutter yaitu *GestureDetector*, dimana widget tersebut memberikan informasi interaksi sentuhan pada widget anak, seperti *tap*, *double tap*, *long press*, dan geser (*drag*), yang bisa digunakan untuk mengatur perilaku interaktif di *UI*. Hasil dari pemanfaatan dan pengolahan data tersebut disimpan kedalam bentuk *matrix* 4x4.

```

void onDragUpdate(ScaleUpdateDetails details) {
    // Reset matriks transformasi sementara ke nilai identitas (tidak ada transformasi)
    translationDeltaMatrix = Matrix4.identity();
    scaleDeltaMatrix = Matrix4.identity();
    rotationDeltaMatrix = Matrix4.identity();

    // Hitung pergeseran berdasarkan titik fokus sentuhan
    Offset translationDelta = translationUpdater.update(details.focalPoint);
    // Buat matriks translasi dari delta tersebut
    translationDeltaMatrix = _translate(translationDelta);
    // Terapkan translasi ke matriks transformasi utama
    matrix = translationDeltaMatrix * matrix;

    // Simpan posisi baru setelah translasi
    currentPosition = currentPosition + translationDelta;

    // Proses scaling jika diizinkan dan nilai skala berubah dari 1.0 (default)
    if (widget.shouldScale && details.scale != 1.0) {
        double newScale = recordOldScale * details.scale;
        // Hanya terapkan scaling jika masih dalam batas minimal dan maksimal
        if (newScale > widget.minScale && newScale < widget.maxScale) {
            recordScale = newScale; // Simpan skala yang baru
            double scaleDelta = scaleUpdater.update(details.scale);
            // Buat matriks skala dan terapkan ke posisi titik fokus lokal
            scaleDeltaMatrix = _scale(scaleDelta, details.localFocalPoint);
            // Terapkan skala ke matriks utama
            matrix = scaleDeltaMatrix * matrix;
        }
    }

    // Proses rotasi jika diizinkan dan nilai rotasi berubah dari 0
    if (widget.shouldRotate && details.rotation != 0.0) {
        double rotationDelta = rotationUpdater.update(details.rotation);
        // Buat matriks rotasi berdasarkan delta dan titik fokus lokal
        rotationDeltaMatrix = _rotate(rotationDelta, details.localFocalPoint);
        // Terapkan rotasi ke matriks utama
        matrix = rotationDeltaMatrix * matrix;

        // Simpan sudut rotasi terkini
        currentRotationAngle += rotationDelta;
    }

    // Konversi posisi visual ke koordinat logis dalam canvas
    Offset canvasPosition = visualToLogic(
        widget.canvas,
        currentPosition,
        recordScale,
        widget.qrcode,
    );

    // Panggil callback untuk mengirim update posisi, skala, dan rotasi
    widget.onUpdate(
        recordScale,
        matrix,
        rotation: currentRotationAngle,
        position: canvasPosition,
    );
}

```

Gambar 4.2. Implementasi kode untuk tata letak watermark

- Membaca *qr*code atau *barcode* lewat kamera atau file gambar.

Menggunakan *package mobile\_scanner* versi 6.0.10.

```
MobileScanner(
  onDetect: (capture) {
    // Ambil daftar barcode yang terdeteksi dari hasil pemindaian
    final List<Barcode> barcodes = capture.barcodes;
    // Jika ada barcode yang terdeteksi dan belum pernah ditemukan sebelumnya
    if (barcodes.isNotEmpty && !founded) {
      founded = true; // Set flag agar tidak mendeteksi ulang
      // Ambil nilai barcode pertama, atau beri teks fallback jika null
      final String code = barcodes.first.rawValue ?? 'Tidak terbaca';
      // Tutup halaman scanner dan kembali ke halaman sebelumnya
      Navigator.of(context).pop();
      // Jika terdapat callback result, panggil dengan nilai barcode
      if (widget.result != null) {
        widget.result!(code);
      }
    }
  },
),
```

Gambar 4.3. Implementasi kode untuk membaca *qr*code dan *barcode*

- Melakukan proses unggah dokumen ke *object storage*. Menggunakan *package* resmi dari *minio* versi 3.5.7.

```
Future<String?> putObject(PlatformFile file, String path) async {
  // Buat nama objek dengan path tujuan dan nama file asli
  final objectName = '$path/${file.name}';
  try {
    // Buat objek File dari path file yang diterima
    File newF = File(file.path ?? "");

    // Upload file ke MinIO sebagai stream byte
    await Constant.minioConfig.putObject(
      Constant.minioBucket, // Nama bucket MinIO
      objectName,           // Nama objek (lokasi + nama file)
      Stream<Uint8List>.value(newF.readAsBytesSync()), // Data file dalam bentuk stream
    );
    // Jika berhasil, kembalikan nama objek sebagai hasil
    return objectName;
  } catch (e) {
    // Jika gagal, kembalikan null
    return null;
  }
}
```

Gambar 4.4. Implementasi kode untuk unggah ke *Object Storage*

- Berkomunikasi secara langsung dengan *backend*. Menggunakan *package dio* versi 5.8.0+1.

```
Dio dio = Dio(); // Inisialisasi instance Dio untuk request HTTP
var userAgent = localData.read(KeyStorage.userAgent) ?? ''; // Baca User-Agent dari storage lokal
if (userAgent != '') {
  dio.options.headers['User-Agent'] = userAgent; // Set header User-Agent jika ada
}
if (['get'].contains(method)) { // Jika metode request GET
  methodAllowed = true; // Tandai metode diizinkan
  var param = '';
  Map<String, dynamic> params = data['params'] ?? {}; // Ambil parameter dari data
  // Buat query string dari params
  param = params.entries.map((e) => '${e.key}=${e.value ?? ''}').join('&');
  try {
    dio.options.headers = headers; // Set header tambahan
    final response = await dio.get("$url?$param"); // Kirim GET request dengan query string
    responseJson = _response(response); // Proses response
  } on DioException catch (e) { // Tangani error khusus Dio
    print(e);
    if (e.response != null) {
      responseJson = _response(e.response!); // Proses response error jika ada
    } else {
      print('Error: ${e.message}');
    }
  }
} on SocketException { // Tangani error koneksi jaringan
  print('Tidak terhubung ke server');
}
} else if (['post'].contains(method)) { // Jika metode request POST
  methodAllowed = true; // Tandai metode diizinkan
  try {
    Map<String, dynamic> body = data['body'] ?? {}; // Ambil body dari data
    dio.options.headers = headers; // Set header tambahan
    FormData formData = FormData.fromMap(body); // Konversi body ke FormData
    final response = await dio.post(
      url,
      data: formData,
      options: Options(headers: {"Content-Type": "multipart/form-data"}), // Set content type
    );
    responseJson = _response(response); // Proses response
  } on DioException catch (e) { // Tangani error khusus Dio
    if (e.response != null) {
      responseJson = _response(e.response!); // Proses response error jika ada
    } else {
      print('Error: ${e.message}');
    }
  }
} on SocketException { // Tangani error koneksi jaringan
  print('Tidak terhubung ke server');
}
}
```

Gambar 4.5. Implementasi kode untuk interaksi dengan Webservice

## 2. Aplikasi Web (Frontend)

Menggunakan *ReactJS* yaitu sebuah *library javascript* untuk membangun aplikasi web. Aplikasi ini secara khusus dirancang untuk proses verifikasi

dokumen, dan tidak digunakan untuk pengelolaan pengguna atau administrasi lainnya.

Fungsi utama aplikasi web:

- Verifikasi melalui *upload file*.
- Verifikasi melalui tautan (*link*) yang dihasilkan dari *qrcode reader*.

### 3. Backend (*Microservices Architecture*)

Pembangunan Backend aplikasi dibangun sesuai dengan konsep *microservices*, yang memecah fungsionalitas sistem menjadi beberapa layanan kecil yang saling terpisah namun saling terhubung. Setiap layanan memiliki tanggung jawab spesifik dan berkomunikasi melalui *RESTful API*.

Layanan yang dibangun adalah:

- *Base Service*: Mengelola autentikasi dan otorisasi pengguna.
- *Public Service*: Mengelola data-data yang bersifat umum.
- *Mobile Service*: Mengelola data-data yang dibutuhkan aplikasi mobile.
- *Document Service*: Mengelola data utama, yaitu memproses pembuatan keamanan dokumen dan memeriksa dokumen saat diverifikasi

Seluruh layanan ini dibangun menggunakan *AdonisJS* versi 6 dengan *TypeScript*, *framework* berbasis *Node.js* yang mendukung arsitektur modular dan *scalable*.

Proses pengamanan dokumen terdapat pada *document service*, dimana ada dua proses utama, yaitu proses pembuatan saat rilis dokumen dan verifikasi saat mau memverifikasi dokumen.

- Proses rilis dokumen. Dilakukan pada *endpoint POST*

*/v1/process/processing.json*, memanfaatkan *library pdf-lib* versi 1.17.1.

```
// Buat objek metadata dokumen
const amaninSecret = {
  // Ambil UUID dokumen dari detail
  uuid: detail.doc_uuid,
  // Ambil kode dokumen
  code: detail.doc_code,
  // Ambil subjek dokumen
  subject: detail.doc_subject,
  // Ambil UUID pembuat dokumen
  from: detail.doc_created_user_uuid,
  // Ambil penerima dokumen
  to: detail.doc_applies_to,
  // Ambil tanggal sekarang dalam format ISO string
  date: new Date().toISOString(),
}

// Konversi objek ke JSON string dengan indentasi rapi
const amaninSecretString = JSON.stringify(amaninSecret, null, 2)
// Encode JSON string jadi Uint8Array (byte array)
const jsonBytes = new TextEncoder().encode(amaninSecretString)
// Buat stream terkompresi berisi JSON bytes
const jsonStream = pdfDoc.context.flateStream(jsonBytes, {
  // Tentukan tipe sebagai file yang disematkan
  Type: PDFName.of('EmbeddedFile'),
  // Tentukan subtype MIME JSON
  Subtype: PDFName.of('application/json'),
})

// Daftarkan stream ke konteks PDF, dapatkan referensi
const jsonRef = pdfDoc.context.register(jsonStream)
// Buat dictionary file specification untuk file embedded
const filespecDict = pdfDoc.context.obj({
  // Tipe file spec PDF
  Type: PDFName.of('Filespec'),
  // Nama file dalam format heksadesimal
  F: PDFHexString.fromText('amanin-meta.json'),
  // Referensi ke stream file embedded
  EF: pdfDoc.context.obj({ F: jsonRef }),
})

// Buat objek daftar file yang disisipkan
const embeddedFiles = pdfDoc.context.obj({
  // Pasangan nama file dan file spec-nya
  Names: [PDFHexString.fromText('amanin-meta.json'), filespecDict],
})

// Buat dictionary 'Names' untuk katalog PDF
const nameDict = pdfDoc.context.obj({
  // Tetapkan objek EmbeddedFiles ke properti ini
  EmbeddedFiles: embeddedFiles,
})

// Set dictionary 'Names' di katalog PDF
pdfDoc.catalog.set(PDFName.of('Names'), nameDict)
```

Gambar 4.6. Implementasi kode untuk menambah metadata

```

// Buat objek metadata dokumen
const amaninSecret = {
  // Ambil UUID dokumen dari detail
  uuid: detail.doc_uuid,
  // Ambil kode dokumen
  code: detail.doc_code,
  // Ambil subjek dokumen
  subject: detail.doc_subject,
  // Ambil UUID pembuat dokumen
  from: detail.doc_created_user_uuid,
  // Ambil penerima dokumen
  to: detail.doc_applies_to,
  // Ambil tanggal sekarang dalam format ISO string
  date: new Date().toISOString(),
}

// Ambil context PDF dari objek pdfDoc
const context = pdfDoc.context
// Encode string JSON metadata menjadi array byte
const secretBytes = new TextEncoder().encode(amaninSecretString)
// Buat stream terkompresi berisi byte metadata sebagai XObject Form PDF
const secretStream = context.flateStream(secretBytes, {
  // Tipe objek PDF adalah XObject
  Type: PDFName.of('XObject'),
  // Subtype XObject adalah Form
  Subtype: PDFName.of('Form'),
  // FormType 1 menandakan form standar PDF
  FormType: 1,
  // Bounding box form, di sini ukuran sangat kecil (1x1)
  BBox: context.obj([0, 0, 1, 1]),
})

// Daftarkan stream ke konteks PDF dan dapatkan referensinya
const secretStreamRef = context.register(secretStream)
// Ambil halaman pertama dari daftar halaman
const firstPage = pages[0]
// Ambil resource dictionary halaman pertama, jika tidak ada buat objek baru
const res = firstPage.node.Resources() || context.obj({})
// Set resource dictionary di node halaman pertama
firstPage.node.set(PDFName.of('Resources'), res)
// Ambil dictionary XObject dari resource dictionary
let xObjectDict = res.get(PDFName.of('XObject')) as PDFDict
// Jika XObject dictionary belum ada, buat dan set di resource dictionary
if (!xObjectDict) {
  xObjectDict = context.obj({})
  res.set(PDFName.of('XObject'), xObjectDict)
}

// Set XObject dengan key 'StegoSecret' ke referensi stream metadata
xObjectDict.set(PDFName.of('StegoSecret'), secretStreamRef)

```

Gambar 4.7. Implementasi kode untuk menambah steganografi

- Proses verifikasi dokumen

```
// Muat dokumen PDF dari buffer stream
const pdfDoc = await PDFDocument.load(bufferStream)
// Ambil context PDF dari objek pdfDoc
const context = pdfDoc.context

// Inisialisasi objek metadata kosong untuk nanti diisi
var metadataObject: any = {}
try {
  // Ambil dictionary 'Names' dari katalog PDF
  const namesDict = pdfDoc.catalog.get(PDFName.of('Names')) as PDFDict

  // Jika tidak ada dictionary 'Names', lempar error
  if (!namesDict) throw new Error('Tidak ada /Names di catalog.')

  // Ambil dictionary 'EmbeddedFiles' dari 'Names'
  const embeddedFiles = namesDict.get(PDFName.of('EmbeddedFiles')) as PDFDict

  // Jika tidak ada 'EmbeddedFiles', lempar error
  if (!embeddedFiles) throw new Error('Tidak ada /EmbeddedFiles.')

  // Ambil array 'Names' dari 'EmbeddedFiles'
  const namesArray = embeddedFiles.get(PDFName.of('Names')) as any

  // Jika array 'Names' kosong atau ukurannya kurang dari 2, lempar error
  if (!namesArray || namesArray.size < 2) throw new Error('Array /Names kosong.')

  // Ambil dictionary file specification pada indeks 1 dari array 'Names'
  const filespecDict = namesArray.get(1) as PDFDict

  // Ambil dictionary 'EF' (Embedded File) dari file specification
  const efDict = filespecDict.get(PDFName.of('EF')) as PDFDict

  // Ambil referensi file dari dictionary 'EF' dengan key 'F'
  const fileRef = efDict.get(PDFName.of('F'))

  // Lookup isi stream file dari referensi file
  const fileStream = context.lookup(fileRef) as PDFRawStream

  // Ambil isi konten stream file (terkompresi)
  const fileContent = fileStream.contents

  // Dekompresi konten file yang terkompresi
  const fileData = inflate(fileContent)

  // Decode bytes menjadi string dan parsing JSON metadata
  metadataObject = JSON.parse(new TextDecoder().decode(fileData))
} catch (e) {
  // Tangani error (tidak melakukan apa-apa)
}
```

Gambar 4.8. Implementasi kode untuk mengambil metadata

```

// Muat dokumen PDF dari buffer stream
const pdfDoc = await PDFDocument.load(bufferStream)
// Ambil context PDF dari objek pdfDoc
const context = pdfDoc.context

// Inisialisasi variabel teks steganografi kosong
var steganographyText = ''
try {
  // Ambil semua halaman PDF
  const pages = pdfDoc.getPages()

  // Ambil halaman pertama PDF
  const firstPage = pages[0]

  // Ambil node dictionary dari halaman pertama
  const dict = firstPage.node as any

  // Ambil dictionary Resources dari halaman
  const resources = dict.Resources()

  // Jika Resources tidak ditemukan, lempar error
  if (!resources) throw new Error('Resources not found')

  // Ambil dictionary XObject dari Resources
  const xObjectDict = resources.get(PDFName.of('XObject'))

  // Jika XObject tidak ditemukan, lempar error
  if (!xObjectDict) throw new Error('XObject not found')

  // Ambil referensi objek steganografi dengan nama 'StegoSecret'
  const steganographyRef = xObjectDict.get(PDFName.of('StegoSecret'))

  // Jika referensi steganografi tidak ditemukan, lempar error
  if (!steganographyRef) throw new Error('StegoSecret not found')

  // Lookup stream steganografi dari referensi
  const steganographyStream = context.lookup(steganographyRef)

  // Jika objek bukan stream PDFRawStream, lempar error
  if (!(steganographyStream instanceof PDFRawStream)) {
    throw new Error('Stream is not valid')
  }

  // Ambil isi konten stream steganografi (terkompresi)
  const steganographyContent = steganographyStream.contents

  // Dekompresi isi konten steganografi
  const steganographyData = inflate(steganographyContent)

  // Decode hasil dekomposisi menjadi teks
  steganographyText = new TextDecoder().decode(steganographyData)
} catch (e) {
  // Tangani error tanpa aksi
}

```

Gambar 4.9. Implementasi kode untuk mengambil steganografi

#### 4. Object Storage: MinIO

Sistem penyimpanan file tidak menggunakan *filesystem* lokal, tetapi menggunakan *MinIO*, sebuah *object storage* yang kompatibel dengan *Amazon S3*. *File* dokumen hasil *watermark* dan *file* hasil *upload* pengguna disimpan dalam *MinIO* untuk memastikan:

- Keamanan dan Skalabilitas: *File* dapat diakses dengan *URL* terbatas dan disimpan dalam *bucket* yang terisolasi.
- Kemudahan Integrasi: *Backend* dapat mengakses *file* secara efisien menggunakan *SDK S3*.
- Redundansi: *File* disimpan dengan konfigurasi yang mendukung replikasi dan *backup*.

#### 5. Database

Sistem basis data menggunakan *PostgreSQL*, menyimpan informasi penting seperti:

- Data pengguna
- *Metadata* dokumen
- *Log* aktivitas proses *watermark* dan verifikasi
- *UUID* dan *hash* dokumen sebagai identifikasi unik

##### 4.1.2. Implementasi Tampilan Antarmuka

Tampilan antarmuka (*user interface*) merupakan aspek penting dalam pengembangan aplikasi karena menjadi titik awal interaksi antara pengguna dan

sistem. Desain antarmuka yang baik dapat meningkatkan kenyamanan, efisiensi, serta mempermudah pengguna dalam memahami dan menggunakan fitur-fitur yang tersedia.

Pada implementasi ini, antarmuka aplikasi dibagi menjadi dua bagian utama, yaitu:

#### 1. Tampilan *Mobile*

Tampilan ini ditujukan bagi pengguna aplikasi dalam bentuk *mobile*, khususnya di perangkat *Android* dan *iOS*. Desain dirancang dengan pendekatan minimalis dan responsif menggunakan *framework Flutter*, serta mengedepankan kemudahan akses terhadap fitur *watermark*, verifikasi dokumen, serta pemindaian.

##### A. *Splashscreen*



Gambar 4.10. *Splashscreen*

## B. Onboarding



Gambar 4.11. Onboarding

## C. Pendaftaran



Gambar 4.12. Pendaftaran

#### D. Verifikasi Kode *OTP*



01.18

Aman.in

**OTP Verifikasi**

Masukkan kode verifikasi yang kami kirim ke alamat surel Anda saat mendaftar

Surel: testbab4@gmail.com

1 2 3

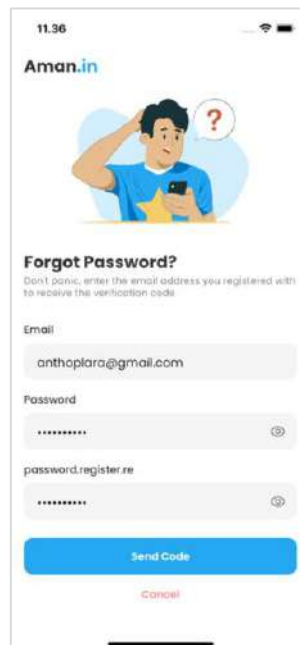
4 5 6

7 8 9

0

Gambar 4.13. Verifikasi kode *OTP*

#### E. Lupa Sandi



11.36

Aman.in

**Forgot Password?**

Don't panic, enter the email address you registered with to receive the verification code

Email

anthoplara@gmail.com

Password

password.register.re

Send Code

Cancel

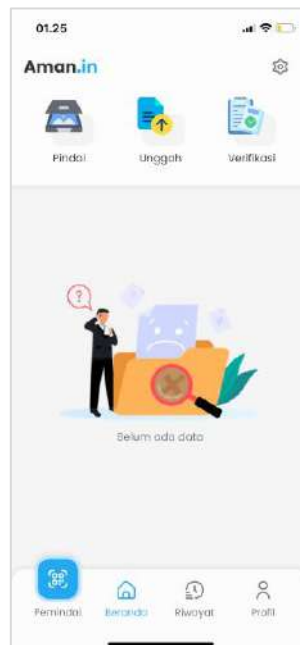
Gambar 4.14. Lupa Sandi

## F. Login

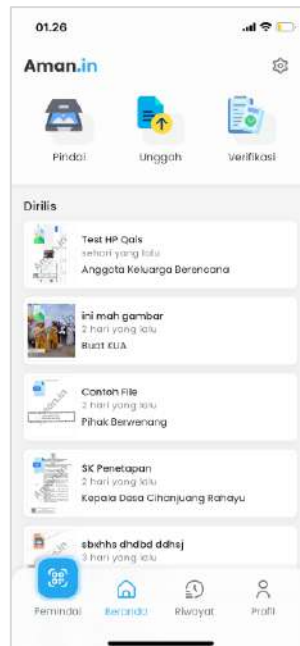


Gambar 4.15. Lupa Sandi

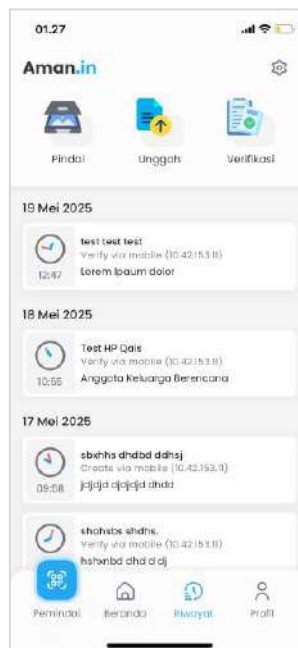
## G. Halaman Utama



Gambar 4.16. Halaman Utama Kosong

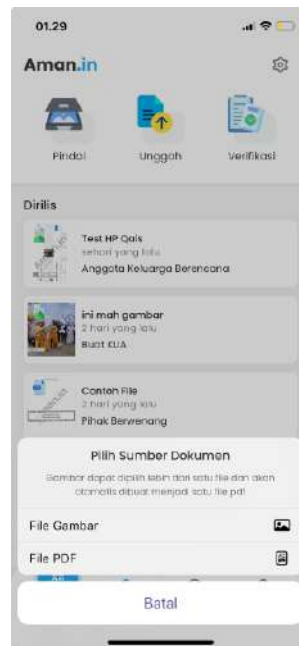


Gambar 4.17. Halaman Utama Isi

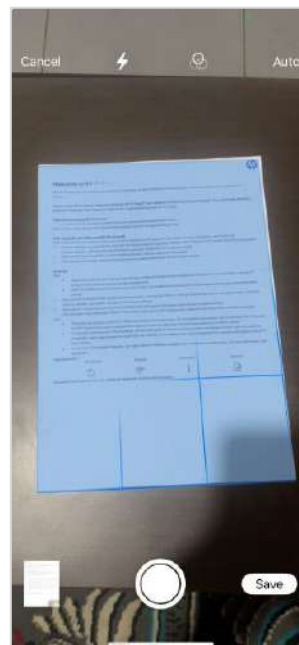


Gambar 4.18. Halaman Utama Riwayat

## H. Proses Pilih Sumber



Gambar 4.19. Pilih Sumber dari File



Gambar 4.20. Pilih Sumber dari Kamera

## I. Pratinjau

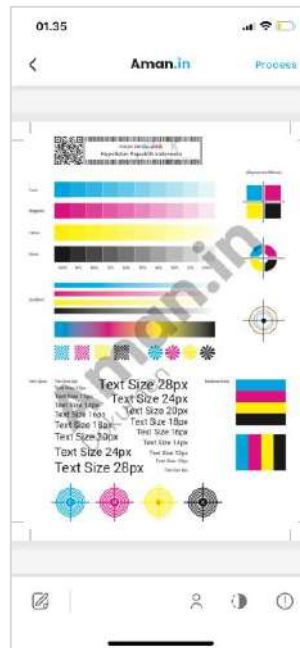


Gambar 4.21. Pratinjau PDF

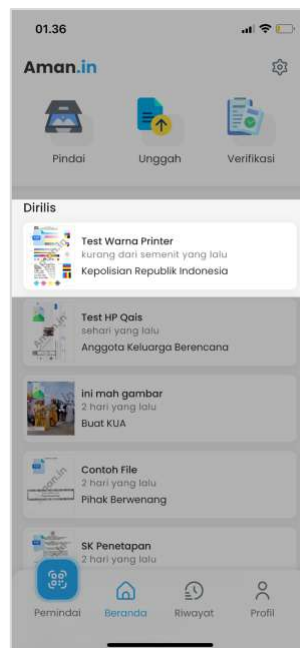


Gambar 4.22. Pratinjau Kirim

## J. Proses *Watermark*



Gambar 4.23. Proses *Watermark*

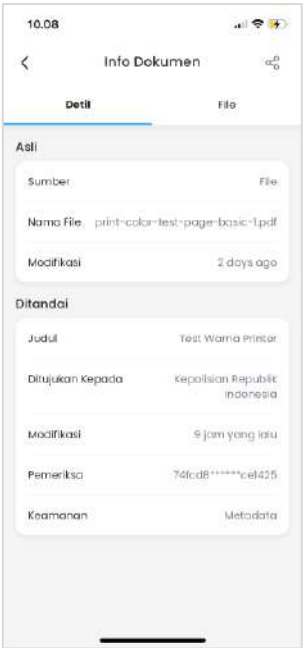


Gambar 4.24. Dokumen Rilis

K. Proses Verifikasi

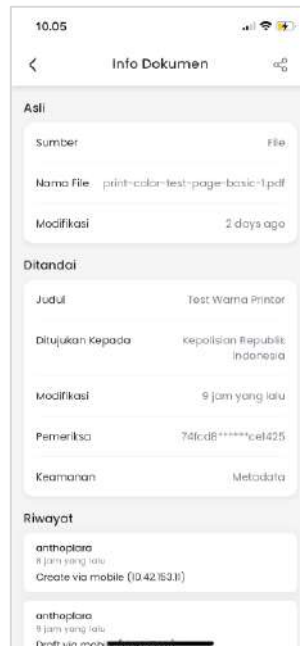


Gambar 4.25. Verifikasi Gagal



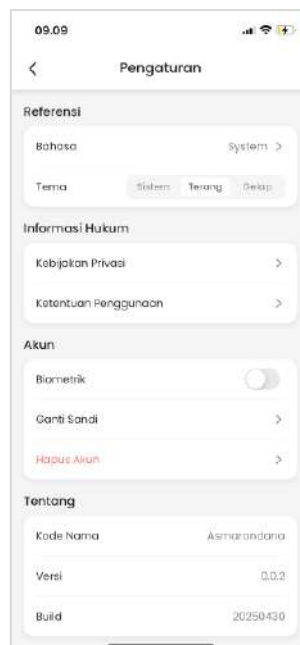
Gambar 4.26. Verifikasi Berhasil

## L. Informasi Dokumen yang Dirilis



Gambar 4.27. Detil Informasi Dokumen

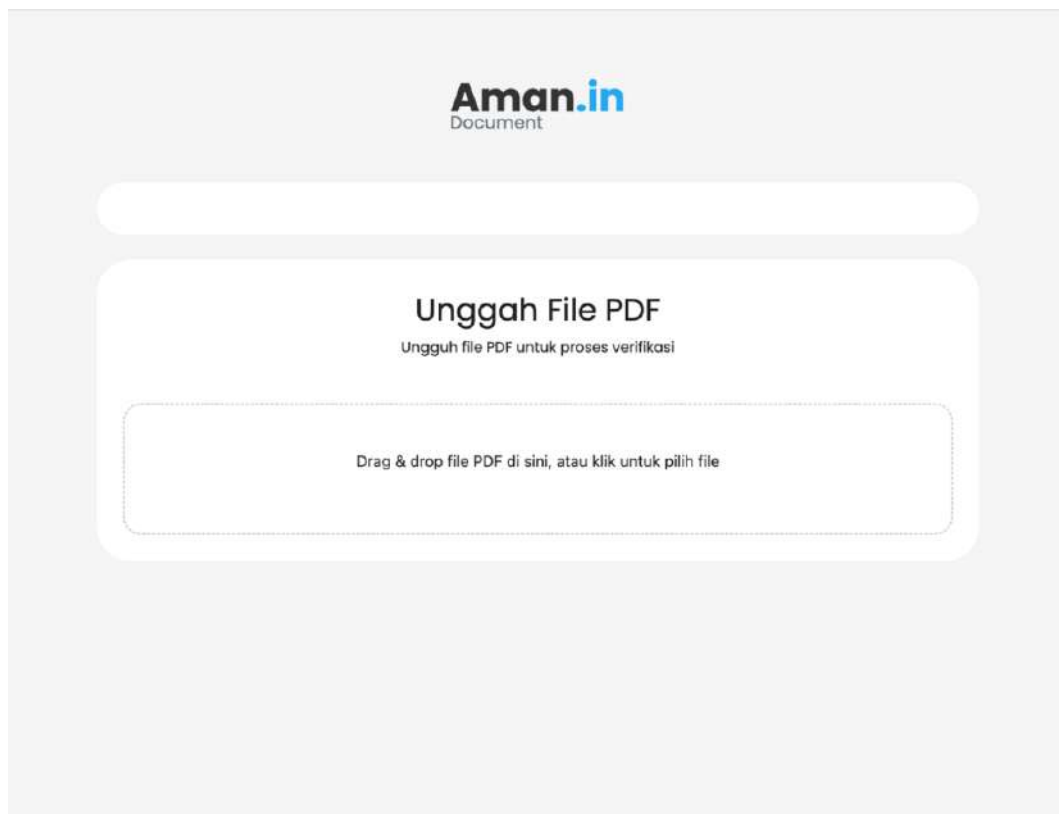
## M. Pengaturan



Gambar 4.28. Pengaturan

## 2. Tampilan *Web*

Tampilan *web* digunakan untuk kebutuhan manajemen dan validasi dokumen melalui antarmuka berbasis *browser*. Pengembangan dilakukan dengan memanfaatkan teknologi *web* modern dan menyesuaikan tata letak agar informatif, ringan, dan mudah digunakan oleh pengguna dari berbagai perangkat.



Gambar 4.29. Tampilan Web

### 4.2. Pengujian Sistem (*verification*)

Pada tahap ini, dilakukan pengujian untuk memastikan bahwa sistem yang dikembangkan dalam aplikasi yang dibangun telah berjalan sesuai dengan fungsionalitas yang telah ditentukan. Pengujian ini bertujuan untuk memverifikasi

bahwa setiap fitur utama bekerja sebagaimana mestinya dan mampu menangani masukan sesuai skenario yang diharapkan.

Metode pengujian yang digunakan adalah *blackbox testing*, yaitu pengujian yang difokuskan pada keluaran sistem berdasarkan input tertentu tanpa memperhatikan struktur internal dari kode program. Pengujian ini menilai apakah sistem mampu memproses data dengan benar, menampilkan hasil yang tepat, dan merespon interaksi pengguna sesuai rancangan.

#### **4.2.1. Fokus Pengujian**

Adapun fokus pengujian meliputi:

- Proses Login dan Autentikasi Pengguna

Memastikan bahwa pengguna dapat masuk ke dalam aplikasi menggunakan kredensial yang *valid*, serta sistem menolak akses jika informasi yang diberikan tidak sesuai. Fitur ini juga diuji untuk mengantisipasi input tidak valid dan menjaga keamanan akses.

- Fungsi *Watermark*

Memastikan dokumen (gambar atau *PDF*) dapat diberi tanda berupa teks dan/atau *barcode* dan *qrcode* dengan benar, serta dapat menyesuaikan posisi dan transparansi *watermark* sesuai pengaturan pengguna.

- Verifikasi Dokumen

Menguji kemampuan sistem dalam membaca dan memverifikasi dokumen baik digital maupun cetak, melalui pemindaian *qrcode* dan *barcode* atau pembacaan *metadata* tersembunyi (steganografi).

- Pengambilan *File*

Memastikan sistem dapat mengambil gambar dari kamera atau memilih *file* dari penyimpanan perangkat, serta mendukung berbagai format file umum seperti *JPG*, *PNG*, dan *PDF*.

- Pembacaan *Metadata* dan Informasi Tersembunyi

Memastikan sistem dapat mendeteksi dan menampilkan informasi pemilik serta penerima dokumen dari data yang tertanam, baik secara metadata maupun metode tersembunyi lainnya.

- Penyimpanan dan Hasil Akhir

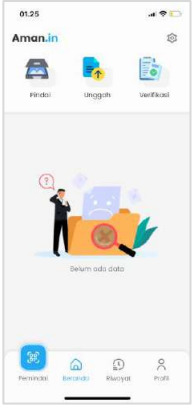
Menguji proses penyimpanan hasil file yang telah diberi watermark atau tanda, serta memastikan dokumen tersebut tidak rusak dan siap digunakan kembali, baik dalam format digital maupun dicetak.

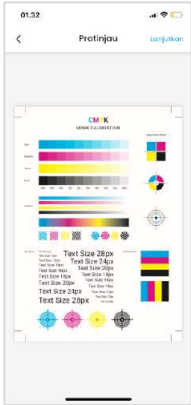
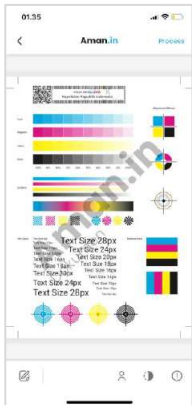
#### **4.2.2. Proses Pengujian**

Sebelum laporan pengujian resmi dikeluarkan, tim pengembang telah melakukan serangkaian proses pengujian secara sistematis dan menyeluruh terhadap sistem yang telah dibangun. Pengujian ini bertujuan untuk memastikan bahwa sistem berfungsi sesuai dengan spesifikasi, mampu menangani beban pengguna secara optimal, dan memiliki tingkat keamanan yang memadai untuk digunakan oleh pengguna akhir.

## A. Fungsional Mobile

Table 4.1. Table Pengujian Fungsional Mobile

| No | Tampilan                                                                                                            | Kasus Uji                                                                                                   | Hasil Yang Diharapkan                                                            | Validitas |       |
|----|---------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|-----------|-------|
|    |                                                                                                                     |                                                                                                             |                                                                                  | Ya        | Tidak |
| 1  | <b>Login.</b><br>                 | Ketika data yang diinput valid dan sesuai:<br>Alamat email dan <i>password</i> benar                        | - Berhasil <i>login</i> .<br>- Masuk ke halaman utama.                           | √         |       |
|    |                                                                                                                     | Ketika data yang diinput tidak valid dan tidak sesuai :<br><i>username</i> atau <i>password</i> tidak valid | - Gagal <i>login</i> .<br>- Muncul informasi kesalahan login dan batas kesalahan | √         |       |
| 2  | <b>Menu di halaman utama</b><br> | Klik masing-masing menu di halaman utama                                                                    | Muncul data dari menu yang dipilih sesuai dengan fungsi masing-masing            | √         |       |

|   |                                                                                                                    |                                                                                               |                                                                                              |   |  |
|---|--------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|---|--|
| 3 | <p><i>Draf watermark</i></p>      | Klik menu-menu untuk memilih sumber dokumen, setelah dipilih akan masuk ke halaman pratinjau  | Muncul pratinjau dokumen yang dipilih, baik dari kamera, gambar atau file                    | √ |  |
|   |                                                                                                                    | Lanjutkan dan masukkan informasi yang dibutuhkan                                              | Akan memproses draf <i>watermark</i> , dan dokumen akan muncul di halaman utama sebagai draf | √ |  |
| 4 | <p><i>Proses Watermark</i></p>  | Klik salah satu data yang masih berstatus draf, akan masuk ke halaman proses <i>watermark</i> | Muncul dokumen sebelumnya sesuai dengan source, dan muncul watermark aplikasi                | √ |  |
|   |                                                                                                                    | Melakukan pengaturan posisi dan pilihan lain di proses                                        | Semua alat edit berfungsi dan qrcode bisa di atur skala, rotasi dan posisi                   | √ |  |

|   |                                                                                                 |                                                           |                                                                                                                            |   |  |
|---|-------------------------------------------------------------------------------------------------|-----------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|---|--|
| 5 | Verifikasi<br> | Mengambil gambar yang akan diverifikasi                   | Berhasil melakukan scan qrcode dan gambar, jika tidak ada code, akan mengeluarkan informasi tidak ditemukan                | √ |  |
|   |                                                                                                 | Mengambil PDF yang akan diverifikasi                      | Berhasil mengambil file PDF                                                                                                | √ |  |
|   |                                                                                                 | Pengambilan berhasil akan diarahkan ke halaman verifikasi | Akan muncul informasi dokumen beserta informasi keamanan yang digunakan, jika gagal verifikasi akan muncul informasi gagal | √ |  |

Berdasarkan hasil pengujian menggunakan metode blackbox, diperoleh kesimpulan bahwa seluruh fitur utama pada aplikasi yang dibangun telah berjalan dengan baik sesuai dengan fungsionalitas yang dirancang. Setiap menu dan proses dalam sistem, seperti login, pemberian watermark, pengambilan file, verifikasi dokumen digital maupun cetak, serta pembacaan metadata, telah diuji dan menunjukkan kinerja yang stabil serta responsif terhadap input pengguna.

Tidak ditemukan kesalahan signifikan selama proses pengujian berlangsung, baik dari sisi fungsionalitas maupun alur interaksi pengguna. Hal ini menunjukkan bahwa sistem yang dibangun secara umum telah memenuhi kebutuhan dasar dalam hal pengamanan dokumen digital dan fisik.

#### B. 3a (load testing)

Tujuan dari pengujian performa adalah untuk memastikan bahwa aplikasi ini dapat memberikan respons yang cepat, stabil, dan dapat menangani beban pengguna dalam kondisi normal maupun ekstrem.

##### Detil Pengujian Login

- *Endpoint* Diuji:

POST /v1/auth/login/base.json

- *Isi request:*

```
1  {
2    "username": "pru8hejKWYxKNY86IGdSY2qNPD0izrfSz6ITYCGnk48=",
3    "password": "F7jJgDOXtiQZ4jgoxMveaw=="
4  }
```

Gambar 4.30. Payload Request Login

- *Isi response:*

```
1  {
2    "status": true,
3    "message": "",
4    "language": "en-US",
5    "payload": {
6      "next": "login-userdata",
7      "secret": "qQgPEbUGgHwj52uvBAF/X78DfufAc0jmkSW0r...+EWFpRk=",
8      "checksum": "93468b4f4b7cdef6895189efaa7fe392"
9    },
10   "generated": 0.028,
11   "tokenExpire": 1747967223763,
12   "serverTime": 1747963623782
13 }
```

Gambar 4.31. Payload Response Login

- Jumlah *sampel* dan *rump-up*:  
100 permintaan dalam 1 detik
- Tujuan:  
Menguji performa endpoint login terhadap 100 permintaan simultan.

### Hasil Pengujian

| Label          | # Samples | Average | Min | Max | Std. Dev | Error % | Throughput | Received KB/sec | Sent KB/sec | Avg. Bytes |
|----------------|-----------|---------|-----|-----|----------|---------|------------|-----------------|-------------|------------|
| Endpoint Login | 100       | 337     | 78  | 470 | 91.72    | 0.00%   | 80.3/sec   | 64.42           | 170.31      | 821.8      |
| TOTAL          | 100       | 337     | 78  | 470 | 91.72    | 0.00%   | 80.3/sec   | 64.42           | 170.31      | 821.8      |

Gambar 4.32. Hasil Pengujian Login

### Analisis Singkat

- Tidak ada error, menunjukkan sistem login cukup stabil.
- Rata-rata response time masih tergolong baik (337 ms), dengan penyebaran (standard deviasi) wajar untuk 100 request.
- Jumlah permintaan yang berhasil diproses sebanyak 80.3 request per detik menandakan bahwa sistem dapat melayani permintaan dengan kecepatan cukup tinggi.
- Rata-rata ukuran data yang diterima per request adalah sekitar 822 byte.

### Kesimpulan

Table 4.2. Kesimpulan Pengujian Endpoint Login

| Faktor         | Nilai & Analisis                                                        |
|----------------|-------------------------------------------------------------------------|
| Responsiveness | Rata-rata response time hanya 18 ms per request.                        |
| Stability      | Stabil, Hampir tidak ada deviasi signifikan; request time konsisten.    |
| Error Rate     | Semua request berhasil, menunjukkan keandalan dan logika yang solid.    |
| Throughput     | 80.3 request/detik, ini sangat tinggi dan menunjukkan efisiensi tinggi. |

|                 |                                                                                |
|-----------------|--------------------------------------------------------------------------------|
| Ukuran Response | Sekitar 300 byte, sangat kecil, efisien untuk komunikasi ringan seperti login. |
| Skalabilitas    | Siap menerima lonjakan trafik, dapat diandalkan untuk high concurrency.        |

## Detil Pengujian Proses

- *Endpoint* Diuji:  
POST /v1/process/processing.json
- Isi *request*

```

1  {
2    "draft": {
3      "subject": "Dokumen Catatan",
4      "steganography": 1,
5      "to": "PT. Codilabs Indonesia"
6    },
7    "page": {
8      "page": 1,
9      "watermark": {
10       "opacity": 0.25
11     },
12     "qrcode": {
13       "opacity": 0.8,
14       "rotation": 0,
15       "left": -67.5,
16       "top": 226.4495798319328,
17       "matrix": "[1.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,1.0,0.0,-67.5,226.4495798319328,0.0,1.0]"
18     }
19   },
20   "uuid": "3bde4df9-40df-4a70-b8e0-6bebe3d450f0",
21   "minio": {
22     "overlay": "document/overlay/3bde4df9-40df-4a70-b8e0-6bebe3d450f0.png",
23     "thumbnail": "document/thumbnail/3bde4df9-40df-4a70-b8e0-6bebe3d450f0.png"
24   }
25 }

```

Gambar 4.33. Payload Request Proses

- Isi *response*:

```

1  {
2    "status": true,
3    "message": "",
4    "language": "en-US",
5    "payload": "3bde4df9-40df-4a70-b8e0-6bebe3d450f0",
6    "generated": 0.171,
7    "tokenExpire": 1747968678902,
8    "serverTime": 1747965079063
9  }

```

Gambar 4.34. Payload Response Proses

- Jumlah *sampel* dan *rump-up*:  
100 permintaan dalam 1 detik dan 60 detik
- Tujuan:  
Menguji performa endpoint proses terhadap 100 permintaan.

### Hasil Pengujian *rump-up* 1 detik

| Label            | # Samples | Average | Min  | Max   | Std. Dev | Error % | Throughput | Received KB/sec | Sent KB/sec | Avg. Bytes |
|------------------|-----------|---------|------|-------|----------|---------|------------|-----------------|-------------|------------|
| Endpoint Process | 100       | 12010   | 1400 | 14604 | 2081.79  | 47.00%  | 6.7/sec    | 21.52           | 17.88       | 3283.8     |
| TOTAL            | 100       | 12010   | 1400 | 14604 | 2081.79  | 47.00%  | 6.7/sec    | 21.52           | 17.88       | 3283.8     |

Gambar 4.35. Hasil Pengujian Proses (1 detik)

### Hasil Pengujian *rump-up* 60 detik

| Label            | # Samples | Average | Min | Max | Std. Dev | Error % | Throughput | Received KB/sec | Sent KB/sec | Avg. Bytes |
|------------------|-----------|---------|-----|-----|----------|---------|------------|-----------------|-------------|------------|
| Endpoint Process | 100       | 163     | 117 | 699 | 59.82    | 0.00%   | 1.7/sec    | 0.08            | 4.67        | 412.9      |
| TOTAL            | 100       | 163     | 117 | 699 | 59.82    | 0.00%   | 1.7/sec    | 0.08            | 4.67        | 412.9      |

Gambar 4.36. Hasil Pengujian Proses (60 detik)

### Analisis Singkat (*rump-up* 1 detik)

- Waktu tanggapan tergolong tinggi, dengan rata-rata 12 detik per request. Ini dapat menyebabkan pengalaman pengguna yang buruk.
- Hampir setengah dari total request gagal (47%), yang menunjukkan ada masalah serius di sisi server atau logika proses.
- Jumlah permintaan yang berhasil diproses hanya 6.7 request per detik, ini rendah dibandingkan endpoint login (80.3/detik), menandakan bahwa proses ini berat atau belum optimal.
- Response cukup besar (rata-rata 3.2 KB per request).

### Analisis Singkat (*rump-up* 60 detik)

- Waktu tanggap turun drastis menjadi rata-rata 163 ms, menunjukkan performa bisa baik jika beban datang secara bertahap.
- Tidak ada error, menunjukkan sistem stabil saat tidak dibebani secara mendadak.
- Jumlah permintaan yang berhasil diproses sebanyak 1.7 permintaan per detik, masih rendah, tapi wajar karena beban tidak simultan.
- *Response* rata-rata hanya 412.9 bytes, lebih ringan dari pengujian sebelumnya, kemungkinan karena variasi input atau response.

### Kesimpulan Umum (untuk endpoint proses)

Table 4.3. Kesimpulan Umum Pengujian Endpoint Proses

| Faktor                | Ramp-up 1 detik                                                                                                             | Ramp-up 60 detik                   |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------|------------------------------------|
| <i>Responsiveness</i> | Lambat & tidak stabil                                                                                                       | Cepat dan stabil                   |
| <i>Stability</i>      | Banyak error (47%)                                                                                                          | Tidak ada error                    |
| Kesiapan <i>Load</i>  | Tidak siap beban serentak                                                                                                   | Siap jika beban bertahap           |
| Optimisasi            | Perlu perbaikan logika dan resource handling                                                                                | Cukup efisien untuk beban bertahap |
| Skalabilitas          | Perlu ditingkatkan                                                                                                          | Perlu monitoring saat traffic naik |
| Optimasi Tambahan     | Tambahkan pod ( <i>horizontal scaling</i> ) agar sistem bisa menangani lonjakan permintaan secara paralel tanpa bottleneck. |                                    |

### C. Pengujian Keamanan (*penetration test*)

Proses pengujian dengan metode sederhana, yaitu dengan mengidentifikasi potensi celah SQL Injection pada fitur login aplikasi yang memungkinkan penyerang melewati autentikasi atau mengakses data yang tidak seharusnya.

#### Skema Uji Coba

- *Endpoint* Diuji:

*POST /v1/auth/login/basic.json*

- *Penetrasi:*

admin' UNION SELECT null, version(), null –

(mencoba mengeksekusi query tambahan untuk mengambil informasi versi database)

#### Hasil Pengujian

```
1  {
2    "status": false,
3    "message": "Username or password doesn't match, you have 4 attempts remaining",
4    "language": "en-US",
5    "payload": null,
6    "generated": 0.024,
7    "tokenExpire": 1747971061626,
8    "serverTime": 1747967461640
9  }
```

Gambar 4.37. Response SQL Injection Login (Postman)



Gambar 4.38. Response SQL Injection Login (Mobile)

## Kesimpulan

- Aplikasi tidak rentan terhadap *Union-Based SQL Injection*.
- *Input* tidak diproses sebagai bagian dari *query SQL* mentah.

## **BAB V**

### **PENUTUP**

#### **5.1. Penutup**

Setelah melalui serangkaian proses penelitian, perancangan, hingga pengujian aplikasi yang dibangun, penulis sampai pada sejumlah kesimpulan yang dapat menggambarkan hasil dari keseluruhan perjalanan ini.

Aplikasi yang dibangun hadir sebagai sebuah solusi dari keresahan yang muncul akibat maraknya pemalsuan dan penyalahgunaan dokumen, baik dalam bentuk digital maupun fisik. Dengan memanfaatkan teknologi mobile melalui Flutter, serta backend AdonisJS yang kokoh dan fleksibel, aplikasi ini berhasil dikembangkan untuk membantu masyarakat menjaga keaslian dokumen secara praktis.

Fitur-fitur utama yang dibangun seperti penambahan *watermark*, penyisipan *metadata*, dan teknik steganografi tidak hanya memperkuat identitas sebuah dokumen, tapi juga membuka cara baru dalam menjaga jejak dan autentikasi. Dokumen yang sudah diberi tanda melalui aplikasi ini bisa diverifikasi kembali, baik dengan mengunggah file digitalnya, maupun dengan memotret dokumen cetaknya.

Fitur verifikasi menjadi poin kunci. Dengan adanya sistem yang mengarah ke *server* untuk melakukan pengecekan keaslian berdasarkan *watermark* atau *metadata*, pengguna tidak hanya memiliki kontrol terhadap dokumen miliknya, tapi

juga mendapatkan jaminan bahwa apa yang mereka bagikan tidak bisa dimanipulasi seenaknya oleh pihak lain.

Pengujian membuktikan bahwa perpaduan antara *watermark*, *metadata*, dan steganografi memberikan hasil yang cukup menjanjikan. Dokumen menjadi lebih sulit untuk dipalsukan, namun tetap mudah untuk digunakan dan diverifikasi oleh orang yang berhak. Pendek kata, aplikasi ini berhasil memenuhi tujuannya: membantu masyarakat menjaga dokumen penting agar tidak disalahgunakan.

## **5.2. Saran**

Meskipun aplikasi ini sudah berhasil dibangun dan diuji secara fungsional, ada beberapa hal yang perlu dipertimbangkan untuk pengembangan ke depan. Teknologi dan kebutuhan masyarakat selalu berkembang, maka aplikasi ini pun harus bisa beradaptasi.

- Pertama, integrasi dengan tanda tangan digital yang legal dan diakui secara hukum akan menjadi langkah penting selanjutnya. Hal ini akan memperkuat posisi dokumen dari sisi hukum, tidak hanya secara teknis.
- Kedua, perlu dipikirkan tentang enkripsi file, terutama untuk dokumen yang sensitif. Dengan begitu, selain keasliannya terjaga, kerahasiaannya pun terlindungi.
- Ketiga, akan sangat membantu bila ada *dashboard* admin berbasis *web*, khususnya untuk institusi atau perusahaan yang memproses dokumen dalam jumlah banyak. Dengan sistem manajemen terpusat, pengawasan dan pengelolaan dokumen bisa lebih efisien.

- Keempat, pengembangan ke arah *platform web* (selain mobile) juga patut dipertimbangkan. Banyak pengguna yang lebih nyaman bekerja di laptop atau komputer, dan kehadiran versi web akan menjangkau lebih banyak orang.
- Kelima, untuk mengetahui sejauh mana efektivitas sistem ini, uji coba di lapangan secara lebih luas perlu dilakukan. Dokumen yang digunakan bisa lebih bervariasi, kondisi perangkat pun lebih beragam. Dari situ akan muncul masukan-masukan nyata yang bisa memperkaya aplikasi ini secara fungsional dan teknis.

Demikianlah penutup dari penelitian ini. Penulis menyadari bahwa pekerjaan ini masih jauh dari sempurna. Namun harapannya, Aplikasi yang dibangun dapat menjadi langkah awal yang berarti dalam menciptakan sistem pengamanan dokumen yang lebih cerdas, mudah diakses, dan relevan dengan kebutuhan zaman.

## DAFTAR PUSTAKA

- Mokodompit, R. N., Mustain, M. R., & Feybelia, F. (2020). Aplikasi Mobile Layanan Bantuan Pengadilan Negeri Tondano. *JOINTER*, 44.
- Al Fatih, M. T. (2024, Oktober). *Memahami Apa Itu Aplikasi dan Fungsinya di Era Digital*. Retrieved from [www.telkomuniversity.ac.id](http://www.telkomuniversity.ac.id): <https://dif.telkomuniversity.ac.id/memahami-apa-itu-aplikasi-dan-fungsinya-di-era-digital/>
- Imron, Azizah, N., Nurhayati, S., & Wijonarko, B. (2021). Perancangan Aplikasi Mobile Zakat dan Infaq Berbasis Android Pada Baznas Kabupaten Tangerang. *JIUBJ*, 198.
- Annisa, R., & Baihaqi, M. R. (2021). PENERAPAN METODE WATERFALL DALAM MERANCANG APLIKASI MOBILE GURU DAN ORANG TUA. *Jurnal Teknik Informatika Kaputama*, 196.
- Wahyuni, T., Sopiandi, i., & Raharjo, S. (2020). SISTEM INFORMASI GEOGRAFIS WISATA KULINER BERBASIS ANDROID. *INFOTECH journal*, 38.
- Suharyanto, E. (2022). PERANCANGAN APLIKASI PENGENALAN BUDAYA NUSANTARA BERBASIS ANDROID DENGAN METODE RAD. *Jurnal Ilmu Komputer JIK*, 31.
- Nugroho, A. H., & Rohimi, T. (2020). Perancangan Aplikasi Sistem Pengolahan DataPenduduk Dikelurahan Desa Kaduronyok Kecamatan Cisata, Kabupaten Pandeglang Berbasis Web. *JUTIS*, 4.
- Anggraini, Y., Pasha, D., Damayanti, & Setiawan, A. (2020). SISTEM INFORMASI PENJUALAN SEPEDA BERBASIS WEB MENGGUNAKAN FRAMEWORK CODEIGNITER (STUDI KASUS : ORBIT STATION) . *Jurnal Teknologi dan Sistem Informasi* , 65.
- Kurniawan, H., Apriliah, W., Kurniawan, I., & Firmasyah, D. (2020). PENERAPAN METODE WATERFALL DALAM PERANCANGAN SISTEM INFORMASI PENGGAJIAN PADA SMK BINA KARYA KARAWANG. *Jurnal Interkom*, 160.
- Nurlaela, L., Dharmalau, A., & Parida, N. T. (2020). RANCANGAN SISTEM INFORMASI INVENTORY BARANG BERBASIS WEB STUDI KASUS PADA CV. LIMOPLAST . *Syntax Idea*, 76.

- Ardian, V. D., Haniyah, & Saktiawan, P. (2025). PERLINDUNGAN HUKUM PENGGUNAAN DIGITAL WATERMARK SEBAGAI HAK ATAS KEKAYAAN INTELEKTUAL MULTIMEDIA. *Jurnal Ilmu Hukum, Sosial, dan Humaniora*, 147.
- Gultoma, C. E., & Suhartanaa, K. G. (2023). Penerapan Steganografi dan Visible Watermarking Pada Gambar Digital Untuk Perlindungan Hak Cipta . *Jurnal Elektronik Ilmu Komputer Udayana* , 377.
- Putri, A. E., Kartikadewi, A., & Abdul Rosyid, L. A. (2020). Implementasi Kriptografi Dengan Algoritma Advanced Encryption Standard (AES) 128 Bit Dan Steganografi Menggunakan Metode End Of File (EOF) Berbasis Java Desktop Pada Dinas Dalam Steganografi (steganography) adalah ilmu dan seni. *Applied Information Systems and Management (AISM)*, 73.
- Lubis, R. A., & Nasution, Z. (2024). Metadata dan Perannya dalam Sistem Informasi Digital. *Jurnal Teknologi dan Sistem Informasi*, 25–33.
- Yudhanto, D., & Mayesti, F. (2021). Peran Metadata dalam Repositori Penelitian: Studi Kasus pada SINTA dan RIN. *Jurnal Ilmu Perpustakaan dan Informasi*, 88-95.
- Putra, A., & Santoso, B. (2022). Pengantar Keamanan Informasi dalam Sistem Digital. *Jurnal Teknologi Informasi dan Keamanan*, 15-22.
- Rahman, C., & Lestari, D. (2023). Strategi dan Implementasi Keamanan Data. *Jurnal Sistem Informasi dan Komunikasi*, 50.
- Saifurrohman, A., Poetro, B. S., & Taufik, M. (2023). Implementasi Steganografi Line Shifting untuk Pengamanan Dokumen PDF. *Seri Risalah Informatika dan Manajemen (SERIMA)*, 1.
- Loekito, C. H., Indriyani, T., & Rozi, N. F. (2020). Aplikasi Pengamanan Dokumen PDF dengan Teknik Watermarking Menggunakan Metode Serpent Chiper. *Jurnal Teknologi dan Manajemen*, 28.
- Pramudia, D., Fitriansyah, A., & Ramliyana, R. (2021). Aplikasi Hybrid Steganografi EOF dan Enkripsi AES-128 untuk Keamanan File PDF pada Android. *Jurnal Riset dan Aplikasi Mahasiswa Informatika (JRAMI)*, 1-8.
- Ridwan, M. K., Pattipeilohy, W. F., & Sanwani, S. (2020). Aplikasi Keamanan Dokumen Digital Menggunakan Algoritma Steganografi Discrete Cosine Transform (DCT) pada Perusahaan Alat Berat. *Jurnal Ilmu Pengetahuan dan Teknologi Komputer (JITK)*, 177.

- Apple, I. (2023). *Apple platform security*. Retrieved from Apple: [https://help.apple.com/pdf/security/id\\_ID/apple-platform-security-guide-id.pdf](https://help.apple.com/pdf/security/id_ID/apple-platform-security-guide-id.pdf)
- Mulyani, R. (2022). Keamanan Sistem Operasi iOS dan Android dalam Perbandingan. *Jurnal Literasi Sains*, 6(1), 55.
- Arnomo, S. A., & H. (2019). Perbandingan Fitur Smartphone, Pemanfaatan dan Tingkat Usability Pada Android dan iOS Platforms. *Jurnal Nasional Informatika dan Teknologi Jaringan*, 184.
- Barros Assunção, J. V., Rodrigues Pantoja, R. P., Pereira Gonçalves, B., Barbosa de Alencar, D., & Lobo de Oliveira, J. M. (2019). Comparative Analysis Between Native and Hybrid Mobile Applications. *International Journal for Innovation Education and Research*, 133.
- Khoirudin, K., Daru, A. F., & Nugroho, A. (2019). Hybrid Mobile Application Dengan Metode Service Oriented Architecture. *Jurnal Informatika Upgris*, 1-10.
- Pradana, D. I., & Waspada, I. (2019). Aplikasi Hybrid pada Sistem Informasi Penyewaan Buku. *Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, 1-10.
- Sinlae, D. (2024). Menjelajahi Dunia Web: Panduan Pemula Untuk Pemrograman Web. *Jurnal Sains dan Matematika Digital*, 45.
- Pratama, R., & Siregar, H. (2024). Pengembangan Aplikasi Random Team Generator Berbasis Web Menggunakan HTML, CSS, dan JavaScript. *Jurnal Manajemen dan Informatika*, 60.
- Fauzan, D. R., & Ramadhani, R. (2023). Pengembangan Aplikasi Web Dinamis Menggunakan React JS dan Firebase. *Jurnal Teknik Komputer AMIK BSI*, 54-61.
- Pratama, R., & Hidayat, S. (2021). Pengembangan Aplikasi Mobile Cross-Platform Menggunakan Flutter. *Jurnal Teknologi Informasi dan Komunikasi*, 99-106.
- Santoso, A., Putra, R., & Utomo, D. (2022). Studi Implementasi Flutter dalam Pengembangan Aplikasi Mobile. *Jurnal Sistem Informasi*, 45.
- Wibowo, T. (2023). Pemanfaatan Flutter untuk Pengembangan Aplikasi Web dan Desktop. *Jurnal Informatika Terapan*, 10.18.

- Sebastian, A., & Rahman, F. (2020). Dasar-dasar Bahasa Pemrograman untuk Pemula. *Jurnal Ilmu Komputer*, 99-105.
- Sari, D. P., & Nugroho, L. (2022). Perbandingan Bahasa Tingkat Tinggi dan Tingkat Rendah dalam Pengembangan Software. *Jurnal Sistem Komputer*, 45.
- Juniarti, J. A., Pratama, K. I., Utami, C. C., Merlina, N., & Mayangky, N. A. (2023). Digitalisasi Sistem Informasi Pelayanan Perizinan Ekspor dan Impor Benih Tanaman. *Journal of Hypermedia & Technology-Enhanced Learning*, 51.
- Prasetya, M., & Lestari, S. (2021). Manajemen Database dan Aplikasinya dalam Sistem Informasi. *Jurnal Teknologi Informasi*, 120.
- Setiawan, A., & Hartono, B. (2022). Pemanfaatan Docker dalam pengembangan aplikasi modern. *Jurnal Teknologi Informasi Terapan*, 42.
- Lubis, F., & Santoso, D. (2021). Studi implementasi Docker untuk containerization pada lingkungan cloud. *Jurnal Sistem Komputer*, 110.
- Dragoni, N., Lanese, I., Larsen, S. T., Mazzara, M., Mustafin, R., & Safina, L. (2020). Microservices: Yesterday, today, and tomorrow. In *Present and Ulterior Software Engineering*. Springer, 195.
- Richardson, C. (2021). *Microservices patterns: With examples in Java*. Retrieved from Manning Publications: <https://www.manning.com/books/microservices-patterns>
- Aljawarneh, S. (2021). Cloud computing security challenges and solutions: A survey. *Journal of Network and Computer Applications*, 178.
- Khan, S. U., Uddin, M., & Gani, A. (2022). Resource management in cloud computing: Taxonomy and survey. *Journal of Network and Computer Applications*, 190.
- Magazine, A. (2023). *An open-source object storage solution*. Retrieved from ADMIN Magazine: <https://www.admin-magazine.com/Archive/2023/75/An-open-source-object-storage-solution>
- Malhotra, S., Yashu, F., Saqib, M., Mehta, D., Jangid, J., & Dixit, S. (2025). Evaluating Fault Tolerance and Scalability in Distributed File Systems: A Case Study of GFS, HDFS, and MinIO. *arXiv*.
- AdonisJS. (2025). *What is AdonisJS?* Retrieved from AdonisJS.com: <https://docs.adonisjs.com/guides/preface/introduction>

- Pavithran, K., & Ganapathy, B. (2022). Building Secure and Scalable Node.js Applications using AdonisJS Framework. *International Journal of Engineering Research & Technology (IJERT)*, 153.
- Behrang, M., & Ali, K. (2021). Comparative Study of Node.js Web Frameworks for Enterprise Application Development. *Journal of Software Engineering and Applications*, 115.
- Judd, J. (2018). *Benchmarking Web Framework Response Times (Part 2) — Enter Node.js*. Retrieved from medium.com: [https://medium.com/@jamesjudd\\_21057/benchmarking-web-framework-response-times-part-2-enter-node-js-16a635e36428](https://medium.com/@jamesjudd_21057/benchmarking-web-framework-response-times-part-2-enter-node-js-16a635e36428)
- GoodFirms. (2025). *Flutter 2025: Definition, Key Trends, and Statistics*. Retrieved from GoodFirms: <https://www.goodfirms.co/blog/flutter-2025-definition-key-trends-statistics>
- linkedin.com. (2023). *Hybrid app framework with reference to M- Learning*. Retrieved from linkedin.com: <https://www.linkedin.com/pulse/hybrid-app-framework-reference-m-learning-dr-aniket-srivastava/>
- Pusiknas. (2021). *Rata-rata, tujuh kasus pemalsuan surat ditangani Polri setiap hari*. Retrieved from Pusiknas Polri: [https://pusiknas.polri.go.id/detail\\_artikel/rata-rata%2C\\_tujuh\\_kasus\\_pemalsuan\\_surat\\_ditangani\\_polri](https://pusiknas.polri.go.id/detail_artikel/rata-rata%2C_tujuh_kasus_pemalsuan_surat_ditangani_polri)
- Nugroho, A., & Putra, D. (2023). Leveraging Dart Language Features for Efficient Flutter Application Development. *Journal of Software Engineering and Mobile Computing*, 45-53.
- Sari, N., & Widodo, A. (2022). The role of JavaScript in modern web application development. *International Journal of Computer Science and Information Technology*, 25.
- Putra, R., & Hartono, D. (2023). Asynchronous programming in JavaScript: Techniques and performance analysis. *Journal of Web Engineering and Technologies*, 78.
- Wijaya, R., & Santoso, A. (2021). Enhancing code quality and maintainability with TypeScript in large-scale web applications. *Journal of Software Engineering and Applications*, 123.
- Putri, S., & Hidayat, T. (2022). Impact of TypeScript adoption on debugging time and developer productivity in front-end development. *International Journal of Computer Science Trends and Technology*, 45-52.

- Barokah, R., & Sauda, M. (2022). Pengembangan aplikasi e-commerce “Localla” menggunakan Node.js dan PostgreSQL. *Jurnal Informatika dan Sistem Informasi*, 45.
- Saputra, F., & Wibowo, A. (2021). Pengembangan aplikasi web menggunakan ReactJS dengan pendekatan component-based. *Jurnal Teknologi Informasi dan Komputer*, 87.
- Hartono, R., & Wijaya, S. (2022). Implementasi ReactJS dalam pengembangan aplikasi e-commerce berbasis web. *Jurnal Sistem Informasi dan Teknologi Informasi*, 15-25.
- Zebua, B., Herwanto, P., & Rosida, R. (2022). Penggunaan Encripsi MD5 Untuk Pencegahan SQL Injection Pada Aplikasi Berbasis Web. *Prosiding Seminar Nasional Inovasi Dan Adopsi Teknologi (INOTEK)*. 2, pp. 22–31. Karawang: STMIK ROSMA. doi:<https://doi.org/10.35969/inotek.v2i1.206>
- Kurniawan, A., Humaira, I., & Rozi, M. (2020). Sistem transaksi jasa elektronik berbasis Android menggunakan Node.js dan REST API. *Jurnal Ilmiah Teknologi Informasi dan Sistem Informasi*, 123.

## LAMPIRAN

### Kode Program



1. *Mobile App*
2. *Web App*
3. *Database Migration*
4. *Base Service*
5. *Public Service*
6. *Mobile Service*
7. *Document Service*