IMPLEMENTASI SISTEM *WATERMARKING* TAK TERLIHAT PADA BAHAN AJAR DIGITAL MENGGUNAKAN KOMBINASI *QR CODE* DAN *STEGANOGRAFI* CITRA (METODE LSB)

SKRIPSI

Diajukan sebagai Salah Satu Syarat untuk Mendapatkan
Gelar Sarjana Komputer (S.Kom) Program Studi Informatika

ARIKAL KHAIRAT
105841108421

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH MAKASSAR
2025

IMPLEMENTASI SISTEM *WATERMARKING* TAK TERLIHAT PADA BAHAN AJAR DIGITAL MENGGUNAKAN KOMBINASI *QR CODE* DAN *STEGANOGRAFI* CITRA (METODE LSB)

Diajukan sebagai Salah Satu Syarat untuk Mendapatkan
Gelar Sarjana Komputer (S.Kom) Program Studi Informatika

Disusun Dan Diajukan Oleh :

ARIKAL KHAIRAT
105841108421

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH MAKASSAR
2025

PENGESAHAN

Skripsi atas nama Arikal Khairat dengan nomor induk Mahasiswa 105841108421, dinyatakan diterima dan disahkan oleh Panitia Ujian Tugas Akhir/Skripsi sesuai dengan Surat Keputusan Dekan Fakultas Teknik Universitas Muhammadiyah Makassar Nomor : 0004/SK-Y/55202/091004/2025, sebagai salah satu syarat guna memperoleh gelar Sarjana Komputer pada Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar pada hari Sabtu, 30 Agustus 2025.

Panitia Ujian :

Makassar, 06 Rabi'ul Awal 1447 H / 30 Agustus 2025 M

1. Pengawas Umum
   a. Rektor Universitas Muhammadiyah Makassar
      Dr. Ir. H. Abd. Rakhim Nanda, ST.,MT.,IPU :
   b. Dekan Fakultas Teknik Universitas Hasanuddin
      Prof. Dr. Eng. Muhammad Isran Ramli, S.T., M.T., ASEAN.,Eng :
2. Penguji
   a. Ketua : Dr.Ir. Zahir Zainuddin, M.Sc. :
   b. Sekertaris : Desi Anggreni, S.Kom., M.T. :
3. Anggota : 1. Ir. Muhammad Faisal, S.SI., M.T., Ph.D., IPM :
   2. Runal Rezkiawan B, S.Kom., M.T :
   3. Darniati, S.Kom., M.T. :

Mengetahui :

Pembimbing I	Pembimbing II
(Titin Wahyuni, S.Pd., M.T.)	(Muhyiddin A M Hayat, S.Kom., M.T.)

Dekan

Ir. Muh. Syaraat S. Kuba, S.T., M.T.
NBM : 795 288

HALAMAN PENGESAHAN

Tugas Akhir ini diajukan untuk memenuhi syarat ujian guna memperoleh gelar Sarjana Komputer (S.Kom) Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar.

Judul Skripsi : **IMPLEMENTASI SISTEM *WATERMARKING* TAK TERLIHAT PADA BAHAN AJAR DIGITAL MENGGUNAKAN KOMBINASI *QR CODE* DAN *STEGANOGRAFI* CITRA (METODE LSB)**

Nama : Arikal Khairat

Stambuk : 105 84 11084 21

Makassar, 30 Agustus 2025

Telah Diperiksa dan Disetujui
Oleh Dosen Pembimbing;

Pembimbing I	Pembimbing II
(Titin Wahyuni, S.Pd., M.T.)	**(Muhyiddin A M Hayat, S.Kom., M.T.)**

Mengetahui,
Ketua Prodi Informatika

Rizki Yusliana Bakti, S.T., M.T.
NBM : 1307 284

ABSTRAK

ARIKAL KHAIRAT. Implementasi Sistem Watermarking Tak Terlihat pada Bahan Ajar Digital Menggunakan Kombinasi QR Code dan Steganografi Citra (Metode LSB) (Dibimbing oleh Titin Wahyuni dan Muhyiddin AM Hayat).

Penelitian ini bertujuan melindungi hak cipta bahan ajar digital dengan mengimplementasikan sistem watermarking tak terlihat. Metode yang digunakan adalah kombinasi QR Code sebagai pembawa informasi dan teknik steganografi Least Significant Bit (LSB) sebagai media penyisipan, yang dikembangkan dalam bentuk aplikasi berbasis web.

Proses sistem terdiri dari tahap penyisipan (embedding) dan ekstraksi (extracting). Pada tahap embedding, QR Code berisi identitas disisipkan ke dalam citra dokumen digital, sementara pada tahap extracting dilakukan pemulihan QR Code untuk validasi. Pengujian sistem dilakukan dengan perhitungan PSNR, MSE, serta uji keberhasilan ekstraksi.

Hasil penelitian menunjukkan bahwa watermark tidak menimbulkan perbedaan visual signifikan, dengan nilai PSNR 55,4–67,1 dB dan MSE 0,02–0,19. Proses ekstraksi berhasil 100% dengan QR Code yang dapat dipindai secara akurat. Kesimpulannya, kombinasi QR Code dan LSB efektif digunakan sebagai solusi perlindungan hak cipta bahan ajar digital.

Kata Kunci: watermarking tak terlihat, QR Code, steganografi, LSB, bahan ajar digital

ABSTRACT

ARIKAL KHAIRAT. Implementation of Invisible Watermarking System on Digital Learning Materials Using a Combination of QR Code and Image Steganography (LSB Method**)** (Supervised by Titin Wahyuni and Muhyiddin AM Hayat).

This study aims to protect the copyright of digital learning materials by implementing an invisible watermarking system. The method combines QR Code as the information carrier and the Least Significant Bit (LSB) steganography technique as the embedding process, developed in a web-based application.

The system consists of two main stages: embedding and extracting. In the embedding stage, a QR Code containing identity information is inserted into images within digital documents, while in the extracting stage the QR Code is recovered for validation. System performance was tested using PSNR, MSE, and extraction success rate.

The results show that the watermark caused no significant visual differences, with PSNR values ranging from 55.4 to 67.1 dB and MSE values between 0.02 and 0.19. The extraction process achieved 100% success, with all QR Codes accurately scanned. In conclusion, the combination of QR Code and LSB is effective as a solution for copyright protection of digital learning materials.

Keywords: invisible watermarking, QR Code, steganography, LSB, digital learning materials

KATA PENGANTAR

Assalamu'Alaikum Warahmatullahi Wabarakatuh

Segala Puji bagi Allah SWT atas limpahan rahmat, kesehatan dan kekuatannya sehingga proposal skripsi dengan judul "IMPLEMENTASI SISTEM *WATERMARKING* TAK TERLIHAT PADA BAHAN AJAR DIGITAL MENGGUNAKAN KOMBINASI *QR CODE* DAN *STEGANOGRAFI* (METODE LSB)" ini dapat penulis selesaikan sebagaimana salah satu syarat untuk penyusunan Skripsi Program Studi Informatika. Shalawat dan junjungan Nabi Besar Muhammad SAW sebagai uswatun hasanah dan rahmatan lil alamin.

Dengan segala kerendahan hati, penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada pihak-pihak yang telah memberikan dukungan, bimbingan, dan doa selama proses penyusunan proposal ini, khususnya kepada:

1. Kedua orang tua serta saudara-saudari tercinta, penulis mengucapkan terima kasih yang tak terhingga atas segala doa yang selalu mengiringi setiap langkah, semangat yang tak pernah padam, serta dukungan moral dan material yang tiada henti. Tanpa kasih sayang, pengorbanan, dan ketulusan mereka, penulis tidak akan mampu mencapai titik ini. Setiap pencapaian yang diraih adalah buah dari cinta dan perjuangan mereka yang luar biasa.
2. **Bapak Muhyiddin AM Hayat, S.Kom., M.T.**, selaku Ketua Program Studi Informatika sekaligus Dosen Pembimbing II, penulis menyampaikan penghormatan dan rasa terima kasih yang sebesar-besarnya atas bimbingan, arahan, serta motivasi yang telah beliau berikan dengan penuh kesabaran dan dedikasi.
3. **Ibu Titin Wahyuni, S.Pd., M.T.**, selaku Dosen Pembimbing I, penulis menghaturkan rasa terima kasih atas kesabaran, ketelitian, dan perhatian beliau dalam membimbing penulis melalui setiap tahapan penyusunan skripsi ini.
4. Seluruh Dosen dan Staf Fakultas Teknik Universitas Muhammadiyah Makassar atas ilmu dan bantuan yang diberikan selama masa studi penulis.
5. Ucapan terima kasih yang tulus penulis sampaikan kepada seorang terkasih, Andi Agung Dwi Arya B, S.Kom, atas segala dukungan dan semangat yang

diberikan sepanjang proses penyusunan skripsi ini. Kehadiran yang selalu memberikan arti, pengertian yang tulus, serta kesabaran yang tak ternilai telah menjadi sumber kekuatan ketika berada di titik lelah, hingga akhirnya skripsi ini dapat terselesaikan dengan baik.

6. Kepada teman-teman seperjuangan, khususnya rekan-rekan angkatan 2021 Informatika Universitas Muhammadiyah Makassar, penulis ingin menyampaikan rasa hormat dan terima kasih yang tulus atas kebersamaan, dukungan, dan semangat yang telah terjalin selama masa perkuliahan. Tawa, lelah, diskusi hingga larut malam, serta perjuangan bersama dalam menyelesaikan setiap tantangan menjadi kenangan yang tak ternilai. Terima kasih telah menjadi bagian dari perjalanan ini dan saling menguatkan satu sama lain di setiap langkah.

Demikian laporan skripsi ini penulis buat, dan penulis sadar bahwa laporan ini masih memiliki banyak kekurangan di dalamnya olehnya itu saran dan kritik yang sifatnya membangun dari pembaca sangat penulis harapkan untuk kesempurnaan kedepannya.

Billahi fi sabililhaq, fastabiqul khairat.
Wassalamu'Alaikum Warahmatullahi Wabarakatuh.

Makassar, 07 Mei 2025

Arikal Khairat

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

DAFTAR ISTILAH

Watermarking	Teknik penyisipan informasi *(watermark)* ke dalam data digital untuk proteksi hak cipta, autentikasi, dan pelacakan integritas data.
Steganografi	Ilmu dan seni menyembunyikan pesan rahasia di dalam media lain sehingga keberadaan pesan tersebut tidak terdeteksi.
Invisible Watermarking	Teknik *watermarking* di mana tanda air disembunyikan sehingga tidak mengganggu persepsi visual media asli.
Least Significant Bit (LSB)	Metode *steganografi* yang mengganti bit terakhir dari data piksel citra penampung dengan bit dari pesan rahasia atau *watermark*.
QR Code	Kode batang dua dimensi yang mampu menyimpan berbagai jenis informasi dalam format visual yang dapat dipindai dengan cepat.
Imperceptibility	Ketidakterlihatan; dalam konteks penelitian ini, mengacu pada kualitas *watermark* yang tidak terlihat oleh mata manusia.
Cover Image	Citra penampung, yaitu media (gambar) yang digunakan untuk

	menyembunyikan pesan atau *watermark*.
Stego Image	Citra yang dihasilkan setelah proses penyisipan *watermark*, yang secara visual identik dengan citra asli namun mengandung data tersembunyi.
Stego Key	Kunci *steganografi*, digunakan untuk menentukan urutan piksel target dalam proses penyisipan atau ekstraksi *watermark*.
Payload	Muatan; dalam konteks penelitian ini, mengacu pada informasi hak cipta yang akan disembunyikan dalam *QR Code*.
MSE (Mean Squared Error)	Metrik untuk mengukur perbedaan kuantitatif antara dua citra.
PSNR (Peak Signal-to-Noise Ratio)	Metrik untuk mengukur kualitas citra setelah penyisipan *watermark*. Nilai PSNR yang tinggi mengindikasikan kualitas yang baik
CRC32 (Cyclic Redundancy Check 32-bit)	Metode pemeriksaan kesalahan dengan menghasilkan nilai *checksum* 32-bit untuk memverifikasi integritas data. Jika nilai CRC32 sama antara data asli dan hasil, berarti data tidak berubah.

BAB I
PENDAHULUAN

A. Latar Belakang

Era digital telah merevolusi cara penyampaian informasi, termasuk dalam dunia pendidikan melalui penggunaan bahan ajar digital yang semakin masif, kemudahan akses dan potensi interaktif yang ditawarkan menjadikan materi pembelajaran digital sebagai pilihan populer, memungkinkan penyajian materi yang lebih menarik (Faisal et al., 2020). Akan tetapi, kemudahan dalam mendistribusikan konten digital ini juga membuka celah terhadap isu keamanan, terutama terkait perlindungan hak kekayaan intelektual. Materi seperti modul elektronik, gambar ilustrasi, atau video pembelajaran menjadi sangat rentan terhadap tindakan penyalinan, perubahan, dan pemanfaatan ilegal tanpa atribusi yang layak kepada pencipta aslinya (Wulandari, 2024). Kondisi ini tidak hanya berpotensi menimbulkan kerugian bagi pengembang konten, tetapi juga dapat membahayakan integritas informasi jika konten tersebut dimanipulasi, misalnya melalui teknik *copy-move, splicing,* atau *retouching* (Fadlika Satria et al., 2021).

Menjawab tantangan ini, berbagai metode pengamanan data digital telah dikembangkan. Menurut Yanti & Budayawan (2023), Teknik kriptografi dapat menyandikan data menggunakan algoritma seperti Vigenere Cipher atau AES, namun keberadaan data tersandi itu sendiri bisa terdeteksi. Sebagai alternatif, teknik *steganografi* dan *Watermarking* bertujuan menyembunyikan informasi autentikasi atau kepemilikan langsung di dalam media digital itu sendiri, termasuk untuk aset budaya seperti batik, *watermarking* secara khusus berfungsi untuk menandai konten digital guna membuktikan keaslian dan kepemilikan (WIDIYONO et al., 2021).

Terdapat dua pendekatan utama dalam *watermarking* yaitu *visible* dan *invisible.* Meskipun *visible watermark* dapat secara jelas menunjukkan kepemilikan, penempatannya sering kali mengurangi nilai estetika, dan kenyamanan pengguna pada bahan ajar. Sebaliknya, *invisible watermarking*

menawarkan solusi dengan menyisipkan data secara tersembunyi tanpa mengubah kualitas visual konten secara kasat mata Gultom & Suhartana (2023), sehingga integritas tampilan bahan ajar tetap terjaga dan keasliannya dapat diverifikasi saat diperlukan.

Least Significant Bit (LSB) adalah teknik *steganografi* yang umum digunakan untuk menyisipkan dan mengekstrak informasi rahasia dalam media digital. Menurut Nur Aqsal Aminullah et al. (2022) Cara kerja metode LSB yaitu mengubah bit terakhir (bit yang paling tidak signifikan) dari data piksel citra penampung *(cover image)* yang tidak berpengaruh signifikan dengan bit dari gambar sebelumnya. Metode ini banyak digunakan dalam *invisible watermarking* karena implementasinya yang relatif mudah dan cepat, kemampuan menyisipkan data dalam jumlah cukup besar, serta efek minim terhadap persepsi visual citra penampung Aditya Permana & Amma (2022). Berbagai studi telah menunjukkan kelayakan LSB untuk penyembunyian informasi pada citra digital, baik untuk pesan teks maupun citra *watermark*, (Afsari et al., 2022).

Seiring dengan teknologi, *Quick Response (QR) Code* telah diadopsi secara luas sebagai cara praktis untuk menyimpan dan mengakses beragam jenis data melalui pemindaian dengan perangkat mobile. Fleksibilitas dan kapasitas penyimpanan *QR Code* menjadikannya komponen yang potensial untuk diintegrasikan dalam sistem keamanan data (Ferdiansyah et al., 2021). Menggabungkan *QR Code* dengan *steganografi* LSB membuka peluang baru untuk proteksi bahan ajar digital, di mana *QR Code* dapat berfungsi sebagai pembawa informasi yang kemudian disisipkan secara tersembunyi.

Penelitian ini mengimplementasikan sebuah sistem *invisible watermarking* yang dirancang khusus untuk melindungi bahan ajar digital. Fokus utama adalah pada implementasi dan mengetahui kinerja sistem dalam hal penyisipan dan ekstraksi *watermark* secara akurat.

B. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana implementasi sistem *watermarking* tak terlihat pada bahan ajar digital menggunakan kombinasi *Quick Response (QR)* code sebagai muatan *watermark* dan metode *steganografi Least Significant Bit (LSB)*?
2. Sejauh mana tingkat kinerja metode kombinasi *QR Code* dan LSB dalam konteks *watermarking* tak terlihat pada bahan ajar digital, khususnya dalam hal *imperceptibility* (ketidakterlihatan)?

C. Tujuan Penelitian

1. Untuk mengimplementasikan sistem *watermarking* tak terlihat pada bahan ajar digital dengan menggabungkan *Quick Response (QR) Code* sebagai muatan *watermark* dan metode *steganografi Least Significant Bit (LSB)* sebagai teknik penyisipan data.
2. Untuk mengetahui tingkat kinerja dari metode kombinasi *QR Code* dan LSB dalam konteks *watermarking* tak terlihat, khususnya dalam aspek *imperceptibility* (ketidakterlihatan) terhadap kualitas visual bahan ajar digital.

D. Manfaat Penelitian

1. Bagi Penulis

   Menambah wawasan dan pemahaman penulis mengenai teknik *watermarking* digital, khususnya yang menggunakan kombinasi *QR Code* dan metode *steganografi Least Significant Bit (LSB).*
2. Bagi Pembaca

   Menjadi referensi bagi peneliti, akademisi, maupun pengembang sistem dalam mengembangkan teknologi perlindungan hak cipta berbasis *steganografi* dan *QR Code.*

E. Ruang Lingkup Penelitian

1. Penelitian ini dibatasi pada bahan ajar digital yang mengandung gambar, karena metode *steganografi* LSB pada penelitian ini diterapkan khusus pada citra digital. Bahan ajar non-gambar, seperti dokumen teks atau PDF tanpa elemen visual, tidak termasuk dalam ruang lingkup penelitian.
2. Fokus penelitian adalah pada perlindungan bahan ajar digital dari tindakan penyalinan secara digital. Penelitian ini tidak mencakup pengamanan

terhadap bahan ajar yang telah dicetak atau dipindai, karena *watermark* LSB tidak dapat dipertahankan pada media non-digital.

F. Sistematika Penulisan

Secara garis besar penulisan laporan tugas akhir ini terbagi menjadi beberapa bab yang tersusun sebagai berikut:

BAB I PENDAHULUAN

Bab ini menerangkan secara singkat dan jelas mengenai latar belakang penulisan penelitian tugas akhir, rumusan masalah, tujuan dan manfaat, batasan permasalahan, metodologi yang digunakan dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada bab ini membahas tentang teori-teori yang melandasi penulis dalam melaksanakan penelitian.

BAB III METODE PENELITIAN

Membahas tentang metode penelitian dan alat yang digunakan untuk pembuatan sistem.

BAB IV HASIL DAN PEMBAHASAN

Bab ini membahas hasil implementasi sistem dan pengujian yang dilakukan. Uraian meliputi deskripsi lingkungan uji, skenario dan prosedur pengujian, penyajian data hasil uji, analisis ketercapaian tujuan penelitian.

BAB V PENUTUP

Bab ini memuat kesimpulan yang diperoleh dari hasil penelitian serta saran untuk pengembangan sistem penelitian selanjutnya.

BAB II
TINJAUAN PUSTAKA

A. Landasan Teori

1. Bahan Ajar Digital

   Bahan ajar merupakan komponen esensial dalam proses pembelajaran, berfungsi sebagai materi atau substansi yang disusun sistematis untuk membantu guru dan siswa mencapai kompetensi pembelajaran. Di era digital, bahan ajar bertransformasi ke dalam format digital, menawarkan keunggulan seperti kemudahan akses, distribusi, dan potensi interaksi melalui multimedia (Antika et al., 2022). Namun, sifat digitalnya juga membawa tantangan terkait kemudahan penggandaan dan modifikasi ilegal, sehingga memerlukan mekanisme perlindungan hak cipta (Fathanudien & Maharani, 2023).

2. *Watermarking*

   Watermarking adalah teknik penyisipan informasi (*watermark*) ke dalam data digital (citra, audio, atau video) dengan tujuan utama untuk proteksi hak cipta, autentikasi (pembuktian keaslian), dan pelacakan integritas data (WIDIYONO et al., 2021). *Watermark* dapat berupa teks, logo, atau data biner lainnya yang mengidentifikasi pemilik atau keaslian konten. Berdasarkan visibilitasnya, *watermarking* dibagi menjadi *visible watermarking*, di mana tanda air terlihat jelas pada media, dan *invisible watermarking* (tak terlihat), di mana tanda air disembunyikan sehingga tidak mengganggu persepsi visual media asli (Gultom & Suhartana, 2023). Penelitian ini berfokus pada penerapan *invisible watermarking* menggunakan metode *Least Significant Bit (LSB)* dalam upaya perlindungan hak cipta pada bahan ajar digital.

3. Citra Digital

   Citra digital adalah gambar dua dimensi yang diubah dari bentuk sinyal analog menjadi bentuk digital melalui proses yang disebut sampling. Proses ini memecah gambar menjadi bagian-bagian kecil yang disebut

piksel. Setiap piksel menyimpan angka yang menunjukkan tingkat kecerahan atau warna, sehingga gambar tersebut bisa disimpan dan diolah oleh komputer.

Pengolahan citra digital adalah proses mengubah atau memanipulasi gambar digital agar kualitasnya menjadi lebih baik atau agar informasi di dalamnya lebih mudah dipahami oleh manusia maupun komputer. Proses ini dimulai dari membaca gambar sebagai input, lalu dilakukan serangkaian pengolahan, dan hasil akhirnya berupa gambar yang sudah dimodifikasi atau informasi penting dari gambar tersebut. Teknik ini memiliki banyak kelebihan, seperti pemrosesan yang cepat, mudah diterapkan, dan tidak merusak gambar aslinya (Devi & Rosyid, 2022).

4. *Steganografi*

Steganografi berasal dari bahasa Yunani yang berarti "tulisan tersembunyi". Ini adalah ilmu dan seni menyembunyikan keberadaan pesan rahasia di dalam media lain (disebut *cover object* atau media penampung) sedemikian rupa sehingga keberadaan pesan tersebut tidak terdeteksi oleh pihak ketiga. Berbeda dengan kriptografi yang menyandikan pesan tetapi tidak menyembunyikan fakta adanya komunikasi rahasia, *steganografi* bertujuan agar komunikasi rahasia itu sendiri tidak diketahui (Irawan & Pujianto, 2020). Teknik *invisible watermarking* sering kali menggunakan prinsip-prinsip *steganografi* untuk menyembunyikan data *watermark*.

5. Metode *Least Significant Bit (LSB)*

Least Significant Bit (LSB) adalah salah satu metode *steganografi* domain spasial yang paling umum dan sederhana. Prinsip kerjanya adalah mengganti bit terakhir (bit yang paling tidak signifikan) dari data piksel citra penampung *(cover image)* dengan bit dari pesan rahasia atau *watermark*. Karena perubahan hanya terjadi pada bit yang memiliki kontribusi paling kecil terhadap nilai total piksel, modifikasi ini umumnya tidak terdeteksi oleh mata manusia *(imperceptible)* (Purbaningrum et al., 2023).

Proses penyisipan dimulai dengan mengubah data rahasia, baik berupa teks, gambar, maupun file lain, ke dalam bentuk biner. Bit-bit biner

tersebut kemudian ditempatkan pada posisi LSB setiap piksel citra penampung. Semakin banyak bit yang digunakan dalam setiap piksel, semakin besar kapasitas penyimpanan, namun berpotensi menurunkan kualitas citra. Oleh karena itu, pemilihan jumlah bit yang dimodifikasi harus mempertimbangkan keseimbangan antara kapasitas dan kualitas visual (*imperceptibility*) (Hasan et al., 2020).

Kelebihan LSB adalah kesederhanaan implementasi dan kapasitas penyisipan yang relatif tinggi (Afsari et al., 2022). Sejumlah penelitian menunjukkan bahwa LSB dapat menghasilkan kualitas stego image yang baik dengan nilai *Peak Signal-to-Noise Ratio* (PSNR) di atas 40 dB, yang menandakan perbedaan visual sangat kecil antara citra asli dan citra hasil penyisipan (Alveda et al., 2024).

6. *Quick Response (QR) Code*

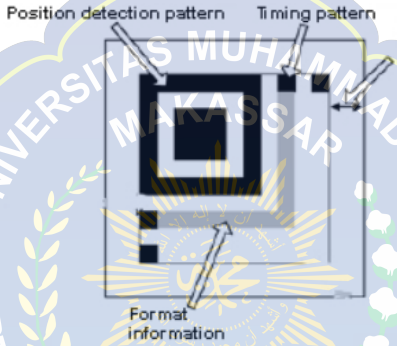
QR Code adalah jenis kode batang dua dimensi yang mampu menyimpan berbagai jenis informasi (numerik, alfanumerik, biner) dalam format visual yang dapat dipindai dengan cepat menggunakan kamera, biasanya pada perangkat seluler. Strukturnya yang khas, termasuk *finder patterns* dan *alignment patterns*, memungkinkan pemindaian dari berbagai sudut (Ferdiansyah et al., 2021). Kemampuannya untuk mengkodekan data dalam jumlah yang relatif besar dalam ruang visual yang kecil menjadikannya media yang menarik untuk membawa informasi *watermark* yang kemudian dapat disembunyikan menggunakan *steganografi*.

Berbeda dengan barcode konvensional yang hanya menyimpan data dalam satu arah, *QR Code* menyimpan informasi secara dua dimensi, sehingga kapasitas datanya jauh lebih besar. *QR Code* mampu menyimpan berbagai jenis informasi, seperti tautan situs web, informasi produk, SMS, hingga detail kontak yang mencakup nama, nomor telepon, dan alamat. Contoh visualnya ditunjukkan pada Gambar 1.

Gambar 1. Contoh *QR Code*

Jenis data yang dapat disimpan dalam *QR Code* meliputi:

1) Numerik (angka): hingga 7.089 karakter
2) Alfanumerik: hingga 4.296 karakter
3) Biner: hingga 2.844 byte
4) Koreksi kesalahan: Level L (7%), M (15%), Q (25%), dan H (30%)



Gambar 2. Struktur Dasar *QR Code*

Seperti yang ditunjukkan pada Gambar 2 Struktur *QR Code* terdiri dari beberapa komponen penting, antara lain:

1) Pola deteksi posisi (position detection patterns) yang terletak pada tiga sudut *QR Code* dan berfungsi untuk memungkinkan pemindaian cepat dari berbagai arah.
2) Margin, yaitu area kosong di sekeliling kode yang minimal terdiri dari empat modul.
3) Pola waktu (timing pattern) berupa modul hitam dan putih bergantian yang membantu menentukan posisi modul dalam *QR Code*.
4) Informasi format, bagian awal yang dibaca saat kode dipindai untuk menerjemahkan data.

Fungsi dasar *QR Code* serupa dengan barcode, yaitu sebagai alat identifikasi suatu objek. Namun, penggunaannya jauh lebih fleksibel,

seperti pada kartu nama, media promosi, hingga dokumen digital lainnya. Secara keseluruhan, *QR Code* memiliki keunggulan signifikan dibandingkan barcode karena kemampuannya menyimpan informasi dalam dua arah, sehingga dapat memuat data yang lebih kompleks dan beragam (Riansah, 2021).

7. Kombinasi *QR Code* dan *Steganografi* LSB

Penggabungan *QR Code* dan LSB menghasilkan metode yang memanfaatkan keunggulan *QR Code* sebagai wadah informasi yang ringkas dan LSB sebagai metode penyembunyian tak terlihat. Informasi *watermark* (data hak cipta, URL verifikasi, identitas pembuat) pertama-tama dikodekan ke dalam sebuah *QR Code*. Citra *QR Code* ini kemudian diperlakukan sebagai data rahasia yang akan disisipkan ke dalam bit-bit LSB dari piksel-piksel citra yang terdapat dalam bahan ajar digital. Sebagaimana dijelaskan oleh Alajmi et al. (2020), *QR Code* memiliki struktur biner yang padat, toleransi tinggi terhadap kerusakan, dan validitas tetap terjaga meskipun membawa pesan tersembunyi, menjadikannya medium yang sangat efektif untuk *steganografi* dan penyisipan payload tak terlihat.

Dibandingkan dengan penyisipan pesan berbentuk teks biasa, *QR Code* menawarkan keunggulan signifikan dalam konteks *steganografi*. Struktur dua dimensinya yang padat memungkinkan penyimpanan data kompleks dalam ukuran kecil, serta tahan terhadap kerusakan parsial yang bisa merusak pesan jika menggunakan teks biasa. Selain itu, *QR Code* dapat langsung diverifikasi secara visual menggunakan pemindai, mempercepat proses autentikasi informasi setelah diekstraksi. Dengan demikian, penggunaan *QR Code* dalam metode LSB tidak hanya meningkatkan efisiensi ruang, tetapi juga memperkuat keamanan dan keandalan dalam sistem *watermarking* tak terlihat (Ferdiansyah et al., 2021).

8. *Python*

Tujuan yang ingin dicapai adalah menciptakan perangkat lunak *steganografi* menggunakan *Python*. Perangkat lunak ini dirancang untuk menyembunyikan pesan atau informasi di dalam file gambar PNG, dengan

harapan file gambar yang dihasilkan tetap mempertahankan kualitas visual yang serupa dengan file aslinya. *Python* sendiri merupakan bahasa pemrograman yang banyak digunakan oleh pengembang dan juga mendukung komputasi serta visualisasi gambar (Aditya Permana & Amma, 2022).

9. *CRC32 (Cyclic Redundancy Check 32-bit)*

   CRC32 adalah fungsi hash 32-bit untuk pemeriksaan integritas dan deteksi kesalahan pada data tersimpan atau ditransmisikan, nilai *CRC32* berperan sebagai *checksum* atau identitas file. Nilainya sangat peka terhadap perubahan sehingga modifikasi sekecil apa pun pada data akan menghasilkan nilai CRC yang berbeda. Pada dokumen digital, pengecekan dilakukan dengan membandingkan *checksum* saat ini dengan catatan sebelumnya dan metode ini cepat namun hanya bersifat deteksi (tidak memperbaiki kesalahan) (Sagala, 2021).

B. Penelitian Terkait

Penelitian ini disusun dengan merujuk pada berbagai studi terdahulu yang relevan dengan permasalahan yang diangkat, khususnya yang berkaitan dengan latar belakang skripsi ini. Adapun beberapa penelitian yang terkait disajikan pada Tabel 1.

Tabel 1. Penelitian Terkait

Penelitian	Tujuan/Kasus	Metode	Hasil
Penyisipan *Watermark* Menggunakan Metode LSB (Least Significant Bit) untuk Autentikasi Citra Medis,	Menyisipkan *watermark* pada citra medis digital untuk tujuan autentikasi, tanpa mengganggu informasi	Menggunakan metode LSB *steganografi* dan menyisipkan *watermark* pada Region of Non-Interest (RONI), yaitu area yang tidak mengandung	*Watermark* berhasil disisipkan dengan baik dan tetap dapat diekstrak setelah citra mengalami gangguan seperti noise,

(Alveda et al., 2024)	penting pada citra tersebut.	informasi penting atau sensitif.	menunjukkan robustness sistem terhadap interferensi eksternal.
Penerapan *Steganografi* dan Visible *Watermark*ing Pada Gambar Digital Untuk Perlindungan Hak Cipta, (Gultom & Suhartana, 2023)	Menganalisis efektivitas kombinasi dua pendekatan *watermarking* dalam melindungi citra digital.	Penggabungan metode *visible watermarking* (yang terlihat langsung) dan *invisible watermarking* (tak terlihat) dengan LSB pada satu citra.	Pendekatan ini menghasilkan perlindungan ganda satu *watermark* terlihat sebagai penanda visual, sementara *watermark* tak terlihat tetap tersembunyi dan aman dari penghapusan atau manipulasi.
Implementasi *Steganografi* Menggunakan Metode Least Significant Bit (Lsb) dalam Pengamanan Informasi Pada Citra Digital, (Yanti &	Meneliti teknik penyembunyian pesan teks pada citra digital untuk menjaga kerahasiaan informasi.	Menggunakan kombinasi metode *steganografi Least Significant Bit* (LSB) untuk menyisipkan pesan dalam citra dan kriptografi *Vigenere Cipher* untuk	Penelitian menunjukkan hasil *imperceptibility* (ketidakterlihatan) yang baik dengan nilai PSNR (*Peak Signal-to-Noise Ratio*) di atas 30

Budayawan, 2023)		mengenkripsi pesan terlebih dahulu sebelum disisipkan.	dB, menandakan bahwa pesan yang disisipkan tidak menyebabkan degradasi signifikan pada citra penampung.
Proteksi Keamanan Data pada Quick Response (QR) Code, (Harits M et al., 2021)	Mengevaluasi berbagai metode pengamanan informasi dalam *QR Code*, baik dari sisi penyembunyian maupun enkripsi.	Studi literatur dan eksperimen terhadap metode enkripsi (AES, Speck) serta teknik *steganografi* untuk menyisipkan data ke dalam *QR Code* atau menyembunyikan *QR Code* ke dalam media digital lain.	Menunjukkan bahwa *QR Code* dapat berfungsi ganda sebagai kontainer informasi dan medium keamanan, terutama jika digabungkan dengan algoritma enkripsi dan teknik penyembunyian.
Analisis Digital *Watermark*ing untuk Otentikasi	Menggunakan *watermarking* untuk autentikasi citra digital dan	Menyisipkan *watermark* dengan metode LSB ke dalam citra asli dan membandingkan	Penelitian menunjukkan bahwa *watermark* akan mengalami

pada Citra Manipulasi Menggunakan Metode Least Significant Bit, (Fadlika Satria et al., 2021)	deteksi perubahan atau manipulasi.	*watermark* yang diekstrak setelah citra dimanipulasi.	kerusakan ketika citra dimanipulasi, sehingga dapat digunakan sebagai indikator integritas data untuk membedakan citra asli dan hasil rekayasa.
Teknik *Watermark*ing Menggunakan Metode Least Significant Bit Pada Citra Untuk Perlindungan Hak Cipta Motif Batik, (WIDIYONO et al., 2021)	Melindungi hak cipta terhadap motif batik digital, yang rawan diduplikasi atau dipalsukan.	Penerapan metode *watermarking* menggunakan LSB pada citra motif batik, dengan menyisipkan *watermark* hak cipta secara tak terlihat.	Hasil eksperimen menunjukkan *watermark* dapat disisipkan tanpa merusak tampilan visual citra, sekaligus dapat diekstraksi kembali untuk membuktikan kepemilikan asli.
Peningkatan Kompetensi Guru Sekolah Dasar dalam	Meningkatkan kemampuan guru dalam mengembangka	Pelatihan berbasis penggunaan platform digital populer seperti	Guru dapat membuat bahan ajar yang lebih menarik dan

Mengembangkan Bahan Ajar Digital di Kabupaten Gowa, (Faisal et al., 2020)	n bahan ajar digital secara interaktif.	Canva untuk desain visual dan Quizziz untuk evaluasi interaktif.	kreatif, meningkatkan partisipasi dan pemahaman siswa dalam pembelajaran daring.

Penelitian-penelitian yang telah dipaparkan pada Tabel 1 memberikan landasan yang kuat bagi penelitian ini. Secara kolektif, studi-studi tersebut menunjukkan relevansi penting dalam pengembangan metode *watermarking* tak terlihat.

1. Kelayakan Metode LSB

   Secara konsisten menunjukkan bahwa metode LSB efektif untuk menyisipkan data secara tak terlihat *(imperceptible)* ke dalam citra digital, yang diukur melalui metrik PSNR dan MSE. Hal ini mendukung pemilihan LSB sebagai teknik *steganografi* dalam penelitian ini untuk menjaga kualitas visual bahan ajar (Alveda et al., 2024).

2. Aplikasi *Watermarking* untuk Keamanan

   Studi-studi tersebut mengaplikasikan *watermarking* untuk berbagai tujuan keamanan, termasuk perlindungan hak cipta, autentikasi dan deteksi manipulasi, serta perlindungan ganda (Gultom & Suhartana, 2023). Ini menggarisbawahi potensi *watermarking* sebagai solusi relevan untuk masalah keamanan bahan ajar digital.

3. Potensi *QR Code* sebagai Pembawa *Watermark*

   Penelitian oleh Harits M et al. (2021) secara spesifik menyoroti *QR Code* sebagai medium yang dapat diamankan dan digunakan untuk membawa informasi, yang sejalan dengan ide penelitian ini untuk menggunakan *QR Code* sebagai muatan *(payload)* informasi hak cipta yang akan disembunyikan.

4. Konteks Bahan Ajar Digital

   Meskipun tidak secara langsung membahas *watermarking*, penelitian Faisal et al. (2020) menegaskan pentingnya dan semakin masifnya penggunaan bahan ajar digital, yang memperkuat urgensi untuk mengembangkan metode perlindungannya.

Dengan demikian, penelitian ini bertujuan untuk menyintesis temuan-temuan tersebut dengan mengimplementasikan dan mengetahui kombinasi spesifik antara *QR Code* (sebagai pembawa informasi hak cipta yang praktis) dan metode *steganografi* LSB (untuk penyisipan tak terlihat) dalam konteks perlindungan bahan ajar digital, dengan fokus utama pada aspek *imperceptibility* dan keberhasilan ekstraksi dasar.

C. Kerangka Berpikir

1. Masalah

   Dimulai dari identifikasi masalah utama yaitu kerentanan bahan ajar digital terhadap penyalinan dan modifikasi ilegal, yang mengarah pada kebutuhan perlindungan hak cipta.

2. Solusi Potensial

   Meninjau berbagai solusi kriptografi dan *watermarking*, kemudian mengerucut pada *invisible watermarking* sebagai pendekatan yang tidak mengganggu tampilan visual.

3. Teknik yang Dipilih

   Memilih metode *Least Significant Bit (LSB)* sebagai teknik *steganografi* karena kelebihannya dalam hal implementasi dan kapasitas, serta *QR Code* sebagai media praktis untuk membawa informasi *watermark*.

4. Solusi

   Menggabungkan *QR Code* (sebagai pembawa informasi hak cipta) dengan metode LSB untuk menyisipkan *QR Code* secara tak terlihat ke dalam citra bahan ajar.

5. Implementasi

   Merancang sistem yang terdiri dari dua proses utama yaitu

penyisipan *(Embedding) watermark QR Code* ke dalam citra dan ekstraksi *(extracting) watermark* dari citra.

6. Pengujian

   Menetapkan metrik dan teknik pengujian untuk mengetahui kinerja sistem, terutama fokus pada *imperceptibility* (apakah *watermark* benar-benar tak terlihat dan tidak merusak citra) dan keberhasilan ekstraksi (apakah informasi *watermark* dapat diambil kembali dengan akurat).

7. Hasil yang Diharapkan

   Menghasilkan sistem *watermarking* tak terlihat untuk melindungi hak cipta bahan ajar digital tanpa mengorbankan kualitas visualnya. Alur konseptual mengenai hubungan antara masalah, solusi, hingga hasil yang diharapkan ditunjukkan pada Gambar 3.

Gambar 3. Kerangka Berpikir

BAB III
METODE PENELITIAN

A. Tempat dan Waktu Penelitian

Tempat Penelitian adalah suatu tempat atau objek yang akan dilakukan suatu penelitian. Penentuan lokasi penelitian merupakan langkah penting dalam proses penelitian karena memudahkan peneliti untuk melakukan penelitian. Lokasi penelitian yang dipilih peneliti adalah di Universitas Muhammadiyah Makassar tepatnya di Laboratorium Informatika Fakultas Teknik.

Waktu penelitian ini akan dilakukan dalam jangka waktu kurang lebih 2 bulan, yaitu dimulai pada bulan Mei 2025 hingga Agustus 2025, sebagaimana ditampilkan pada Tabel 2.

Tabel 2. Jadwal Penelitian

NO	KEGIATAN	MEI		JUNI				JULI			
		III	IV	I	II	III	IV	I	II	III	IV
1	Studi Literatur										
2	Analisis Sistem										
3	Desain Sistem										
4	Implementasi Sistem										
5	Pengujian Sistem										
6	Penulisan Laporan										

B. Alat dan Bahan

Alat penelitian berupa laptop yang akan digunakan untuk mengembangkan sistem cerdas dalam pembuatan *watermark* tak terlihat pada bahan ajar digital. Dalam penelitian ini, peneliti menggunakan perangkat keras dan perangkat lunak:

1. Perangkat Keras (Pengembangan)
    a. *Processor* Intel(R) Celeron(R)
    b. Besar *Memory Ram* 4 GB
    c. Kapasitas SSD 512GB
2. Perangkat Lunak
    a. Linux - Ubuntu
    b. *Text editor Visual Studio Code*
    c. *Python* sebagai bahasa pemrograman

Bahan kajian dalam penelitian ini terdiri dari bahan ajar digital yang telah dikumpulkan dari berbagai sumber, khususnya dari guru secara langsung dan dari platform daring. Pengumpulan data dilakukan melalui teknik dokumentasi**,** yaitu dengan mengumpulkan file-file digital (dokumen) yang digunakan sebagai objek uji dalam proses penyisipan *watermark.*

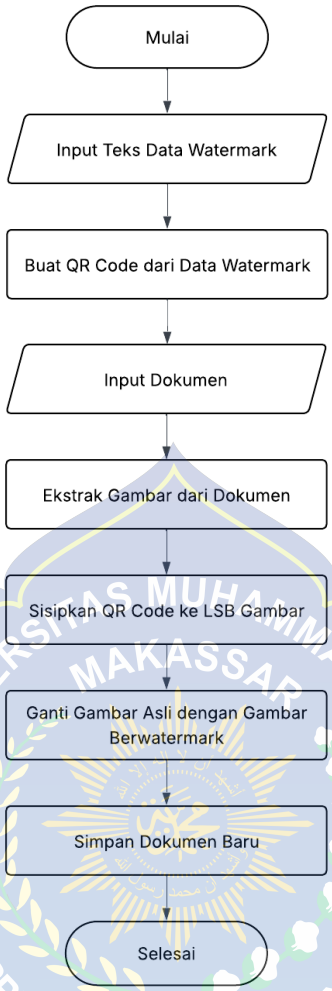
C. Perancangan Sistem

Perancangan sistem dalam penelitian ini bertujuan untuk mengimplementasikan penyisipan informasi hak cipta dalam bentuk *QR Code* ke dalam citra digital pada bahan ajar secara tak terlihat (*invisible watermarking*). Metode *steganografi* yang diterapkan menggunakan algoritma *steganografi Least Significant Bit (LSB).*

Proses ini dibagi menjadi dua tahapan utama, yaitu penyisipan *(Embedding)* dan ekstraksi *(extracting)*

1. Proses Penyisipan *(Embedding)*

Ilustrasi proses penyisipan *watermark* berupa *QR Code* ke dalam gambar yang terdapat pada dokumen digital (DOCX/PDF), menggunakan metode *Least Significant Bit (LSB)* ditampilkan pada Gambar 4. Setiap Langkah dilakukan secara berurutan dan saling terhubung untuk menghasilkan dokumen baru yang telah mengandung *watermark* tersembunyi.



Gambar 4. Flowchart Proses Penyisipan

Uraian setiap langkah pada flowchart Gambar 4 disajikan sebagai berikut:

a. Input Teks Data *Watermark*

Sistem menerima input berupa teks yang akan digunakan sebagai data *watermark*. Teks ini dapat berupa nama pemilik hak cipta, tautan, atau identitas penting lainnya.

Untuk menjaga konsistensi, kemudahan pemindaian, serta efisiensi penggunaan kapasitas QR Code, ditetapkan ketentuan penulisan data sebagai berikut:

1. Panjang Data

Panjang total data dibatasi maksimal 50 karakter. Batasan ini bertujuan agar ukuran QR Code tetap optimal, tidak terlalu padat,

serta dapat dipindai dengan mudah pada berbagai perangkat.

2. Identitas Bahan Ajar

   Data yang dimuat mencakup identitas bahan ajar, yaitu nama lengkap pemilik hak cipta bahan ajar digital.

QR Code semakin banyak digunakan sebagai media penyimpanan informasi dalam sistem keamanan digital karena kemampuannya menyimpan berbagai jenis data seperti teks, tautan, dan metadata. Namun demikian, kapasitas *QR Code* tetap terbatas dan dipengaruhi oleh jenis data serta tingkat koreksi kesalahan yang digunakan. Misalnya, pada format alfanumerik, *QR Code* mampu menampung hingga 4.296 karakter, namun semakin banyak karakter yang dimasukkan, ukuran fisik *QR Code* akan semakin besar.

Dalam konteks *steganografi* menggunakan metode Least Significant Bit (LSB), ukuran *QR Code* yang besar dapat menyebabkan perubahan signifikan pada citra yang digunakan sebagai media penyisipan, sehingga meningkatkan risiko penurunan kualitas visual. Oleh karena itu, pembatasan jumlah karakter dalam *QR Code* menjadi penting untuk menjaga efektivitas dan kualitas stego image. Pendekatan ini diterapkan oleh (Putri Pebriani et al., 2025), yang membatasi panjang pesan hanya sampai 8 karakter untuk memastikan integritas visual gambar tetap terjaga.

Penelitian yang dilakukan oleh Riansah (2021) juga menegaskan bahwa meskipun *QR Code* memiliki kemampuan menyimpan data dalam jumlah besar, semakin panjang data yang dikodekan akan menghasilkan pola *QR Code* yang semakin kompleks. Teks panjang perlu dikonversi ke dalam bentuk biner sebelum diproses menjadi *QR Code*, dan hal ini menyebabkan ukuran citra *QR Code* bertambah besar. Dalam konteks *steganografi* citra digital, kondisi ini dapat menyulitkan proses penyisipan, khususnya jika media citra memiliki resolusi yang terbatas. Oleh sebab itu, membatasi jumlah karakter yang dikodekan ke dalam *QR Code* hanya sampai 50 karakter, menjadi strategi yang efektif

untuk mempertahankan kestabilan visual dan meningkatkan keamanan data tersembunyi.

b. *Generate QR Code*

Data teks *watermark* yang diterima akan dikonversi menjadi citra *QR Code*. *QR Code* digunakan karena dinilai efisien dan praktis dalam menyimpan informasi yang padat serta mudah dibaca secara digital. Selain itu, *QR Code* memiliki tingkat ketahanan kesalahan *(error correction)* yang tinggi sehingga cocok untuk keperluan penyisipan informasi pada media visual (Ferdiansyah et al., 2021).

c. Input Dokumen

Menginput dokumen digital berformat DOCX atau PDF yang akan disisipkan *watermark*. Dokumen ini akan dianalisis untuk menemukan elemen gambar (seperti cover atau ilustrasi) sebagai media penampung *watermark*.

d. Ekstrak Gambar dari Dokumen

Sistem akan mengekstrak gambar dari dokumen. Format citra penampung sebaiknya menggunakan format *lossless* seperti PNG atau BMP untuk menghindari hilangnya bit penting pada LSB (Fadel et al., 2024). Setelah gambar diperoleh, sistem membaca nilai piksel citra dan mengonversinya ke dalam representasi biner.

e. Penyisipan *QR Code* ke LSB Gambar

Citra *QR Code* yang telah dihasilkan sebelumnya akan dikonversi menjadi aliran bit biner. Bit-bit ini kemudian disisipkan ke dalam posisi *Least Significant Bit* (LSB**)** dari piksel gambar penampung. Proses penyisipan dapat dilakukan secara sekuensial atau secara acak dengan menggunakan kunci *steganografi* (*stego key*) untuk menentukan urutan piksel target (Alveda et al., 2024). Perlu dipastikan bahwa kapasitas citra penampung mencukupi untuk menampung seluruh bit data *watermark* .

f. Ganti Gambar Asli dengan Gambar Ber-*watermark*

Setelah proses penyisipan selesai, sistem akan menggantikan gambar asli dalam dokumen dengan gambar hasil *steganografi*.

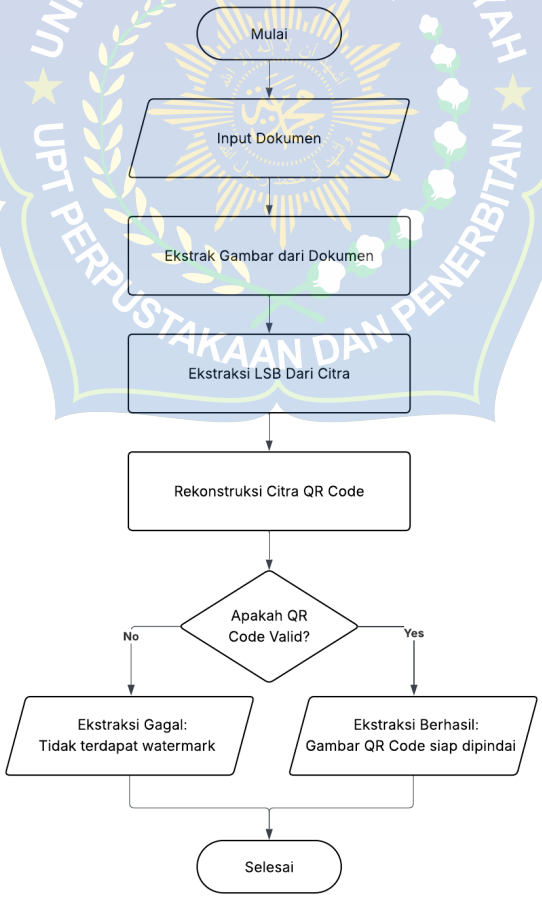
Langkah ini diperlukan karena proses penyisipan dilakukan pada file gambar secara terpisah dari dokumen. Dengan mengganti gambar di dokumen, *watermark* akan tersimpan secara permanen dalam konteks file aslinya.

g. Simpan Dokumen Baru

Dokumen yang telah berisi gambar ber-*watermark* disimpan kembali ke dalam format aslinya (PDF atau DOCX). Format hasil akhir tetap mempertahankan struktur dokumen, namun kini memiliki *watermark* tersembunyi yang dapat divalidasi.

2. Proses Ekstraksi *(Extracting)*

Tahapan ini bertujuan untuk mengekstraksi kembali informasi *watermark* dari *stego image*. Proses ini merupakan invers dari tahap penyisipan dan alur proses ekstraksi ditunjukkan pada Gambar 5.



Gambar 5. Flowchart Proses Ekstraksi

a. Input Dokumen

Sistem menerima masukan berupa dokumen digital berformat PDF atau DOCX yang akan diperiksa. Dokumen tersebut diasumsikan berisi gambar *(stego image)* yang telah disisipkan *watermark QR Code* sebelumnya.

b. Ekstrak Gambar dari Dokumen

Sistem mengekstrak citra digital (berupa gambar sampul atau ilustrasi) dari dalam dokumen. Gambar ini akan digunakan sebagai bahan utama untuk proses ekstraksi. Gambar yang berhasil diekstrak akan digunakan sebagai *stego image* untuk proses selanjutnya. Citra *stego* dibaca dan nilai pikselnya dikonversi ke dalam bentuk representasi biner (Yanti & Budayawan, 2023).

c. Ekstraksi LSB

Sistem melakukan pembacaan terhadap bit paling tidak signifikan ***(****Least Significant Bit)* dari setiap piksel gambar. Bit-bit ini diekstraksi berdasarkan pola penyisipan sebelumnya, baik secara berurutan maupun menggunakan *stego key*. Nilai piksel citra dikonversi ke dalam bentuk representasi biner sebagai dasar pembentukan ulang *watermark* (Alveda et al., 2024).

d. Rekonstruksi *QR Code*

Bit-bit hasil ekstraksi LSB disusun kembali menjadi aliran bit yang kemudian direkonstruksi menjadi citra *QR Code*. Proses ini menentukan apakah informasi *watermark* dapat dikenali atau tidak.

e. Validasi *QR Code*

Validasi *QR Code* merupakan tahap untuk memastikan apakah *watermark* berhasil direkonstruksi dari gambar. Pada tahap ini, sistem akan memeriksa apakah *QR Code* yang terbentuk valid dan dapat dipindai. Jika *QR Code* yang terbentuk valid dan dapat dipindai, maka proses ekstraksi berhasil. Artinya, *watermark* tersembunyi berhasil diambil. Sebaliknya, jika *QR Code* tidak valid atau tidak terbaca, maka proses ekstraksi dianggap gagal, ini menunjukkan bahwa tidak terdapat

watermark atau *watermark* rusak, sehingga sistem menampilkan notifikasi "Ekstraksi gagal: tidak terdapat *watermark*".

f. Keluaran (Output)

Output dari proses ekstraksi adalah *QR Code* yang berhasil direkonstruksi dari citra stego. *QR Code* ini dapat dipindai secara langsung menggunakan perangkat pemindai (scanner) standar, seperti kamera ponsel, untuk memperoleh informasi *watermark* berupa nama pemilik hak cipta, tautan verifikasi, atau data identitas lainnya. Keberhasilan proses ditandai dengan validitas *QR Code* yang dapat terbaca dan memuat informasi sesuai dengan data *watermark* yang disisipkan.

D. Teknik Pengujian Sistem

Pengujian sistem bertujuan untuk memperoleh data guna mengetahui kinerja dari metode *watermarking* tak terlihat yang diimplementasikan. Fokus pengujian adalah pada aspek ketidakterlihatan (*imperceptibility*) *watermark* dan keberhasilan ekstraksi. Teknik pengujian yang akan diterapkan meliputi:

1. Uji *Imperceptibility*
   a. Perbandingan Visual

      Melakukan observasi visual untuk mengidentifikasi perbedaan antara citra asli (*cover image*) dan citra hasil penyisipan (*stego image*) (Fadlika Satria et al., 2021).

   b. Pengukuran Kuantitatif (MSE & PSNR)

      Melakukan pengukuran kuantitatif perbedaan antara kedua citra menggunakan metrik *Mean Squared Error* (MSE) dan *Peak Signal-to-Noise Ratio* (PSNR). Nilai MSE yang rendah dan PSNR yang tinggi mengindikasikan tingkat *imperceptibility* yang baik (Alveda et al., 2024).

2. Uji Keberhasilan Ekstraksi *(Recovery)*

   Memverifikasi bahwa *QR Code* yang diekstraksi dapat dipindai dan menghasilkan informasi yang identik dengan informasi *watermark* awal (Putri Pebriani et al., 2025).

3. Perbandingan Ukuran File

Pengujian dilakukan dengan mencatat ukuran file citra (dalam KB atau MB) sebelum dan sesudah penyisipan *watermark*. Data ini digunakan untuk mengetahui apakah proses penyisipan memengaruhi ukuran file secara langsung (Fadel et al., 2024).

E. Teknik Analisis Data

Analisis dilakukan terhadap data yang dihasilkan dari tahap pengujian, teknik analisis yang digunakan bersifat kuantitatif dan deskriptif dengan menyajikan hasil dalam bentuk angka, persentase, dan perbandingan visual.

Tujuan dari analisis ini adalah untuk mengevaluasi tingkat keberhasilan dan kualitas hasil penyisipan serta ekstraksi *watermark*, berdasarkan tiga aspek utama berikut:

1. Analisis Ketidakterlihatan *(Imperceptibility)*

Analisis ini bertujuan untuk menilai sejauh mana proses penyisipan *watermark* memengaruhi kualitas visual citra digsital.

a. Secara Visual

Jika hasil observasi menunjukkan tidak adanya perbedaan mencolok antara gambar asli dan gambar *stego*, maka dapat disimpulkan bahwa *watermark* berhasil disisipkan secara tak terlihat (*invisible*).

b. Secara Kuantitatif (MSE & PSNR)

Analisis dilakukan berdasarkan hasil perhitungan nilai MSE (*Mean Squared Error*) dan PSNR (*Peak Signal-to-Noise Ratio*). Menurut Yanti & Budayawan (2023), jika nilai PSNR tinggi (umumnya di atas 30 dB) dan nilai MSE rendah, maka kualitas visual citra setelah disisipi *watermark* tetap terjaga dengan baik. Perhitungan tersebut dilakukan menggunakan Persamaan (1) dan (2).

Rumus MSE:

$$MSE = \frac{1}{mn} \sum_{i=1}^{m} \sum_{i=1}^{n} [I(i,j) - K(i,j)]^2 \qquad (1)$$

Keterangan:

$I(i, j)$: piksel citra asli

K(i, j) : piksel citra setelah disisipkan *watermark*

m, n : ukuran gambar

Rumus PSNR:

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX^2}{MSE}\right) \qquad (2)$$

Keterangan:

MSE : *Mean Squared Error*

MAX : nilai maksimum piksel (255 untuk gambar 8-bit)

2. Analisis Keberhasilan Ekstraksi

Analisis ini dilakukan untuk menilai keberhasilan sistem dalam mengekstraksi kembali *watermark* berupa *QR Code* dari citra hasil penyisipan *(stego image)*. *QR Code* yang diekstrak akan diuji menggunakan pemindai digital untuk memastikan bahwa informasi yang terkandung masih dapat dibaca dengan benar. Mengevaluasi apakah *QR Code* yang diekstrak bisa dibaca dengan benar "berhasil" atau tidak "gagal" pada setiap percobaan. Persentase keberhasilan dihitung menggunakan persamaan (3).

$$Persentase\ Keberhasilan = \left(\frac{Jumlah\ Berhasil}{Total\ Sampel}\right) \times 100\% \qquad (3)$$

3. Analisis Ukuran File

Data ukuran file yang diperoleh dari proses pengujian kemudian dianalisis secara deskriptif untuk menentukan apakah metode *watermarking* menyebabkan peningkatan ukuran file yang signifikan atau masih dalam batas wajar (Akmal et al., 2023).

BAB IV
HASIL DAN PEMBAHASAN

A. Hasil Implementasi Sistem

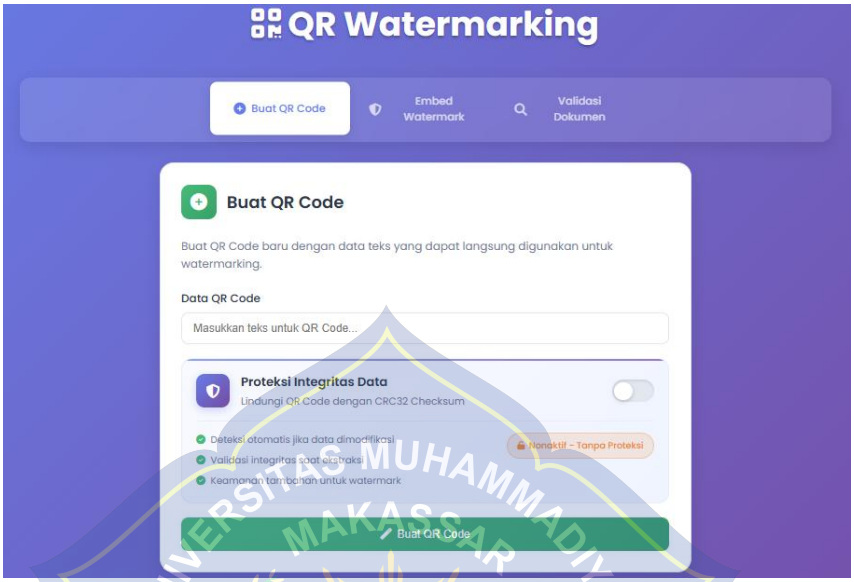
Sistem *watermark*ing *QR Code* telah berhasil diimplementasikan dalam bentuk aplikasi berbasis web yang terdiri atas tiga menu utama, yaitu pembuatan *QR Code*, penyisipan *watermark* pada dokumen, dan validasi *watermark* (lihat Gambar 6). Setiap menu dirancang dengan alur interaksi yang sederhana untuk memastikan proses *watermark*ing dapat berjalan efektif tanpa memerlukan langkah teknis yang kompleks.

Gambar 6. Tampilan Web

Secara keseluruhan, sistem ini diimplementasikan dengan alur kerja yang terintegrasi. Proses dimulai dari tahap unggah dokumen digital dalam format PDF maupun DOCX, yang kemudian diproses untuk menghasilkan *QR Code* berisi identitas atau informasi unik. *QR Code* tersebut disisipkan ke dalam citra pada dokumen menggunakan teknik Least Significant Bit (LSB). Tahap akhir menghasilkan dokumen baru yang telah tertanam *watermark* tersembunyi. Untuk proses validasi, sistem melakukan ekstraksi *QR Code* dari dokumen hasil penyisipan guna memverifikasi keaslian informasi yang dibawa. Seluruh

rangkaian proses ini dijalankan secara otomatis melalui antarmuka web yang dirancang untuk memastikan konsistensi dan keandalan sistem.

1. *Generate QR Code*



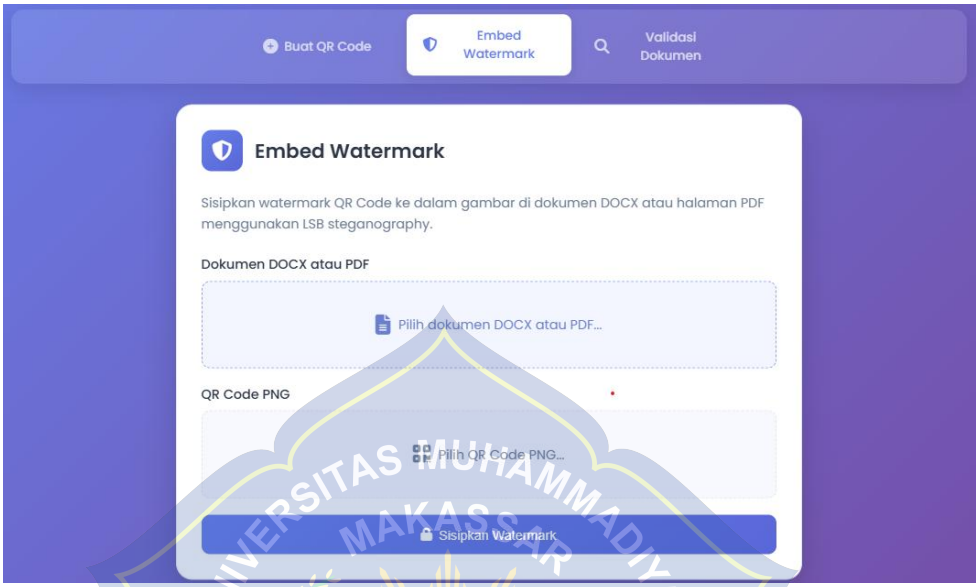
Gambar 7. Tampilan Menu *Generate QR Code*

Gambar 7 menampilkan menu *Generate QR Code*, sistem web yang dibangun mampu memproses data input dan menghasilkan gambar *QR Code* secara cepat dalam satu alur kerja yang terhubung. Saat data dimasukkan, sistem langsung menganalisis panjang dan jenis karakter untuk menentukan pengaturan yang paling sesuai, termasuk tipe *encoding*, ukuran versi *QR Code*, dan tingkat koreksi kesalahan. Semua perubahan pada input akan langsung ditampilkan dalam bentuk pratinjau *QR Code* yang terus diperbarui secara *real-time*.

QR Code yang dihasilkan dari proses ini nantinya akan menjadi komponen utama yang akan diintegrasikan ke dalam metode LSB (*Least Significant Bit*) untuk proses *steganografi*. Dalam implementasinya, sangat disarankan untuk menyertakan informasi tambahan seperti nama pembuat bahan ajar atau identitas pemilik aplikasi ke dalam data *QR Code*. Hal ini tidak hanya berfungsi sebagai *watermark* digital, tetapi juga memudahkan dalam mendeteksi peredaran tidak resmi. Dengan demikian, setiap *QR Code* yang dihasilkan akan memiliki identitas yang jelas dan dapat diverifikasi

keasliannya, sehingga meningkatkan keamanan dan kredibilitas dari sistem yang dikembangkan.

2. Penyisipan (*Embedding*)



Gambar 8. Tampilan Menu *Embedding*

Gambar 8 menunjukkan tampilan menu Embedding, yaitu tahap dimana Sistem menerima input berupa *QR Code* dari tahap sebelumnya dan mengonversinya menjadi citra *QR Code* untuk disisipkan pada bit paling tidak signifikan (LSB) dari piksel citra penampung (*cover image*) yang terdapat pada bahan ajar digital. Dengan metode ini, tampilan visual dokumen tetap terjaga dan tidak mengalami perubahan secara kasat mata.

Proses embedding dilakukan dengan metode Least Significant Bit (LSB) pada kanal biru citra penampung. Metode ini dipilih karena perubahan yang terjadi hanya pada bit terakhir dari setiap piksel, sehingga tidak menimbulkan perbedaan visual yang dapat dideteksi oleh mata manusia. Dengan demikian, tampilan dokumen hasil penyisipan tetap sama seperti dokumen asli meskipun sudah mengandung watermark tersembunyi.

Untuk memperjelas proses yang dilakukan sistem, berikut diberikan contoh perhitungan manual pada sebagian kecil channel biru citra penampung (3×3 piksel). Data watermark berupa teks *"Arikal Khairat-Unismuh"* terlebih dahulu dikonversi menjadi QR Code berukuran 227 ×

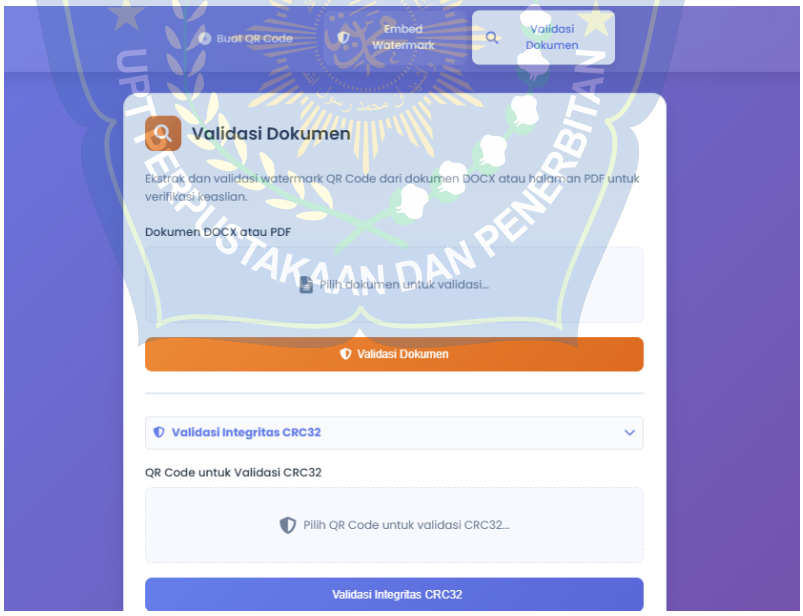
227 pixel, sehingga total terdapat 51.529 bit. Setiap piksel QR Code bernilai hitam (1) atau putih (0), sehingga dapat direpresentasikan dalam bentuk biner. Sebagai contoh, urutan bit awalnya adalah 101100101. Bit-bit inilah yang kemudian disisipkan ke LSB channel biru piksel penampung.

Tabel 3. Contoh Penyisipan Bit QR Code pada Piksel Citra

Nilai Desimal	Biner Asli	Bit QR	Biner Stego	Nilai Stego
218	11011010	1	11011011	219
205	11001101	0	11001100	204
201	11001001	1	11010011	201

Tabel 3 menunjukkan bahwa proses penyisipan hanya mengubah bit terakhir (LSB) sesuai dengan bit QR Code. Perubahan ini membuat sebagian nilai piksel berubah, sementara sebagian lainnya tetap sama, sehingga kualitas visual citra tidak terpengaruh.

3. Ekstraksi (*Extraction*)



Gambar 9. Tampilan Menu Ekstraksi

a. Tab Validasi Dokumen

Pada penelitian ini, tab validasi dokumen dikembangkan sebagai komponen utama yang memungkinkan ekstraksi dan validasi *watermark* dari dokumen digital. Ketika peneliti menguji fungsionalitas

ini, proses dimulai dengan pengunggahan dokumen berformat DOCX atau PDF yang telah mengandung *watermark* tersembunyi. Sistem kemudian menganalisis dokumen tersebut untuk mengidentifikasi dan mengekstrak seluruh gambar yang terdapat di dalamnya, sebagaimana alur yang ditampilkan pada Gambar 9.

Hasil pengujian menunjukkan bahwa setiap gambar yang berhasil diekstrak dapat diproses lebih lanjut menggunakan teknik LSB *steganografi* untuk mengambil bit-bit data yang tersembunyi. Proses rekonstruksi *QR Code* dari bit-bit yang diekstrak memerlukan algoritma yang teliti untuk memastikan data dapat dikembalikan ke bentuk aslinya. Temuan dari penelitian ini mengindikasikan bahwa *QR Code* yang berhasil direkonstruksi dapat langsung dipindai untuk memverifikasi konten *watermark*, sehingga memberikan bukti keaslian dokumen yang dianalisis.

b. komponen unggah dokumen

Bagian ini adalah kotak unggah berkas bertuliskan *"Pilih dokumen untuk validasi…"* yang menerima format DOCX atau PDF. Setelah file dipilih, tombol Validasi Dokumen mengawali proses ekstraksi. Jika format tidak didukung atau file rusak, sistem menampilkan pesan gagal unggah/validasi.

c. pilih *QR Code* untuk validasi *CRC32*

Fitur validasi integritas yang memanfaatkan algoritma *CRC32* sebagai metode tambahan untuk memverifikasi keutuhan data payload. Melalui serangkaian pengujian, ditemukan bahwa pendekatan terpisah untuk validasi integritas memberikan fleksibilitas yang lebih besar dalam proses verifikasi. Sistem yang dikembangkan mampu menerima input berupa gambar *QR Code* dan melakukan kalkulasi hash *CRC32* dari *payload* yang terkandung di dalamnya.

Hasil penelitian menunjukkan bahwa perbandingan nilai *CRC32* dengan referensi yang telah ditetapkan dapat memberikan indikasi yang akurat mengenai integritas data. Pengujian dengan berbagai skenario

modifikasi data menunjukkan bahwa sistem dapat secara konsisten mengidentifikasi perubahan yang terjadi pada *payload*. Output yang dihasilkan berupa status validitas yang komprehensif memungkinkan peneliti untuk memahami tingkat integritas data dengan baik.

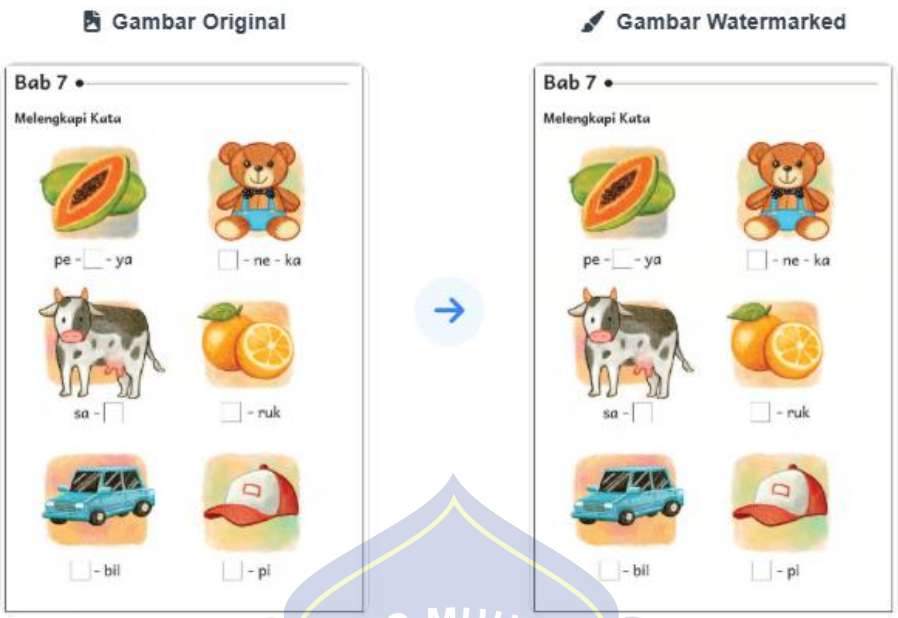
B. Hasil Pengujian

1. Uji *Imperceptibility*

a. Secara Visual

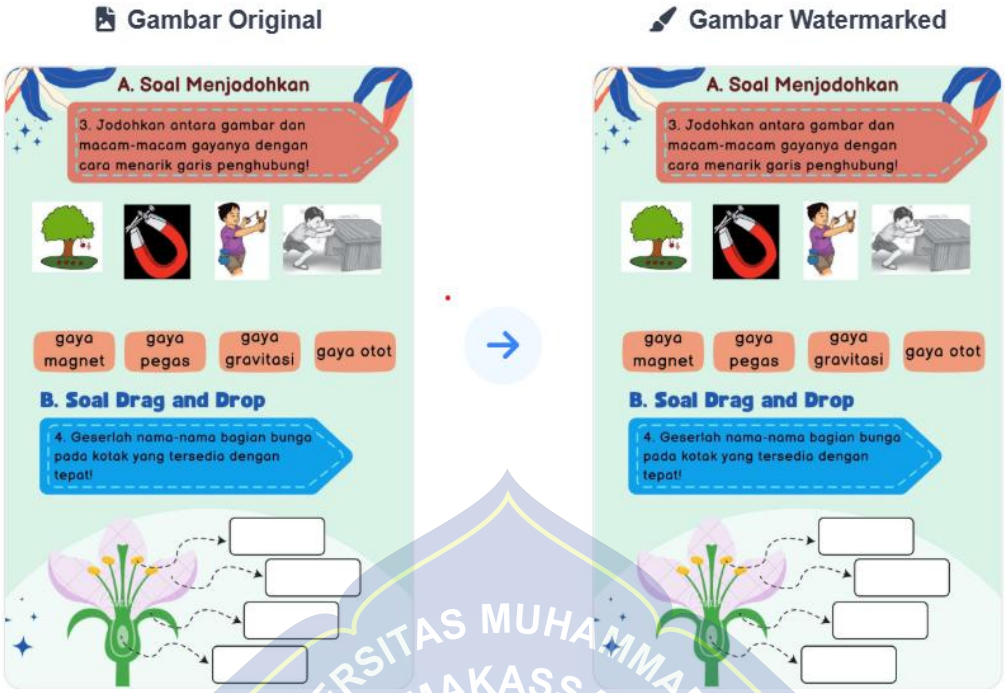
Pengujian visual subjektif dilakukan untuk mengetahui apakah *watermark* terlihat oleh mata manusia setelah disisipkan. Pengujian ini melibatkan perbandingan visual antara gambar asli dan gambar yang telah disisipkan *watermark*. Sebagaimana ditampilkan pada Gambar 10-13.

Gambar 10. Perbandingan visual cover image sebelum dan sesudah penyisipan (Modul Ajar PPKn)



Gambar 11. Perbandingan visual cover image sebelum dan sesudah Penyisipan (Modul Ajar Bahasa Indonesia)

Gambar 12. Perbandingan visual cover image sebelum dan sesudah Penyisipan (Modul Ajar Bahasa Indonesia)



Gambar 13. Perbandingan visual cover image sebelum dan sesudah Penyisipan (Modul Ajar IPAS)

Hasil pengujian pada Gambar 10-13 menunjukkan bahwa proses penyisipan menggunakan metode LSB berjalan dengan baik karena tidak menimbulkan perbedaan visual yang signifikan antara gambar sebelum dan sesudah penyisipan. Kesamaan tampilan ini membuktikan bahwa teknik LSB berhasil mempertahankan kualitas visual gambar sekaligus menyembunyikan *watermark* secara efektif tanpa terdeteksi oleh pengamatan kasat mata.

b. Perhitungan MSE dan PSNR

Pengujian dilakukan pada 10 dokumen uji (5 DOCX, 5 PDF). Masing-masing dokumen diekstrak gambarnya, disisipi *QR Code*, kemudian dihitung nilai *Mean Squared Error (MSE)* dan *Peak Signal-to-Noise Ratio (PSNR)*.

Rumus MSE:

$$MSE = \frac{1}{mn} \sum_{i=1}^{m} \sum_{i=1}^{n} [I(i,j) - K(i,j)]^2 \qquad (4)$$

Keterangan:

I(i, j) : piksel citra asli

K(i, j) : piksel citra setelah disisipkan *watermark*

m, n : ukuran gambar

Rumus PSNR:

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX^2}{MSE}\right) \qquad (5)$$

Keterangan:

MSE : *Mean Squared Error*

MAX : nilai maksimum piksel (255 untuk gambar 8-bit)

Gambar 14. Citra Uji yang Digunakan untuk Perhitungan MSE dan PSNR

Contoh perhitungan MSE dan PSNR pada matriks pixel 3x3 dari citra yang ditunjukkan pada Gambar 14 menggunakan persamaan (4) dan (5).

194	187	191	195	186	191
198	184	200	199	184	200
196	196	205	196	197	204

$(194 - 195)^2 + (187 - 186)^2 + (191 - 191)^2$

$(198 - 199)^2 + (184 - 184)^2 + (200 - 200)^2$

$(196 - 196)^2 + (196 - 197)^2 + (205 - 204)^2$

= (1 + 1 + 0 + 1 + 0 + 0 + 0 + 1 + 1)

= 5

MSE = $\frac{1}{9}\ x\ 5$ = 0.5555

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX^2}{MSE}\right)$$

$$= 10 \cdot \log_{10}\left(\frac{255^2}{0.5555}\right)$$

$$= 10 \cdot \log 10\left(\frac{65025}{0.5555}\right) = 117056$$

$$= 10 \cdot \log 10\,(117056)$$

$$PSNR = 50.68 \text{ dB}$$

Tabel 4. Rata-rata Nilai MSE dan PSNR Hasil Pengujian

No	Nama Dokumen	Jumlah Gambar	MSE	PSNR	MSE (CRC32)	PSNR (CRC32)
1	Doc1.docx	23	0.18	55.6 dB	0.18	55.4
2	Doc2.docx	18	0.18	55.4 dB	0.18	55.3
3	Doc3.docx	13	0.15	56.5 dB	0.19	55.4
4	Doc4.docx	17	0.16	56.0 dB	0.18	55.6
5	Doc5.docx	11	0.17	56.7 dB	0.17	56.3
6	Doc6.pdf	18	0.19	55.4 dB	0.19	55.3
7	Doc.pdf	8	0.12	58.0 dB	0.13	57.2
8	Doc8.pdf	22	0.15	56.8 dB	0.17	56.2
9	Doc9.pdf	25	0.15	58.1 dB	0.15	57.8
10	Doc10.pdf	14	0.13	57.1 dB	0.15	56.4

Berdasarkan hasil pada Tabel 4, seluruh dokumen uji baik dalam format DOCX maupun PDF menunjukkan nilai MSE yang rendah pada rentang 0,12 hingga 0,19. Nilai MSE yang kecil ini menandakan bahwa perbedaan rata-rata kuadrat antara gambar asli dan gambar hasil penyisipan sangat minim, sehingga kualitas visual tetap terjaga.

Nilai PSNR yang diperoleh berada pada kisaran 55,4 dB hingga 58,1 dB. Nilai tertinggi dicapai pada *Doc9.pdf* dengan PSNR sebesar 58,1 dB, sementara nilai terendah terdapat pada *Doc2.docx* dan *Doc6.pdf* dengan PSNR 55,4 dB. Berdasarkan standar kualitas citra,

PSNR di atas 30 dB sudah dianggap baik, sehingga hasil pengujian ini menunjukkan kualitas visual yang sangat baik untuk seluruh sampel.

Selain itu, pengujian dengan validasi CRC32 juga menghasilkan nilai MSE dan PSNR yang tidak berbeda signifikan dibandingkan hasil tanpa CRC32. Misalnya, pada *Doc1.docx* nilai PSNR sedikit menurun dari 55,6 dB menjadi 55,4 dB setelah validasi CRC32. Hal ini menunjukkan bahwa penerapan CRC32 sebagai mekanisme integritas tidak memengaruhi kualitas visual citra secara signifikan, melainkan hanya berfungsi sebagai verifikasi tambahan terhadap keutuhan payload.

Perbedaan nilai PSNR antar dokumen lebih banyak dipengaruhi oleh variasi jumlah dan resolusi gambar dalam masing-masing dokumen. Dokumen dengan resolusi lebih tinggi atau komposisi visual yang stabil cenderung menghasilkan nilai PSNR lebih tinggi. Dengan demikian, dapat disimpulkan bahwa penyisipan watermark dengan metode LSB, baik dengan maupun tanpa validasi CRC32, berjalan efektif pada semua sampel uji tanpa mengorbankan kualitas visual dokumen.

c. Pengujian Dokumen Tanpa Gambar

Pengujian sistem *watermark*ing *QR Code* juga melibatkan evaluasi terhadap dokumen yang tidak mengandung gambar untuk memahami perilaku sistem dalam kondisi tersebut. Hasil pengujian menunjukkan bahwa sistem telah dirancang dengan mekanisme deteksi yang dapat mengidentifikasi keberadaan gambar dalam dokumen PDF maupun DOCX sebelum memulai proses *watermark*ing. Ketika sistem menganalisis dokumen yang tidak mengandung elemen visual atau gambar, algoritma deteksi akan menghentikan proses secara otomatis dan memberikan notifikasi yang sesuai.

Implementasi fitur deteksi gambar ini menjadi komponen penting dalam menjaga stabilitas sistem dan mencegah terjadinya error selama proses operasional. Pengujian dilakukan terhadap jenis dokumen tanpa

gambar, termasuk dokumen teks murni yang hanya berisi konten tekstual. Hasil analisis menunjukkan bahwa sistem dapat secara konsisten mengidentifikasi kondisi ini dan memberikan informasi mengenai ketidaktersediaan media penampung untuk proses penyisipan *watermark*, seperti yang ditunjukkan pada Gambar 15.

Gambar 15. Tampilan Penyisipan Menggunakan Dokumen Tanpa Gambar

2. Uji Keberhasilan Ekstraksi

Proses ekstraksi dilakukan untuk mengambil *watermark* berupa *QR Code* dari dokumen yang telah melalui tahap penyisipan. Pada tahap ini, sistem membaca bit paling tidak signifikan (LSB) pada citra untuk merekonstruksi *QR Code* dan mendekodenya kembali. Validasi integritas menggunakan *CRC32* bersifat opsional, dan hanya dijalankan apabila fitur *integrity check* diaktifkan atau dibutuhkan pada skenario verifikasi. Sebagaimana ditampilkan pada Tabel 5.

Tabel 5. Hasil Pengujian Ekstraksi *QR Code*

No	Nama Dokumen	Jumlah Gambar	Validasi Tanpa CRC32	Status CRC32	Validasi CRC32
1	Doc1.docx	23	Berhasil	Cocok	Berhasil
2	Doc2.docx	18	Berhasil	Cocok	Berhasil
3	Doc3.docx	13	Berhasil	Cocok	Berhasil
4	Doc4.docx	17	Berhasil	Cocok	Berhasil
5	Doc5.docx	11	Berhasil	Cocok	Berhasil
6	Doc6.pdf	18	Berhasil	Cocok	Berhasil
7	Doc7.pdf	8	Berhasil	Cocok	Berhasil
8	Doc8.pdf	22	Berhasil	Cocok	Berhasil
9	Doc9.pdf	25	Berhasil	Cocok	Berhasil
10	Doc10.pdf	14	Berhasil	Cocok	Berhasil

Hasil pengujian ditunjukkan pada Tabel 5 yang menampilkan status validasi tanpa CRC32 sekaligus hasil uji dengan CRC32 untuk 10 dokumen uji (169 gambar). Pada pengujian tanpa CRC, seluruh QR Code berhasil diekstraksi, dapat dipindai, dan isi datanya sesuai dengan payload yang disisipkan. Dengan demikian, status validasi tanpa CRC seluruhnya adalah Berhasil.

Pada pengujian yang sama, sistem juga menghitung nilai CRC32 dari payload asli saat penyisipan, lalu membandingkannya dengan nilai CRC32 hasil ekstraksi. Hasilnya menunjukkan bahwa semua dokumen memiliki status Cocok dan validasi dengan CRC32 juga dinyatakan Berhasil. Hal ini menandakan bahwa tidak ada perubahan data selama proses penyisipan maupun ekstraksi, sehingga integritas payload QR Code tetap terjaga.

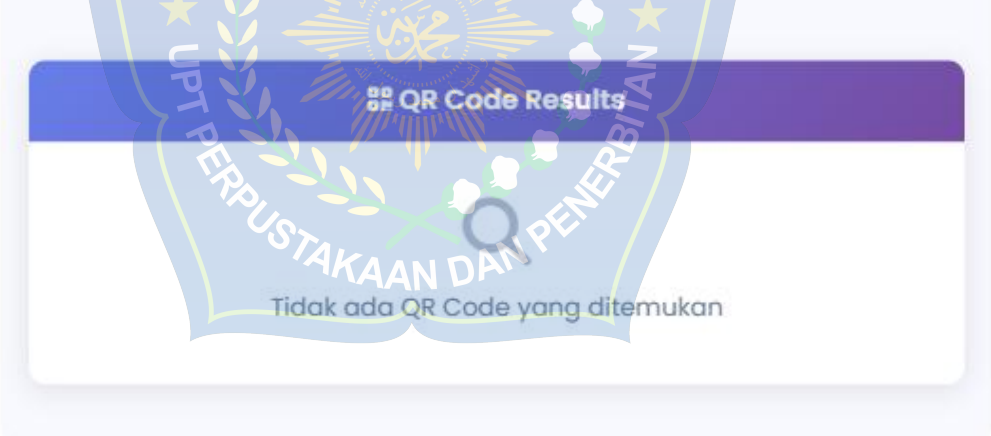
$$\textit{Persentase Keberhasilan} = \left(\frac{Jumlah\ Berhasil}{Total\ Sampel}\right) \times 100\% \qquad (6)$$

$$= \left(\frac{169}{169}\right) \text{ x } 100\ \%$$

$$= 100\%$$

Seluruh sampel (169 dari 169 gambar) memenuhi kriteria "Berhasil", sehingga tingkat keberhasilan ekstraksi = 100% dengan menggunakan rumus (6). Dengan demikian, seluruh sampel dinyatakan "Valid", baik pada pengujian standar maupun saat menggunakan CRC32. Sebuah dokumen dikategorikan "Valid" apabila *QR Code* hasil ekstraksi dapat dipindai dan isi payload sama dengan data yang disisipkan, serta nilai CRC32 hasil ekstraksi identik dengan nilai saat penyisipan. Sebaliknya, status Tidak Valid akan muncul apabila *QR Code* tidak dapat dipindai, data payload berubah, atau nilai CRC32 tidak sesuai.

3. Ektraksi Tanpa *QR Code*



Gambar 16. Tampilan Ekstraksi Tanpa *QR Code*

Tampilan "*QR Code Results*: Tidak ada *QR Code* yang ditemukan" sebagaimana ditunjukkan pada Gambar 16, berarti sistem sudah memeriksa semua gambar yang ada di dalam dokumen untuk mencari *watermark* yang tersembunyi. Sistem mencoba mengambil data dari bagian terkecil setiap titik gambar (disebut bit LSB) dengan cara yang sama seperti saat menyembunyikan *watermark*. Tapi karena tidak ketemu pola *watermark* yang benar, sistem tidak bisa membangun ulang *QR Code* yang

disembunyikan. Ini berarti dokumen tersebut memang tidak pernah diproses dengan sistem *watermarking* ini.

Hasil seperti ini sangat berguna untuk memastikan keaslian dokumen karena membuktikan bahwa sistem bisa membedakan mana dokumen yang sudah punya *watermark* dan mana yang belum. Ketika sistem bilang "tidak ditemukan" pada dokumen biasa, ini menunjukkan sistem tidak akan salah mendeteksi *watermark* yang sebenarnya tidak ada. Jadi setiap kali sistem berhasil menemukan *watermark*, kita bisa yakin bahwa *watermark* itu benar-benar dari sistem ini. Dengan begitu, fitur ini memperkuat kemampuan sistem untuk memverifikasi apakah dokumen bahan ajar benar-benar asli dari pembuat yang sah.

4. Perbandingan Ukuran File

Perbandingan ukuran file sebelum dan sesudah penyisipan menunjukkan bahwa beberapa dokumen mengalami penurunan ukuran, sementara lainnya mengalami peningkatan. Rincian hasil pengujian baik pada kondisi watermark standar maupun dengan validasi CRC32, ditampilkan pada Tabel 6.

Tabel 6. Perbandingan Ukuran File Sebelum dan Sesudah Embedding

No	Nama Dokumen	Asli	Watermark	Watermark (CRC32)
1	Doc1.docx	1.990 KB	1.591 KB	1.608
2	Doc2.docx	1.502 KB	1.206 KB	1.217
3	Doc3.docx	2.196 KB	1.661 KB	1.029
4	Doc4.docx	2.018 KB	1.661 KB	1.352
5	Doc5.docx	567 KB	1.287 KB	1.297
6	Doc6.pdf	515 KB	2.436 KB	2.436
7	Doc7.pdf	817 KB	1.970 KB	1.970
8	Doc8.pdf	789 KB	10.723 KB	10.723
9	Doc9.pdf	941 KB	14.191 KB	14.191
10	Doc10.pdf	513 KB	860 KB	860

Hasil pengujian pada Tabel 6 menunjukkan adanya perbedaan ukuran file sebelum dan sesudah penyisipan watermark, baik dengan maupun tanpa validasi CRC32. Pada dokumen berformat DOCX (Doc1–5), sebagian besar ukuran file justru mengalami sedikit penurunan atau perubahan tidak signifikan setelah proses embedding. Hal ini kemungkinan disebabkan oleh mekanisme penyimpanan ulang yang mengompresi ulang bagian lain dari dokumen, sehingga meskipun ada penambahan data LSB, ukuran total file tidak selalu bertambah.

Sebaliknya, pada dokumen berformat PDF (Doc6–10), khususnya Doc7–9, ukuran file meningkat cukup signifikan setelah proses embedding. Peningkatan ini dipengaruhi oleh karakteristik format PDF yang kurang fleksibel dalam melakukan kompresi ulang gambar setelah dimodifikasi, sehingga penambahan bit LSB untuk QR Code menyebabkan pertambahan ukuran file yang lebih besar.

Sementara itu, hasil pada kolom Watermark (CRC32) menunjukkan bahwa penambahan proses validasi integritas tidak memberikan pengaruh besar terhadap ukuran file dibandingkan hasil watermark tanpa CRC32. Perubahan ukuran yang tercatat tetap berada pada rentang yang relatif kecil dan konsisten dengan pola sebelumnya.

Dengan demikian, meskipun terdapat variasi ukuran file pada beberapa dokumen, baik pada hasil watermark standar maupun dengan CRC32, kualitas visual gambar tetap terjaga. Hal ini menegaskan bahwa metode LSB dengan tambahan validasi CRC32 tetap efektif dalam menyisipkan watermark tanpa mengorbankan keterbacaan maupun kualitas visual bahan ajar digital.

C. Pembahasan Hasil

Sistem *watermarking QR Code* yang dikembangkan telah berhasil diimplementasikan dalam bentuk aplikasi berbasis web dengan tiga menu utama, yaitu pembuatan *QR Code*, penyisipan *watermark*, dan validasi *watermark*. Pada tahap *Generate*, sistem memproses data input secara *real-time* dan menyesuaikan parameter seperti tipe encoding, versi *QR Code*, serta tingkat

error correction sesuai panjang data, lengkap dengan pratinjau yang diperbarui otomatis. Proses penyisipan menggunakan metode LSB pada kanal biru gambar penampung yang diambil dari dokumen PDF maupun DOCX, terbukti berjalan stabil dan tidak menimbulkan perbedaan visual signifikan. Pada tahap ekstraksi, sistem tidak hanya membaca kembali data dari bit LSB tetapi juga menerapkan mekanisme *Cyclic Redundancy Check* (CRC) yang memastikan *QR Code* hasil ekstraksi identik dengan *QR Code* yang diunggah di halaman validasi. Dari pengujian pada 10 dokumen uji, seluruh hasil ekstraksi divalidasi secara optimal tanpa kasus *QR Code* yang tidak terbaca.

Pengujian *imperceptibility* menunjukkan nilai PSNR seluruh sampel berada pada rentang 55.4–58,1 dB, jauh di atas batas 30 dB yang menandakan kualitas visual sangat baik, dengan nilai MSE yang rendah (0.12–0.19) sebagai bukti minimnya perbedaan antara gambar asli dan hasil *embedding*. Perubahan ukuran file sebelum dan sesudah penyisipan juga menjadi indikator bahwa proses LSB berjalan, meskipun perubahan tersebut sangat kecil dan tidak memengaruhi keterbacaan *watermark*.

BAB V
PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, penelitian ini berhasil menjawab seluruh rumusan masalah dan mencapai tujuan yang telah ditetapkan dengan hasil sebagai berikut:

1. Implementasi

   Sistem *watermark*ing tak terlihat berbasis web dengan tiga menu (*Generate QR Code*, *Embedding*, Validasi) berjalan *real-time* dan terintegrasi *end-to-end*. *QR Code* berisi identitas/kode dokumen lalu disisipkan ke kanal biru citra pada PDF/DOCX dengan LSB sehingga tampilan tetap tidak berubah secara kasat mata. Antarmuka sederhana memandu proses hingga menghasilkan dokumen baru ber-*watermark*. *CRC32* bersifat opsional pada tahap ekstraksi: jika diaktifkan, sistem membandingkan *checksum* referensi dan hasil ekstraksi untuk mendeteksi perubahan; jika tidak, sistem hanya merekonstruksi & membaca QR. Skema ini mendukung autentikasi sekaligus pelacakan *(traceability)* hasil bahan ajar.

2. Kinerja

   Tingkat kinerja metode kombinasi *QR Code* dan LSB dalam konteks *watermarking* tak terlihat, khususnya dalam aspek *imperceptibility*, menunjukkan hasil yang sangat baik. Pengujian *imperceptibility* menghasilkan nilai PSNR 55,4 – 58,1 dB (>30 dB) dan MSE 0,12 – 0,19, menandakan distorsi sangat minim. Uji pada 10 dokumen menunjukkan 100% keberhasilan ekstraksi dan QR dapat dipindai tanpa kegagalan. Perubahan ukuran file minimal, sehingga efisien untuk penyimpanan dan distribusi. Aktivasi *CRC32* (opsional) menambah jaminan integritas payload, menjadikan kombinasi *QR Code* + LSB efektif dan andal untuk keamanan, verifikasi, dan pelacakan dokumen digital di praktik nyata.

B. Saran

1. Pengembangan pada Format Non-Gambar

   Disarankan agar penelitian selanjutnya mengembangkan metode penyisipan *watermark* yang dapat diterapkan pada bahan ajar non-gambar, seperti teks atau PDF, untuk memperluas cakupan perlindungan konten digital.

2. Akses secara offline

   Disarankan agar sistem yang dikembangkan di masa depan memiliki kemampuan verifikasi dan ekstraksi *watermark* secara offline, sehingga tetap dapat digunakan tanpa bergantung pada koneksi internet.

DAFTAR PUSTAKA

Aditya Permana, A., & Amma, H. (2022). IMPLEMENTASI *STEGANOGRAFI* FILE CITRA DIGITAL MENGGUNAKAN METODE LEAST SIGNIFICANT BIT. *JT Jurnal Teknik*, *11*(1), 62–72.

Afsari, M., Mulyana, D. I., Damaiyanti, A., & Sa'adah, N. (2022). Implementasi Mode Operasi Kombinasi Cipher Block Chaining dan Metode LSB-1 Pada Pengamanan Data text. *Jurnal Pendidikan Sains Dan Komputer*, *2*(1), 70–82. https://doi.org/10.47709/jpsk.v2i1.1381

Akmal, R. A., Furqan, Mhd. F., & Kurniawan R, R. (2023). Implementasi Metode Least Significant Bit Dalam Teknik *Steganografi* pada Berkas Audio Dengan Stego Citra Digital. *G-Tech: Jurnal Teknologi Terapan*, *7*(2), 543–553. https://doi.org/10.33379/gtech.v7i2.2300

Alajmi, M., Elashry, I., El-Sayed, H. S., & FaragAllah, O. S. (2020). Steganography of Encrypted Messages Inside Valid *QR Code*s. *IEEE Access*, *8*, 27861–27873. https://doi.org/10.1109/ACCESS.2020.2971984

Alveda, A., Rakhmawati, L., & Agustin Tjahyaningtijas Agustin, R. H. P. (2024). *Penyisipan Watermark Menggunakan Metode LSB (Least Significant Bit) untuk Autentikasi Citra Medis*.

Antika, T. L., Kusmana, S., & Gloriani, Y. (2022). BAHAN AJAR DIGITAL TEKS CERPEN UNTUK SMP. In *Jurnal Penelitian Pendidikan Bahasa dan Sastra* (Vol. 7, Issue 2).

Devi, P. A. R., & Rosyid, H. (2022). Pemaparan Materi Dasar Pengolahan Citra Digital untuk Upgrade Wawasan Siswa di SMK Dharma Wanita Gresik. *Jurnal Abdi Masyarakat Indonesia*, *2*(4), 1259–1264. https://doi.org/10.54082/jamsi.405

Fadel, A. S., Saputra, R. D., Putra, R. N., & Fatma, Y. (2024). Analisis keamanan *steganografi* teks dengan metode lsb (least significant bit) pada citra digital. *Jurnal CoSciTech (Computer Science and Information Technology)*, *5*(1), 36–41. https://doi.org/10.37859/coscitech.v5i1.6759

Fadlika Satria, A., Ibnu Adam, R., & carudin. (2021). Analisis Digital *Watermark*ing untuk Otentikasi pada Citra Manipulasi Menggunakan Metode Least Significant Bit. *Edumatic Jurnal Pendidikan Informatika*, *5*(2), 204–213. https://doi.org/10.29408/edumatic.v5i2.3901

Faisal, M., Hotimah, Nurhaedah, AP, N., & Khaerunnisa. (2020). Peningkatan Kompetensi Guru Sekolah Dasar dalam Mengembangkan Bahan Ajar Digital di Kabupaten Gowa. *Jurnal Publikasi Pendidikan*, *10*(3), 266–270. http://ojs.unm.ac.id/index.php/

Fathanudien, A., & Maharani, V. (2023). Perlindungan Hukum Hak Cipta terhadap Buku Elektronik (E-Book) di Era Globalisasi. In *Jurnal Penelitian Universitas Kuningan* (Vol. 14).

Ferdiansyah, Id Hadiana, A., & Rakhmat Umbara, F. (2021). PENGGUNAAN *QR CODE* BERBASIS KRIPTOGRAFI ALGORITMA AES ADVANCED ENCRYPTION STANDARD UNTUK ADMINISTRASI REKAM MEDIS. *JOINT (Journal of Information Technology)*, *03*(2), 20–27.

Gultom, C. E., & Suhartana, K. G. (2023). Penerapan *Steganografi* dan Visible *Watermark*ing Pada Gambar Digital Untuk Perlindungan Hak Cipta. *Jurnal Elektronik Ilmu Komputer Udayana*, *12*(2), 377–384.

Harits M, A. R., Ridwan, R., Hafidzin, A. P., & Taufik, M. (2021). Proteksi Keamanan Data pada Quick Response (QR) Code. *Jurnal Teknologi Dan Rekayasa Manufaktur*, *3*(2), 99–110. https://doi.org/10.48182/jtrm.v3i2.58

Hasan, N. F., Dengen, C. N., & Ariyus, D. (2020). Analisis Histogram *Steganografi* Least Significant Bit Pada Citra Grayscale. *Jurnal Teknologi Informasi & Komunikasi Digital Zone*, *11*, 2086–4884. https://doi.org/10.31849/digitalzone.v11i1.3413ICCS

Irawan, D., & Pujianto. (2020). Menyembunyikan File Kedalam File Gambar Menggunakan Metode *Steganografi*. *JSAI*, *3*(1), 1–6. http://www.jurnal.umb.ac.id/index.php/JSAI

Nur Aqsal Aminullah, M., Yusliana Bakti, R., Hayat, M. A., & Lukman. (2022). *PEMBUATAN VERIFIKASI SERTIFIKAT DIGITAL SEBAGAI BUKTI*

KEABSAHAN MENGGUNAKAN ALGORITMA STEGANOGRAFI DENGAN METODE LEAST SIGNIFICANT BIT INSERTION (LSB) (Vol. 4, Issue 1).

Purbaningrum, A., Silvi Amalia, K., & Ady Saputro, I. (2023). *Penerapan Metode Least Significant Bit (LSB) dalam Menyisipkan Pesan Rahasia pada Citra Digital: Sebuah Pendekatan Steganografi*.

Putri Pebriani, D., Marwati, R., & Rachmatin, D. (2025). Implementasi Kombinasi Secret Sharing dan *Steganografi* Citra Least Significant Bit dengan *QR Code*. *Original Article Indonesian Journal of Applied Mathematics*, *5*(1), 33–41. https://doi.org/10.35472/indoja

Riansah, W. (2021). Aplikasi *QR Code* Generator Dan *QR Code* Reader Menggunakan Metode Stroke Histogram. *J-SISKO TECH Jurnal Teknologi Sistem Informasi Dan Sistem Komputer TGD*, *4*(1), 38–49.

Sagala, S. H. (2021). Penerapan Metode *CRC32* Untuk Mendeteksi Otentikasi Citra Tanda Tangan. *Pelita Informatika : Informasi Dan Informatika*, *9*(4), 276–280.

WIDIYONO, WIBOWO, A. P., & DARMAWAN, A. S. (2021). TEKNIK *WATERMARK*ING MENGGUNAKAN METODE LEAST SIGNIFICANT BIT PADA CITRA UNTUK PERLINDUNGAN HAK CIPTA MOTIF BATIK. *Jurnal Instek Informatika Sains Dan Teknologi*, *6*(1), 37–45.

Wulandari, F. (2024). Problematika Pelanggaran Hak Cipta di Era Digital. *Journal of Contemporary Law Studies*, *2*(2), 99–114. https://doi.org/10.47134/lawstudies.v2i2.2261

Yanti, F., & Budayawan, K. (2023). Implementasi *Steganografi* Menggunakan Metode Least Significant Bit (Lsb) dalam Pengamanan Informasi Pada Citra Digital. *Jurnal Vocational Teknik Elektronika Dan Informatika*, *11*(1), 63–70. http://ejournal.unp.ac.id/index.php/voteknika/index

Lampiran 1. Source Code

```
# File: lsb_steganography.py
# Deskripsi: Fungsi inti untuk menyisipkan dan mengekstrak QR Code
menggunakan LSB.

from PIL import Image
from PIL.Image import Resampling
import itertools
import os
import math

HEADER_TERMINATOR_BIN = '00000000'
HEADER_TERMINATOR_LEN = len(HEADER_TERMINATOR_BIN)


def _int_to_binary(integer: int, bits: int) -> str:
    """
    Konversi integer ke string biner dengan panjang tetap.
    Memastikan output selalu memiliki jumlah bit yang ditentukan dengan
padding '0' di depan.
    Contoh: _int_to_binary(10, 8) -> '00001010'
    """
    return format(integer, f'0{bits}b')


def _binary_to_int(binary_string: str) -> int:
    """Konversi string biner ke integer."""
    return int(binary_string, 2)


def _embed_bit(pixel_value: int, bit: str) -> int:
    """
    Menyisipkan satu bit ('0' atau '1') ke LSB (Least Significant Bit)
    dari sebuah nilai integer (byte piksel).
    Jika bit = '0', LSB di-set ke 0.
    Jika bit = '1', LSB di-set ke 1.
    """
    if bit == '0':
        return pixel_value & 254
    else:
        return pixel_value | 1


def _extract_lsb(pixel_value: int) -> str:
```

```
    """
    Mengekstrak LSB dari sebuah nilai integer (byte piksel).
     Mengembalikan '1' jika nilai ganjil (LSB=1), '0' jika genap
(LSB=0).
    """
    return '1' if pixel_value % 2 == 1 else '0'


def _resize_qr_for_capacity(qr_img, max_capacity: int):
    """
     Menyesuaikan ukuran QR code agar muat dalam kapasitas citra
penampung.

    Args:
        qr_img: Objek Image dari QR code yang perlu disesuaikan.
        max_capacity: Kapasitas maksimum yang tersedia dalam bit.

    Returns:
        Objek Image dari QR code yang telah diresize.
    """
    # Kurangi kapasitas untuk header (16+16+8 bit)
       available_bits_for_qr = max_capacity - (16 + 16 +
HEADER_TERMINATOR_LEN)

    if available_bits_for_qr <= 0:
        raise ValueError("Kapasitas cover image terlalu kecil bahkan
untuk header saja.")

    # Hitung dimensi maksimum berdasarkan akar kuadrat dari kapasitas
      # Kita perlu bilangan bulat yang ketika dikuadratkan <=
available_bits_for_qr
    new_dimension = int(math.sqrt(available_bits_for_qr))

    # Jika QR lebih kecil dari dimensi maksimal, tidak perlu diresize
       if qr_img.width <= new_dimension and qr_img.height <=
new_dimension:
        return qr_img

   # Resize QR secara proporsional tapi tidak lebih besar dari dimensi
maksimum
    new_size = min(new_dimension, new_dimension)
    # Resize QR code dengan tetap mempertahankan mode
          resized_qr = qr_img.resize((new_size, new_size),
Resampling.NEAREST)
```

```
    print(f"[*] QR code diresize dari {qr_img.width}x{qr_img.height}
ke {new_size}x{new_size} agar muat dalam kapasitas.")
    return resized_qr


def embed_qr_to_image(cover_image_path: str, qr_image_path: str,
output_stego_path: str, resize_qr_if_needed: bool = True,
preserve_format: bool = True, quality: int = 95):
    """
    Menyisipkan citra QR Code ke dalam LSB channel Biru dari citra
penampung.

    Args:
        cover_image_path (str): Path ke citra penampung.
           qr_image_path (str): Path ke citra QR Code yang akan
disembunyikan (harus hitam putih).
        output_stego_path (str): Path untuk menyimpan citra hasil.
        resize_qr_if_needed (bool): Jika True, QR code akan diresize
otomatis agar muat dalam kapasitas.
       preserve_format (bool): Jika True, akan mencoba mempertahankan
format asli jika memungkinkan.
        quality (int): Kualitas kompresi untuk format lossy (1-100).

    Raises:
        FileNotFoundError: Jika file input tidak ditemukan.
        ValueError: Jika kapasitas citra penampung tidak cukup atau
format output salah.
        Exception: Jika terjadi error lain selama proses.
    """

    print("[*] Memulai proses embed_qr_to_image")  # Log awal fungsi

    # Validasi keberadaan file input
    if not os.path.exists(cover_image_path):
            raise FileNotFoundError(f"File cover tidak ditemukan:
{cover_image_path}")
    if not os.path.exists(qr_image_path):
          raise FileNotFoundError(f"File QR Code tidak ditemukan:
{qr_image_path}")

    # Deteksi format asli untuk preserve_format option
    original_format = None
    try:
        with Image.open(cover_image_path) as temp_img:
            original_format = temp_img.format
```

```
    except:
        original_format = 'PNG'  # Fallback ke PNG

    # Untuk LSB steganography, kita memerlukan format lossless
     # PNG tetap jadi pilihan utama, tapi kita bisa preserve format
jika memungkinkan
    requires_lossless = True  # LSB memerlukan lossless format

    print(f"[*] Format asli: {original_format}")
    print(f"[*] Preserve format: {preserve_format}")
    print(f"[*] Requires lossless: {requires_lossless}")

    try:
        # 1. Buka kedua citra
          cover_img = Image.open(cover_image_path).convert('RGB')  #
Pastikan format RGB
         qr_img = Image.open(qr_image_path).convert('1')  # Konversi
QR ke mode 1-bit (hitam/putih)

        # Cek apakah file cover dan QR sama
                    if    os.path.abspath(cover_image_path)    ==
os.path.abspath(qr_image_path):
              print("[!] Warning: File cover dan QR sama. Ini dapat
menyebabkan masalah kapasitas.")

        cover_width, cover_height = cover_img.size
        qr_width, qr_height = qr_img.size

        # Hitung kapasitas citra penampung
        max_capacity = cover_width * cover_height

        # 2. Buat aliran bit dari QR Code
        # Cek dulu jika perlu resize QR
        original_qr_size = (qr_width, qr_height)

        # Perkiraan kebutuhan bit untuk header dan data QR
        header_bits_len = 16 + 16 + HEADER_TERMINATOR_LEN
        qr_bits_len = qr_width * qr_height
        total_bits_needed = header_bits_len + qr_bits_len

        # Jika kapasitas tidak cukup, resize QR jika opsi diaktifkan
        if total_bits_needed > max_capacity:
            if resize_qr_if_needed:
               qr_img = _resize_qr_for_capacity(qr_img, max_capacity)
                qr_width, qr_height = qr_img.size
```

```
                print("[*] QR code diresize untuk menyesuaikan dengan
kapasitas.")
            else:
                    raise ValueError(f"Kapasitas citra tidak cukup.
Dibutuhkan:  {total_bits_needed}  bits,  Tersedia:  {max_capacity}
bits.")

        # Iterasi piksel QR, '1' untuk hitam (nilai 0 di mode '1'),
'0' untuk putih (nilai 255)
        qr_bits = "".join(['1' if qr_img.getpixel((x, y)) == 0 else
'0'
                                for y in range(qr_height) for x in
range(qr_width)])
        num_qr_bits = len(qr_bits)

        # 3. Buat header: 16 bit untuk lebar QR, 16 bit untuk tinggi
QR, + terminator
                header_bits  =  _int_to_binary(qr_width,  16)  +
_int_to_binary(qr_height, 16) + HEADER_TERMINATOR_BIN
        num_header_bits = len(header_bits)

        # Total bit yang perlu disisipkan
        total_bits_to_embed = num_header_bits + num_qr_bits

        # 4. Cek kapasitas final citra penampung setelah resize (jika
ada)
        if total_bits_to_embed > max_capacity:
             raise ValueError(f"Kapasitas citra tidak cukup bahkan
setelah  resize.  Dibutuhkan:  {total_bits_to_embed}  bits,  Tersedia:
{max_capacity} bits.")

        # Informasi proses
        print(f"[*] Ukuran QR Code: {qr_width}x{qr_height}")
        if original_qr_size != (qr_width, qr_height):
                        print(f"[*]   QR   Code   diresize   dari
{original_qr_size[0]}x{original_qr_size[1]}                        ke
{qr_width}x{qr_height}")
        print(f"[*] Jumlah bit QR Code: {num_qr_bits}")
        print(f"[*] Jumlah bit Header: {num_header_bits}")
                print(f"[*]   Total   bit   untuk   disisipkan:
{total_bits_to_embed}")
         print(f"[*] Kapasitas citra penampung (Blue channel LSB):
{max_capacity} bits")

        # 5. Siapkan data untuk disisipkan dan citra output
```

```
        data_bits_iterator = iter(header_bits + qr_bits)  # Iterator
untuk bit header + QR
          stego_img  =  cover_img.copy()  #  Salin  citra  asli  untuk
dimodifikasi
        pixels_processed = 0

        # 6. Proses penyisipan bit ke LSB channel Biru
        for y in range(cover_height):
            for x in range(cover_width):
                try:
                    # Ambil bit berikutnya dari iterator
                    bit_to_embed = next(data_bits_iterator)
                    # Dapatkan nilai RGB piksel saat ini
                    r, g, b = stego_img.getpixel((x, y))
                      # Modifikasi hanya channel Biru (b) dengan bit
yang akan disisipkan
                    new_b = _embed_bit(b, bit_to_embed)
                    # Update piksel di citra stego
                    stego_img.putpixel((x, y), (r, g, new_b))
                    pixels_processed += 1
                except StopIteration:
                    # Jika iterator habis (semua bit sudah disisipkan)
                    print(f"[*] Penyisipan selesai. {pixels_processed}
piksel dimodifikasi.")

                    # Simpan dengan format yang optimal
                      save_format  =  "PNG"  # Default  untuk  LSB
steganography
                    save_options = {}

                    # Untuk LSB steganography, kita tetap menggunakan
PNG untuk memastikan
                      # tidak ada kehilangan data, tapi kita bisa
mengoptimalkan kompresi
                    if preserve_format and original_format in ['PNG',
'BMP', 'TIFF']:
                        # Format lossless yang aman untuk LSB
                      save_format = original_format if original_format
!= 'BMP' else 'PNG'
                        if save_format == 'PNG':
                                save_options  =  {'optimize':  True,
'compress_level': 6}
                    else:
                        # Gunakan PNG dengan optimasi
```

```
                                save_options = {'optimize': True,
'compress_level': 6}

                print(f"[*] Menyimpan dengan format: {save_format}")
                     stego_img.save(output_stego_path, save_format,
**save_options)
                            print(f"[*] Stego image disimpan di:
{output_stego_path}")
                    return  # Keluar dari fungsi setelah penyimpanan
berhasil

        # Baris ini seharusnya tidak tercapai jika kapasitas cukup
       print("[!] Warning: Loop selesai tapi tidak semua bit tersisip?
Cek logika kapasitas.")

    # Menangani error spesifik dan umum
    except FileNotFoundError as e:
        print(f"[!] Error: {e}")
        raise
    except ValueError as e:
        print(f"[!] Error: {e}")
        raise
    except Exception as e:
        print(f"[!] Error saat proses embedding: {e}")
        raise


def extract_qr_from_image(stego_image_path: str, output_qr_path:
str):
    """
    Mengekstrak citra QR Code yang tersembunyi dari LSB channel Biru
stego image.

    Args:
        stego_image_path (str): Path ke stego image (harus PNG).
         output_qr_path (str): Path untuk menyimpan citra QR hasil
ekstraksi (akan dibuat PNG).

    Raises:
        FileNotFoundError: Jika file stego tidak ditemukan.
           ValueError: Jika header tidak valid, terminator tidak
ditemukan, atau data tidak cukup.
        Exception: Jika terjadi error lain selama proses.
    """
```

```
    print("[*] Memulai proses extract_qr_from_image")  # Log awal fungsi

    # Validasi file input
    if not os.path.exists(stego_image_path):
            raise FileNotFoundError(f"File stego tidak ditemukan: {stego_image_path}")
    # Menyesuaikan output path jika tidak diakhiri .png
    if not output_qr_path.lower().endswith('.png'):
        print("[!] Warning: Output path disarankan .png, akan disimpan sebagai PNG.")
        output_qr_path = os.path.splitext(output_qr_path)[0] + ".png"

    try:
        # Buka stego image dalam mode RGB
        stego_img = Image.open(stego_image_path).convert('RGB')
        width, height = stego_img.size

          extracted_bits = ""  # String untuk menampung bit yang diekstrak
        pixels_processed = 0  # Counter piksel yang diproses
        header_found = False  # Flag penanda header sudah ditemukan
        qr_width = 0
        qr_height = 0
         # Total panjang header = 16 (lebar) + 16 (tinggi) + panjang terminator
        num_header_bits = 16 + 16 + HEADER_TERMINATOR_LEN

        # 1. Ekstrak Header (Dimensi QR)
        print("[*] Mengekstrak header...")
        # Iterasi piksel stego image
        for y in range(height):
            for x in range(width):
                # Dapatkan nilai RGB
                r, g, b = stego_img.getpixel((x, y))
                # Ekstrak LSB dari channel Biru
                extracted_bits += _extract_lsb(b)
                pixels_processed += 1

               # Cek apakah sudah cukup bit untuk header + terminator
                if len(extracted_bits) >= num_header_bits:
                        # Cek apakah bagian akhir bit cocok dengan terminator
                    if extracted_bits.endswith(HEADER_TERMINATOR_BIN):
                        # Ambil bagian header (sebelum terminator)
```

```
                                    header_data  =  extracted_bits[:-
HEADER_TERMINATOR_LEN]
                        # Pastikan panjangnya 32 bit (16+16)
                        if len(header_data) == 32:
                              # Konversi bit header ke integer untuk
lebar dan tinggi
                          qr_width = _binary_to_int(header_data[:16])
                         qr_height = _binary_to_int(header_data[16:])
                            header_found = True
                            print(f"[*] Header ditemukan! Dimensi QR:
{qr_width}x{qr_height}")
                            break  # Keluar dari loop x karena header
sudah ketemu
                        else:
                            # Error jika panjang header tidak sesuai
                             raise ValueError("Panjang header tidak
sesuai setelah terminator ditemukan.")
                    # Jika bit sudah banyak tapi terminator belum
ketemu, mungkin error
                    elif len(extracted_bits) > num_header_bits +
500:  # Toleransi batas pencarian
                        raise ValueError("Terminator header tidak
ditemukan dalam batas wajar piksel.")

            if header_found:
                break # Keluar dari loop y jika header sudah ketemu

        # Jika setelah iterasi seluruh piksel header tidak ditemukan
        if not header_found:
             raise ValueError("Gagal menemukan header QR Code dalam
citra.")

         # 2. Hitung jumlah bit QR yang perlu diekstrak berdasarkan
dimensi
        num_qr_bits_expected = qr_width * qr_height
        total_bits_expected = num_header_bits + num_qr_bits_expected

                print(f"[*]  Jumlah  bit  QR  yang  diharapkan:
{num_qr_bits_expected}")
           print(f"[*] Total bit yang diharapkan (header + QR):
{total_bits_expected}")

        # 3. Lanjutkan ekstraksi untuk data QR
        qr_bits_list = []  # List untuk menampung bit QR
        # Indeks piksel tempat ekstraksi header berhenti
```

```
        start_pixel_index = pixels_processed

         # Konversi indeks piksel linear ke koordinat (x, y) untuk
melanjutkan
        start_y = start_pixel_index // width
        start_x = start_pixel_index % width

          print(f"[*] Melanjutkan ekstraksi dari piksel ({start_x},
{start_y})")

        # Buat iterator piksel yang dimulai dari piksel setelah header
        # dan berhenti setelah mengekstrak sejumlah bit yang diperlukan
(num_qr_bits_expected)
        pixel_iterator = itertools.islice(
             ((x_, y_) for y_ in range(start_y, height) for x_ in
range(width) if y_ > start_y or (y_ == start_y and x_ >= start_x)),
            num_qr_bits_expected
        )

        bits_extracted_count = 0  # Counter bit QR yang diekstrak
        # Iterasi menggunakan iterator piksel yang sudah dibuat
        for x, y in pixel_iterator:
            r, g, b = stego_img.getpixel((x, y))
            # Ekstrak LSB dari channel Biru dan tambahkan ke list
            qr_bits_list.append(_extract_lsb(b))
            bits_extracted_count += 1

            print(f"[*] Jumlah bit QR yang berhasil diekstrak:
{bits_extracted_count}")

        # Cek apakah jumlah bit yang diekstrak sesuai harapan
        if bits_extracted_count < num_qr_bits_expected:
                    raise ValueError(f"Data tidak cukup. Hanya
{bits_extracted_count} dari {num_qr_bits_expected} bit QR yang bisa
diekstrak.")

        # Gabungkan list bit menjadi satu string
        qr_bits = "".join(qr_bits_list)

        # 4. Rekonstruksi citra QR Code dari aliran bit
        print("[*] Merekonstruksi citra QR Code...")
        # Buat citra baru mode '1' (hitam/putih) dengan dimensi yang
didapat dari header
        reconstructed_qr = Image.new('1', (qr_width, qr_height))
        bit_index = 0
```

```
        # Iterasi koordinat citra QR yang akan dibuat
        for y in range(qr_height):
            for x in range(qr_width):
                # Jika bit = '1' (representasi hitam), set piksel ke
0 (hitam di mode '1')
                if qr_bits[bit_index] == '1':
                    reconstructed_qr.putpixel((x, y), 0)
                # Jika bit = '0' (representasi putih), set piksel ke
255 (putih di mode '1')
                else:
                    reconstructed_qr.putpixel((x, y), 255)
                bit_index += 1

        # Simpan citra QR hasil rekonstruksi
        reconstructed_qr.save(output_qr_path, "PNG")
          print(f"[*] Citra QR Code hasil ekstraksi disimpan di:
{output_qr_path}")

    # Menangani error spesifik dan umum
    except FileNotFoundError as e:
        print(f"[!] Error: {e}")
        raise
    except ValueError as e:
        print(f"[!] Error: {e}")
        raise
    except Exception as e:
        print(f"[!] Error saat proses extracting: {e}")
        raise
```

Lampiran 2. SK Bebas Plagiat

MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH MAKASSAR
UPT PERPUSTAKAAN DAN PENERBITAN
Alamat kantor: Jl.Sultan Alauddin N0.259 Makassar 90221 Tlp.(0411) 866972,881593, Fax.(0411) 865588

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

SURAT KETERANGAN BEBAS PLAGIAT

UPT Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar,
Menerangkan bahwa mahasiswa yang tersebut namanya di bawah ini:

Nama : Arikal Khairat
Nim : 105841108421
Program Studi : Teknik Informatika

Dengan nilai:

No	Bab	Nilai	Ambang Batas
1	Bab 1	9%	10 %
2	Bab 2	9%	25 %
3	Bab 3	10%	10 %
4	Bab 4	3%	10 %
5	Bab 5	5%	5 %

Dinyatakan telah lulus cek plagiat yang diadakan oleh UPT- Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar Menggunakan Aplikasi Turnitin.

Demikian surat keterangan ini diberikan kepada yang bersangkutan untuk dipergunakan seperlunya.

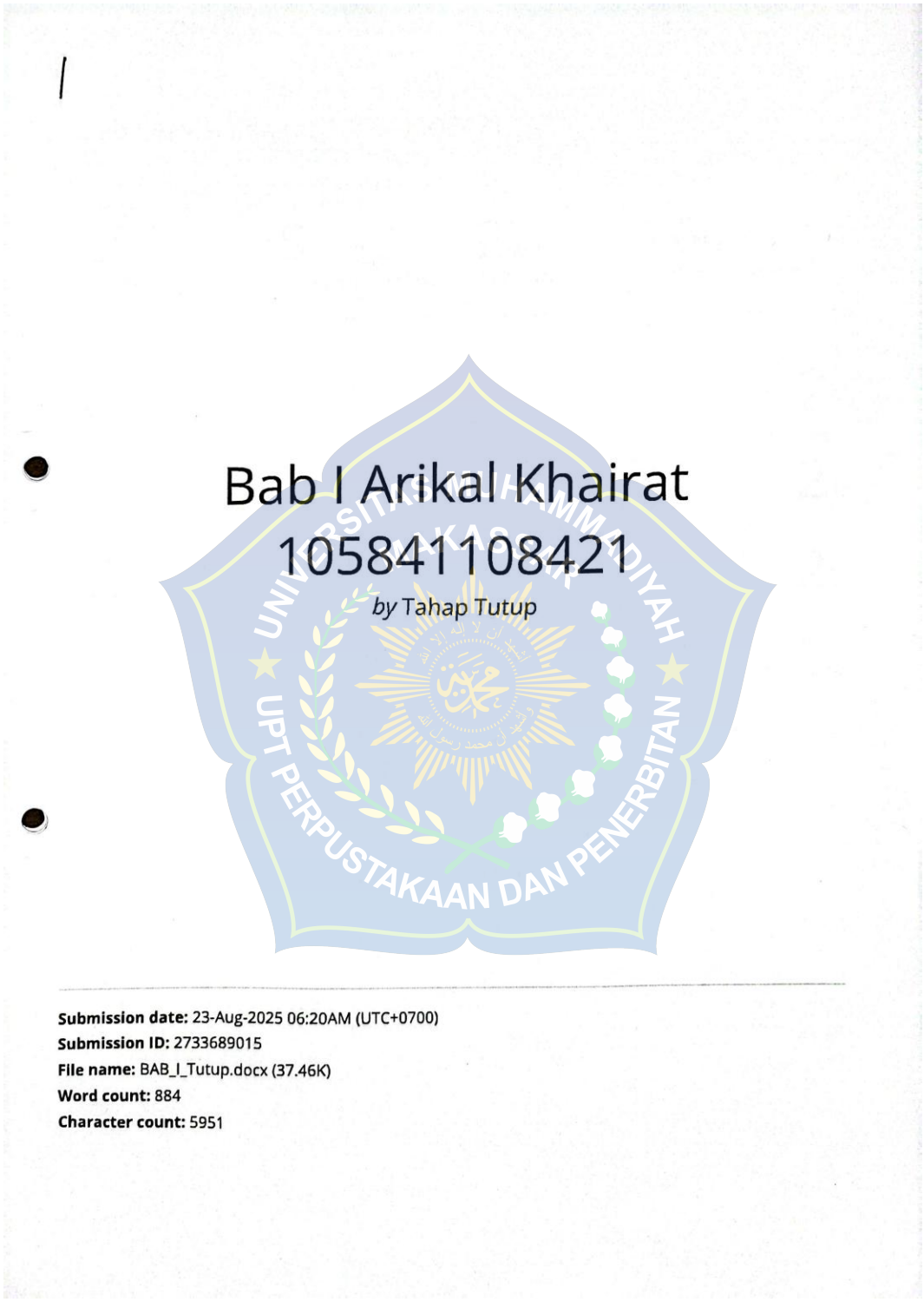
Makassar, 26 Agustus 2025
Mengetahui,

Kepala UPT- Perpustakaan dan Pernerbitan,

Nursmaliah S.Hum, M.I.P
NBM. 964 591

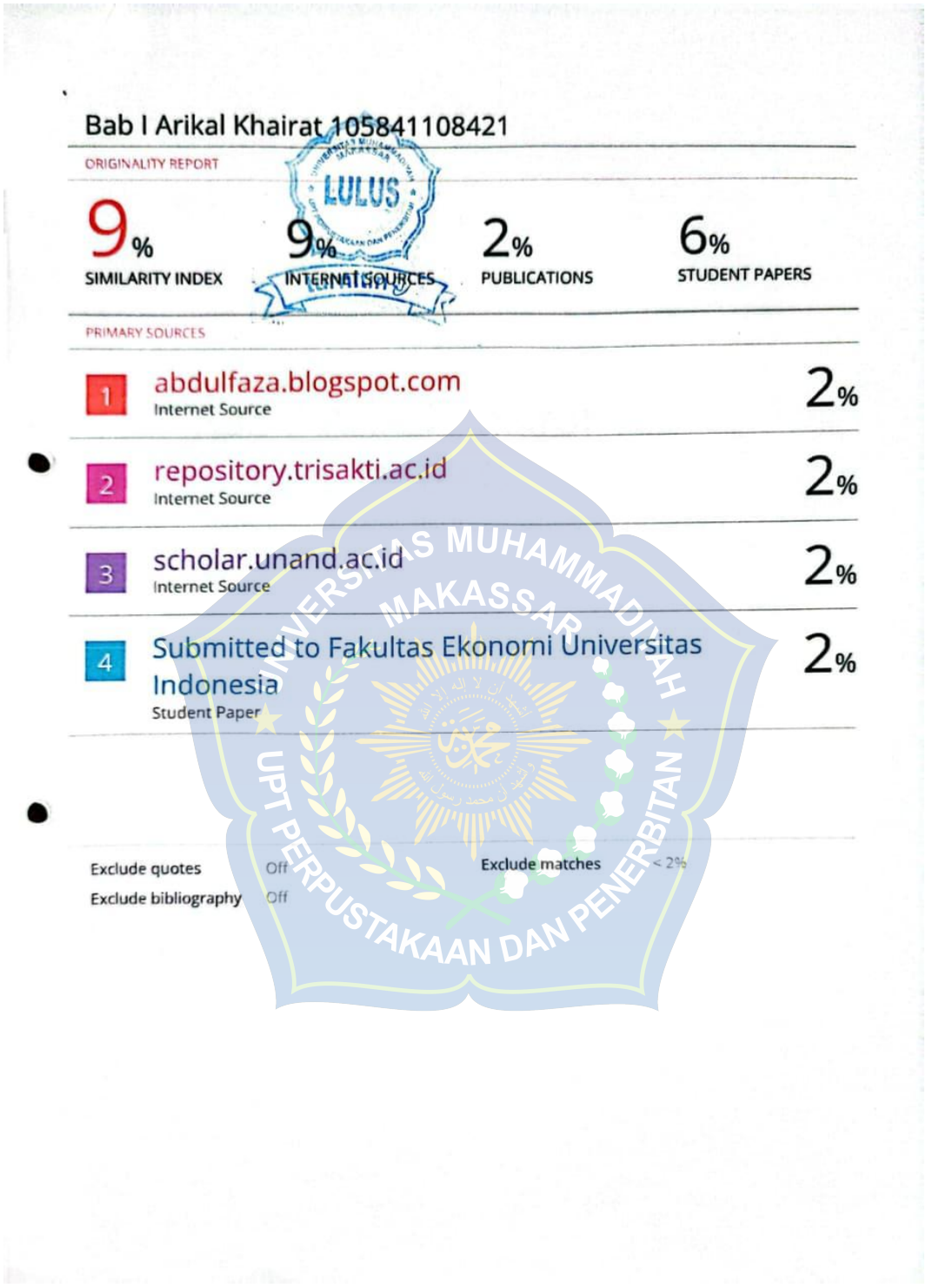
Jl. Sultan Alauddin no 259 makassar 90222
Telepon (0411)866972,881 593,fax (0411)865 588
Website: www.library.unismuh.ac.id
E-mail : perpustakaan@unismuh.ac.id

Lampiran 3. Uji Plagiat



Bab I Arikal Khairat
10584110842l
by Tahap Tutup

Submission date: 23-Aug-2025 06:20AM (UTC+0700)
Submission ID: 2733689015
File name: BAB_I_Tutup.docx (37.46K)
Word count: 884
Character count: 5951



Bab I Arikal Khairat 105841108421

ORIGINALITY REPORT

LULUS

9%	9%	2%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	abdulfaza.blogspot.com Internet Source	2%
2	repository.trisakti.ac.id Internet Source	2%
3	scholar.unand.ac.id Internet Source	2%
4	Submitted to Fakultas Ekonomi Universitas Indonesia Student Paper	2%

Exclude quotes Off

Exclude bibliography Off

Exclude matches < 2%

Bab II Arikal Khairat 105841108421

by Tahap Tutup

Submission date: 23-Aug-2025 06:21AM (UTC+0700)
Submission ID: 2733689227
File name: BAB_II_Tutup.docx (98.37K)
Word count: 2235
Character count: 14766

Bab II Arikal Khairat 105841108421

by Tahap Tutup

Submission date: 23-Aug-2025 06:21AM (UTC+0700)
Submission ID: 2733689227
File name: BAB_II_Tutup.docx (98.37K)
Word count: 2235
Character count: 14766

Bab II Arikal Khairat 105841108421

ORIGINALITY REPORT

9%	7%	4%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Fakultas Teknik Student Paper	1%
2	Submitted to Surabaya University Student Paper	1%
3	journal.uin-alauddin.ac.id Internet Source	1%
4	docplayer.info Internet Source	1%
5	ejournal.unp.ac.id Internet Source	1%
6	Maliatul Fitriyasari. "Tren dan Inovasi dalam Image Hiding untuk Keamanan Informasi Medis: Tinjauan Literatur", RIGGS: Journal of Artificial Intelligence and Digital Business, 2025 Publication	1%
7	Submitted to Universitas Musamus Merauke Student Paper	1%
8	journal.upgris.ac.id Internet Source	1%
9	ojs.unud.ac.id Internet Source	1%
10	lib.unnes.ac.id Internet Source	1%


11 e-journal.hamzanwadi.ac.id
Internet Source
1%
12 repository.radenintan.ac.id
Internet Source
1%
Exclude quotes
Exclude bibliography
Exclude matches
< 1%
UNIVERSITAS MUHAMMADIYAH MAKASSAR
UPT PERPUSTAKAAN DAN PENERBITAN

Bab III Arikal Khairat 105841108421

by Tahap Tutup

Submission date: 23-Aug-2025 06:23AM (UTC+0700)
Submission ID: 2733689786
File name: BAB_III_Tutup.docx (92K)
Word count: 1976
Character count: 12537

Bab III Arikal Khairat 105841108421

ORIGINALITY REPORT

10%	10%	5%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

LULUS

turnitin

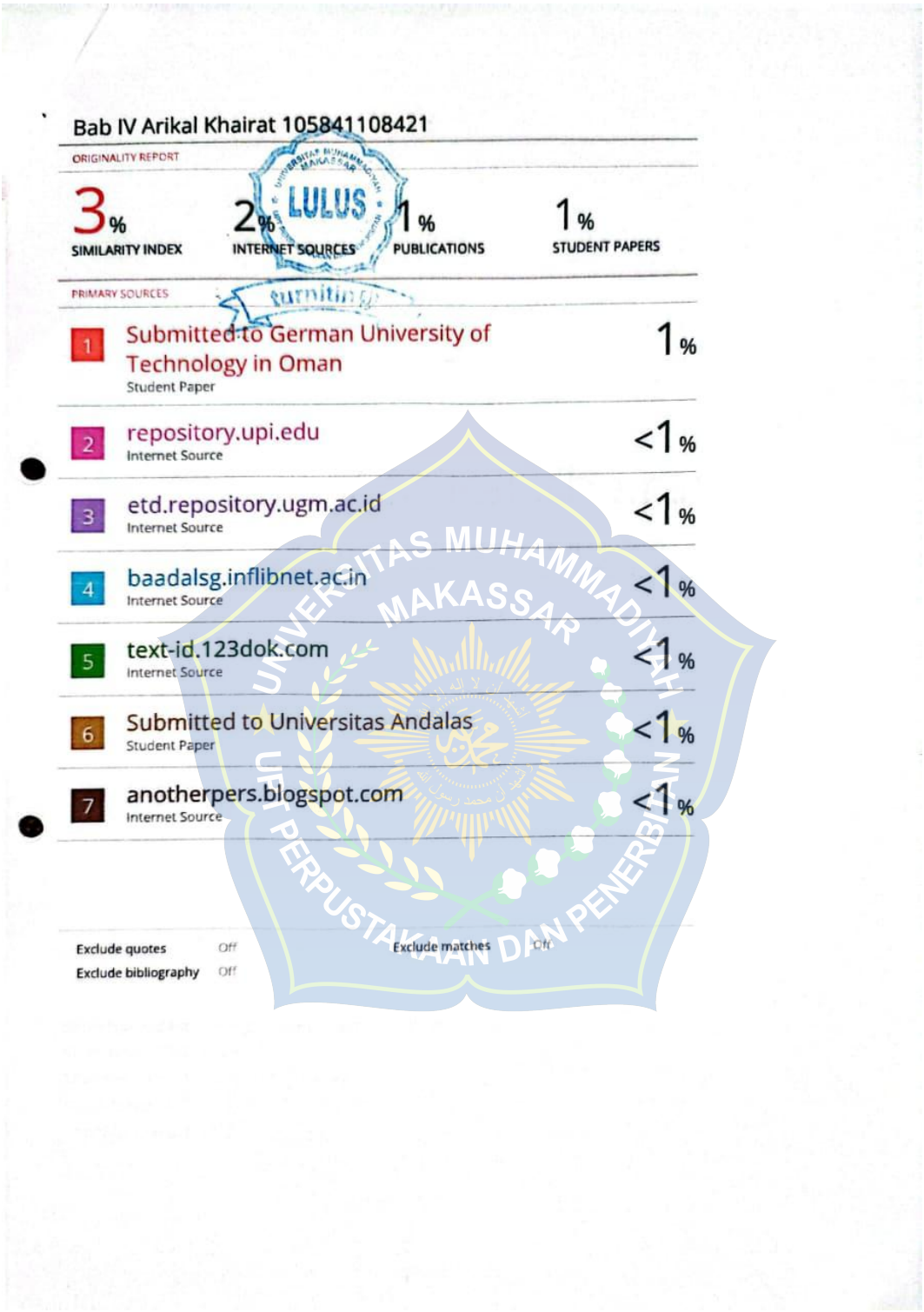
PRIMARY SOURCES

1	Submitted to Universitas Muhammadiyah Makassar Student Paper	3%
2	123dok.com Internet Source	1%
3	repository.unpkediri.ac.id Internet Source	1%
4	text-id.123dok.com Internet Source	1%
5	docplayer.info Internet Source	1%
6	www.scribd.com Internet Source	1%
7	Submitted to Sriwijaya University Student Paper	1%
8	jtera.polteksmi.ac.id Internet Source	1%
9	www.idncitizen.com Internet Source	1%
10	repository.uinjkt.ac.id Internet Source	<1%

Bab IV Arikal Khairat 105841108421

by Tahap Tutup

Submission date: 23-Aug-2025 06:24AM (UTC+0700)
Submission ID: 2733690000
File name: BAB_IV_Tutup.docx (1.4M)
Word count: 2683
Character count: 16596



Bab IV Arikal Khairat 105841108421

ORIGINALITY REPORT

3%	2%	1%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to German University of Technology in Oman Student Paper	1%
2	repository.upi.edu Internet Source	<1%
3	etd.repository.ugm.ac.id Internet Source	<1%
4	baadalsg.inflibnet.ac.in Internet Source	<1%
5	text-id.123dok.com Internet Source	<1%
6	Submitted to Universitas Andalas Student Paper	<1%
7	anotherpers.blogspot.com Internet Source	<1%

Exclude quotes Off
Exclude bibliography Off
Exclude matches Off

Bab V Arikal Khairat 105841108421

by Tahap Tutup

Submission date: 23-Aug-2025 06:25AM (UTC+0700)
Submission ID: 2733690403
File name: BAB_V_Tutup.docx (15.92K)
Word count: 272
Character count: 1810

Bab V Arikal Khairat 105841108421

ORIGINALITY REPORT

5%	5%	3%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	etheses.uinmataram.ac.id Internet Source	5%

Exclude quotes	Off	Exclude matches	< 1%
Exclude bibliography	Off		