

APLIKASI WATERMARK ROBUST REVERSIBLE DENGAN ANTARMUKA

Afi Athallah Syamsulhadi Putra¹, Ledy Novamizanti², Gelar Budiman³

^{1, 2, 3}Universitas Telkom

nygrazium@student.telkomuniversity.ac.id¹, ledyaldn@telkomuniversity.ac.id², gelarbudiman@telkomuniversity.ac.id³

Abstrak

Dalam era digital, keaslian dan keamanan citra digital menjadi sangat penting mengingat mudahnya konten digital diduplikasi dan dimodifikasi tanpa izin. Penulisan ini bertujuan untuk mengembangkan aplikasi robust reversible watermarking citra serta antarmuka pengguna yang intuitif. Fokus utama pengembangan aplikasi adalah pada kemudahan penggunaan agar dapat diakses oleh berbagai kalangan. Aplikasi yang dikembangkan merupakan aplikasi *desktop* yang memiliki antarmuka dan fungsi-fungsinya. Hasil pengujian dengan dua perangkat yaitu laptop dan PC menunjukkan bahwa aplikasi Wmarks mampu menunjukkan responsivitas *latency* yang cukup bagus dengan selisih rata-rata 58.079 ms. Berdasarkan pengujian *black box*, Wmarks berhasil menjalankan fitur dengan baik. Memiliki performa yang cukup baik dengan memperoleh hasil CPU *usage* dengan kategori “baik” dengan rata-rata 0-4% pada kondisi *idle* dan memperoleh kategori “buruk” dengan rata-rata 24-68% pada kondisi fungsi aplikasi tereksekusi. Berdasarkan hasil pengujian *system usability* yang dilakukan oleh 30 responden, aplikasi Wmarks mendapatkan total nilai SUS (*System Usability Scale*) sebesar 80.35 dengan *grade A* yang menunjukkan bahwa aplikasi dapat diterima dan digunakan oleh pengguna dengan sistem operasi *desktop*.

Kata kunci— teknologi watermarking, citra digital, antarmuka pengguna, *latency responsiveness*, *black box*, *spread spectrum adaptive*.

I. PENDAHULUAN

Dalam era digital, keaslian dan keamanan citra digital menjadi sangat penting mengingat mudahnya konten digital digandakan dan dimodifikasi tanpa izin. Teknologi watermarking menawarkan solusi dengan menyisipkan informasi yang tidak terlihat ke dalam citra, yang membantu dalam memverifikasi keaslian dan status kepemilikan citra [1]. Teknologi digital watermarking semakin signifikan dalam berbagai sektor karena memberikan solusi yang efisien untuk berbagai pendekatan dan metode [2]. Watermark adalah tanda yang ditempatkan pada dokumen atau citra untuk menunjukkan kepemilikan, sumber, atau sebagai perlindungan terhadap plagiarisme [3]. Teknik ini memungkinkan penyematan informasi yang tidak mengubah persepsi visual citra tetapi masih efektif dalam menjaga hak cipta dan keaslian citra. Watermarking dirancang untuk tahan terhadap upaya modifikasi atau penghapusan, tanpa mengorbankan kualitas visual citra yang dilindungi [4].

Robust Reversible Watermarking (RRW) adalah sebuah teknik yang memungkinkan menyematankan dan ekstraksi watermark pada citra digital tanpa menyebabkan kehilangan atau kerusakan permanen pada citra asli [5]. Banyak penelitian tentang watermarking citra yang telah dilakukan seperti spread spectrum (SS) [6]. Spread spectrum adaptive (SSA) diusulkan untuk memastikan ketahanan dan reversibilitas proses penandaan air, terutama pada aplikasi yang sensitif seperti pencitraan medis [7]. Teknik penyematan dan amplitudo SSA disesuaikan dengan sinyal host, membantu mengurangi gangguan sumber sambil menjaga keseimbangan antara ketahanan dan kualitas citra yang diberi tanda air. Selain itu, amplitudo disusun sebagai kuantisasi dithered, sehingga dapat diperkirakan secara akurat pada penerima [8].

Banyak aplikasi watermarking yang telah dikembangkan, tetapi *user friendly* dan efisien untuk pengguna umum masih terbatas. Antarmuka pengguna yang intuitif sangat penting

untuk memungkinkan pengguna melakukan proses watermarking tanpa memerlukan pengetahuan teknis yang mendalam. Teknologi ini diimplementasikan dengan menggunakan aplikasi desktop dan pemrograman MATLAB. Aplikasi desktop adalah perangkat lunak yang dirancang untuk berjalan di komputer atau laptop. Aplikasi desktop adalah perangkat lunak yang dirancang untuk berjalan di komputer atau laptop, menyediakan pengguna dengan fungsi-fungsi tertentu seperti pengolahan kata, manajemen spreadsheet, atau alat pengembangan [9]. MATLAB adalah bahasa pemrograman yang digunakan dalam komputasi teknis, memungkinkan para peneliti dan insinyur untuk menganalisis data, mengembangkan algoritma, dan membuat model matematika dengan lebih efisien [10]. Salah satu keunggulan utama MATLAB adalah kemampuannya untuk menangani operasi matriks dan vektor secara langsung, yang mempercepat proses komputasi, terutama pada aplikasi yang membutuhkan perhitungan matematis yang kompleks.

II. KAJIAN TEORI

A. Robust Reversible Watermark (RRW)

Robust Reversible Watermarking adalah sebuah teknik yang memungkinkan penyematan dan ekstraksi watermark pada citra digital tanpa menyebabkan kehilangan atau kerusakan permanen pada citra asli [5]. Teknik ini memanfaatkan kelebihan di antara piksel-piksel citra untuk menyematankan dan mengekstrak watermark, untuk memastikan citra asli dapat dipulihkan dengan sempurna tanpa serangan apa pun. Teknik ini merupakan teknik yang menjanjikan untuk melindungi citra digital karena dapat memulihkan citra asli tanpa kehilangan data. Robust Reversible Watermarking bertujuan untuk memberikan kinerja *imperceptibility*, kapasitas, dan ketahanan yang optimal pada citra digital. RRW dapat memulihkan watermark dan citra host ketika citra yang diberi watermark tidak ada atau sedang diserang.

B. Aplikasi Desktop

Aplikasi *desktop* adalah perangkat lunak yang dirancang untuk dijalankan di komputer desktop atau laptop. Aplikasi ini diinstal langsung pada sistem operasi perangkat pengguna dan umumnya menawarkan kinerja yang lebih cepat dan stabil dibandingkan aplikasi berbasis web. Aplikasi desktop sering kali digunakan dalam lingkungan yang memerlukan fungsionalitas lanjutan atau akses yang lebih besar ke sumber daya perangkat keras [9]. Aplikasi ini biasanya dirancang untuk memanfaatkan sepenuhnya perangkat keras komputer, seperti CPU, memory, dan kartu grafis, yang memungkinkan mereka untuk menangani tugas-tugas yang memerlukan performa tinggi.

III. METODE

A. Robust Reversible Watermark (RRW)

1. Proses *Embedding Watermark*

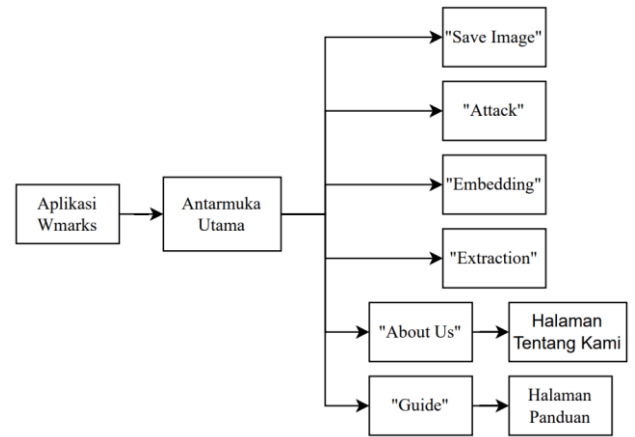
Proses ini dimulai dengan citra host *grayscale* 512×512 yang menjalani transformasi *wavelet*, yang menguraikan citra ke dalam berbagai sub frekuensi. Dari sub-band ini, sub-band tertentu dipilih untuk diproses lebih lanjut. Citra kemudian dibagi menjadi beberapa blok untuk memudahkan penyematan. Dengan menggunakan kunci, komponen F0 dan G0 dihitung untuk menyandikan *watermark*. *Watermark* disematkan ke dalam blok citra yang telah diubah. Setelah itu, blok-blok yang telah dimodifikasi digabungkan dan transformasi *wavelet invers* diterapkan, sehingga menghasilkan citra ber*watermark*. Metode ini memastikan bahwa citra asli dapat dipulihkan, membuat prosesnya dapat *reversible*.

2. Proses *Ekstraksi Watermark*

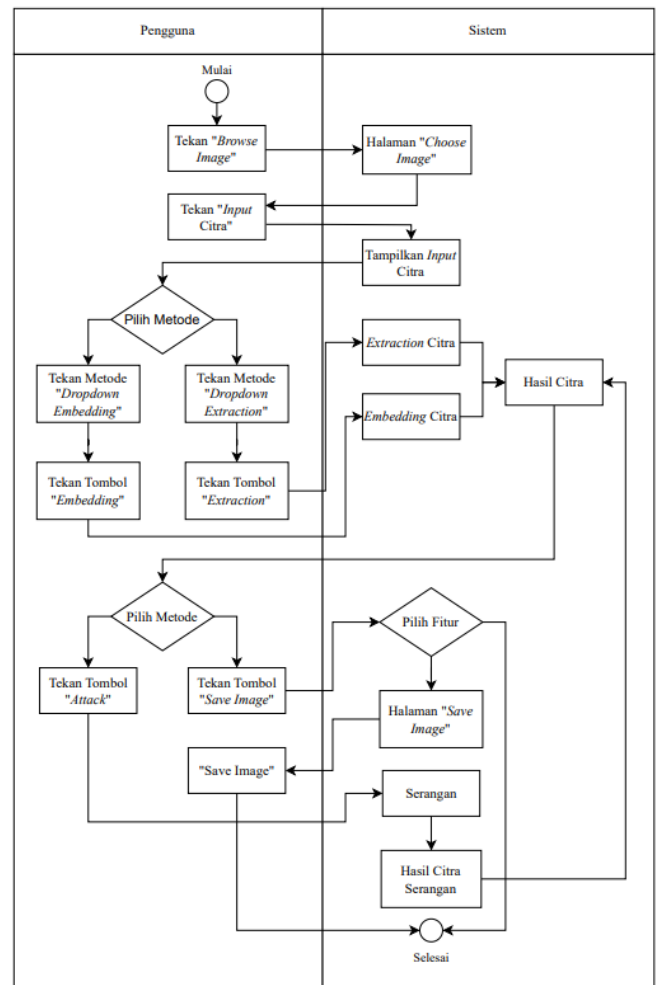
Dimulai dengan citra yang diberi *watermark*, yang mengalami transformasi *wavelet* untuk menguraikannya menjadi berbagai sub. Sub-band tertentu dipilih, dan citra dibagi menjadi beberapa blok agar lebih mudah dimanipulasi. Dengan menggunakan kunci yang sama, vektor yang disematkan diekstraksi. Parameter T dan α diperkirakan untuk membantu membalikkan perubahan yang dibuat selama penyematan. Blok-blok tersebut kemudian digabungkan, dan transformasi *wavelet invers* diterapkan untuk merekonstruksi *watermark* asli dan citra yang dipulihkan. Proses ini memastikan bahwa citra asli dapat dipulihkan sepenuhnya tanpa kehilangan data, membuat teknik ini *reversible*.

B. Desain Sistem Aplikasi

Secara umum, aplikasi desktop ini akan didesain dengan MATLAB menjadi bentuk aplikasi desktop standalone yang bernama Wmarks. Gambar 1 menampilkan desain global dari aplikasi yang dikembangkan, yang secara komprehensif merepresentasikan alur kerja dari berbagai fungsi utama dalam aplikasi. Diagram ini menggambarkan bagaimana antarmuka utama aplikasi berinteraksi dengan fitur dalam sistem termasuk yaitu *About Us*, *Guide*, *Attack*, *Extraction*, dan *Embedding*.



GAMBAR 1
GLOBAL DESAIN APLIKASI



GAMBAR 2
USE CASE DIAGRAM APLIKASI

Gambar 2 menunjukkan *case diagram embedding* dan *extraction* aplikasi Wmarks. Terdapat opsi pengambilan citra dari *directory* pengguna pada antarmuka utama. Ketika citra dipilih, data citra akan diteruskan ke bagian mode proses *embedding* dan *extraction* citra. Proses *Embedding* yaitu proses penyisipan atau penggabungan bit *watermark* ke dalam citra *host* dengan algoritma tertentu, sedangkan proses *extraction* yaitu proses pengambilan atau pemisahan *watermark* yang telah disisipkan ke dalam citra *host*. Citra yang telah ber*watermark* dapat diberikan serangan (*attack*)

untuk mengetahui kualitas proses *watermarking* tersebut. dilakukan perhitungan parameter PSNR untuk mengetahui kualitas citra baik citra *watermark* dan citra hasil *Recovery*. serta perhitungan parameter BER untuk mengetahui kualitas *watermark* yang telah diekstraksi setelah terserang serangan. Setelah itu citra hasil dapat disimpan ke dalam *directory* yang ingin diinginkan.

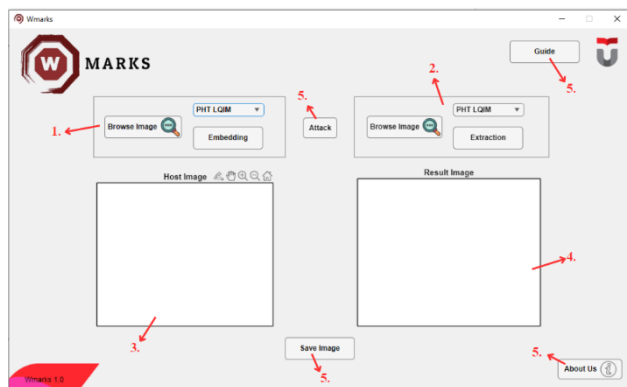
C. Desain Antarmuka

Bagian ini membahas desain antarmuka dari aplikasi Wmarks, yang berfungsi sebagai elemen penting dalam memudahkan interaksi pengguna dengan berbagai fitur yang tersedia. Antarmuka ini dirancang dengan tampilan sederhana namun fungsional, menempatkan berbagai opsi navigasi dan fungsi utama aplikasi di posisi yang mudah diakses oleh pengguna. Fitur seperti nama aplikasi yang ditempatkan di bagian atas kiri serta opsi-opsi utama seperti "guide", "embedding", "extraction", "attack", "save image", dan "about us" diatur secara strategis untuk mendukung alur kerja pengguna yang efisien dan intuitif.

IV. HASIL DAN PEMBAHASAN

A. Implementasi Aplikasi

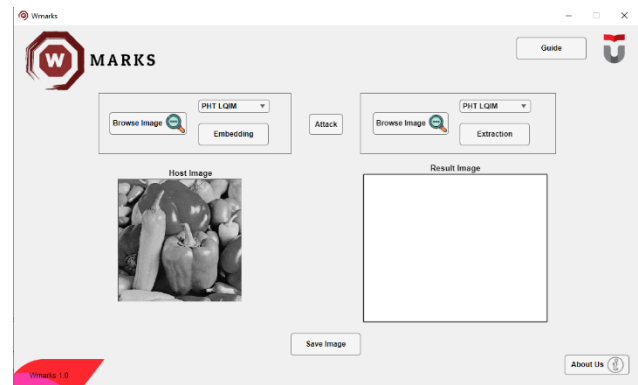
UI yang telah dibuat di App Designer MATLAB menjadi antarmuka utama aplikasi. Pengguna akan masuk kedalam antarmuka utama aplikasi. Fungsi dari halaman ini adalah memberikan beberapa fitur yang bisa dipakai oleh pengguna. Antarmuka ini terdapat 5 fitur untuk pengguna dapat menggunakan fungsi yang diinginkan yaitu ada 1. panel embedding, 2. panel extraction, 3. panel penampilan citra host, 4. panel penampilan citra hasil dan 5. Fitur dengan fungsi pendukung lainnya.



GAMBAR 3
ANTARMUKA UTAMA APLIKASI

1. Citra Embedding

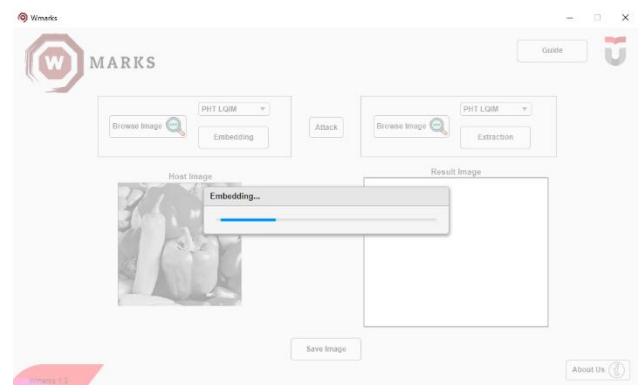
Pada Gambar 3 bagian 1, ketika Fitur tombol "browse image" dipilih, metode "BrowseImagebutton" akan terpanggil. Pertama, variabel "image1" dideklarasikan sebagai variabel *global* untuk memungkinkan akses di seluruh aplikasi. Selanjutnya, fungsi "uigetfile" digunakan untuk membuka jendela dialog agar pengguna dapat memilih citra. Jika *file* citra berhasil dipilih, aplikasi akan mencoba membaca *file* tersebut menggunakan "imread", pada panel penampilan citra *host* "UIAxes" di antarmuka utama seperti pada Gambar 4 akan menampilkan citra yang di input dengan menggunakan "imshow".



GAMBAR 4
PANEL PENAMPILAN CITRA *HOST* PADA ANTARMUKA UTAMA

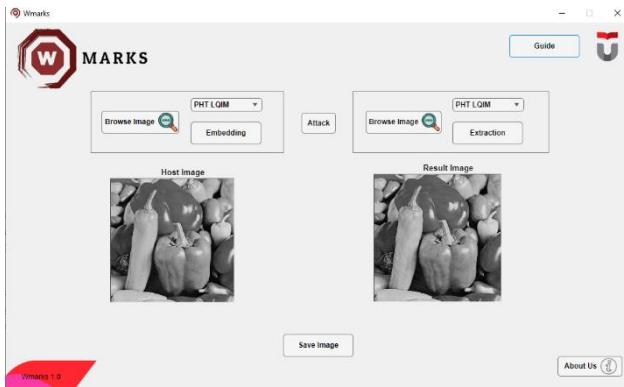
Jika pengguna tidak memilih citra atau menekan tombol "Cancel", maka aplikasi akan menampilkan peringatan melalui dialog dengan pesan "No image selected. Please choose an image.". jika terjadi kesalahan dalam proses pembacaan atau penampilan citra karena format *file* yang tidak didukung, blok "catch" akan bertindak dengan menampilkan pesan *error* yaitu "Error loading the image. Please try again."

Selanjutnya mengimplementasikan proses *embedding* dalam aplikasi Wmarks yang menggabungkan citra dengan teknik *spread spectrum adaptive* Fungsi "EmbeddingButton" dipanggil ketika tombol "Embedding" ditekan oleh pengguna, dilanjutkan dengan memilih metode dengan fitur "dropdown". Tombol embedding di klik dan sebuah dialog progress "(uiprogressdlg)" ditampilkan untuk menunjukkan bahwa proses *embedding* sedang berlangsung seperti pada Gambar 5.



GAMBAR 5
DIALOG PROGRESS EMBEDDING

Dalam prosesnya fungsi ini mengubah ukuran citra input "image1" menjadi resolusi 512x512 *pixel* menggunakan "imresize". Selanjutnya, citra diperiksa apakah merupakan citra RGB atau *grayscale*. Jika citra merupakan citra RGB, konversi ke *grayscale* dilakukan menggunakan fungsi "rgb2gray". Citra hasil yang telah berwatermark "embed.bmp" dibaca dan ditampilkan seperti pada Gambar 3. bagian 4 pada panel penampilan citra hasil "UIAxes2" menampilkan citra hasil proses menggunakan fungsi "imshow". Akhirnya, dialog progress ditutup untuk menandakan bahwa proses *embedding* telah selesai, seperti pada Gambar 6.



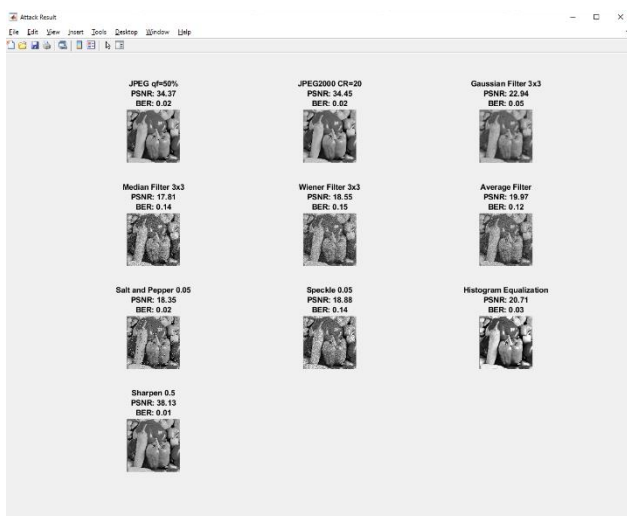
GAMBAR 6
ANTARMUKA PROSES EMBEDDING BERHASIL

2. Ekstraksi Citra

Proses *extraction* citra dapat dilakukan dengan mengklik tombol “browse image” untuk memilih citra yang ingin diproses, citra yang di input akan muncul pada panel penampilan citra *host*. Setelah itu pengguna mengklik tombol “dropdown” dan memilih metode *extraction* yang diinginkan. Pengguna mengklik tombol “extraction” untuk melakukan proses *extraction* pada citra sesuai dengan metode yang dipilih. Hasil berupa citra yang telah terekstrak akan ditampilkan pada panel penampilan citra hasil.

3. Fitur Pendung lainnya

a) *Citra Attack*: Setelah memproses Citra, pengguna dapat memilih untuk menyerang citra yang diproses. Citra yang diproses diserang dengan berbagai serangan, seperti JPEG dan JPEG2000, untuk serangan kompresi. Selain itu, berbagai bentuk serangan lain diterapkan, seperti menambahkan “Gaussian noise”, memfilter dengan filter “Gaussian”, filter “median”, filter “Wiener”, filter “Average”, menambahkan “salt & pepper noise”, “speckle noise”, “histogram equalization”, dan “image sharpen”. Gambar 7. Halaman hasil serangan akan muncul dan menampilkan citra hasil serangan beserta informasi PSNR dan BER.



GAMBAR 7
HALAMAN HASIL SERANGAN

b) *Panduan*: Pengguna dapat memilih fitur panduan untuk melihat halaman panduan aplikasi. Ketika fungsi ini dipanggil, dialog baru akan muncul dengan judul “Guide”. Di

dalam dialog tersebut, terdapat teks yang menjelaskan bahwa aplikasi bernama “Wmarks” digunakan untuk menyematkan dan mengekstrak *watermark* pada Citra digital dengan metode *spread spectrum adaptive* yang menjelaskan tujuan aplikasi tersebut.

c) *About Us*: Pengguna dapat memilih fitur “about us” untuk melihat halaman tentang kami. Fitur “about us” digunakan untuk menampilkan dialog “About Us” yang berisi informasi mengenai judul proyek, anggota tim, dan mentor.

d) *Save Image*: Pengguna dapat memilih fitur simpan citra untuk menyimpan citra yang telah diproses di dalam aplikasi. Fungsi “SaveImageButton” dipanggil ketika tombol “Save Image” ditekan oleh pengguna.

B. Kondisi Eksperimen

Pada penulisan ini, eksperimen dilakukan untuk menguji aplikasi. Kondisi eksperimen yang digunakan dalam penulisan ini meliputi berbagai aspek, seperti sumber data set, jumlah citra *host*, spesifikasi perangkat keras dan perangkat lunak, serta pengaturan uji coba. Semua kondisi ini dipilih untuk memastikan bahwa eksperimen dapat mencerminkan situasi yang realistis dalam aplikasi *watermarking* pada citra digital

1. Spesifikasi Perangkat Keras dan Perangkat Lunak

Eksperimen ini dilaksanakan pada perangkat keras dan perangkat lunak yang dirancang untuk mendukung pemrosesan citra digital secara optimal. Perangkat keras yang digunakan meliputi PC CPU AMD Ryzen 5 5600, RAM 16GB DDR4, GPU NVIDIA RTX 3060, penyimpanan SSD berkapasitas 512 GB dan laptop CPU AMD Ryzen 7 3750h, RAM 16GB DDR4, GPU NVIDIA RTX 2060, penyimpanan SSD berkapasitas 512 GB. Perangkat keras ini dipilih untuk memastikan bahwa aplikasi dapat dijalankan dengan kinerja yang optimal, terutama dalam pengolahan citra yang memerlukan sumber daya komputasi yang cukup besar. Selain itu, perangkat lunak yang digunakan mencakup sistem operasi windows 10 (64-bit).

C. Pengujian dan Analisis Aplikasi

Pengujian dilakukan dengan menggunakan metode *black box testing*, pengukuran *latency*, performa, kompatibilitas dan *system usability scale* untuk mengevaluasi fungsionalitas aplikasi tanpa mempertimbangkan detail implementasi internal. Pengujian hanya didasarkan pada spesifikasi dan persyaratan yang telah ditentukan sebelumnya, dengan tujuan untuk memastikan bahwa aplikasi berfungsi sesuai dengan yang diharapkan dari sudut pandang pengguna.

1. Black Box Testing

Makalah ini menggunakan metode pengujian black box untuk menguji aplikasi Wmarks. Pengujian ini akan memastikan aplikasi Wmarks bekerja dengan baik. Pengujian akan dilakukan dengan menjalankan setiap fitur yang ada pada aplikasi Wmarks untuk memastikan semua fungsi bekerja dengan benar [11]. Hasil pengujian yang didapatkan terdapat pada Tabel I.

Setelah melakukan pengujian fungsi *black box* pada setiap fitur tombol yang ada di antarmuka utama aplikasi Wmarks, hasilnya menunjukkan bahwa aplikasi ini dapat beroperasi dengan baik dan berjalan dengan lancar tanpa masalah.

TABEL I
HASIL PENGUJIAN *BLACK BOX TESTING*

No.	Fitur	Skenario Tes	Hasil Tes	Status
1	Pembuka	Membuka aplikasi Wmarks	Aplikasi mulai dijalankan dan menampilkan logo Wmarks..	Berhasil
2	<i>Embedding</i>	Melakukan input image	Pengguna berhasil untuk memasukkan citra ke aplikasi	Berhasil
3	<i>Embedding</i>	Memilih metode <i>embedding</i>	Pengguna berhasil untuk memilih metode <i>embedding</i> yang diinginkan	Berhasil
4	<i>Embedding</i>	Proses <i>embedding</i> pada citra	Aplikasi berhasil untuk melakukan <i>embedding</i> pada citra..	Berhasil
5	Panel Citra Host	Menampilkan citra	Panel citra berhasil menampilkan citra	Berhasil
6	Panel Citra Hasil	Menampilkan citra	Panel citra berhasil menampilkan citra	Berhasil
7	<i>Attack</i>	Melakukan serangan terhadap hasil proses citra	Aplikasi berhasil melakukan serangan terhadap hasil proses citra dan menampilkan hasilnya.	Berhasil
8	<i>Extraction</i>	Melakukan <i>input image</i>	Pengguna berhasil untuk memasukkan citra ke aplikasi.	Berhasil
9	<i>Extraction</i>	Memilih metode <i>extraction</i>	Pengguna berhasil untuk memilih metode <i>extraction</i> yang diinginkan.	Berhasil
10	<i>Extraction</i>	Proses <i>extraction</i> pada citra	Aplikasi tidak berhasil untuk melakukan <i>extraction</i> pada citra.	Berhasil
11	<i>Save Image</i>	Menyimpan hasil citra ke <i>directory</i> pengguna	Pengguna berhasil menyimpan hasil proses citra ke <i>directory</i> yang diinginkan.	Berhasil
12	<i>Guide</i>	Menjalankan fungsi <i>guide</i>	Aplikasi berhasil menampilkan halaman <i>guide</i>	Berhasil
13.	<i>About Us</i>	Menjalankan fungsi <i>about us</i>	Aplikasi berhasil menampilkan halaman <i>about us</i>	Berhasil

2. Pengukuran Latency

Metode pengujian dengan pengukuran *latency*, yaitu waktu responsif yang dibutuhkan aplikasi untuk mengeksekusi sebuah tombol fungsi. Pengukuran dilakukan pada desktop dengan PC dan laptop pada Tabel II dan Tabel III. Pengukuran menggunakan fungsi “tic” dan “toc” untuk mengukur waktu komputasi sebuah tombol fungsi pada aplikasi. Dengan acuan pengukuran baik 10-50 ms, cukup baik 50-200 ms, dapat ditoleransi 200-500 ms, buruk >500 ms [12].

TABEL II
LAPTOP *LATENCY TEST*

No.	Fungsi Tombol	Rata-rata waktu (ms)	Kategori
1.	<i>Guide</i>	99.4	Cukup Baik
2.	<i>Browse Image</i>	124.4	Cukup Baik
3.	<i>Embedding</i>	97.6	Cukup Baik
4.	<i>Extraction</i>	95.4	Cukup Baik
5.	<i>Attack</i>	744	Buruk
6	<i>About Us</i>	105	Cukup Baik

TABEL III
PC *LATENCY TEST*

No.	Fungsi Tombol	Rata-rata waktu (ms)	Kategori
1.	<i>Guide</i>	0.484	Baik
2.	<i>Browse Image</i>	0.052	Baik
3.	<i>Embedding</i>	0.276	Baik
4.	<i>Extraction</i>	0.082	Baik
5.	<i>Attack</i>	298.8	Dapat Ditoleransi
6	<i>About Us</i>	48.8	Baik

Dari pengujian menggunakan laptop dan PC, hasil yang didapat menggunakan laptop memperoleh lima kategori “cukup baik” dan satu kategori “buruk”. Sedangkan pada PC didapatkan lima kategori “baik” dan satu kategori “dapat ditoleransi”. Fitur “*attack*” yang terdapat pada aplikasi mendapatkan kategori “buruk” dan “dapat ditoleransi”, membuat fungsi fitur ini menjadi salah satu fitur yang memiliki performa kurang baik pada aplikasi. Pada pengujian juga terdapat perbedaan yang cukup besar antara penggunaan aplikasi pada perangkat laptop dan PC, dimana perangkat PC dengan CPU yang lebih unggul mendapatkan performa yang lebih baik dibandingkan aplikasi yang berjalan pada laptop. Rata-rata perbedaan selisih *latency* yang didapatkan setidaknya hingga 58.079 ms. Hal ini bisa terjadi karena perangkat PC lebih unggul untuk menjalankan aplikasi dibandingkan dengan laptop. Jadi spesifikasi minimum yang disarankan untuk menjalankan aplikasi Wmarks adalah dengan CPU AMD Ryzen 5 5600.

3. Pengukuran Performa

Metode pengujian dengan pengukuran performa, yaitu penggunaan CPU dan penggunaan *memory* yang dibutuhkan aplikasi untuk menjalankannya [13]. Pengukuran dilakukan pada desktop dengan spesifikasi laptop dan PC dengan hasil pengujian yang terdapat pada Tabel IV.

TABEL IV
HASIL PERFORMA LAPTOP DAN PC

Usage	Kondisi		Kategori	
	<i>Idle</i>	Fungsi Dijalankan	<i>Idle</i>	Fungsi Dijalankan
Laptop CPU (%)	0-4	24-68	Baik	Buruk
Laptop Memory (MB)	820-924	895-944	Cukup Baik	Cukup Baik
PC CPU (%)	0-2	33-65	Baik	Buruk
PC Memory (MB)	1800-1815	1800-1863	Cukup Baik	Cukup Baik

Pada pengujian yang telah dilakukan pada dua perangkat yaitu Laptop dan PC, didapatkan hasil penggunaan CPU dengan menggunakan perangkat laptop memperoleh kategori “baik” pada kondisi *idle* dan memperoleh kategori “buruk” pada kondisi menjalankan fungsi aplikasi. Perangkat PC memperoleh hasil penggunaan CPU dengan kategori “baik” pada kondisi *idle* dan memperoleh kategori “buruk” pada kondisi menjalankan fungsi aplikasi. Dari hasil kedua perangkat yang telah menjalankan aplikasi Wmarks, tidak terdapat perbedaan yang signifikan. Oleh karena itu perangkat tersebut dapat memanfaatkan CPU yang cukup besar untuk menjalankan aplikasi Wmarks ini. Untuk

pengujian penggunaan memori pada kedua perangkat, didapatkan hasil bahwa setiap perangkat menggunakan penggunaan memori sesuai dengan kebutuhannya masing-masing. Tidak ada perbedaan yang signifikan saat kedua perangkat dalam keadaan idle atau menjalankan fungsi aplikasi. Hanya saja penggunaan memori yang dibutuhkan oleh perangkat PC lebih besar pada kisaran 1800-1863 dibandingkan dengan perangkat laptop dengan kisaran 820-944. Hal ini dapat terjadi karena perangkat PC lebih unggul dibandingkan dengan laptop, sehingga dapat memanfaatkan aplikasi dengan lebih optimal dibandingkan dengan laptop.

5. System Usability Scale

Penelitian ini memberikan kuesioner yang terdiri dari 10 pertanyaan. Responden mengisi kuesioner berdasarkan pengalaman mereka saat menjalankan tugas yang diberikan. Setiap pertanyaan yang diberikan dirancang untuk mengetahui sejauh mana usability aplikasi dari sudut pandang pengguna dengan skala penilaian 1 (sangat tidak setuju) hingga 5 (sangat setuju) seperti pada Tabel V [14].

TABEL V
PERTANYAAN SYSTEM USABILITY SCALE

No.	Question
Q1	Saya merasa akan menggunakan sistem ini lagi
Q2	Saya merasa sistem ini rumit untuk digunakan
Q3	Saya merasa sistem ini mudah digunakan
Q4	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini
Q5	Saya merasa fitur-fitur sistem ini berjalan dengan semestinya
Q6	Saya merasa ada banyak hal yang tidak konsisten pada sistem
Q7	Saya merasa orang lain dapat memahami cara menggunakan sistem ini dengan cepat.
Q8	Saya merasa sistem ini membingungkan
Q9	Saya merasa tidak ada hambatan dalam menggunakan sistem ini
Q10	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini

Untuk menghitung nilai skor, perlu mempertimbangkan aturan berikut.

1. Nilai akhir pertanyaan bernomor ganjil, diperoleh dengan mengurangi satu dari nilai pengguna.
2. Nilai akhir pertanyaan bernomor genap, diperoleh dengan mengurangi nilai pengguna dengan 5.
3. Pembobotan skor SUS diperoleh dari jumlah total skor pengguna dikalikan dengan 2,5.

Rumus untuk menghitung skor total SUS, diterapkan pada kuesioner yang diperoleh dengan (1).

$$\text{SUS score} = \left(((Q1 - 1) + (5 - Q2) + (Q3 - 1) + (5 - Q4) + (Q5 - 1) + (5 - Q6) + (Q7 - 1) + (5 - Q8) + (Q9 - 1) + (5 - Q10)) * 2,5 \right) \quad (1)$$

$$X = \sum_n x \quad (2)$$

Persamaan (2) digunakan untuk menghitung nilai rata-rata skor SUS [9]. Sebanyak 10 pertanyaan diberikan kepada 30 responden. Nilai SUS yang didapatkan menunjukkan tingkat penerimaan pengguna terhadap aplikasi Wmarks dengan nilai 80,35 yang termasuk dalam kategori acceptable dengan Grade A. Dengan acuan pengukuran $A \geq 80,3$, $B \leq 74$, $C \leq 68$, $D \leq 51$, dan $E < 51$ [14].

V. KESIMPULAN

Pengembangan aplikasi *robust reversible watermarking* berbasis desktop dengan *user interface* yang intuitif telah berhasil diwujudkan melalui aplikasi Wmarks yang memanfaatkan lingkungan pemrograman MATLAB. Aplikasi ini dirancang untuk memudahkan proses *embedding* dan *Extraction watermark* pada citra digital dengan menggunakan metode *adaptive spread spectrum*. Hasil pengujian menunjukkan bahwa fitur-fitur pada aplikasi ini dapat berfungsi dengan baik. Pengujian lebih lanjut bahwa aplikasi ini memiliki performa yang cukup baik dalam hal responsivitas. Selain itu, aplikasi Wmarks kompatibel dengan sistem operasi windows 10 dan seterusnya. Hasil evaluasi *System Usability Scale* (SUS) menunjukkan bahwa aplikasi ini mendapatkan nilai yang baik dari pengguna, hal ini menunjukkan bahwa aplikasi ini diterima dengan baik dari segi kegunaan.

REFERENSI

- [1] L. Novamizanti, G. Budiman, and E. N. F. Astuti, "Robust audio watermarking based on transform domain and SVD with compressive sampling framework," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 18, no. 2, pp. 1079-1088, 2020, doi: 10.1093/icon/moab009.
- [2] L. Novamizanti, A. B. Suksmono, D. Danudirdjo, and G. Budiman, "Robust and Imperceptible Watermarking on Medical Images using Coefficient Pair Modification," *IIUM Engineering Journal*, vol. 24, no. 1, pp. 88-105, Jan. 2023, doi: 10.31436/iiujme.v24i1.2597
- [3] Y. HAFIZHANA, I. SAFITRI, L. NOVAMIZANTI, and N. IBRAHIM, "Image Watermarking pada Citra Medis menggunakan Compressive Sensing berbasis Stationary Wavelet Transform," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 8, no. 1, p. 43, Jan. 2020, doi: 10.26760/elkomika.v8i1.43.
- [4] L. Novamizanti, I. Wahidah, and N. Wardana, "A Robust Medical Images Watermarking Using FDCuT-DCT-SVD," *Int. J. Intell. Eng. Syst.*, vol. 13, no. 6, pp. 266-278, 2020, doi: 10.22266/ijies2020.1231.24.
- [5] L. Novamizanti, A. B. Suksmono, D. Danudirdjo, & G. Budiman, "Robust Reversible Watermarking using Stationary Wavelet Transform and Multibit Spread Spectrum in Medical Images", *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 3, pp. 343-354, 2022. doi:10.22266/ijies2022.0630.29.
- [6] Y. Yunawan, I. Safitri, and L. Novamizanti, "Compressive Sensing for Image Watermarking Discrete Wavelet Transform and Spread Spectrum," in *2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*, IEEE, Dec. 2018, pp. 99-103. doi: 10.1109/ICCEREC.2018.8712090
- [7] L. Novamizanti, A. B. Suksmono, D. Danudirdjo, and G. Budiman, "Robust Reversible Image Watermarking based on Independent Embedding Domain and Pixel Value Ordering," in *2023 IEEE 8th International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, IEEE, Dec. 2023, pp. 1-6. doi: 10.1109/ICRAIE59459.2023.10468312.

- [8] Z. Huang, B. Feng, S. Xiang, “*Robust reversible image watermarking scheme based on spread spectrum*”, J. Vis. Commun. Image Represent., vol. 93, May 2023, doi: 10.1016/j.jvcir.2023.103808.
- [9] R. S. Pressman, and B. R. Maxim, “*Software Engineering: A Practitioner's Approach*.” McGraw-Hill Educatio, 2014.
- [10] Moler, C. B. “*Numerical Computing with MATLAB. Society for Industrial and Applied Mathematics*”, 2010.
- [11] Verma A, Khatana, & Chaudhary. “*A Comparative Study of Black Box Testing and White Box Testing*”, 2017.
- [12] Start stopwatch timer - MATLAB tic. (n.d.). MathWorks. Retrieved July 19 2024, from <https://www.mathworks.com/help/matlab/ref/tic.html>
- [13] D. Mishra and P. Kulkarni, “*A survey of memory management techniques in virtualized systems*,” *Comput Sci Rev*, vol. 29, pp. 56–73, Aug. 2018, doi: 10.1016/j.cosrev.2018.06.002.
- [14] Z. Sharfina and H. B. Santoso, “*An Indonesian adaptation of the System Usability Scale (SUS)*,” in *2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, IEEE, Oct. 2016, pp. 145–148. doi: 10.1109/ICACSIS.2016.7872776