Pengembangan Website pada Sistem Deteksi Kemiripan Karya Citra Digital untuk Seleksi Pendaftaran Hak Cipta Berbasis *Artificial Intelligence*

1st Hasna Putri Rachadian
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
hasnaputrirachadian@student.telkomuniversity.ac.id

2nd Iwan Iwut Tritoasmoro
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
iwaniwut@telkomuniversity.ac.id

3rd Eng Danu Dwi Sanjoyo
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
danudwj@telkomuniversity.ac.id

***Abstrak* — Pengembangan sistem deteksi kemiripan karya citra digital tidak hanya memerlukan metode deteksi yang akurat, tetapi juga website yang mampu menyediakan antarmuka yang responsif, mudah digunakan, dan mendukung interaksi pengguna secara efektif. Penelitian ini bertujuan mengembangkan website pada sistem deteksi kemiripan karya citra digital untuk seleksi pendaftaran hak cipta menggunakan ReactJS dan Tailwind CSS. *Frontend* dikembangkan dengan menerapkan arsitektur *component-based* dan diintegrasikan dengan *backend* melalui *REST API* untuk mendukung fitur *image upload*, *threshold settings*, *similarity result*, *generate report*, serta pengelolaan metadata. Evaluasi sistem dilakukan melalui pengujian fungsional, pengujian performa menggunakan Chrome DevTools dan *response time* API, serta pengujian *User Experience* menggunakan kuesioner skala Likert. Hasil pengujian menunjukkan bahwa seluruh fitur *frontend* berhasil berjalan sesuai dengan rancangan sistem. Pengujian performa memperoleh nilai *Performance* sebesar 100% pada *Plagiarism Page* dan 99% pada *Metadata Page*, sedangkan *Accessibility* dan *Best Practices* masing-masing memperoleh nilai hingga 100%. Selain itu, *endpoint upload* memiliki waktu respons sekitar 9,29 detik, sedangkan *endpoint* pengelolaan metadata memiliki waktu respons kurang dari 100 ms. Pengujian *User Experience* memperoleh nilai keseluruhan sebesar 98,18%, sehingga dapat disimpulkan bahwa *website* yang dikembangkan memiliki performa yang baik, mudah digunakan, dan mampu mendukung proses deteksi kemiripan karya citra digital secara efektif.**

***Kata kunci*— Deteksi kemiripan citra, hak cipta, ReactJS, REST API, Tailwind CSS, website.**

I. PENDAHULUAN

Perkembangan teknologi informasi mendorong transformasi digital pada berbagai sektor, termasuk dalam pengelolaan hak kekayaan intelektual. Digitalisasi layanan pendaftaran hak cipta memberikan kemudahan dalam proses pengajuan, penyimpanan, dan pengelolaan dokumen secara elektronik yang dapat meningkatkan efisiensi layanan [1]. Seiring meningkatnya jumlah karya digital yang didaftarkan, proses pemeriksaan kemiripan karya sebelum pendaftaran menjadi semakin penting untuk membantu proses verifikasi dan meminimalkan potensi kemiripan dengan karya yang telah terdaftar sebelumnya [2].

Keberhasilan sistem berbasis web tidak hanya ditentukan oleh kemampuan sistem dalam menjalankan fungsinya, tetapi juga oleh kualitas antarmuka yang digunakan pengguna untuk berinteraksi dengan sistem [3]. *Frontend* yang dirancang secara responsif, konsisten, dan mudah digunakan dapat meningkatkan efektivitas interaksi pengguna, mempercepat penyelesaian tugas, serta memberikan pengalaman penggunaan yang lebih baik [4]. Oleh karena itu, pengembangan *frontend* menjadi salah satu aspek penting dalam menghasilkan aplikasi web yang tidak hanya fungsional, tetapi juga nyaman digunakan.

Beberapa penelitian telah mengembangkan *website* menggunakan teknologi modern. Kroons dan Dewi mengembangkan *dashboard* berbasis website menggunakan ReactJS dan Golang untuk mempermudah proses pemindaian kerentanan *container image* melalui antarmuka yang lebih ramah pengguna dibandingkan *command line interface*. Namun, penelitian tersebut belum mengevaluasi kualitas frontend dari aspek performa halaman web maupun pengalaman pengguna secara komprehensif [5]. Penelitian lain oleh Benz dan Testiana menerapkan Tailwind CSS dalam pengembangan antarmuka website institusi sehingga menghasilkan tampilan yang responsif dan konsisten. Akan tetapi, implementasi yang dilakukan masih berorientasi pada penyajian informasi sehingga interaksi *frontend* dengan layanan *backend* untuk pengelolaan data secara dinamis belum menjadi fokus utama [6].

Berdasarkan kondisi tersebut, penelitian ini bertujuan mengembangkan *website* pada sistem deteksi kemiripan karya citra digital untuk seleksi pendaftaran hak cipta menggunakan ReactJS dan Tailwind CSS. *Frontend* diintegrasikan dengan *backend* melalui REST API untuk mendukung proses unggah citra, pengelolaan metadata, pengaturan *threshold*, serta penyajian hasil deteksi kemiripan secara dinamis. Selain itu, *frontend* yang dikembangkan dievaluasi melalui pengujian fungsional, pengujian performa menggunakan Chrome DevTools, dan pengujian *User Experience* untuk mengetahui kualitas antarmuka yang dihasilkan.

II. KAJIAN TEORI

Kajian teori menyajikan konsep dan teori yang menjadi landasan penelitian, meliputi *ReactJS* sebagai *library* pengembangan *frontend*, *Tailwind CSS* sebagai *framework* CSS, *REST API* sebagai mekanisme komunikasi antara *frontend* dan *backend*, serta *User Experience* sebagai dasar evaluasi pengalaman pengguna.

A. ReactJS

ReactJS merupakan *library* JavaScript yang berfungsi untuk mengembangkan antarmuka pengguna pada aplikasi web. ReactJS mengadopsi konsep *component-based architecture*, yaitu pendekatan yang membagi antarmuka menjadi komponen-komponen yang dapat digunakan kembali (*reusable*) sehingga proses pengembangan maupun pemeliharaan aplikasi menjadi lebih terstruktur. Selain itu, ReactJS memanfaatkan *Virtual DOM* untuk mengoptimalkan proses pembaruan tampilan dengan hanya merender bagian antarmuka yang mengalami perubahan [7][8].

B. Tailwind CSS

Tailwind CSS Adalah *framework* CSS yang menerapkan pendekatan *utility-first* dengan menyediakan berbagai kelas utilitas untuk membangun antarmuka pengguna langsung pada elemen HTML. Pendekatan tersebut mempercepat proses pengembangan antarmuka, menjaga konsistensi tampilan, serta mendukung desain yang responsif tanpa memerlukan banyak penulisan kode CSS terpisah [6]. Dengan karakteristik tersebut, Tailwind CSS menjadi salah satu framework CSS yang banyak digunakan dalam pengembangan aplikasi web modern.

C. REST API

Representational State Transfer Application Programming Interface merupakan arsitektur layanan web yang digunakan untuk mendukung komunikasi dan pertukaran data antara frontend dan backend melalui protokol HTTP [9]. REST API memanfaatkan metode HTTP, seperti GET, POST, PUT, dan DELETE, sehingga proses integrasi dapat dilakukan tanpa bergantung pada platform maupun bahasa pemrograman yang digunakan [10]. REST API dirancang dengan prinsip stateless sehingga setiap permintaan dapat diproses secara mandiri.

D. *User Experince* (UX)

User Experience adalah persepsi dan tanggapan pengguna yang muncul saat menggunakan atau sebelum menggunakan suatu produk, sistem, atau layanan. UX tidak hanya berfokus pada kemudahan penggunaan (*usability*), tetapi juga mencakup kenyamanan, efisiensi, kepuasan, serta pengalaman emosional pengguna selama berinteraksi dengan sistem. Oleh karena itu, evaluasi *User Experience* dilakukan untuk mengetahui kualitas aplikasi berdasarkan pengalaman pengguna serta menilai apakah sistem telah mampu memenuhi kebutuhan penggunanya [11].

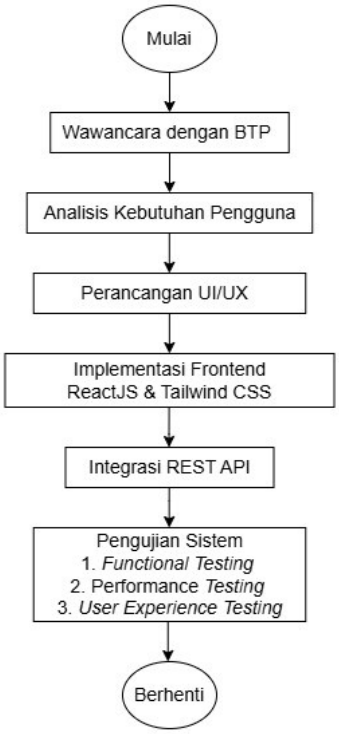
III. METODE

Penelitian ini menggunakan metode pengembangan aplikasi berbasis web yang terdiri atas tahapan analisis kebutuhan, perancangan *frontend*, implementasi antarmuka pengguna, integrasi dengan *backend* melalui *REST API*, serta pengujian sistem. Metode tersebut diterapkan untuk menghasilkan *website* yang mampu mendukung proses deteksi kemiripan karya citra digital secara interaktif, responsif, dan mudah digunakan.

A. Tahapan Penelitian

Tahapan penelitian diawali dengan analisis kebutuhan pengguna melalui proses wawancara dengan pihak Bandung Techno Park (BTP) sebagai calon pengguna sistem. Wawancara dilakukan untuk mengidentifikasi kebutuhan fungsional, alur penggunaan sistem, serta fitur-fitur yang diperlukan pada *website* dalam mendukung proses pemeriksaan kemiripan karya citra digital.

Berdasarkan hasil analisis kebutuhan, dilakukan perancangan antarmuka pengguna (*user interface*) yang terdiri atas *Plagiarism Page* dan *Metadata Page*. Selanjutnya *frontend* diimplementasikan menggunakan ReactJS dan Tailwind CSS dengan menerapkan pendekatan *component-based architecture*. Setelah implementasi selesai, *frontend* diintegrasikan dengan *backend* melalui *REST API* untuk mendukung proses pertukaran data secara *real-time*. Tahap terakhir adalah melakukan pengujian terhadap *frontend* yang meliputi pengujian fungsional, pengujian performa, dan evaluasi *User Experience*. Tahapan penelitian ditunjukkan pada Gambar 1 berikut.

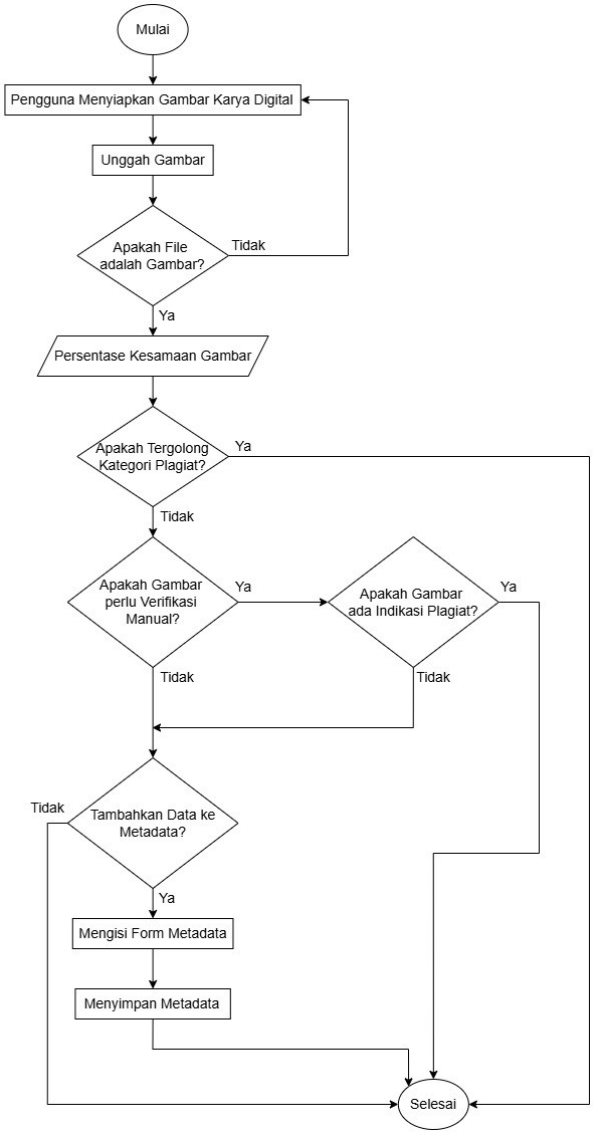


GAMBAR 1
Tahapan Penelitian

B. Perancangan *Website*

Website dikembangkan menggunakan ReactJS sebagai *JavaScript library* dan Tailwind CSS sebagai *CSS framework*. Pengembangan *frontend* menerapkan konsep *component-based architecture* sehingga antarmuka tersusun atas komponen-komponen yang bersifat *reusable* dan mudah dipelihara. *Frontend* berfungsi sebagai media interaksi antara pengguna dan sistem, sedangkan seluruh proses pengolahan data dilakukan pada *backend* melalui komunikasi menggunakan *REST API*.

Frontend terdiri atas dua halaman utama, yaitu *Plagiarism Page* dan *Metadata Page*. *Plagiarism Page* digunakan untuk melakukan *upload image*, pengaturan *threshold*, menampilkan hasil *similarity*, serta menghasilkan *report*. Sementara itu, *Metadata Page* digunakan untuk menampilkan, mencari, memperbarui, dan menghapus metadata karya digital. Alur penggunaan *website* ditunjukkan pada Gambar 2 berikut.



GAMBAR 2
Activity Diagram Frontend

C. Pengujian *Website*

Pengujian dilakukan untuk memastikan bahwa *website* telah memenuhi kebutuhan fungsional maupun nonfungsional. Pengujian terdiri atas pengujian fungsional, pengujian performa, dan evaluasi *User Experience*. Setiap pengujian dilakukan menggunakan skenario yang berbeda sesuai dengan tujuan evaluasi.

1. Pengujian Fungsional

   Pengujian fungsional dilakukan untuk memastikan setiap fitur pada *frontend* berjalan sesuai dengan rancangan sistem. Pengujian mencakup fitur *image upload*, *threshold settings*, *similarity result*, *generate report*, pengelolaan metadata, *error handling*, serta keamanan penyimpanan data pada sisi *browser*. Keberhasilan pengujian ditentukan berdasarkan kesesuaian fungsi yang dijalankan dengan respons yang diterima dari *backend* melalui *REST API*.

2. Pengujian Performa

   Pengujian performa dilakukan untuk mengevaluasi keandalan *website* melalui pengukuran kualitas halaman dan waktu respons komunikasi antara *frontend* dan *backend*. Kualitas halaman dievaluasi menggunakan Chrome DevTools berdasarkan metrik *Performance*, *Accessibility*, dan *Best Practices*. Selain itu, dilakukan pengujian *response time* API untuk mengukur waktu respons pada proses *upload image* maupun pengelolaan metadata.

3. Evaluasi *User Experience*

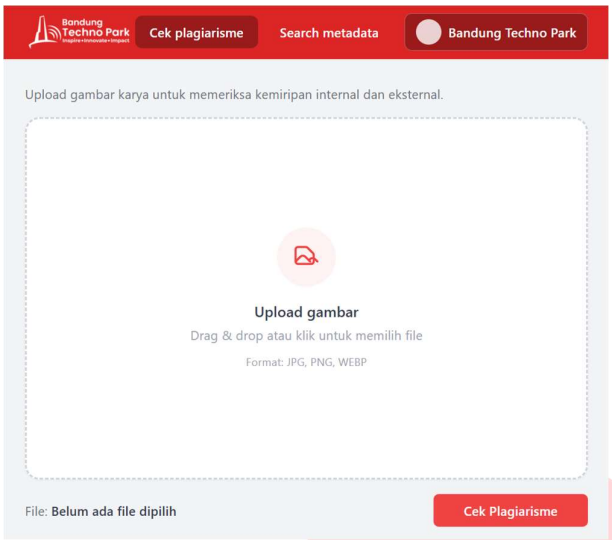
   Evaluasi *User Experience* dilakukan menggunakan kuesioner berbasis skala Likert setelah responden menggunakan *website*. Evaluasi bertujuan untuk mengukur tingkat kemudahan penggunaan, kejelasan antarmuka, kemudahan navigasi, serta kepuasan pengguna terhadap fitur-fitur yang tersedia. Hasil evaluasi digunakan untuk mengetahui tingkat penerimaan pengguna terhadap *website* yang dikembangkan.

IV. HASIL DAN PEMBAHASAN

Pada bagian ini dibahas hasil implementasi *website* yang telah dikembangkan serta hasil pengujian untuk mengevaluasi kualitas sistem. Pembahasan meliputi implementasi antarmuka pengguna, hasil pengujian fungsional, pengujian performa, serta evaluasi *User Experience*. Setiap hasil pengujian dianalisis untuk mengetahui apakah *frontend* yang dikembangkan telah memenuhi kebutuhan pengguna dan mendukung proses deteksi kemiripan karya citra digital.

A. Implementasi *Plagiarism Page*

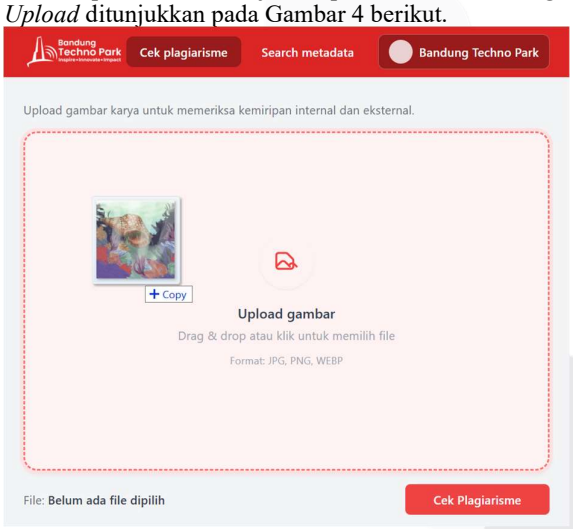
Plagiarism Page merupakan halaman utama yang digunakan pengguna untuk melakukan pemeriksaan kemiripan karya citra digital. Halaman ini mengintegrasikan seluruh proses deteksi kemiripan, mulai dari unggah citra hingga penyajian hasil pemeriksaan. Selain itu, halaman ini juga menyediakan pengaturan parameter deteksi sehingga pengguna dapat menyesuaikan proses pemeriksaan sesuai kebutuhan. Implementasi *Plagiarism Page* ditunjukkan pada Gambar 3 berikut.



GAMBAR 3
Implementasi *Plagiarism Page*

1. Fitur *Image Upload*

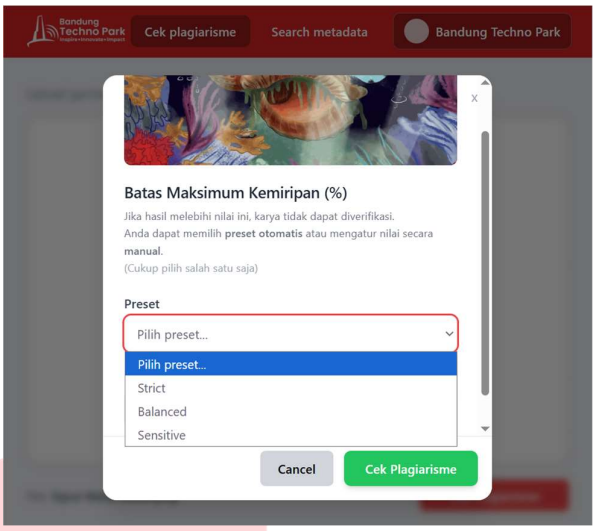
Fitur *Image Upload* digunakan untuk mengunggah citra yang akan diperiksa tingkat kemiripannya. Pengguna dapat memilih berkas citra berformat JPG, JPEG, atau PNG melalui perangkat yang digunakan. Setelah proses validasi format berhasil dilakukan, citra akan dikirimkan ke *backend* menggunakan *REST API* untuk diproses lebih lanjut. Implementasi fitur *Image Upload* ditunjukkan pada Gambar 4 berikut.



GAMBAR 4
Implementasi Fitur *Image Upload*

2. Fitur *Threshold Settings*

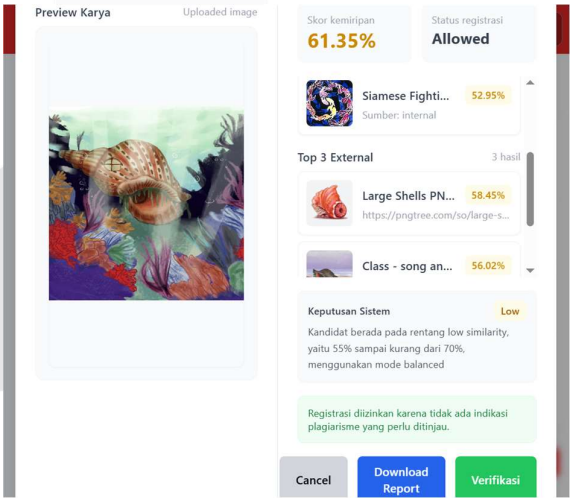
Fitur *Threshold Settings* digunakan untuk menentukan tingkat sensitivitas proses deteksi kemiripan Pengguna dapat memilih konfigurasi *preset*, yaitu *Strict*, *Balanced*, atau *Sensitive*, yang masing-masing memiliki nilai *threshold* berbeda. Selain menggunakan *preset*, pengguna juga dapat mengatur *threshold* secara manual sesuai dengan kebutuhan pemeriksaan. Fleksibilitas tersebut memungkinkan pengguna menyesuaikan tingkat sensitivitas deteksi berdasarkan kebutuhan dan karakteristik citra yang diperiksa. Implementasi fitur *Threshold Settings* ditunjukkan pada Gambar 5 berikut.



GAMBAR 5
Implementasi Fitur *Threshold Settings*

3. Fitur *Similarity Result*

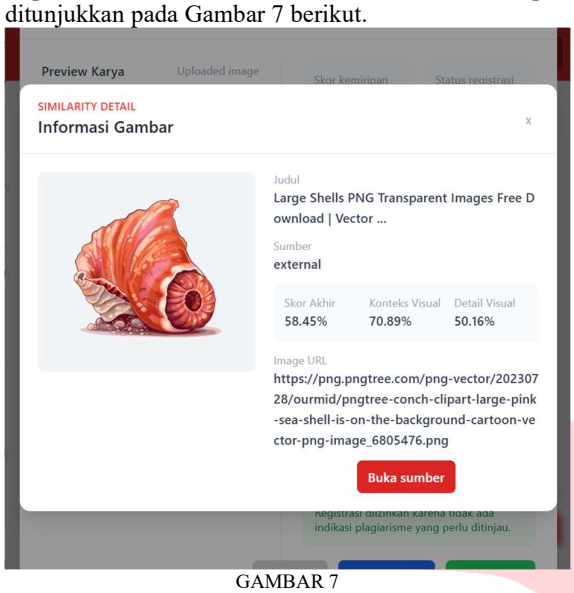
Fitur *Similarity Result* menampilkan hasil pemeriksaan kemiripan setelah proses deteksi selesai dilakukan. Informasi yang ditampilkan meliputi skor kemiripan, status registrasi, daftar citra internal maupun eksternal dengan tingkat kemiripan tertinggi, serta keputusan sistem berdasarkan nilai *threshold* yang dipilih. Pada halaman ini juga tersedia tombol *Download Report* yang digunakan untuk mengunduh laporan hasil pemeriksaan kemiripan sebagai dokumentasi, serta tombol Verifikasi yang digunakan untuk melanjutkan proses verifikasi dan pengambilan keputusan terhadap hasil deteksi. Implementasi fitur *Similarity Result* ditunjukkan pada Gambar 6 berikut.



GAMBAR 6
Implementasi Fitur *Similarity Result*

Pengguna dapat melihat informasi lebih rinci dari setiap hasil kemiripan dengan memilih salah satu citra yang ditampilkan pada daftar hasil. Sistem akan menampilkan informasi berupa judul karya, sumber citra, nilai *Final Score*, *Visual Context*, *Visual Detail*, serta alamat *Image URL* dari citra pembanding. Selain itu, tersedia tombol Buka Sumber yang berfungsi mengarahkan pengguna ke halaman asal citra sehingga

proses verifikasi dapat dilakukan secara lebih mendalam. Implementasi halaman detail hasil kemiripan ditunjukkan pada Gambar 7 berikut.



GAMBAR 7
Implementasi *Detail Similarity Result*

4. Fitur *Metadata Form*

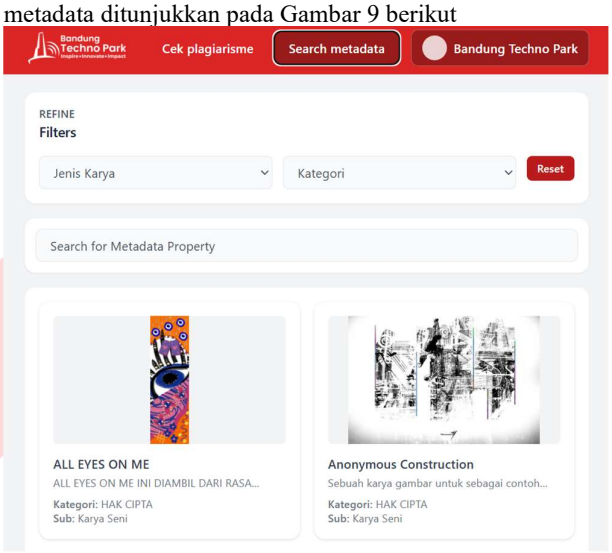
Fitur *Metadata Form* digunakan untuk melengkapi informasi metadata karya digital sebagai tahap akhir setelah proses pemeriksaan kemiripan selesai dilakukan. Formulir ini memuat beberapa informasi yang harus diisi oleh pengguna, seperti judul, deskripsi, kategori, subkategori, kategori HKI, dan subkategori HKI. Selain itu, sistem secara otomatis menampilkan *Check ID* sebagai identitas hasil pemeriksaan yang telah dilakukan. Setelah seluruh informasi terisi, pengguna dapat menyimpan metadata melalui tombol Simpan Metadata sehingga data karya beserta hasil pemeriksaan kemiripan tersimpan ke dalam sistem. Implementasi fitur *Metadata Form* ditunjukkan pada Gambar 8 berikut.



GAMBAR 8
Implementasi Fitur *Metadata Form*

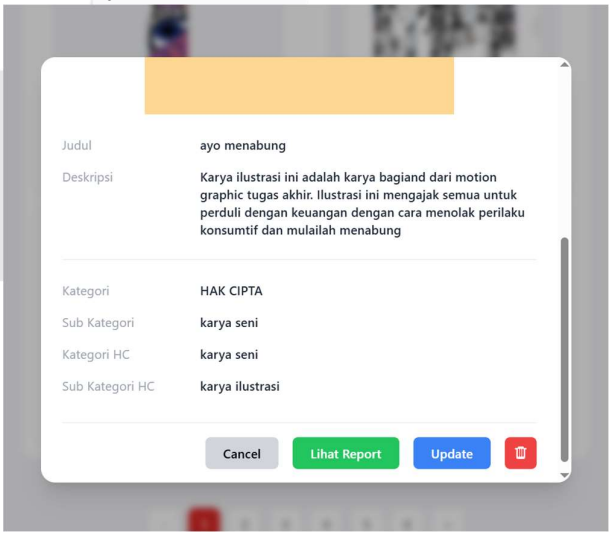
B. Implementasi *Metadata Page*

Metadata Page digunakan untuk mengelola informasi metadata karya digital yang tersimpan pada sistem. Pada halaman ini, pengguna dapat melihat daftar metadata yang telah tersimpan dalam bentuk *card*, melakukan pencarian berdasarkan kata kunci, serta memanfaatkan *filter* untuk mempersempit hasil pencarian berdasarkan jenis karya maupun kategori. Implementasi halaman pengelolaan metadata ditunjukkan pada Gambar 9 berikut



GAMBAR 9
Implementasi *Metadata Page*

Setelah pengguna memilih salah satu metadata, sistem akan menampilkan informasi secara lebih rinci melalui jendela *pop-up*. Halaman ini menampilkan informasi seperti judul, deskripsi, kategori, subkategori, serta kategori hak cipta dari karya yang dipilih. Selain itu, tersedia fitur *update* untuk memperbarui metadata, *delete* untuk menghapus metadata, dan *report* untuk menampilkan laporan hasil pemeriksaan kemiripan yang berkaitan dengan karya tersebut. Implementasi detail metadata ditunjukkan pada Gambar 10 berikut.



GAMBAR 10
Implementasi *Detail Metadata*

C. Hasil Pengujian Fungsional

Pengujian fungsional dilakukan untuk memastikan bahwa seluruh fitur pada *website* dapat berjalan sesuai dengan rancangan sistem. Pengujian meliputi proses *upload image*, pengaturan *threshold*, tampilan hasil *similarity*, halaman metadata, fitur *report*, *error handling*, serta keamanan penyimpanan data pada sisi *browser*. Berdasarkan hasil pengujian, seluruh fitur berhasil dijalankan tanpa ditemukan kesalahan fungsional. Hasil pengujian ditunjukkan pada Tabel 1 berikut.

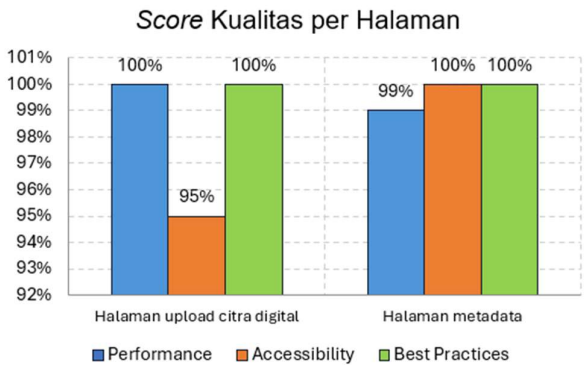
TABEL 1
Hasil Pengujian Fungsional *Frontend*

No	Skenario Pengujian	Hasil
1	*Upload Image*	Citra berhasil diunggah dan diteruskan ke *backend*
2	*Threshold Settings*	*Threshold* berhasil diterapkan sesuai konfigurasi
3	*Similarity Result*	Nilai *similarity* dan *risk level* berhasil ditampilkan sesuai respons API.
4	*Detail Similarity*	Informasi detail citra pembanding berhasil ditampilkan.
5	*Metadata Page*	Fitur pencarian, pembaruan, dan penghapusan metadata berjalan dengan baik.
6	*Generate Report*	Laporan hasil deteksi dapat ditampilkan dan disimpan
7	*Error Handling*	Sistem menampilkan pesan kesalahan ketika terjadi kegagalan proses atau input tidak valid.
8	Keamanan Data	Tidak ditemukan penyimpanan data sensitif secara permanen pada *browser*.

Berdasarkan Tabel 1, seluruh skenario pengujian berhasil dijalankan sesuai dengan fungsi yang telah dirancang. Proses *upload image* mampu mengirimkan citra ke *backend* dengan baik, sedangkan pengaturan *threshold* berhasil diterapkan sesuai konfigurasi yang dipilih pengguna, baik menggunakan *preset* maupun nilai ambang yang ditentukan secara manual. Selain itu, hasil *similarity* dan informasi detail citra pembanding dapat ditampilkan sesuai dengan respons API yang diterima. Fitur pengelolaan metadata, *generate report*, serta *error handling* juga berfungsi dengan baik tanpa ditemukan kendala selama proses pengujian. Hasil tersebut menunjukkan bahwa seluruh fungsi pada *frontend* telah terintegrasi dengan *backend* dan mampu mendukung proses deteksi kemiripan karya citra digital secara optimal.

D. Hasil Pengujian Performa

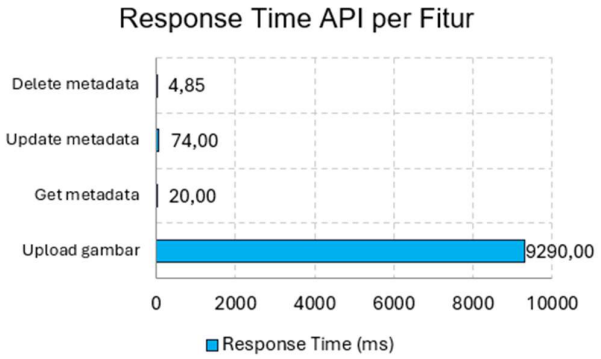
Pengujian performa dilakukan untuk mengevaluasi kualitas *website* dari sisi kualitas halaman (*page quality*) dan waktu respons komunikasi antara *frontend* dengan *backend*. Evaluasi kualitas halaman dilakukan menggunakan Chrome DevTools berdasarkan metrik *Performance*, *Accessibility*, dan *Best Practices*. Hasil pengujian kualitas halaman pada *Plagiarism Page* dan *Metadata Page* ditunjukkan pada Gambar 11 berikut.



GAMBAR 11
Hasil Pengujian Kualitas Halaman

Berdasarkan Gambar 11, *Plagiarism Page* memperoleh nilai *Performance* sebesar 100%, *Accessibility* sebesar 95%, dan *Best Practices* sebesar 100%. Sementara itu, *Metadata Page* memperoleh nilai *Performance* sebesar 99%, sedangkan *Accessibility* dan *Best Practices* masing-masing memperoleh nilai 100%. Hasil tersebut menunjukkan bahwa kedua halaman memiliki performa yang sangat baik, mudah diakses oleh pengguna, serta telah memenuhi praktik pengembangan web modern. Perbedaan kecil pada nilai *Accessibility* di *Plagiarism Page* tidak memengaruhi fungsi utama sistem dan masih berada pada kategori sangat baik.

Selain pengujian kualitas halaman, dilakukan pula pengujian *response time* untuk mengukur waktu respons komunikasi antara *frontend* dan *backend* melalui setiap *endpoint* API. Pengujian dilakukan pada *endpoint upload* serta *endpoint* pengelolaan metadata. Hasil pengujian *response time* ditunjukkan pada Gambar 12 berikut.

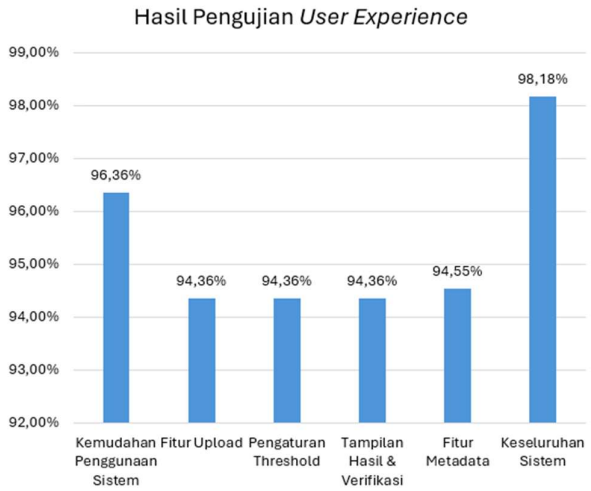


GAMBAR 12
Hasil Pengujian *Response Time API*

Berdasarkan Gambar 12, *endpoint upload* memiliki waktu respons paling tinggi, yaitu sekitar 9,29 detik, karena proses tersebut mencakup unggah citra, ekstraksi fitur, pencarian kemiripan, serta penyusunan hasil deteksi sebelum dikirimkan kembali ke *frontend*. Sebaliknya, *endpoint* pengelolaan metadata memiliki waktu respons di bawah 100 ms, sehingga proses seperti pencarian, pembaruan, dan penghapusan metadata dapat dilakukan secara cepat. Hasil ini menunjukkan bahwa komunikasi antara *frontend* dan *backend* telah berjalan dengan baik, sementara waktu respons yang lebih tinggi pada *endpoint upload* masih sesuai dengan kompleksitas proses yang dilakukan.

E. Hasil Evaluasi *User Experince*

Evaluasi *User Experience* dilakukan untuk mengetahui tingkat kemudahan penggunaan *website* berdasarkan pengalaman pengguna setelah menggunakan sistem. Pengujian dilakukan menggunakan kuesioner skala Likert yang mencakup aspek kemudahan penggunaan sistem, fitur *upload*, pengaturan *threshold*, tampilan hasil *similarity*, serta fitur metadata. Hasil evaluasi ditunjukkan pada Gambar 13 berikut.



GAMBAR 13
Hasil Pengujian *User Experience*

Berdasarkan Gambar 13, seluruh aspek memperoleh nilai di atas 94,798%, sedangkan nilai keseluruhan sistem mencapai 98,18%. Hasil tersebut menunjukkan bahwa pengguna menilai *website* mudah digunakan, memiliki tampilan antarmuka yang jelas, serta mendukung proses pemeriksaan kemiripan citra secara efektif. Nilai yang tinggi pada seluruh aspek menunjukkan bahwa fitur-fitur yang dikembangkan telah memenuhi kebutuhan pengguna.

V. KESIMPULAN

Penelitian ini berhasil mengembangkan *website* untuk sistem deteksi kemiripan karya citra digital menggunakan ReactJS dan Tailwind CSS. *Frontend* yang dikembangkan menyediakan fitur *image upload*, *threshold settings*, *similarity result*, *generate report*, serta pengelolaan metadata yang telah terintegrasi dengan *backend* melalui *REST API*. Berdasarkan hasil pengujian fungsional, seluruh fitur berhasil diimplementasikan sesuai rancangan dan mampu mendukung proses deteksi kemiripan karya citra digital.

Hasil pengujian performa menunjukkan bahwa *Plagiarism Page* memperoleh nilai *Performance* 100%, *Accessibility* 95%, dan *Best Practices* 100%, sedangkan *Metadata Page* memperoleh nilai *Performance* 99%, *Accessibility* 100%, dan *Best Practices* 100%. Selain itu, *endpoint upload* memiliki waktu respons sekitar 9,29 detik, sedangkan *endpoint* pengelolaan metadata memiliki waktu respons kurang dari 100 ms. Pengujian *User Experience* juga menunjukkan hasil yang sangat baik dengan nilai keseluruhan sebesar 98,18%, Sehingga dapat disimpulkan bahwa website yang dikembangkan memiliki performa yang baik, mudah digunakan, dan mampu mendukung proses deteksi kemiripan karya citra digital.

REFERENSI

[1] N. A. Rajab, A. Marianta, I. Lestari, N. F. Azhar and B. Prihasto, "Evolution of the Intellectual Property Information System at the Kalimantan Institute of Technology Using the Waterfall Method and Design Thinking," *Sebatik,* vol. 29, no. 2, 2025.

[2] A. Sukardi, D. Trisnawarman and N. J. Perdana, "Perancangan Sistem Informasi Pendaftaran Hak Cipta," *JUTISI: Jurnal Ilmiah Teknik Informatika dan Sistem Informasi,* vol. 13, no. 3, p. 2349, 2024.

[3] M. F. S. Lazuardy and D. Anggraini, "Modern Front End Web Architectures with React.Js," *International Research Journal of Advanced Engineering and Science,* vol. 7, no. 1, p. 132, 2022.

[4] O. N. Faizah, N. R. Oktadini, B. W. Putra, P. E. Sevtiyuni, P. Putra and A. Meiriza, "UI/UX Design of Web-based Software License Management System using User-Centered Design and System Usability Scale," *Jurnal Nasional Teknologi dan Sistem Informasi,* vol. 9, no. 3, p. 255, 2023.

[5] A. A. Kroons and C. Dewi, "Pengembangan Dashboard Trivy Berbasis Website Menggunakan React JS dan Golang," *Jurnal Indonesia: Manajemen Informatika dan Komunikasi,* vol. 4, no. 3, pp. 1037-1049, 2023.

[6] R. A. Benzi and G. Testiana, "Penerapan Tailwind CSS Dalam Pengembangan *User interface* Website Di Poltekpar Palembang," *Journal of Artificial Intelligence and Digital Business (RIGGS),* vol. 4, no. 6635–6643, p. 4, 2025.

[7] P. S. Maratkar and P. Adkar, "React JS – An Emerging Frontend Javascript Library," *IRE Journals,* vol. 4, no. 12, pp. 99-102, 2021.

[8] M. Kothapalli, "The Evolution of Component-Based Architecture in Front-End Development," *Journal of Scientific and Engineering Research,* vol. 8, no. 7, pp. 261-264, 2021.

[9] S. U. Meshram, "Evolution of Modern Web Services – REST API with its Architecture and Design," *International Journal of Research in Engineering, Science and Management,* vol. 4, no. 7, pp. 83-86, 2021.

[10] M. A. Novianto and S. Munir, "Analisis dan Implementasi RESTful API guna Pengembangan Sistem Informasi Akademik pada Perguruan Tinggi," *Jurnal Informatika Terpadu,* vol. 8, no. 1, pp. 47-61, 2022.

[11] A. S. Newton, S. March, N. D. Gehring, A. K. Rowe and A. D. Radomski, "Establishing a Working Definition of User Experience for eHealth Interventions of Self-reported User Experience Measures With eHealth Researchers and Adolescents: Scoping Review," *Journal of Medical Internet Research,* vol. 23, no. 12, p. e25012, 2021.