

BAB III METODE PENELITIAN

3.1 Desain Penelitian

Penelitian ini menggunakan pendekatan *Design and Development* (D&D). Menurut Richey and Klein (2014), *Design and Development* didefinisikan “*The systematic study of design, development and evaluation processes with the aim of establishing an empirical basis for the creation of instructional and non-instructional products and tools and new or enhanced models that govern their development*”. Dengan kata lain, D&D merupakan penelitian sistematis yang bertujuan untuk menghasilkan dasar yang dapat diuji secara nyata dalam menciptakan produk, alat maupun model yang baru atau ditingkatkan, melalui proses desain, pengembangan dan evaluasi. Metode ini dipilih karena sesuai dengan tujuan penelitian, yaitu bertujuan untuk mengembangkan produk aplikasi berbasis *website* yang dapat digunakan untuk mengembangkan sistem deteksi gambar asli berbasis *website* yang mengimplementasikan hasil komputasi ELA dengan arsitektur CNN. Gambar 3.1 merupakan prosedur penelitian menurut Ellis & Levy (2010).



Gambar 3. 1 Prosedur Penelitian (Ellis & Levy., 2010)

3.2 Identifikasi Masalah

Tahap identifikasi masalah dalam metodologi penelitian *Design and Development* (D&D) merupakan langkah awal yang bertujuan untuk mengidentifikasi kebutuhan sistem, permasalahan yang ada, serta persyaratan teknis yang harus dipenuhi sebelum proses perancangan dilakukan. Pada tahapan ini, peneliti menyadari bahwa perkembangan teknologi digital yang semakin pesat dalam memberikan kemudahan dalam memproduksi, mengedit, dan menyebarkan informasi visual. Namun, kemajuan ini justru memunculkan permasalahan baru, khususnya terkait manipulasi gambar. Gambar yang telah dimodifikasi dapat digunakan untuk tujuan negatif, seperti pencemaran nama

baik, penyebaran berita palsu, dan tindakan asusila. Hal seperti ini menjadi tantangan serius dalam menjaga keaslian informasi digital di era digital.

Dalam konteks forensik digital, deteksi terhadap gambar manipulasi memerlukan metode yang mampu mengidentifikasi perbedaan antara gambar asli dan manipulasi. Metode konvensional sering kali kurang efektif ketika berhadapan dengan manipulasi yang halus yang sulit dikenali oleh mata manusia. Berdasarkan permasalahan tersebut, diperlukan suatu sistem yang dapat mendeteksi keaslian gambar secara otomatis, akurat, dan efisien. Penelitian ini mengusulkan penggunaan *Error Level Analysis* (ELA) untuk mengungkap perbedaan tingkat kompresi pada gambar, yang kemudian dikombinasikan dengan *Convolutional Neural Network* (CNN) untuk melakukan klasifikasi apakah gambar tersebut asli atau hasil manipulasi. Sistem ini diharapkan dapat membantu proses identifikasi bukti digital pada kasus-kasus pencemaran nama baik maupun kasus asusila, sehingga dapat mendukung penegakan hukum dan mencegah penyalahgunaan teknologi.

3.3 Deskripsi Tujuan

Berdasarkan permasalahan yang sudah diidentifikasi pada poin sebelumnya didapati tujuan dari penelitian ini secara umum adalah merancang dan mengembangkan sebuah sistem untuk forensik citra digital yang dapat mendeteksi gambar manipulasi berbasis *website* yang memanfaatkan metode ELA dan CNN sehingga permasalahan yang telah dipaparkan dalam poin sebelumnya dapat diatasi. Adapun tujuan yang lebih rinci dalam penelitian ini sebagai berikut:

1. Merancang arsitektur sistem deteksi gambar manipulasi yang mengintegrasikan proses ekstraksi fitur menggunakan ELA dengan model klasifikasi CNN.
2. Mengimplementasikan sistem deteksi keaslian gambar dalam bentuk aplikasi berbasis *website* yang mudah digunakan.
3. Melakukan pengujian kinerja sistem menggunakan dataset gambar asli dan gambar manipulasi untuk memperoleh matrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *f1-score*.

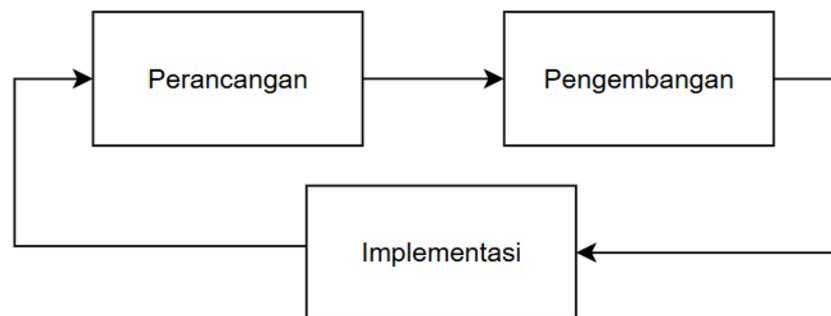
4. Mengevaluasi hasil pengujian untuk mengetahui performa sistem dalam mendeteksi manipulasi gambar.

3.4 Desain dan Pengembangan Sistem

Tahap ini membahas perancangan dan pengembangan sistem yang akan dilakukan dalam penelitian ini. Untuk memastikan proses pengembangan berjalan secara terstruktur. Tahap ini menentukan struktur, alur logika, serta komponen-komponen utama dari sistem deteksi keaslian citra digital yang akan dikembangkan. Desain akan membantu mengurangi ketergantungan antar komponen sistem, sehingga apabila terjadi kesalahan atau *bug* pada salah satu fitur, dampaknya tidak akan mempengaruhi keseluruhan sistem.

3.4.1 Metode Pengembangan Model

Pada tahap ini membahas perancangan atau kerangka kerja menyeluruh sistem *Artificial Intelligence* (AI) yang mencakup perancangan, pengembangan dan penerapan sistem AI seperti pada Gambar 3.2.

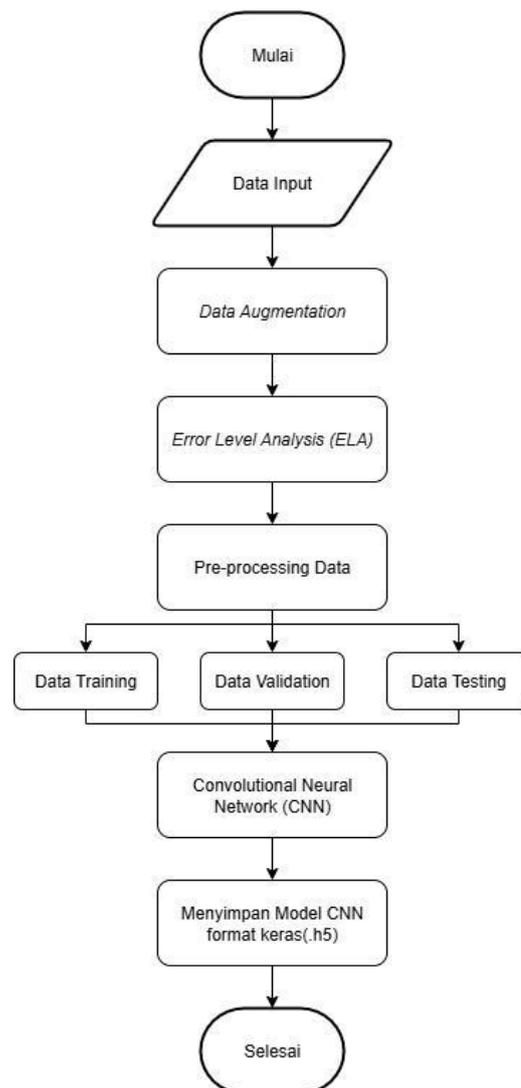


Gambar 3. 2 *Artificial Intelligence Life Cycle* (Silva, D. D., & Alahakoon, D., 2022).

Pada Gambar 3.2 metode perancangan model untuk sistem deteksi ada tiga tahapan dalam *Artificial Intelligence Life Cycle* (AILC) yaitu perancangan, pengembangan dan implementasi. Model yang diterapkan adalah *Convolutional Neural Network* (CNN) salah satu arsitektur pengolahan citra digital.

3.4.1.1. Perancangan Model

Perancangan arsitektur model dilakukan sebelum pengembangan sistem untuk memastikan struktur dan alur kerja yang akan diimplementasikan secara sistematis dan terstruktur. Gambar 3.3 menunjukkan diagram alir pengembangan model.



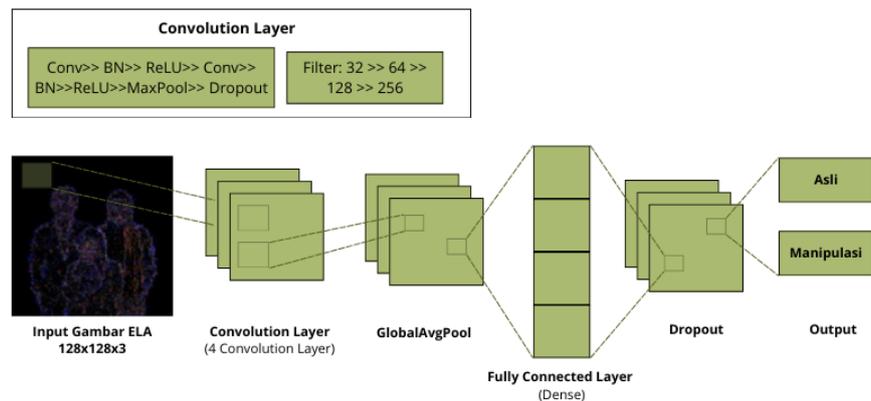
Gambar 3. 3 *Flowchart* Pengembangan Model

Flowchart pengembangan model pada 3.3 menggambarkan tahapan untuk membangun model dalam sistem deteksi manipulasi citra menggunakan gabungan metode ELA dan

CNN. Proses dimulai dengan tahap data *input*, di mana dataset akan digunakan untuk pelatihan model. Sebelum dilakukan pelatihan, dataset dilakukan augmentasi data untuk memperbanyak variasi dataset guna meningkatkan generalisasi model dan mengatasi keterbatasan jumlah data pelatihan.

Tahap berikutnya adalah ELA, yang berfungsi untuk mengidentifikasi area-area dalam citra yang kemungkinan telah mengalami manipulasi dengan menganalisis tingkat kesalahan kompresi JPEG. Hasil analisis ELA kemudian akan memasuki tahap *pre-processing* data, di mana citra dinormalisasi dan dipersiapkan dalam format yang sesuai untuk input model CNN. Dataset akan dibagi menjadi tiga bagian: data *training* untuk melatih model, data *validation* untuk *validasi* selama proses pelatihan, dan data *testing* untuk evaluasi performa model akhir. Proses akan dilanjutkan dengan pelatihan CNN untuk mempelajari pola-pola karakteristik dari citra hasil ELA. Setelah proses *training* selesai, model CNN akan disimpan dalam format keras (.h5) untuk kemudian diimplementasikan kedalam *website*.

Kemudian pengembangan arsitektur model CNN yang digunakan untuk klasifikasi keaslian citra digital seperti Gambar 3.4. Model ini dirancang untuk mengekstraksi fitur visual dari citra hasil ELA untuk membedakan antara gambar yang asli dan gambar yang telah dimanipulasi.



Gambar 3. 4 Arsitektur Model CNN

Gambar 3.4 proses dimulai dengan input berupa citra ELA berukuran 128×128 piksel dengan 3 *channel* warna RGB, yang mewakili spektrum warna dari gambar. Citra ini kemudian diproses melalui empat blok utama konvolusional untuk melakukan ekstraksi fitur secara bertahap, mulai dari pola dasar hingga pola kompleks.

Blok pertama terdiri dari dua lapisan konvolusi dengan 32 filter berukuran 3×3 , masing-masing diikuti oleh *batch normalization* untuk menstabilkan distribusi aktivasi. Setelah itu, dilakukan operasi *max pooling* untuk mereduksi dimensi spasial serta *dropout* sebesar 25% untuk mengurangi risiko *overfitting*. Blok kedua memiliki struktur serupa dengan jumlah filter yang ditingkatkan menjadi 64. Pola ini dilanjutkan pada blok ketiga dan keempat dengan jumlah filter masing-masing sebesar 128 dan 256. Setiap blok selalu terdiri atas dua lapisan konvolusi berurutan, *batch normalization*, *max pooling*, serta *dropout*, dengan tingkat *dropout* sebesar 25% pada blok ketiga dan blok keempat 30%. Penambahan jumlah filter pada setiap blok bertujuan untuk menangkap fitur yang lebih kompleks pada citra manipulasi. Setelah keempat blok konvolusi, arsitektur ini menggunakan *GlobalAveragePooling2D*, yang berfungsi untuk mereduksi data hasil konvolusi menjadi representasi vektor

yang lebih padat, sekaligus mengurangi jumlah parameter secara signifikan dibandingkan penggunaan *Flatten*. Hasil *polling* kemudian diteruskan ke *fully connected layer* dengan 256 unit *neuron* dan aktivasi *ReLU*. Di lapisan ini juga diterapkan *batch normalization* dan *dropout* sebesar 60% untuk mendukung generalisasi model. Akhirnya, lapisan *output* berupa *dense* dengan jumlah *neuron* sesuai dengan jumlah kelas (2 kelas: asli dan manipulasi) dan fungsi aktivasi *softmax* digunakan untuk menghasilkan probabilitas klasifikasi.

3.4.1.2. Pengembangan Model

Pengembangan model CNN yang dirancang untuk mengenali pola-pola tertentu pada gambar hasil ELA. Pengembangan model dilakukan secara bertahap, mencakup desain arsitektur, kompilasi model, serta proses pelatihan dan evaluasi.

1. Data Collection

Data yang digunakan dalam penelitian ini berasal dari CASIA, yaitu salah satu dataset standar yang banyak digunakan dalam penelitian forensik digital, khususnya untuk mendeteksi gambar hasil manipulasi. Dataset ini dikembangkan oleh *National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences*. Dataset CASIA menyediakan beragam gambar digital yang telah melalui proses manipulasi secara manual maupun digital dengan berbagai teknik, seperti *copy-move*, *splicing*, dan *retouching*. Setiap gambar dalam dataset dilengkapi dengan pasangan gambar asli dan gambar hasil manipulasi yang sangat mendukung kebutuhan pelatihan dan pengujian sistem deteksi manipulasi. Dalam penelitian ini penulis menggunakan dataset utama dari CASIA v2.0 yang mencakup dua kelas berjumlah 14.000 dengan

masing-masing gambar 7.000 dengan label Au (*authentic*) berisi gambar-gambar yang asli dan Tp (*tampered*) berisi gambar-gambar yang telah dimanipulasi.

2. *Data Exploration dan Augmentation*

Tahap eksplorasi data merupakan langkah awal yang dilakukan untuk memahami karakteristik dan struktur dataset yang akan digunakan dalam penelitian. Pada tahap ini, akan dilakukan pemeriksaan menyeluruh terhadap struktur folder, format file, distribusi jumlah gambar, serta kualitas data dari masing-masing kelas (Au dan Tp). Proses eksplorasi data dimulai dengan identifikasi berbagai format file yang mungkin tidak relevan untuk proses Error Level Analysis (ELA). Selanjutnya, akan dilakukan analisis distribusi data untuk memahami keseimbangan dataset antar kelas. Tahap ini juga mencakup proses pembersihan data dengan cara menyaring file-file yang tidak sesuai dengan kebutuhan penelitian dan mempertahankan hanya file gambar berformat JPG/JPEG yang kompatibel dengan metode ELA.

Kemudian data *augmentation* adalah teknik yang digunakan untuk memperbanyak jumlah data pelatihan dengan cara melakukan berbagai transformasi pada gambar-gambar. Tujuan utama dari teknik ini adalah untuk meningkatkan kemampuan generalisasi model, mengatasi ketidakseimbangan dataset, serta mengurangi risiko *overfitting*. Dalam penelitian ini, data *augmentation* akan diterapkan khususnya pada kelas yang memiliki jumlah sampel lebih sedikit untuk mencapai keseimbangan distribusi data antar kelas.

3. *Data Preprocessing dan Splitting*

Data preprocessing merupakan tahapan penting yang dilakukan sebelum data citra digunakan dalam pelatihan model *deep learning*. Tahapan ini bertujuan untuk mempersiapkan data agar berada dalam kondisi yang optimal, baik dari sisi ukuran, format, maupun nilai-nilai pikselnya. Tahapan *preprocessing* yang meliputi ELA, *resize image* dan normalisasi piksel. Proses ELA digunakan sebagai teknik awal untuk mengekstraksi fitur dari citra digital guna mendeteksi adanya manipulasi. ELA bekerja dengan menghasilkan gambar *lossy* berformat JPEG. Ketika sebuah gambar disimpan ulang dalam format JPEG dengan kualitas tertentu, bagian-bagian gambar yang telah dimanipulasi akan menunjukkan tingkat kesalahan yang berbeda dibandingkan dengan bagian asli.

- a) Konversi ke format JPEG, gambar asli disimpan ulang dalam format JPEG dengan kualitas kompresi sebesar 90%. Hal ini bertujuan untuk memunculkan perbedaan *error* antara bagian gambar yang telah dimanipulasi dan yang tidak.
- b) Perhitungan perbedaan, proses penyimpanan ulang gambar hasil kompresi dibandingkan dengan gambar asli menggunakan metode *ImageChops.difference()*, yang akan menghasilkan selisih pada gambar.
- c) Skalasi dan penyesuaian *brightness*, untuk memperjelas perbedaan *error*, hasil selisih pada gambar akan disesuaikan Tingkat kecerahannya menggunakan *ImageEnchance.Brightness()* dan dilakukan berdasarkan nilai perbedaan maksimum (*max_diff*).
- d) *Resize* dan normalisasi, gambar hasil ELA kemudian diubah ukurannya menjadi 128×128 piksel dan

dinormalisasikan dalam rentang $[0,1]$ agar dapat diproses oleh model CNN.

Setelah itu data memasuki proses *data splitting* yang bertujuan untuk membagi dataset dan memastikan evaluasi model yang objektif serta menghindari bias dalam performa. Dataset akan dibagi menjadi tiga subset yaitu *training*, *validation*, dan *test*.

Secara spesifik, 80% dari total data digunakan sebagai data pelatihan (*training set*). Data ini digunakan untuk membangun model, di mana CNN akan belajar mengenali pola-pola penting dari gambar hasil ELA, baik yang tergolong asli maupun manipulasi. Model berlatih dari data ini untuk menyesuaikan bobot-bobot internalnya agar dapat melakukan klasifikasi dengan akurasi tinggi.

Selanjutnya, 10% dari dataset dialokasikan sebagai data validasi (*validation set*). Data ini digunakan selama proses pelatihan untuk memantau performa model secara berkala, khususnya dalam mendeteksi *overfitting*. Jika performa model di data pelatihan meningkat namun justru menurun di data validasi, maka dapat disimpulkan bahwa model mulai kehilangan generalisasi, dan penyesuaian parameter perlu dilakukan.

Sisa 10% lainnya dialokasikan untuk pengujian akhir (*test set*). Dataset ini tidak pernah digunakan selama proses pelatihan maupun validasi, dan hanya digunakan satu kali pada tahap akhir untuk mengevaluasi sejauh mana model mampu mengenali gambar yang benar-benar baru. Ini memberikan gambaran akurat mengenai kemampuan generalisasi model terhadap data dunia nyata yang belum pernah dilihat sebelumnya.

4. *Training Model*

Tahap pelatihan model CNN merupakan proses inti dalam penelitian ini, di mana model akan dilatih menggunakan data *training* yang telah melalui tahapan preprocessing. Dalam proses pelatihan, konfigurasi *hyperparameter* menjadi aspek krusial yang harus ditentukan untuk mengoptimalkan performa pembelajaran model (Wiharja. S., dkk., 2024) *Hyperparameter* yang akan digunakan dalam pelatihan model meliputi pengaturan *optimizer*, *learning rate*, *fungsi loss*, dan berbagai parameter *control* pelatihan yang akan mempengaruhi proses konvergensi model.

3.4.1.3. Implementasi Model

Implementasi model merupakan tahapan akhir dalam metodologi penelitian yang bertujuan untuk mengintegrasikan model CNN yang telah dilatih ke dalam sistem aplikasi *website* yang dapat diakses oleh pengguna. Tahapan ini dirancang untuk menjembatani hasil penelitian dengan implementasi praktis yang dapat digunakan. Model CNN yang telah melalui proses pelatihan dan evaluasi akan disimpan dalam format Keras (.h5) yang memungkinkan penyimpanan struktur arsitektur jaringan beserta seluruh bobot yang telah dioptimalkan.

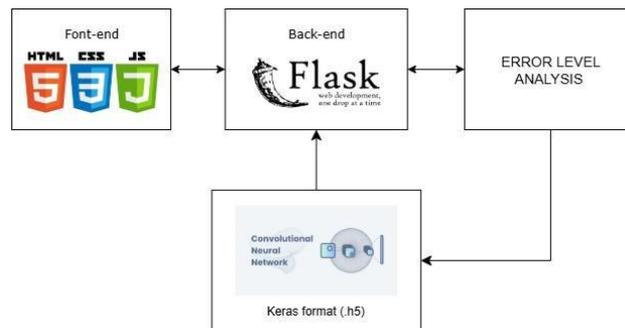
3.4.2 Metode Pengembangan Aplikasi

Pengembangan aplikasi dalam penelitian ini untuk memastikan semua komponen bekerja sesuai kebutuhan sistem. Proses ini mencakup pembuatan desain sistem menggunakan diagram arsitektur, *Use Case Diagram*, *Flowchart* Aplikasi, dan Desain sistem antarmuka pengguna.

3.4.2.1. Diagram Arsitektur Aplikasi

Rancangan diagram arsitektur berfungsi sebagai alat visual untuk memetakan struktur komponen dan hubungan antar bagian dalam suatu sistem. Diagram ini memudahkan proses

perencanaan, komunikasi, pengambilan Keputusan, serta dokumentasi dalam proses pengembangan dan pemeliharaan sistem secara efisien. Gambar 3.5 menunjukkan rancangan arsitektur aplikasi deteksi gambar asli berbasis *website* yang dikembangkan dalam penelitian ini.



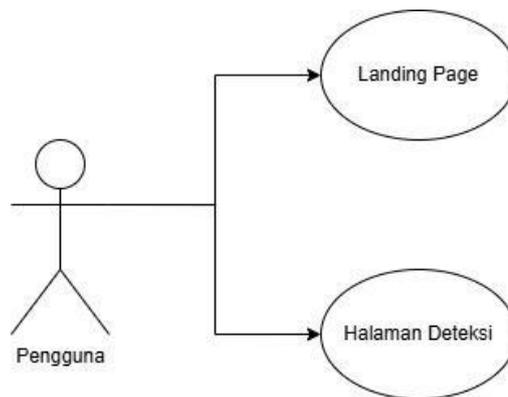
Gambar 3. 5 Arsitektur Sistem

Gambar 3.5 menunjukkan arsitektur sistem terdiri dari empat komponen utama, yaitu *front-end*, *back-end*, ELA, dan model (.h5). Pada bagian *front-end*, terdapat halaman utama dan halaman deteksi. *Front-end* dibangun menggunakan *HTML*, *CSS* dan *JavaScript*. Pada bagian *back-end*, sistem dikembangkan menggunakan *framework Flask (Python)* yang berfungsi sebagai pusat kendali dari seluruh proses sistem. *Back-end* bertanggung jawab untuk mengatur alur data termasuk *input* gambar dari pengguna, melakukan proses ELA untuk menghasilkan representasi citra, menjalankan proses klasifikasi gambar, dan mengelola interaksi pengguna melalui antarmuka *website*.

3.4.2.2. Use Case Diagram Aplikasi

Use case diagram adalah salah satu jenis diagram dalam *Unified Modeling Language (UML)* yang digunakan untuk menggambarkan interaksi antara pengguna (*actor*) dengan sistem dalam mencapai tujuan tertentu. Diagram ini menekankan pada fungsi-fungsi utama yang tersedia dalam

sistem dari sudut pandang pengguna akhir. Gambar 3.6 menunjukkan *use case* diagram dari aplikasi deteksi keaslian gambar berbasis *website* yang dikembangkan dalam penelitian ini.



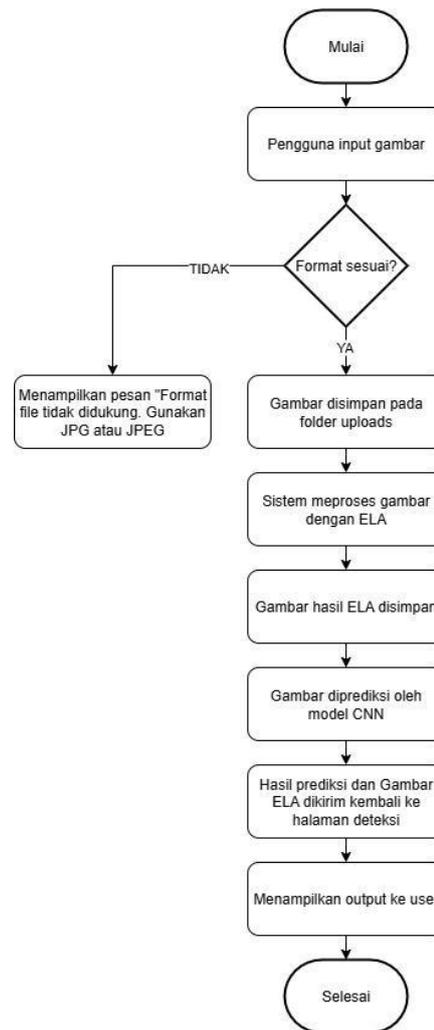
Gambar 3. 6 Use Case diagram sistem

Berikut merupakan penjelasan mengenai masing-masing *use case* pada *website* ini:

- a. Landing page merupakan halaman awal yang berisi informasi seperti, *Call To Action* (CTA) dan informasi pentingnya deteksi gambar.
- b. Halaman deteksi untuk pengguna dapat mengunggah gambar yang dipilih untuk dideteksi dan klasifikasi oleh model. Lalu akan muncul hasil klasifikasi dan gambar hasil ELA. Di halaman ini pengguna juga memungkinkan untuk mengunduh gambar hasil ELA.

3.4.2.3. *Flowchart* Aplikasi

Flowchart merupakan bentuk visualisasi dari alur proses atau algoritma yang ditunjukkan dengan simbol-simbol untuk merepresentasikan tahapan awal, proses, pengambilan keputusan, hingga akhir alur. Dengan adanya *flowchart*, setiap langkah dalam proses dapat dipahami lebih jelas karena tersusun secara runtut dan sistematis. Gambar 3.7 memperlihatkan *flowchart* keseluruhan sistem.

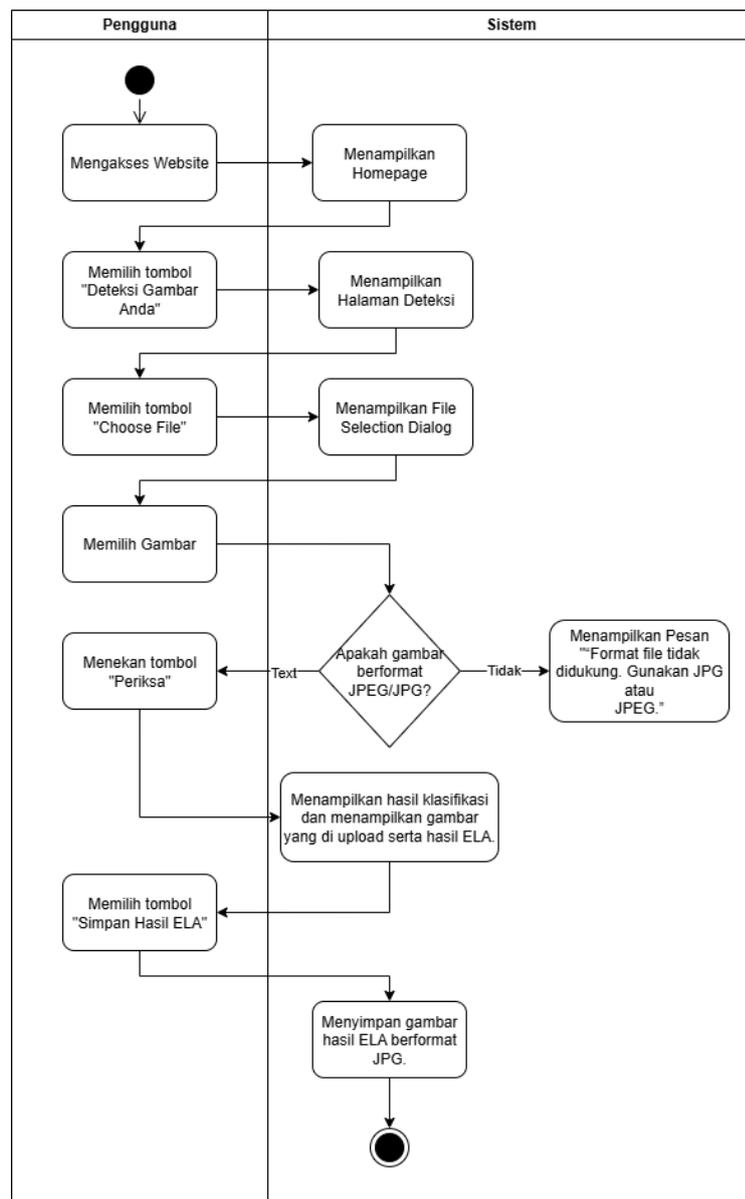


Gambar 3. 7 Flowchart sistem

3.4.2.4. Activity Diagram Aplikasi

Activity Diagram adalah salah satu jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan alur proses dalam sebuah sistem. Dalam sistem deteksi gambar asli dan manipulasi menggunakan *Error Level Analysis* (ELA) dan *Convolutional Neural Network* (CNN), Gambar 3.8 menunjukkan langkah-langkah mulai dari pengguna mengakses *website*, pengunggahan gambar oleh pengguna, proses analisis otomatis oleh sistem, sampai hasil deteksi yang ditampilkan kembali ke pengguna. Dengan

diagram ini, memperjelas bagaimana aktivitas-aktivitas tersebut saling berurutan dan berinteraksi, baik di sisi sistem maupun dari sisi pengguna, sehingga alur kerja sistem menjadi lebih mudah dipahami.



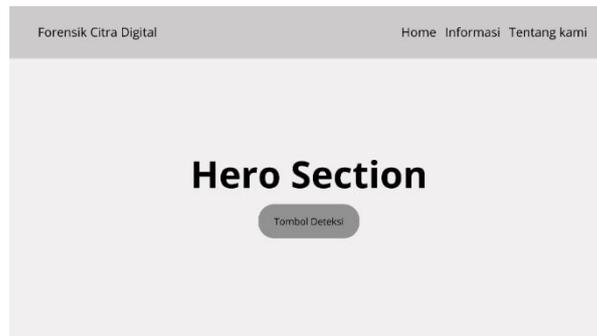
Gambar 3. 8 Activity Diagram Aplikasi

Pada Gambar 3.8 menunjukkan diagram aktivitas yang menggambarkan alur kerja sistem deteksi manipulasi gambar. Proses diawali dengan pengguna yang mengakses *website* dan

sistem menampilkan halaman utama (*homepage*). Selanjutnya untuk mendeteksi pengguna memilih tombol “Deteksi Gambar Anda” yang akan membawa pengguna ke halaman deteksi. Di halaman ini, pengguna menekan tombol “*Choose File*” untuk membuka dialog pemilihan file dari perangkat pengguna. Setelah memilih gambar, sistem memeriksa format file gambar tersebut apakah sudah sesuai dengan format yang didukung yaitu JPG dan JPEG. Jika format tidak sesuai, sistem akan menampilkan pesan “Format file tidak didukung. Gunakan format JPG atau JPEG”. Jika format gambar sesuai, pengguna dapat menekan tombol “Periksa” untuk memulai proses analisis gambar. Sistem kemudian melakukan klasifikasi gambar dan menampilkan hasil klasifikasi serta gambar hasil ELA kepada pengguna. Terakhir, pengguna memiliki opsi untuk menyimpan gambar hasil ELA dengan menekan tombol "Simpan Hasil ELA". Sistem akan menyimpan gambar tersebut dalam format JPG sebagai output akhir dari proses ini.

3.4.2.5. Desain Antarmuka Pengguna

Desain antarmuka pengguna (*User Interface*) pada aplikasi deteksi forensik citra digital ini dirancang dengan mengutamakan kesederhanaan, kemudahan penggunaan, dan kejelasan informasi. Pendekatan ini dilakukan agar pengguna, baik yang memiliki latar belakang teknis maupun non-teknis, dapat dengan mudah memahami cara kerja aplikasi dan memperoleh hasil deteksi secara cepat. Gambar 3.9 merupakan desain dari antarmuka halaman utama pada *website*.



Gambar 3. 9 Desain halaman utama

Pada Gambar 3.9 halaman utama terdapat beberapa *section*, salah satu yang menjadi bagian penting selain *hero section* adalah *section* informasi, yang terdapat informasi pentingnya forensik citra digital. Gambar 3.10 adalah desain halaman deteksi.



Gambar 3. 10 Desain halaman deteksi

Pada Gambar 3.10 halaman deteksi yang merupakan fitur utama dalam sistem ini. Pengguna dapat memasukkan gambar yang ingin diperiksa pada halaman ini.

3.5 Pengujian Sistem

Tahap pengujian dilakukan setelah seluruh proses perancangan dan pengembangan sistem selesai dilaksanakan. Tahap ini bertujuan untuk menguji dan menilai performa sistem secara keseluruhan. Pengujian sistem sangat penting untuk menentukan apakah hasil rancangan memenuhi tujuan penelitian. Pengujian dilakukan pada beberapa bagian sistem yaitu pengujian model yang bertujuan untuk mengukur kinerja algoritma deteksi gambar yang digunakan dalam aplikasi dan fungsionalitas *website*.

3.5.1 Pengujian Model

Tahap pengujian model dalam penelitian ini dilakukan untuk menilai sejauh mana model CNN mampu mengklasifikasikan antara gambar asli dan gambar yang telah mengalami manipulasi berdasarkan data uji. Dalam proses ini, digunakan *confusion matrix* sebagai alat utama dalam mengevaluasi performa klasifikasi. *Confusion matrix* merupakan sebuah tabel yang memperlihatkan perbandingan antara label hasil prediksi model dengan label sebenarnya. Melalui tabel ini, dapat diketahui jumlah prediksi yang tepat maupun yang keliru pada setiap kategori, sehingga memungkinkan analisis yang lebih detail terhadap kesalahan klasifikasi (Krstinić dkk, 2020). Berdasarkan informasi dari *confusion matrix*, dapat dihitung sejumlah metrik evaluasi penting guna mengetahui tingkat efektivitas model secara keseluruhan. Adapun penjelasan dari masing-masing komponen dan metrik evaluasi sebagai berikut.

1. Accuracy

Accuracy menunjukkan persentase prediksi yang benar terhadap keseluruhan data. Metrik ini memperhitungkan jumlah prediksi positif dan negatif. Berikut merupakan persamaan dalam menghitung *Accuracy*:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Keterangan:

- a. *True Positive* (TP). Jumlah prediksi benar untuk kelas positif
- b. *True Negative* (TN). Jumlah prediksi benar untuk kelas negatif
- c. *False Positive* (FP). Jumlah prediksi salah untuk kelas positif

d. *False Negative* (FN). Jumlah prediksi salah untuk kelas negatif

2. *Precision*

Precision mengukur ketepatan model dalam memprediksi kasus positif. Nilai *precision* yang tinggi berarti model jarang salah dalam mengklasifikasikan kasus positif. Berikut merupakan persamaan dalam menghitung *Precision*:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

3. *Recall*

Recall menunjukkan seberapa baik model dalam mendeteksi semua kasus positif. Berikut merupakan persamaan dalam menghitung *Recall*:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

4. *F1-score*

F1-score merupakan rata-rata harmonik dari *precision* dan *recall*, metrik ini memberikan gambaran seberapa baik model dalam menyeimbangkan antara *precision* dan *recall* secara bersamaan. *F1-score* digunakan untuk menilai kinerja model secara keseluruhan, khususnya dalam kondisi data yang tidak seimbang. Adapun rumus untuk menghitung *F1-score* adalah sebagai berikut:

$$F1 \text{ score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

Proses pengujian dilakukan menggunakan 100 gambar uji yang terdiri dari dua kelas, yakni 50 gambar asli dan 50 gambar hasil manipulasi. Setelah semua gambar diprediksi oleh model, hasilnya akan dianalisis menggunakan matrik *accuracy*. *Accuracy* dihitung dengan membandingkan jumlah prediksi yang benar dengan total jumlah sampel uji. Metrik ini digunakan

sebagai indikator utama untuk mengevaluasi sejauh mana model mampu membedakan gambar asli dari gambar manipulasi secara akurat.

3.5.2 Pengujian Fungsionalitas *Website*

Tahap pengujian fungsionalitas pada sistem berbasis *website* yang dikembangkan dalam penelitian ini dilakukan dengan menggunakan metode *black box*. Pengujian ini difokuskan pada pemeriksaan fungsional dari setiap komponen antarmuka tanpa melihat struktur internal program atau kode yang digunakan. Pendekatan ini digunakan untuk menilai apakah setiap fitur dalam aplikasi dapat berjalan sesuai dengan kebutuhan dan spesifikasi sistem yang telah ditetapkan. Tujuan dari pengujian ini adalah untuk memastikan bahwa sistem mampu merespon input dari pengguna secara tepat dan memberikan *output* yang sesuai dengan harapan. Seluruh pengujian difokuskan pada interaksi pengguna dengan fitur-fitur utama, seperti *input* gambar, proses klasifikasi menggunakan model, serta tampilan hasil deteksi. Pengujian dilakukan dengan mendesain serangkaian skenario yang mencerminkan penggunaan sistem secara nyata. Setiap skenario terdiri dari uji kasus, langkah atau aksi pengguna, *output* yang diharapkan, dan hasil aktual yang ditampilkan oleh sistem. Hasil dari tiap pengujian dicatat, baik dalam kondisi berhasil maupun gagal, untuk kemudian dievaluasi apakah sistem telah berfungsi sesuai dengan rancangan awal. Berikut merupakan format instrumen pengujian *black box* yang digunakan.

Tabel 3. 1 Instrumen pengujian fungsionalitas dengan metode *blackbox*

| No | Uji Kasus | Skenario | Hasil Yang Diharapkan |
|----|---------------------|------------------------------|---------------------------|
| 1 | Menekan tombol home | Pengguna menekan tombol home | Kembali ke halaman utama. |

| No | Uji Kasus | Skenario | Hasil Yang Diharapkan |
|----|---|---|--|
| 2 | Menekan tombol informasi | Pengguna menekan tombol informasi | Halaman <i>scrolling</i> pada <i>section</i> informasi. |
| 3 | Menekan tombol tentang kami | Pengguna menekan tombol tentang kami | Halaman <i>scrolling</i> pada <i>section</i> tentang kami. |
| 4 | Menekan tombol “Deteksi Gambar Anda” | Pengguna menekan tombol “Deteksi Gambar Anda” | Sistem mengarahkan ke halaman deteksi. |
| 5 | Proses upload gambar | Pengguna menekan tombol “ <i>Choose File</i> ” | Sistem mengarahkan pengguna ke perangkat pengguna. |
| 6 | Proses deteksi gambar berformat .jpeg atau .jpg | Pengguna menekan tombol “Periksa” | Sistem menampilkan hasil klasifikasi (asli atau manipulasi) dan menampilkan gambar upload serta hasil ELA. |
| 7 | Proses simpan gambar hasil ELA | Pengguna menekan tombol “Simpan Hasil ELA” | Sistem menyimpan gambar hasil ELA berformat JPG. |
| 8 | Proses deteksi gambar berformat .png | Pengguna memasukkan gambar dan menekan tombol “Periksa” | Sistem menampilkan “Format file tidak didukung. Gunakan JPG atau JPEG.” |

| No | Uji Kasus | Skenario | Hasil Yang Diharapkan |
|----|---------------------------------|--|---|
| 9 | Proses Kembali ke halaman utama | Pengguna menekan tombol “Kembali ke Halaman Utama” | Sistem mengarahkan pengguna ke halaman utama. |

3.6 Evaluasi Sistem

Tahap evaluasi sistem dilakukan untuk menilai kinerja dan keandalan aplikasi deteksi forensik citra digital. Evaluasi ini tidak hanya berfokus pada performa model deteksi yang memanfaatkan kombinasi *Error Level Analysis* (ELA) dan *Convolutional Neural Network* (CNN), tetapi juga pada aspek fungsionalitas dan kemudahan penggunaan aplikasi berbasis *Flask* yang telah dibangun. Evaluasi teknis dilakukan dengan menguji kinerja model CNN pada dataset uji yang berbeda dari dataset pelatihan. Beberapa metrik yang digunakan dalam evaluasi ini meliputi akurasi, presisi, *recall*, dan *F1-score*. Penggunaan *confusion matrix* akan digunakan dalam menganalisis seberapa baik model mampu membedakan antara citra asli dan citra manipulasi.

3.7 Penyampaian Hasil Pengujian

Setelah melakukan perancangan, pengembangan, pengujian dan evaluasi sistem, tahap selanjutnya adalah penyampaian hasil pengujian. Tahap ini mencakup penyusunan laporan hasil penelitian dalam bentuk skripsi untuk kemudian dikomunikasikan dalam sidang skripsi kepada dosen penguji. Dengan melakukan tahap ini, hasil dari penelitian yang telah dilakukan dapat terdokumentasi sehingga diharapkan dapat memberikan informasi bagi perkembangan ilmu forensik digital dan *deep learning*.