

BAB III

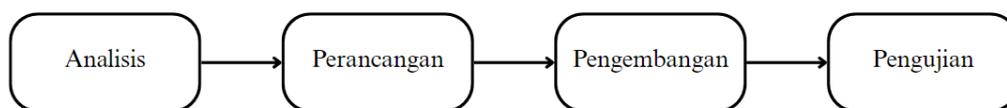
METODE PENELITIAN

3.1 Desain Penelitian

Penelitian ini menggunakan metode *Design and Development* (D&D). Metode D&D merupakan metode penelitian yang berfokus pada tiga tahap utama, yaitu perancangan (*design*), pengembangan (*development*), dan evaluasi sistem untuk menyelesaikan masalah spesifik dalam konteks nyata (Richey & Klein, 2014). Metode ini dipilih karena sesuai dengan tujuan penelitian, yaitu mengembangkan aplikasi *chatbot* berbasis *website* yang mengintegrasikan model *transfer learning* EfficientNetB3 dan GPT-4o.

3.2 Prosedur Penelitian

Gambar 3.1 menunjukkan prosedur penelitian ini yang dibagi menjadi beberapa tahapan. Pertama adalah tahap analisis untuk menganalisis kebutuhan sistem. Selanjutnya tahap perancangan untuk merancang struktur sistem berdasarkan hasil analisis kebutuhan. Selanjutnya tahap Pengembangan untuk mengimplementasikan rancangan yang telah dibuat menjadi sebuah sistem yang berfungsi. Selanjutnya tahap pengujian untuk memastikan bahwa sistem yang dikembangkan telah berfungsi sesuai dengan spesifikasi dan memenuhi kebutuhan yang telah diidentifikasi.



Gambar 3.1 Tahapan penelitian dengan menggunakan metode D&D (Richey & Klein, 2014)

3.3 Tahap Analisis

Tahap analisis dalam metodologi penelitian *Design and Development* (D&D) adalah fase awal yang bertujuan untuk mengidentifikasi kebutuhan, masalah, dan persyaratan sistem sebelum merancang solusi. Pada tahap ini, peneliti melakukan eksplorasi mendalam terhadap konteks permasalahan, keterbatasan solusi yang sudah ada, serta kebutuhan pengguna dan *technical requirements* yang harus dipenuhi. Untuk memenuhi kebutuhan pengguna dan kebutuhan teknis,

peneliti melakukan dua tahapan yaitu studi literatur dan eksplorasi laman resmi OpenAI.

Tahap ini diawali dengan studi literatur berbagai penelitian terdahulu mengenai penerapan model EfficientNetB3 untuk klasifikasi gambar kanker kulit dan literatur terkait performa *Large Language Model* (LLM). Dari hasil studi, ditemukan bahwa solusi yang sudah ada masih dapat dikembangkan, khususnya pada aspek penjelasan interaktif dalam aplikasi deteksi kanker kulit berbasis AI. Hal ini mendorong perlunya pengembangan fitur penjelasan yang lebih komprehensif agar hasil klasifikasi dapat dipahami dengan lebih baik oleh tenaga medis maupun pasien.

Tahap selanjutnya, membaca dokumentasi mengenai model GPT-4o melalui laman resmi OpenAI. Hal tersebut dilakukan guna memahami kapabilitas dari model GPT-4o dalam melakukan tugas *text generation* dan *image analysis*. Selain memahami kapabilitas model, informasi seperti batasan jumlah *token*, harga setiap *token*, mekanisme respons, serta cara penggunaannya menjadi landasan dalam merancang teknik *prompt engineering* yang tepat. Misalnya, pemberian struktur instruksi yang jelas dan penambahan konteks yang spesifik. Dengan demikian, kedua metode ini dapat membentuk kerangka analisis yang komprehensif, baik dari sisi teoretis maupun praktis.

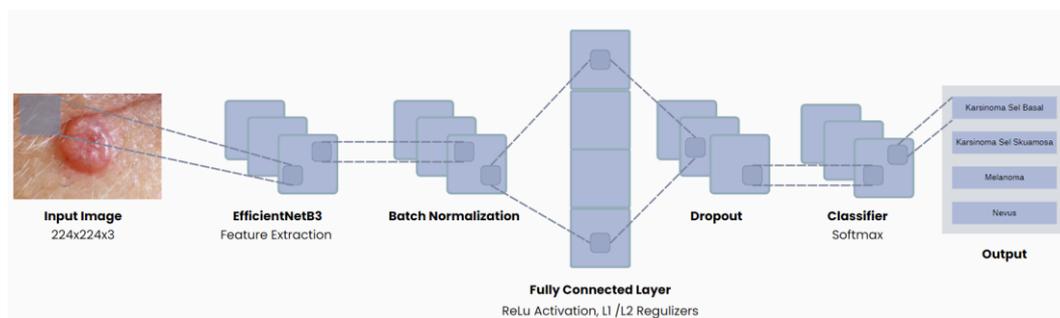
3.4 Tahap Perancangan

Tahap perancangan memainkan peran penting dalam mendefinisikan alur logika, struktur, dan interaksi dari sistem yang akan dibangun. Desain yang baik akan membantu meminimalkan ketergantungan antar komponen sistem, sehingga jika terjadi kesalahan atau *bug* pada salah satu fitur, dampaknya tidak akan mempengaruhi keseluruhan sistem. Tahap perancangan ini bertujuan untuk merinci spesifikasi yang telah diidentifikasi pada tahap analisis kebutuhan, agar proses pengembangan dapat berjalan lancar.

3.4.1 Perancangan Arsitektur Model

Gambar 3.2 menunjukkan perancangan arsitektur model yang akan digunakan untuk klasifikasi kanker kulit dengan *EfficientNetB3* sebagai *feature extraction*. Proses dimulai dengan gambar *input* berukuran $224 \times 224 \times 3$, yang

mewakili ukuran gambar dan *channel* warna RGB dari *input*. Selanjutnya, gambar tersebut melewati tahap *feature extraction* menggunakan EfficientNetB3, di mana model ini secara otomatis mengekstrak fitur-fitur penting seperti tekstur, warna, dan pola dari lesi kulit. Setelah fitur diekstraksi, model menerapkan *batch normalization* untuk menstabilkan distribusi nilai dalam jaringan, yang membantu mempercepat konvergensi selama pelatihan. Setelah *batch normalization*, fitur yang telah dinormalisasi diteruskan ke *fully connected layer*, yang berisi unit-unit neuron dengan ReLU *activation* serta *regulizer* L1/L2 untuk mencegah *overfitting*. Model ini juga menggunakan lapisan *dropout* untuk lebih meningkatkan generalisasi dengan mengurangi ketergantungan berlebih pada beberapa neuron untuk mencegah *overfitting*. Akhirnya, lapisan *classifier* dengan fungsi *softmax* digunakan untuk mengklasifikasikan gambar ke dalam empat kategori, yaitu karsinoma sel basal, karsinoma sel skuamosa, melanoma, dan *nevus*.



Gambar 3.2 Perancangan arsitektur model

3.4.2 Perancangan *Prompt*

3.4.2.1 GPT-4o *Vision*

Prompt yang diimplementasikan dalam penelitian ini dirancang dengan pendekatan terstruktur dan spesifik untuk mengoptimalkan kinerja model GPT-4o *Vision* untuk memvalidasi *input* gambar yang diberikan pengguna terindikasi kanker kulit atau tidak. Struktur *prompt* yang dikembangkan berdasarkan praktik terbaik dalam *prompt engineering*, berikut merupakan struktur *prompt* yang digunakan.

1. Mendefinisikan *role* yang spesifik. Model diminta berperan sebagai spesialis citra dermatologi yang terlatih untuk menganalisis lesi kulit.

2. Mendefinisikan *rules* yang jelas. Model diminta untuk mengikuti aturan spesifik dalam menganalisis *input* gambar. Berikut merupakan *rules* yang telah ditentukan.
 - a. Jika gambar menunjukkan kemungkinan kanker kulit (dengan ciri-ciri spesifik seperti batas tidak teratur, variasi warna, pertumbuhan cepat) atau lesi jinak yang memerlukan pemantauan, respons adalah '*valid*'.
 - b. Jika gambar jelas menunjukkan luka/cedera, jaringan parut, atau kondisi kulit yang tidak terkait dengan kanker kulit (seperti jerawat, eksim, psoriasis, kutil), atau jika gambar tidak relevan atau kurang detail, respons adalah '*invalid*'.
 - c. Jika gambar ambigu seperti *early-stage lesion*, gambar resolusi rendah, atau tidak memberikan informasi yang cukup untuk menentukan dengan yakin, respons adalah '*uncertain*'.
3. Mendefinisikan format respons. Secara eksplisit model diminta untuk mengembalikan respons hanya dengan 1 kata antara '*valid*', '*invalid*', atau '*uncertain*'.

3.4.2.2 Chatbot GPT-4o

Dalam penelitian ini, *prompt chatbot* GPT-4o diimplementasikan menggunakan pendekatan yang terstruktur. Struktur yang diterapkan mencakup teknik formulasi pertanyaan yang spesifik serta penyusunan instruksi yang jelas untuk mengoptimalkan kinerja model GPT-4o dalam menghasilkan respons yang lebih relevan dan bermakna. Berikut merupakan struktur *prompt* yang digunakan.

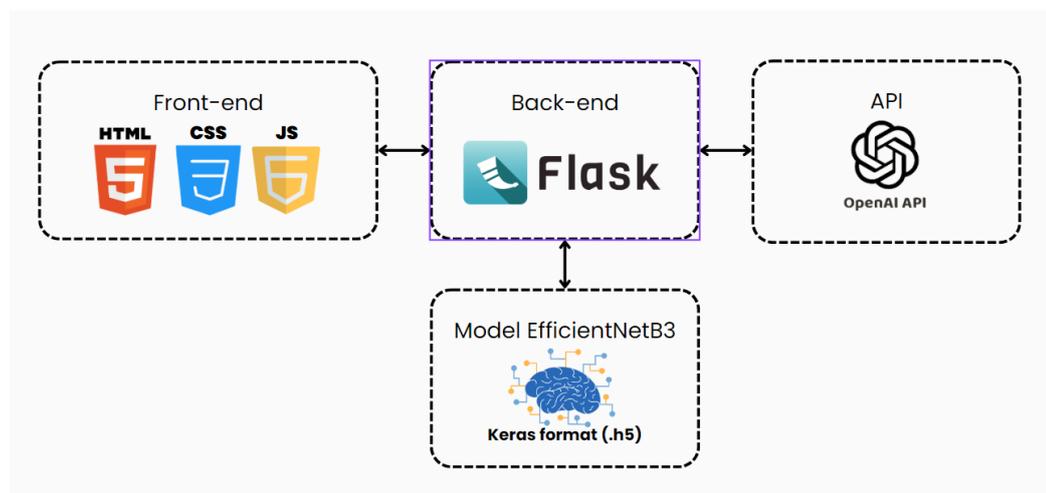
1. Mengambil hasil deteksi sehingga sistem dapat memberikan saran sesuai dengan keadaan pengguna.
2. Mendefinisikan *role* yang spesifik. Model diminta untuk berperan sebagai asisten medis ahli yang berspesialisasi dalam kanker kulit, khususnya karsinoma sel basal, karsinoma sel skuamosa, melanoma, dan *nevus*.
3. Mendefinisikan *guidelines* yang jelas. Model diminta untuk mengikuti aturan spesifik dalam menjawab pertanyaan dari pengguna. Berikut merupakan *guidelines* yang telah ditentukan.

- a. Memberikan informasi yang jelas dan berbasis bukti mengenai gejala kanker kulit, faktor risiko, pencegahan, dan deteksi dini.
 - b. Dorong pengguna untuk mencari evaluasi medis profesional untuk setiap masalah terkait lesi kulit atau perubahan pada kulit mereka.
 - c. Hindari merekomendasikan perawatan, selalu arahkan pengguna ke penyedia layanan kesehatan yang berkualifikasi untuk evaluasi lebih lanjut.
 - d. Ingat dan rujuklah hasil deteksi sebelumnya dan riwayat percakapan jika relevan.
 - e. Berikan konteks mengenai jenis kanker kulit spesifik yang terdeteksi dan karakteristik umum dari jenis tersebut.
 - f. Saat membahas hasil deteksi, selalu sebutkan bahwa hasil tersebut berasal dari model AI dan perlu dikonfirmasi oleh dokter spesialis kulit.
 - g. Menolak dengan sopan untuk menjawab pertanyaan yang tidak terkait dengan deteksi kanker kulit.
4. Mendefinisikan tujuan. Secara eksplisit, model diminta untuk memberikan informasi yang dapat meningkatkan kesadaran pengguna tentang kanker kulit, serta berikan rekomendasi untuk berkonsultasi dengan tenaga medis profesional guna mendapatkan diagnosis yang akurat dan penanganan yang tepat.
 5. Menyimpan pertanyaan pengguna dan respons *chatbot* sehingga nanti sistem dapat mengambil informasi tersebut untuk memberikan *output* yang sesuai dengan konteks yang sedang dibicarakan.

3.4.3 Perancangan Aplikasi

3.4.3.1 Diagram Arsitektur Aplikasi

Rancangan diagram arsitektur berfungsi sebagai alat visual untuk memetakan struktur, komponen, dan hubungan dalam suatu sistem. Ini memudahkan perencanaan, komunikasi, pengambilan keputusan, dan dokumentasi, sehingga mendukung pengembangan dan pemeliharaan sistem secara efisien. Gambar 3.3 menunjukkan rancangan arsitektur aplikasi berbasis *website*.



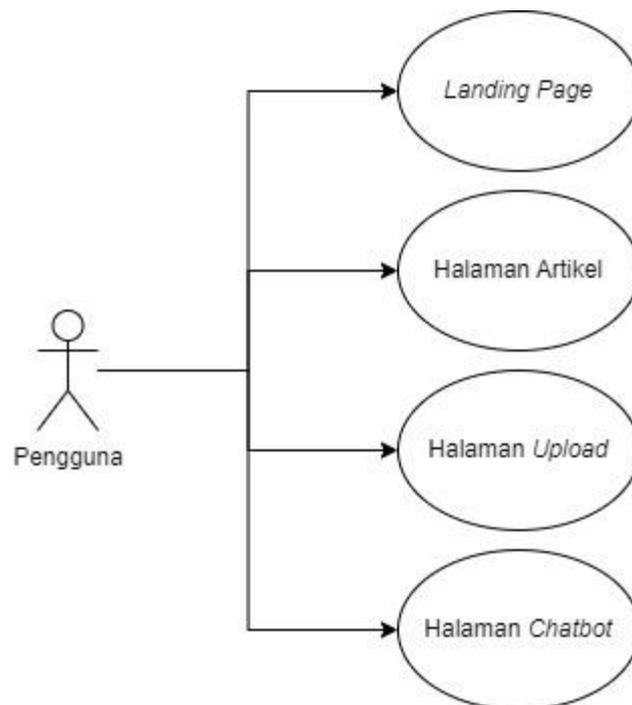
Gambar 3.3 Diagram arsitektur aplikasi berbasis *website*

Arsitektur aplikasi *website* ini terdiri dari empat komponen utama, yaitu *front-end*, *back-end*, model klasifikasi gambar dalam format Keras (.h5), dan API GPT-4o. Pada bagian *front-end*, terdapat empat halaman utama, yaitu *landing page*, halaman artikel, halaman *upload*, dan halaman *chatbot*, yang dibangun menggunakan HTML, CSS, dan JavaScript.

Back-end yang menggunakan Flask bertanggung jawab untuk memproses seluruh operasi sistem, termasuk *upload* gambar, klasifikasi kanker kulit, dan interaksi *chatbot*. Selain itu *Back-end* juga digunakan untuk mengelola komunikasi antara *front-end* dan API GPT-4o. Melalui integrasi ini, pengguna tidak hanya memperoleh hasil klasifikasi gambar kanker kulit, tetapi juga mendapatkan informasi medis tambahan melalui *chatbot* AI yang didukung oleh teknologi GPT-4o.

3.4.3.2 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 (Pooley & King, 1999). Gambar 3.4 menunjukkan *use case diagram* aplikasi berbasis *website* ini.



Gambar 3.4 *Use case diagram* aplikasi

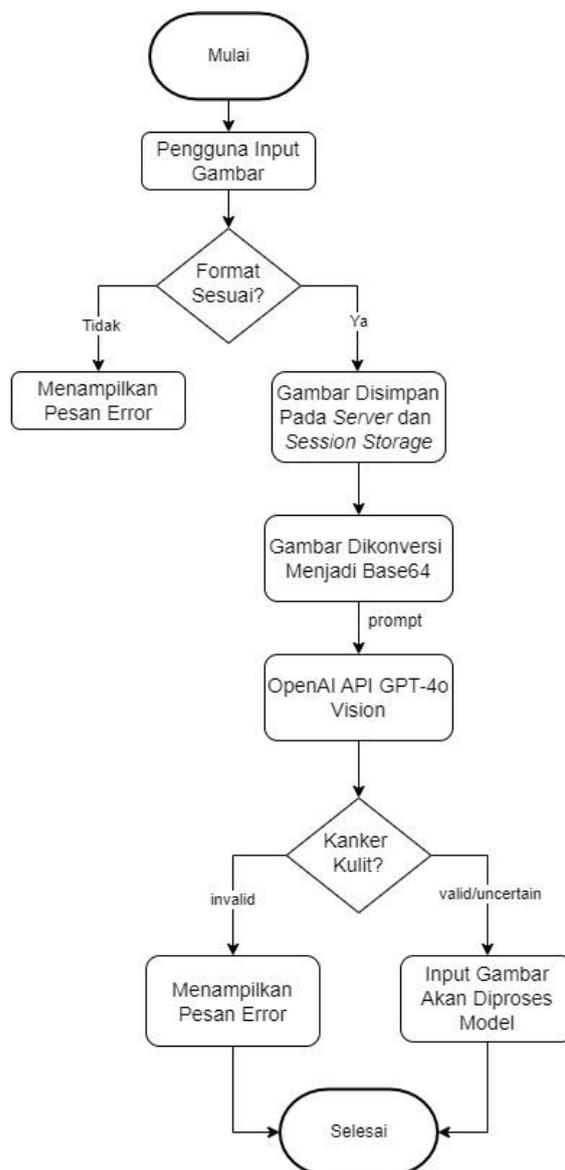
Berikut merupakan penjelasan lebih lanjut mengenai masing-masing *use case* pada aplikasi ini:

- a. *Landing page* merupakan halaman awal yang berisi informasi seperti, *Call To Action* (CTA), pentingnya deteksi kanker kulit, alur kerja *chatbot*, artikel mengenai kanker kulit, serta *Frequently Asked Questions* (FAQ).
- b. Halaman artikel menyajikan berbagai artikel edukatif terkait kanker kulit, metode pencegahan, dan informasi kesehatan lainnya.
- c. Halaman *upload* memungkinkan pengguna untuk mengunggah gambar kanker kulit yang akan divalidasi sebelum diproses lebih lanjut.
- d. Halaman *chatbot* menyediakan layanan konsultasi berbasis model GPT-4o setelah hasil klasifikasi gambar tersedia.

3.4.3.3 *Flowchart* Aplikasi

Flowchart merupakan representasi visual dari suatu aliran proses atau algoritma yang digambarkan menggunakan simbol-simbol untuk menunjukkan tahap awal, tahap proses, keputusan, serta akhir suatu alur (Chaudhuri, 2020). Dengan menggunakan *flowchart*, langkah-langkah dalam suatu proses menjadi

lebih mudah dipahami karena tersaji secara urut dan sistematis. Gambar 3.5 menunjukkan *flowchart* validasi *input* dengan GPT-4o Vision.



Gambar 3.5 *Flowchart* validasi *input* dengan GPT-4o Vision

Flowchart tersebut menjelaskan tahapan penerimaan dan pemrosesan gambar oleh sistem untuk memvalidasi *input* terindikasi kanker kulit atau tidak. Pertama, pengguna mengunggah gambar yang akan dicek kesesuaiannya dengan format yang didukung yaitu JPG, JPEG, dan PNG. Jika format tidak valid, sistem langsung menampilkan pesan *error*. Namun, jika formatnya valid, gambar akan disimpan pada server dan *session storage*, setelah itu gambar diubah menjadi format

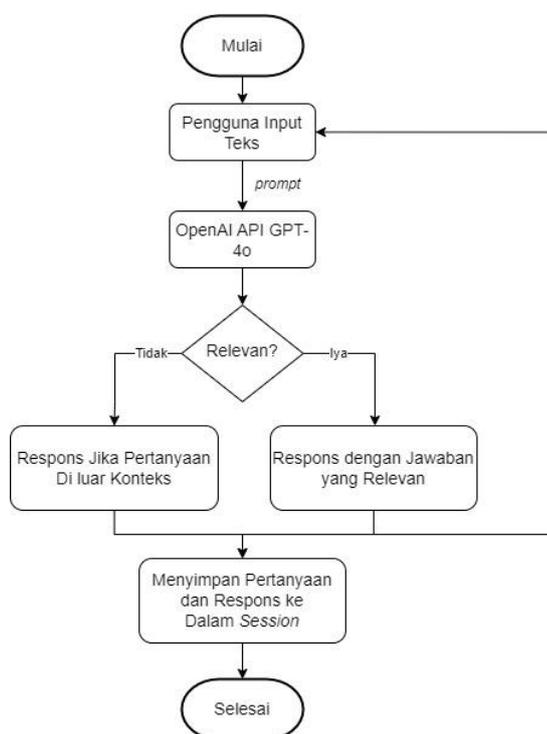
Base64 lalu dikirimkan sebagai *prompt* ke API GPT-4o *Vision* untuk memvalidasi gambar apakah ada indikasi kanker kulit atau tidak. Jika terdeteksi tidak ada indikasi kanker kulit, maka ditampilkan pesan *error*. Sebaliknya, bila hasil deteksinya *valid* atau *uncertain*, sistem akan melanjutkan pemrosesan lebih lanjut melalui model klasifikasi gambar.

Gambar 3.6 menunjukkan *flowchart deteksi* gambar dengan EfficientNetB3. Proses dimulai dengan pengambilan gambar dari server, kemudian dilanjutkan dengan tahapan pra-pemrosesan data agar gambar sesuai dengan format yang dibutuhkan oleh model. Setelah itu, gambar diproses oleh model EfficientNetB3 untuk mendeteksi jenis kanker kulit. Hasil deteksi tersebut disimpan ke dalam *session* agar *chatbot* dapat memiliki konteks deteksi. Setelah hasil disimpan, data gambar yang ada di server akan dihapus. Selanjutnya, sistem akan mengirimkan notifikasi kepada pengguna bahwa gambar telah berhasil diproses. Terakhir, pengguna akan diarahkan ke halaman *chatbot* untuk melakukan interaksi lanjutan.



Gambar 3.6 *Flowchart* deteksi gambar dengan EfficientNetB3

Gambar 3.7 menunjukkan *flowchart chatbot* yang memanfaatkan OpenAI API GPT-4o untuk memproses dan merespons *input* teks dari pengguna. Pertama, pengguna memasukkan teks yang kemudian dikirimkan sebagai *prompt* ke GPT-4o. Selanjutnya, sistem akan menilai apakah pertanyaan atau pernyataan tersebut relevan dengan konteks yaitu mengenai kanker kulit. Jika pertanyaan dianggap tidak relevan atau berada di luar konteks, maka *chatbot* akan menyampaikan bahwa pertanyaan tersebut tidak dapat dijawab karena di luar cakupan. Namun, apabila pertanyaannya relevan, *chatbot* akan memberikan respons berisi jawaban yang sesuai dengan pertanyaan yang dikirimkan. Setiap pertanyaan dan respons *chatbot* akan disimpan ke dalam *session* agar *chatbot* dapat mengingat konteks percakapan sebelumnya.

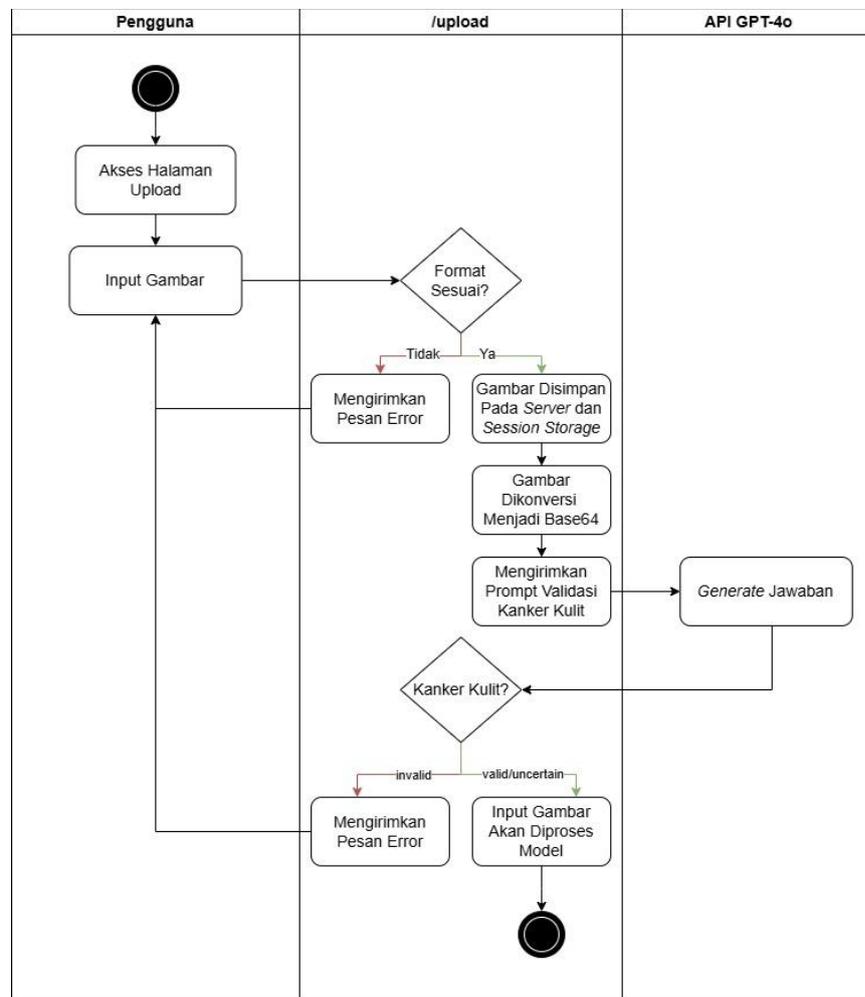


Gambar 3.7 *Flowchart chatbot*

3.4.3.4 *Activity Diagram Aplikasi*

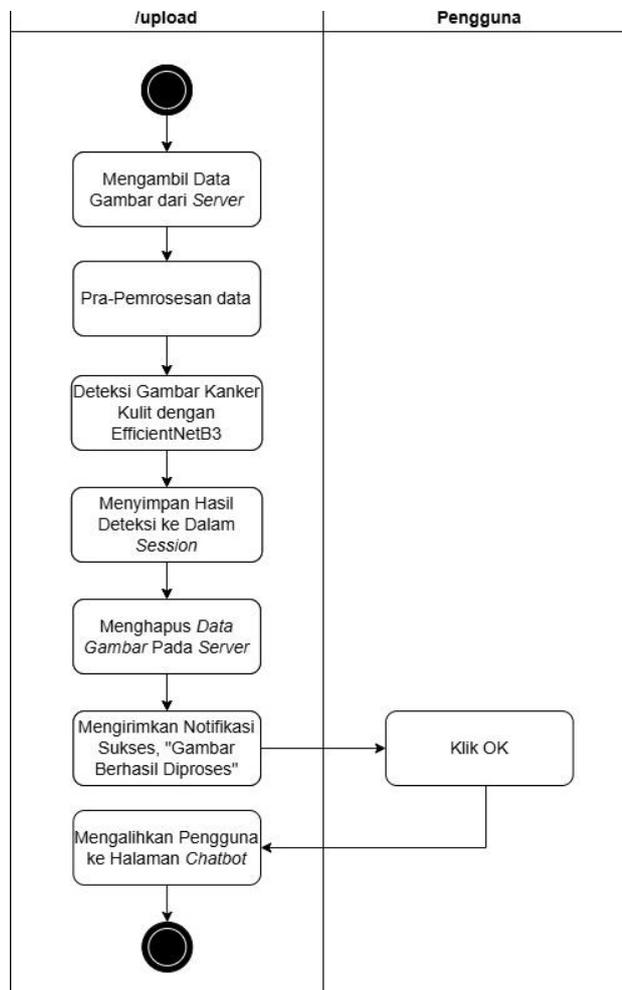
Activity diagram adalah salah satu jenis diagram dalam *Unified Modeling Language* (UML) yang memvisualisasikan alur kerja atau aktivitas dalam sebuah sistem secara terstruktur (Pooley & King, 1999). Diagram ini menyoroti rangkaian tindakan, pengambilan keputusan, dan kemungkinan percabangan proses, serta

memetakan kondisi awal hingga kondisi akhir dalam suatu proses bisnis atau aliran kerja. Gambar 3.8 menunjukkan *activity diagram* proses validasi *input* di halaman *upload* yang menunjukkan langkah-langkah yang terjadi ketika pengguna mengunggah gambar pada halaman *upload*. Dimulai dari pengguna mengakses halaman tersebut, kemudian memasukkan gambar yang ingin dianalisis. Sistem selanjutnya memeriksa apakah format *file* sesuai ketentuan, jika tidak, proses berhenti dan pesan *error* dikirimkan ke pengguna. Namun, jika format *file* sesuai, gambar akan disimpan pada *server* dan *session storage* lalu dikonversi ke bentuk Base64 sebelum dikirimkan ke API GPT-4o untuk divalidasi apakah mengindikasikan kanker kulit atau tidak. Jika dinyatakan *invalid*, pengguna mendapat pesan *error*. Namun, jika *valid* atau *uncertain*, proses dilanjutkan ke model klasifikasi gambar untuk menentukan kelas kanker kulit.



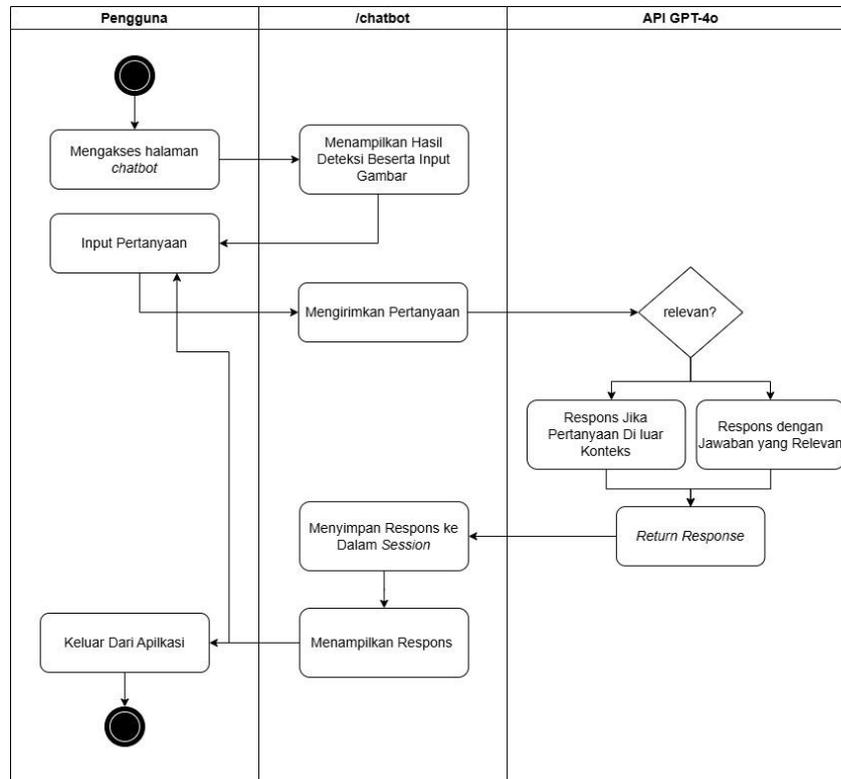
Gambar 3.8 *Activity diagram* validasi *input*

Gambar 3.9 menunjukkan *activity diagram* deteksi kanker kulit dengan model EfficientNetB3. Proses diawali ketika sistem mengambil data gambar dari server. Setelah itu, sistem akan melakukan pra-pemrosesan untuk menyiapkan data agar sesuai dengan kebutuhan model deteksi. Tahap berikutnya adalah mendeteksi jenis kanker kulit pada gambar menggunakan model EfficientNetB3. Setelah proses deteksi selesai, hasilnya akan disimpan ke dalam *session* agar *chatbot* memiliki pengetahuan mengenai konteks hasil deteksi kanker kulit. Setelah gambar diproses, gambar kemudian dihapus dari *server*. Setelah semua tahapan selesai, sistem mengirimkan notifikasi kepada pengguna berupa pesan “Gambar Berhasil Diproses” sebagai tanda bahwa proses telah berjalan dengan sukses. Pengguna kemudian cukup menekan tombol OK untuk beralih ke halaman *chatbot* untuk berkonsultasi lebih lanjut mengenai hasil deteksi dengan LLM GPT-4o.



Gambar 3.9 Activity diagram deteksi kanker kulit dengan EfficientNetB3

Gambar 3.10 menunjukkan *activity diagram* halaman *chatbot* yang menampilkan alur kerja antara pengguna dan sistem pada halaman *chatbot*.



Gambar 3.10 Activity diagram halaman chatbot

Pengguna memulai dengan melihat hasil deteksi jenis kanker kulit, lalu mengirimkan pertanyaan yang dikirimkan sebagai *prompt* ke *endpoint* GPT-4o. Setelah itu, GPT-4o memproses *input* tersebut dan memeriksa relevansinya dengan konteks yang sudah ditetapkan. Jika tidak relevan, GPT-4o akan merespons bahwa pertanyaan pengguna berada di luar konteks. Namun, bila relevan, GPT-4o menghasilkan jawaban yang sesuai. Lalu pertanyaan dan respons akan disimpan ke dalam *session*. Pada akhirnya, pengguna dapat meninjau respons yang diberikan dan dapat bertanya kembali atau mengakhiri sesi percakapan dengan keluar dari aplikasi.

3.5 Tahap Pengembangan

3.5.1 Pengembangan Model

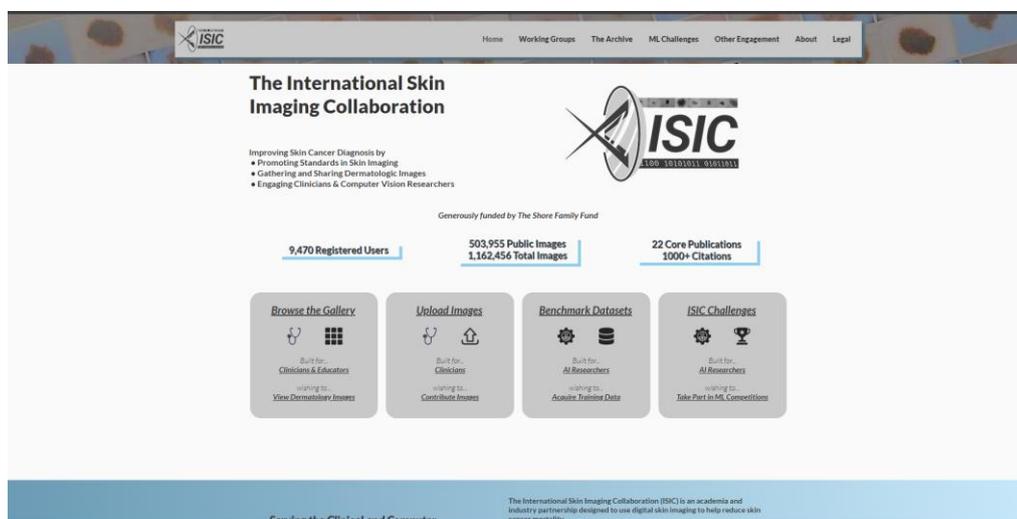
Tahap pengembangan dalam penelitian ini adalah tahap implementasi model *transfer learning* EfficientNetB3. Pada tahap ini, model EfficientNetB3

digunakan untuk mendeteksi kemungkinan kanker kulit melalui klasifikasi gambar, dengan menggunakan *framework deep learning* TensorFlow dan Keras dalam pengembangan AI. Berikut merupakan tahapan dalam pengembangan model.

3.5.1.1 Data Collection

Data yang digunakan dalam penelitian ini berasal dari laman web resmi International Skin Imaging Collaboration (ISIC), yang menyediakan *dataset* terbesar di dunia untuk mendukung penelitian deteksi kanker kulit. ISIC menyediakan gambar-gambar lesi kulit yang telah diberi label secara akurat oleh ahli dermatologi, mencakup berbagai kategori seperti melanoma, karsinoma sel basal, karsinoma sel skuamosa, dan *nevus*. *Dataset* yang tersedia di ISIC diperoleh melalui kolaborasi internasional antara berbagai institusi medis, dengan gambar diambil menggunakan perangkat *dermoscopy* khusus yang memungkinkan pengambilan gambar kulit dengan detail yang tinggi. Setiap citra dalam *dataset* disertai dengan *metadata* yang mencakup informasi klinis pasien seperti diagnosis, usia, jenis kelamin, serta lokasi lesi. Seluruh *dataset* dan informasi terkait dapat diakses secara terbuka oleh peneliti melalui platform ISIC.

Gambar 3.11 merupakan tampilan utama situs web ISIC yang menampilkan berbagai fitur utama seperti akses ke galeri gambar kanker kulit, unggah citra, *benchmark dataset*, serta ISIC Challenges.

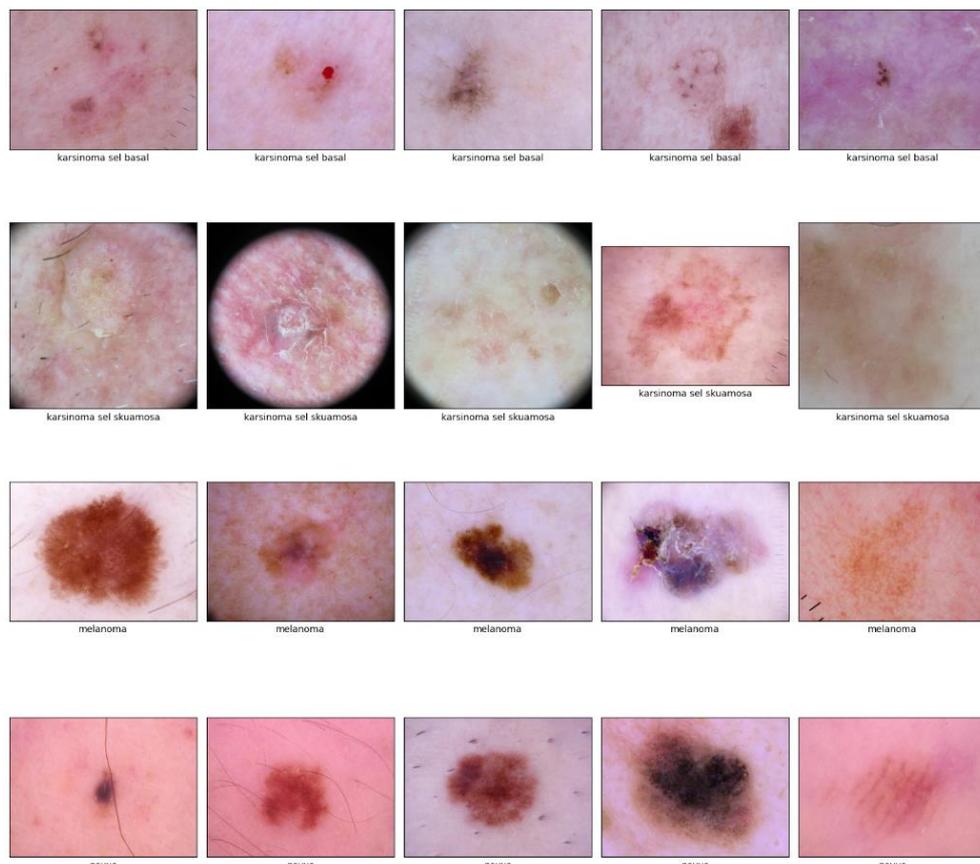


Gambar 3.11 Tampilan *website* ISIC

Dataset utama yang digunakan dalam penelitian ini adalah *dataset* HAM10000 (Tschandl dkk., 2018). Selain itu, peneliti juga menambahkan *dataset* untuk kelas karsinoma sel basal sebanyak 50 data, karsinoma sel skuamosa sebanyak 407 data, dan melanoma sebanyak 50 data. Data tersebut berasal dari laman web *gallery* ISIC (gallery.isic-archive.com). Penggunaan *dataset* tersebut bertujuan untuk meningkatkan keragaman data dan memastikan representasi yang kuat terhadap masing-masing kelas kanker kulit.

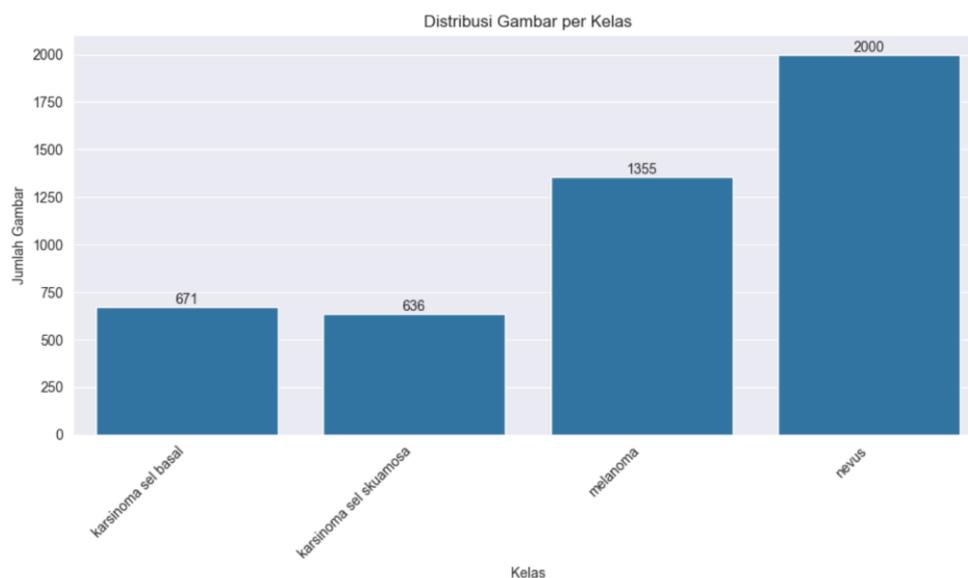
3.5.1.2 Data Exploration

Tahap eksplorasi data berfokus pada pemahaman mendalam terhadap data yang akan digunakan dalam pengembangan model klasifikasi gambar. Pada tahap ini, data gambar kanker kulit akan dieksplorasi untuk mengidentifikasi jenis-jenis lesi kulit yang ingin akan dibuatkan model klasifikasinya, mulai dari karsinoma sel basal, karsinoma sel skuamosa, melanoma, dan *nevus*. Gambar 3.12 menunjukkan sampel gambar dari masing-masing kelas dari *dataset* kanker kulit



Gambar 3.12 Sampel gambar jenis kanker kulit

Selain itu, pada *data exploration* dilakukan juga untuk mengetahui distribusi gambar masing-masing kelas kanker kulit. Hal ini dilakukan agar keseimbangan data dapat dianalisis, sehingga jika terdapat kelas yang memiliki jumlah sampel jauh lebih sedikit dibandingkan kelas lainnya, dapat dilakukan teknik penyeimbangan data seperti *data augmentation*. Dari diagram batang yang ditampilkan pada Gambar 3.13, terlihat bahwa distribusi gambar pada setiap kelas kanker kulit tidak merata. Karsinoma sel skuamosa memiliki jumlah gambar paling sedikit, yaitu 636 gambar, diikuti oleh karsinoma sel basal dengan 671 gambar. Sementara itu, kelas melanoma memiliki 1.355 gambar, dan kelas *nevus* merupakan kelas dengan jumlah gambar terbanyak, yaitu 2.000 gambar. Ketidakeimbangan ini dapat mempengaruhi performa model klasifikasi, terutama dalam mengenali kelas dengan jumlah sampel yang lebih sedikit. Oleh karena itu, diperlukan teknik penyeimbangan data seperti *data augmentation* mengatasi perbedaan distribusi gambar agar model dapat belajar secara optimal dari seluruh kelas yang ada.



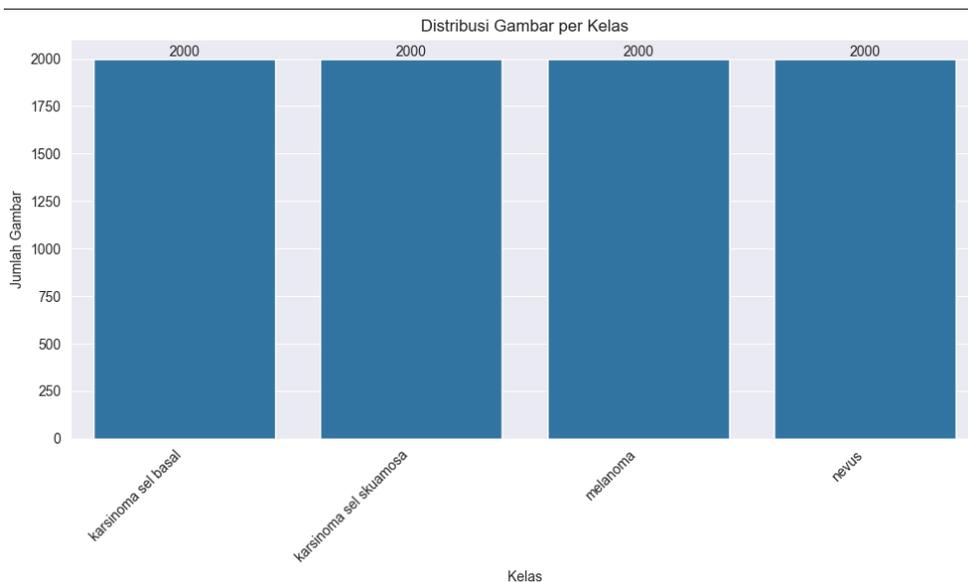
Gambar 3.13 Distribusi gambar setiap kelas kanker kulit

3.5.1.3 Data Augmentation

Data augmentation adalah teknik yang digunakan untuk memperbesar atau memperkaya jumlah data pelatihan dengan memodifikasi gambar asli melalui berbagai transformasi. Tujuan utama dari *data augmentation* dalam penelitian ini adalah untuk mengatasi ketidakeimbangan data dengan menambahkan variasi

pada gambar dari kelas yang memiliki jumlah sampel lebih sedikit, yaitu karsinoma sel basal, karsinoma sel skuamosa, dan melanoma, sehingga masing-masing kelas memiliki 2.000 gambar. Teknik *data augmentation* yang digunakan meliputi *horizontal flip*, *vertical flip*, dan rotasi sebesar 20 derajat, yang bertujuan untuk meningkatkan keragaman data tanpa mengubah karakteristik utama dari gambar asli.

Gambar 3.14 menampilkan diagram batang yang menunjukkan hasil *data augmentation*, di mana distribusi jumlah gambar pada setiap kelas kanker kulit telah diseimbangkan. Sebelum dilakukan *data augmentation*, kelas karsinoma sel skuamosa, karsinoma sel basal, dan melanoma memiliki jumlah gambar yang lebih sedikit dibandingkan kelas *nevus*. Namun, setelah proses *data augmentation* menggunakan teknik *horizontal flip*, *vertical flip*, dan rotasi sebesar 20 derajat, jumlah gambar pada ketiga kelas tersebut berhasil ditingkatkan hingga mencapai 2.000 gambar, menyamai jumlah gambar pada kelas *nevus*.



Gambar 3.14 Distribusi gambar setelah *data augmentation*

3.5.1.4 Data Preprocessing

Data preprocessing merupakan tahapan penting dalam pemrosesan data sebelum model *deep learning* dilatih. Proses ini mencakup berbagai teknik untuk mempersiapkan data agar dapat digunakan secara optimal oleh model, seperti normalisasi, dan konversi ke format yang sesuai. Tujuan utama dari *data*

preprocessing adalah untuk mengkonversi data ke format yang sesuai serta mengurangi *noise*, serta memastikan bahwa model dapat belajar dari pola yang relevan tanpa terpengaruh oleh variasi yang tidak signifikan dalam data. Dalam konteks *deep learning*, *data preprocessing* juga membantu dalam mempercepat proses pelatihan dan meningkatkan akurasi model dengan mengoptimalkan distribusi data yang masuk ke dalam jaringan saraf tiruan pada *deep learning*. *Preprocessing* pada penelitian ini dilakukan menggunakan mengubah *input* gambar menjadi ukuran 224x224 piksel dan melakukan *rescale* gambar. *Rescale* merupakan proses atau normalisasi nilai piksel gambar dari rentang 0–255 menjadi 0–1. Ini dilakukan dengan cara membagi setiap nilai piksel dalam gambar dengan angka 255. Gambar 3.15 merupakan *output* setelah proses data *preprocessing* selesai dilakukan.

```
Ditemukan 4 kelas: karsinoma sel basal, karsinoma sel skuamosa, melanoma, nevus
Memproses kelas: karsinoma sel basal
  Berhasil memproses 2000/2000 gambar untuk kelas karsinoma sel basal
Memproses kelas: karsinoma sel skuamosa
  Berhasil memproses 2000/2000 gambar untuk kelas karsinoma sel skuamosa
Memproses kelas: melanoma
  Berhasil memproses 2000/2000 gambar untuk kelas melanoma
Memproses kelas: nevus
  Berhasil memproses 2000/2000 gambar untuk kelas nevus

Total gambar yang berhasil dipreprocessing: 8000
Jumlah gambar per kelas setelah preprocessing:
  karsinoma sel basal: 2000 gambar
  karsinoma sel skuamosa: 2000 gambar
  melanoma: 2000 gambar
  nevus: 2000 gambar
```

Gambar 3.15 Hasil *data preprocessing*

3.5.1.5 Data Splitting

Data splitting adalah proses membagi *dataset* menjadi beberapa bagian untuk memastikan model dapat belajar secara efektif dan diuji dengan data yang belum pernah dilihat sebelumnya. Tujuan utama dari *data splitting* dalam penelitian ini adalah untuk membagi *dataset* menjadi tiga subset, yaitu:

1. *Training*: Digunakan untuk melatih model EfficientNetB3. Bagian ini biasanya mencakup sekitar 80% dari total *dataset* dan berfungsi untuk mengajarkan model cara mengenali fitur dari gambar kulit.

2. *Validation*: 10% dari *dataset* dialokasikan untuk data validasi. Data validasi ini digunakan untuk mengukur kinerja model selama proses pelatihan, membantu dalam penyetelan *hyperparameter*, dan memantau *overfitting*.
3. *Test*: Sisa 10% dari *dataset* dialokasikan sebagai data uji. Data *test* digunakan hanya pada tahap akhir untuk evaluasi model setelah pelatihan selesai, memastikan bahwa model dapat bekerja dengan baik pada data yang belum pernah dilihat sebelumnya.

Pembagian ini dilakukan agar model dapat dilatih menggunakan mayoritas data yang tersedia, divalidasi selama proses pelatihan untuk menghindari *overfitting*, serta diuji dengan data yang benar-benar baru untuk mengukur performa sebenarnya. Setelah dilakukan proses pembagian, setiap kelas kanker kulit memiliki proporsi data yang sesuai dengan persentase yang telah ditentukan. Gambar 3.16 menunjukkan hasil dari proses *data splitting*, di mana *dataset* telah terbagi dengan masing-masing kelas terdiri dari 1.600 gambar dalam *data training*, 200 gambar dalam *data validation*, dan 200 gambar dalam *data test*.

```

Train set:
karsinoma sel basal: 1600 gambar
karsinoma sel skuamosa: 1600 gambar
melanoma: 1600 gambar
nevus: 1600 gambar

Validation set:
karsinoma sel basal: 200 gambar
karsinoma sel skuamosa: 200 gambar
melanoma: 200 gambar
nevus: 200 gambar

Test set:
karsinoma sel basal: 200 gambar
karsinoma sel skuamosa: 200 gambar
melanoma: 200 gambar
nevus: 200 gambar

```

Gambar 3.16 Hasil *data splitting*

3.5.1.6 Training Model

Langkah utama yang dilakukan pada tahapan ini adalah melatih model EfficientNetB3 yang telah dirancang. Model ini dilatih dengan menggunakan *data*

training yang telah diproses sebelumnya. Dalam pelatihan, *hyperparameter* merupakan parameter yang harus ditentukan sebelum proses dimulai dan berperan krusial dalam mengontrol proses *training* model. Gambar 3.17 memperlihatkan konfigurasi *hyperparameter* yang digunakan untuk mengoptimalkan proses pelatihan model klasifikasi kanker kulit berbasis EfficientNetB3.

```

batch_size = 32 # set batch size for training
epochs = 50 # number of all epochs in training
patience = 1 # number of epochs to wait to adjust lr if monitored value does not improve
stop_patience = 3 # number of epochs to wait before stopping training if monitored value does not improve
threshold = 0.9 # if train accuracy is < threshold adjust monitor accuracy, else monitor validation loss
factor = 0.5 # factor to reduce lr by
ask_epoch = 15 # number of epochs to run before asking if you want to halt training
batches = int(np.ceil(len(train_generator.labels) / batch_size)) # number of training batch to run per epoch

callbacks = [EarlyStopping(monitor='val_loss', patience=patience, stop_patience=stop_patience, threshold=threshold,
                           factor=factor, batches=batches, epochs=epochs, ask_epochs=ask_epoch)]

```

Gambar 3.17 Konfigurasi *hyperparameter*

Beberapa *hyperparameter* penting yang digunakan meliputi *batch_size* sebesar 32, yang menentukan jumlah sampel yang diproses dalam satu iterasi pelatihan, serta *epochs* sebanyak 50, yang menandakan jumlah total iterasi melalui *dataset* selama pelatihan. Selain itu, model menerapkan *patience* sebesar 1 untuk menentukan jumlah *epoch* yang akan ditunggu sebelum mengurangi *learning rate* jika performa tidak meningkat.

Pelatihan model ini juga menggunakan mekanisme *early stopping* dengan *stop_patience* sebesar 3, yang berarti pelatihan akan dihentikan jika dalam 3 *epoch* berturut-turut tidak ada peningkatan signifikan dalam metrik evaluasi. *Threshold* ditetapkan pada 0.9, yang menunjukkan bahwa jika akurasi pelatihan sudah mencapai angka ini, maka penyesuaian *learning rate* akan lebih agresif. Faktor pengurangan *learning rate* sebesar 0.5 digunakan untuk menyesuaikan kecepatan pembelajaran model, sehingga dapat menghindari konvergensi yang terlalu cepat atau terlalu lambat. *Hyperparameter tuning* ini dilakukan dengan memanfaatkan *callback*, yang mengotomatisasi proses pemantauan dan penyesuaian parameter pelatihan, memastikan model dapat mencapai performa optimal dengan efisiensi komputasi yang baik.

3.5.1.7 Model Deployment

Pada tahap *model deployment*, model EfficientNetB3 yang telah dilatih dan dievaluasi akan disimpan dalam format Keras (.h5) untuk menjaga struktur dan bobot model. Selanjutnya, model ini akan langsung dimuat ke dalam aplikasi

berbasis Flask, sehingga prediksi dapat dijalankan secara internal di server. Dengan pendekatan ini, Flask akan menangani pemrosesan *input* dan *output* secara langsung di sisi *backend*, tanpa perlu komunikasi berbasis *endpoint* seperti pada integrasi API.

3.5.2 Implementasi *Prompt Engineering*

Prompt engineering merupakan teknik merancang dan mengoptimalkan perintah atau instruksi yang diberikan kepada model *generative AI* untuk mendapatkan *output* yang lebih akurat, relevan, dan sesuai dengan kebutuhan spesifik (Ekin, 2023). Tujuan utama penggunaan teknik *prompt engineering* pada penelitian ini adalah untuk dapat memvalidasi gambar serta mengoptimalkan interaksi antara pengguna dan model AI dalam aplikasi *chatbot* agar menghasilkan jawaban yang lebih akurat, relevan, dan sesuai dengan kebutuhan spesifik aplikasi.

3.5.2.1 GPT-4o *Vision*

Teknik *prompt engineering* pada GPT-4o *Vision* diimplementasikan untuk memvalidasi gambar kanker kulit dengan tujuan memastikan bahwa GPT-4o dapat memahami dan menganalisis konteks visual secara optimal. Proses ini melibatkan perancangan *prompt* yang spesifik dan terstruktur, sehingga model dapat menginterpretasikan informasi dari gambar dengan lebih akurat.

3.5.2.2 *Chatbot* GPT-4o

Teknik *prompt engineering* dalam model GPT-4o *text generation* berperan penting dalam memastikan bahwa keluaran yang dihasilkan bersifat relevan, informatif, dan berada pada ruang lingkup kanker kulit. Dengan pendekatan yang sistematis, teknik ini memungkinkan *chatbot* untuk memberikan respons yang tidak hanya sesuai dengan konteks pertanyaan, tetapi juga bersifat edukatif. Selain itu, *prompt engineering* dapat diterapkan dalam skenario interaksi yang lebih kompleks, seperti menyusun dialog yang adaptif berdasarkan tingkat pemahaman pengguna atau mengklarifikasi pertanyaan sebelum memberikan jawaban akhir. Strategi ini bertujuan untuk meningkatkan kualitas interaksi dengan pengguna serta meminimalkan kesalahpahaman dalam komunikasi berbasis AI.

3.5.3 Pengembangan Aplikasi

Tahap pengembangan aplikasi berfokus pada pembangunan aplikasi *chatbot* berbasis *website* yang akan menjadi *platform* utama bagi pengguna untuk mengakses layanan deteksi dini kanker kulit. Pengembangan ini melibatkan dua komponen utama yaitu *frontend* dan *backend*, yang masing-masing memiliki peran penting dalam menyediakan antarmuka yang mudah digunakan serta integrasi antara model EfficientNetB3 dan GPT-4o.

3.6 Tahap Pengujian

Tahap pengujian merupakan tahapan yang paling penting untuk memastikan bahwa model EfficientNetB3, GPT-4o, dan aplikasi *chatbot* berbasis *website* yang dikembangkan berjalan dengan optimal sesuai fungsi dan kebutuhan yang sudah ditentukan dari awal.

3.6.1 Pengujian Model

Tahap pengujian model dalam penelitian ini bertujuan untuk mengukur kinerja model EfficientNetB3 dalam mengklasifikasi jenis kanker kulit berdasarkan data pengujian. Pengujian ini akan dilakukan dengan menggunakan *confusion matrix* dalam menganalisis kinerja model klasifikasi. *Confusion matrix* adalah tabel yang menggambarkan hasil prediksi model dengan membandingkan label yang diprediksi dengan label yang sebenarnya. Tabel ini memuat jumlah prediksi yang benar dan salah untuk setiap kelas, yang memungkinkan identifikasi kesalahan klasifikasi secara rinci (Krstinić dkk, 2020). Berdasarkan komponen tersebut, beberapa metrik evaluasi dapat dihitung. Penjelasan dari masing-masing komponen dan metrik evaluasi adalah sebagai berikut:

a. Accuracy

Accuracy mengukur persentase prediksi yang benar secara keseluruhan, baik positif maupun 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

b. *Precision*

Precision menunjukkan sejauh mana model dapat dipercaya dalam memprediksi kasus positif. Semakin tinggi nilainya, semakin sedikit kesalahan dalam memprediksi kasus positif. Berikut merupakan persamaan dalam menghitung *Precision*:

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

c. *Recall*

Recall menghitung proporsi kasus positif yang berhasil diidentifikasi oleh model, yaitu berapa banyak kasus positif yang dapat dideteksi. Berikut merupakan persamaan dalam menghitung *Recall*:

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

d. *F1-score*

F1-score adalah nilai rata-rata harmonik antara *precision* dan *recall*, yang berfungsi untuk menilai keseimbangan keduanya, khususnya ketika data yang digunakan tidak seimbang. Metrik ini memberikan gambaran mengenai seberapa efektif model dalam mengelola kedua metrik tersebut secara bersamaan. Berikut merupakan persamaan dalam menghitung *F1-score*:

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

3.6.2 Pengujian GPT-4o Vision

Pengujian ini bertujuan untuk memvalidasi kemampuan model GPT-4o *Vision* dalam mengidentifikasi apakah *input* gambar terindikasi kanker kulit atau tidak. Dengan melakukan pengujian ini, diharapkan model dapat memberikan klasifikasi yang akurat sehingga dapat digunakan sebagai *validator* sebelum gambar diproses oleh model klasifikasi jenis kanker kulit. Validasi ini penting untuk memastikan bahwa sistem yang dikembangkan dapat memberikan hasil yang andal.

Pengujian dilakukan menggunakan data yang terdiri dari 100 gambar, yang terbagi menjadi dua kategori utama yaitu 50 gambar yang telah dipastikan itu merupakan kanker kulit yang berasal dari data uji dan 50 gambar yang merupakan bukan kanker kulit. Setelah seluruh gambar diuji, hasil prediksi akan dianalisis menggunakan metrik *accuracy*. *Accuracy* dihitung dengan membandingkan jumlah prediksi yang benar terhadap total jumlah sampel yang diuji. Metrik ini digunakan sebagai indikator utama dalam menilai performa model dalam tugas klasifikasi kanker kulit. Berikut merupakan persamaan dari metrik *accuracy*:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

3.6.3. Pengujian Fungsionalitas *Chatbot*

Tahap pengujian fungsionalitas *chatbot* dilakukan dengan menggunakan metode *black box* untuk menguji fungsi-fungsi utama pada aplikasi *chatbot* berbasis *website*. Pengujian ini berfokus pada validasi keluaran sistem tanpa melihat struktur internal kode atau algoritma yang digunakan (Nidhra & Dondeti, 2012). Tujuannya adalah memastikan bahwa sistem bekerja sesuai dengan spesifikasi yang telah ditentukan dan memberikan hasil yang akurat serta responsif kepada pengguna. Tabel 3.1 merupakan instrumen pengujiannya yang terdiri dari uji kasus, skenario, *output* yang diharapkan, dan *output* yang dihasilkan. Di mana setiap hasil pengujian, baik yang berhasil maupun yang gagal akan dicatat. Kemudian berdasarkan hasil pengujian yang telah dicatat, dilakukan evaluasi untuk menyimpulkan apakah *chatbot* telah berjalan sesuai dengan rancangan awal atau tidak.

Tabel 3.1 Instrumen pengujian fungsionalitas dengan metode *black box*

No	Uji Kasus	Skenario	Hasil Yang Diharapkan
1	<i>Upload</i>	Menekan tombol <i>home</i>	Kembali ke <i>landing page</i>
2	<i>Upload</i>	Menekan tombol deteksi gambar sebelum <i>upload</i> gambar	Tombol deteksi gambar tidak aktif

No	Uji Kasus	Skenario	Hasil Yang Diharapkan
3	<i>Upload</i>	Menekan tombol ikon matahari	Tampilan <i>website</i> menjadi <i>dark mode</i>
4	<i>Upload</i>	Menekan tombol ikon bulan	Tampilan <i>website</i> menjadi <i>light mode</i>
5	<i>Upload</i>	Menekan tombol pilih gambar dan mengunggah gambar	Ada <i>preview</i> gambar dan tombol deteksi gambar aktif
6	<i>Upload</i>	<i>Drag and drop</i> gambar ke <i>area upload</i>	Ada <i>preview</i> gambar dan tombol deteksi gambar aktif
7	<i>Upload</i>	Menekan tombol deteksi gambar	Gambar akan diproses dan muncul <i>preloader</i> “memproses gambar”
8	<i>Upload</i>	<i>Upload</i> format gambar selain JPG, JPEG, PNG	Muncul Pesan <i>Error</i> : “Format <i>file</i> tidak didukung”
9	<i>Upload</i>	<i>Upload</i> gambar selain gambar kanker kulit	Muncul Pesan <i>Error</i> : “Gambar yang diunggah bukan gambar kanker kulit”
10	<i>Upload</i>	<i>Upload</i> gambar kanker kulit	Muncul Pesan Sukses: “Gambar berhasil diproses”
11	<i>Chatbot</i>	Menekan tombol kembali	Kembali ke halaman <i>upload</i>
12	<i>Chatbot</i>	Menekan tombol kirim ketika tidak ada <i>input</i>	<i>Input</i> tidak terikirim
13	<i>Chatbot</i>	Menekan tombol ikon matahari	Tampilan <i>website</i> menjadi <i>dark mode</i>

No	Uji Kasus	Skenario	Hasil Yang Diharapkan
14	<i>Chatbot</i>	Menekan tombol ikon bulan	Tampilan <i>website</i> menjadi <i>light mode</i>
15	<i>Chatbot</i>	<i>Input</i> pertanyaan di luar konteks kanker kulit	<i>Chatbot</i> akan memberikan respons bahwa pertanyaan pengguna di luar konteks
16	<i>Chatbot</i>	<i>Input</i> pertanyaan relevan dengan konteks kanker kulit	<i>Chatbot</i> akan memberikan respons yang relevan dengan pertanyaan pengguna
17	<i>Chatbot</i>	Menekan tombol rekomendasi pertanyaan	Pertanyaan akan langsung terkirim dan <i>chatbot</i> akan langsung memberikan respons
18	<i>Chatbot</i>	Menekan tombol <i>icon</i> kirim atau menekan <i>enter</i> pada <i>keyboard</i>	Pertanyaan akan langsung terkirim dan <i>chatbot</i> akan langsung memberikan respons
19	<i>Chatbot</i>	Mengirim <i>input</i> lagi ketika <i>input</i> sebelumnya masih diproses	Pertanyaan baru tidak akan terkirim