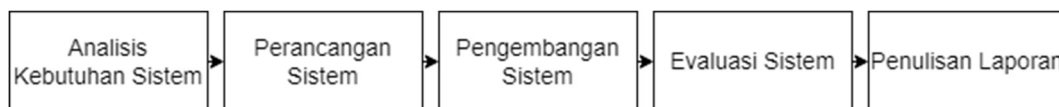


BAB III METODE PENELITIAN

3.1 Metode Penelitian

Berdasarkan analisis berdasarkan kebutuhan pada latar belakang, tujuan, dan rumusan masalah penelitian, maka penelitian ini akan dilakukan menggunakan metode penelitian pengembangan D&D (*Design and Development*). Metode penelitian pengembangan D&D merupakan metode yang berfokus pada luaran yang bermanfaat untuk kebutuhan ilmu pengetahuan, tidak berfokus kepada keuntungan komersial dari aplikasi tersebut (Ellis & Levy, 2010). Oleh karena itu, penulis menggunakan metode D&D sebagai metode penelitian pengembangan utama dalam melakukan penelitian aplikasi filtrasi Spam dan *Phising* pesan Whatsapp dengan metode TF – IDF. Pada gambar 3.1, dijelaskan tahap – tahap metode penelitian *Design and Development*.



Gambar 3. 1 Metode Penelitian

3.1.1 Analisis Kebutuhan Sistem

Tahapan awal penelitian dilakukan dengan melakukan identifikasi masalah yang terjadi pada latar belakang dan melakukan kajian pustaka terhadap penelitian – penelitian yang terkait dengan masalah yang terjadi agar penelitian memberikan hasil yang sesuai dengan tujuan yang telah ditetapkan saat penelitian dimulai. Pada tahap ini, peneliti juga menganalisis kebutuhan yang diperlukan selama penelitian ini berlangsung.

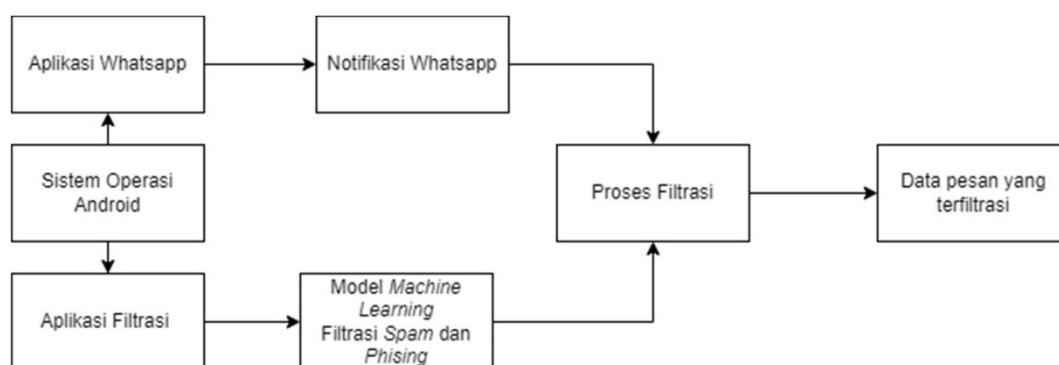
Sesuai dengan apa yang telah dijelaskan pada latar belakang dan kajian pustaka, maka peneliti akan mengembangkan model filtrasi Spam dan *Phising* menggunakan metode *pre-processing* TF – IDF dan menggunakan algoritma Random Forest. Aplikasi yang dibangun merupakan aplikasi berbasis sistem operasi Android yang dibangun menggunakan *framework* dan IDE Android Studio. Aplikasi yang dibangun akan diintegrasikan dengan model yang telah dikembangkan.

Aplikasi yang dibangun akan memiliki kemampuan untuk melakukan filtrasi terhadap pesan – pesan yang masuk ke dalam perangkat pengguna. Data yang masuk ke dalam perangkat pengguna nantinya akan ditampilkan ke dalam masing – masing

tampilan yang telah dibuat pada aplikasi. Aplikasi juga akan melakukan notifikasi kepada pengguna apabila terdapat pesan – pesan yang terdeteksi sebagai pesan yang berbahaya oleh model yang terintegrasi di dalam aplikasi. Sehingga pengguna dapat lebih berhati – hati dan waspada dalam membuka pesan yang diterima.

3.1.2 Perancangan Sistem

Tahapan selanjutnya adalah perancangan model dan sistem berdasarkan hasil dari analisis kebutuhan. Untuk memecahkan permasalahan yang terjadi pada latar belakang, penelitian ini akan merancang dan mengembangkan model menggunakan metode *pre-processing* TF – IDF dan algoritma Random Forest dan mengembangkan aplikasi berbasis android. Setelah pengembangan model dan aplikasi selesai, yang selanjutnya dilakukan adalah implementasi model ke dalam aplikasi yang telah dikembangkan. Arsitektur sistem utama aplikasi yang akan dikembangkan dapat dilihat pada gambar 3.2.



Gambar 3. 2 Diagram Sistem Aplikasi Filtrasi Pesan Spam Dan *Phising*

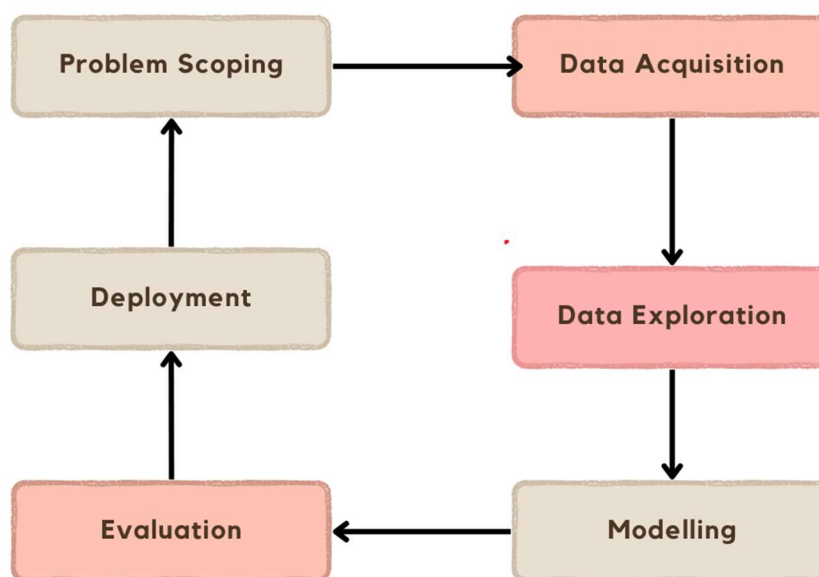
3.1.2.1 Perancangan Model Filtrasi Pesan Spam dan *Phising*

Perancangan model filtrasi pesan Spam dan *Phising* pada penelitian yang dilakukan menggunakan metode yang umum digunakan untuk pengembangan model *Artificial Intelligence* yaitu metode AI Project cycle. Dalam jurnal berjudul “An Artificial Intelligence Life Cycle: From Conception to Production” yang ditulis oleh Daswin De Silva dkk pada tahun 2022, menjelaskan bahwa pada pengembangan sebuah model *artificial intelligence*, terdapat *role* yang dibagi menjadi 3 dengan tahapan tugasnya masing – masing.

Design, merupakan tahap di mana *Data Scientist* mengidentifikasi masalah, melakukan eksplorasi terhadap data yang akan dijadikan sebagai bahan utama untuk model yang akan dibangun. *Develop*, merupakan tahap di mana *ML Scientist*

mulai melakukan pengolahan data terhadap data yang telah dikumpulkan pada tahap sebelumnya, seperti *pre – processing* data , pengembangan awal model, augmentasi data, dan melakukan evaluasi metrics utama model yang dikembangkan.

Deploy, merupakan tahap *ML Engineer* mulai melakukan evaluasi metrics sekunder, *review* model yang telah dikembangkan, operasionalisasi model menggunakan *pipelines* sesuai dengan kebutuhan, dan *monitoring* serta evaluasi performa model yang telah diimplementasi. Azimah dkk pada tahun 2022 merangkum *AI project cycle* menjadi 6 tahapan, yaitu *problem scoping*, *data acquisition*, *data exploration*, *modelling*, *evaluation* dan *deployment*. Pada gambar 3.2, diperlihatkan gambar alur pengembangan model AI berdasarkan *AI project cycle* yang dirangkum oleh Azimah.



Gambar 3. 3 Metode Pengembangan AI Project Cycle

Pada tahap *problem scoping*, peneliti menentukan batasan masalah yang akan diselesaikan. Pembatasan masalah yang diperlukan dalam pengembangan model filtrasi Spam dan *Phising* tersebut adalah model ini akan digunakan untuk melakukan filtrasi terhadap pesan - pesan yang masuk ke dalam perangkat pengguna, di mana pesan – pesan tersebut merupakan pesan yang kontennya merupakan percakapan sehari – hari antar pengguna.

Di tahap *data acquisition*, peneliti melakukan *scrapping* terhadap dataset yang sudah ada sebelumnya di laman *website* kaggle, *link* dataset yang diberikan pada penelitian terdahulu, dan *Github*. Data yang dikumpulkan merupakan pesan – pesan yang dikirim melalui SMS. Selain itu, data juga didapatkan dari perangkat pribadi peneliti dengan melakukan ekstrasi terhadap SMS yang diterima pada perangkat peneliti dan pesan yang diterima pada aplikasi *Whatsapp* akun peneliti.

Dengan data – data yang sudah terkumpul, langkah selanjutnya adalah melakukan *data exploration*. *Data exploration* merupakan tahap di mana data yang dikumpulkan diverifikasi dan dibersihkan menjadi data yang baik untuk digunakan dalam pengembangan model. Dalam pengembangan model filtrasi Spam dan *Phising*, data yang tepat merupakan data pesan yang tidak berbahaya dan pesan – pesan dengan konten promosi dan pesan yang berisi *file* dan link berbahaya. *Dataset* yang terbentuk adalah data yang terbagi menjadi 2 kelas, yaitu kelas 0, sebagai data pesan yang aman dan kelas 1, sebagai pesan yang berbahaya.

Pada tahap *modelling*, dataset dibagi menjadi 80% sebagai data yang akan digunakan dalam proses *training* model, dan 20% sebagai data yang akan digunakan pada proses *testing*.

Pada tahap *evaluation*, dataset yang telah dibagi untuk testing sebesar 20%, digunakan sebagai data untuk mengetahui performa model yang telah dibuat. Untuk mengetahui performa dari model yang telah dibangun, peneliti menggunakan metrik penghitungan *Confusion Matrix*. Tahap ini akan dijelaskan lebih jauh pada bagian metode pengujian model.

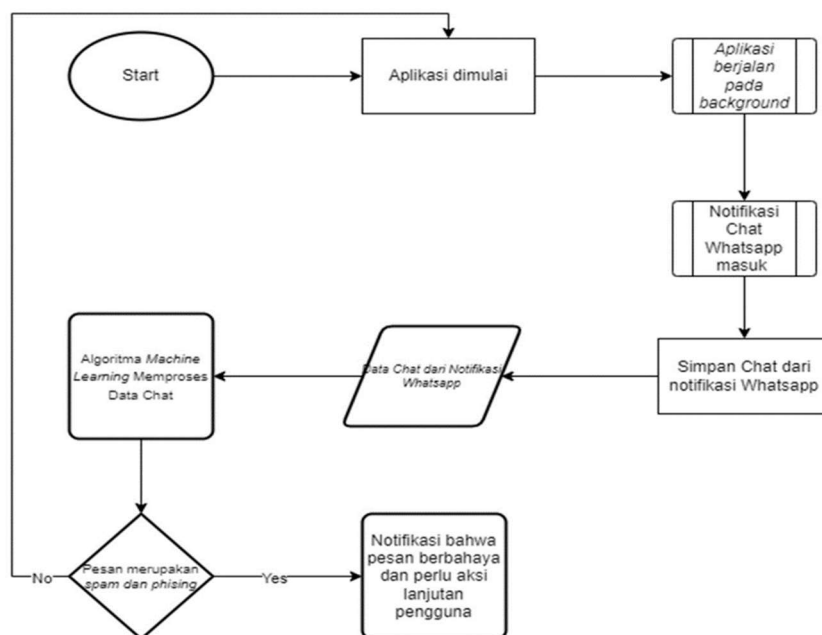
Pada tahap terakhir yaitu *deployment*, model yang telah dibuat akan diimplementasi pada aplikasi android. Model yang telah dilatih akan dikonversi menjadi *file* dengan ekstensi yang dapat terpasang pada aplikasi android. Dalam tahap ini, peneliti menggunakan library *onnx*, yang dapat digunakan untuk mengkonversi model yang dihasilkan menggunakan library *Scikit-learn* mejadi *file* dengan ekstensi *.onnx*. Agar lebih mudah digunakan maka model dengan ekstensi *.onnx* dikonversi menjadi *file* runtime agar dapat berjalan dengan lebih baik pada

3.1.2.2 Perancangan Aplikasi

Dalam penelitian ini, Perancangan aplikasi dilakukan untuk membangun aplikasi untuk sistem berbasis android. Sebelum melakukan penulisan kode

menggunakan IDE Android Studio untuk pengembangan aplikasi, peneliti merumuskan terlebih dahulu diagram alir, blok diagram sistem aplikasi, desain antar muka pengguna, *use case diagram*, dan arsitektur aplikasi yang akan dibangun. Pada gambar 3.4, Terlihat rancangan aplikasi filtrasi pesan Spam dan *Phising* menggunakan diagram sistem.

Seperti yang telah digambarkan pada Blok Diagram Sistem Aplikasi Filtrasi Pesan Spam dan *Phising*, Aplikasi yang dibangun pada Sistem Operasi Android memiliki algoritma *machine learning* yang terintegrasi secara langsung dengan aplikasi. Semua pemrosesan yang berjalan dalam aplikasi, berjalan secara *background processing*, di mana pemrosesan dilakukan tanpa ada dibutuhkan masukkan langsung dari pengguna, melainkan membutuhkan masukkan dari notifikasi pesan masuk dari aplikasi Whatsapp. Berikut merupakan diagram alir dari yang akan dibuat yang digambarkan pada gambar 3.5.



Gambar 3. 4 Diagram Alir Aplikasi

Dengan diagram alir yang dirancang sedemikian rupa maka berikut merupakan *use case diagram* yang dapat dirancang untuk aplikasi yang akan dikembangkan. Gambar 3. 6 adalah desain antarmuka pengguna aplikasi yang akan dikembangkan memperhitungkan kebutuhan aplikasi berdasarkan diagram sistem dan diagram alir yang telah dibuat.



Gambar 3. 5 Desain Antarmuka Pengguna Halaman Utama Aplikasi (Kiri)
Halaman Detail Pesan Aplikasi (Kanan)

3.1.3 Pengembangan Sistem

Penelitian ini menggunakan metode pengembangan prototipe (*prototyping*). Metode pengembangan prototipe merupakan metode yang menggabungkan kebutuhan pada aplikasi dengan kerangka kerja dari aplikasi. Metode ini mengutamakan korespondensi antara aplikasi dengan target pengguna aplikasi sehingga hasil dari aplikasi yang dibuat dari metode ini dapat berubah menyesuaikan dengan apa yang dibutuhkan oleh target pengguna. Maka, hasil dari aplikasi ini merupakan versi mode awal yang dibuat dengan tujuan mengeksplorasi aspek yang dibutuhkan oleh aplikasi agar dapat bekerja dengan baik (Kathryn, 2021). Data yang dikumpulkan selama penelitian berlangsung terbagi menjadi 2 jenis data, yaitu data primer dan juga data sekunder.

3.1.3.1 Tahap Komunikasi

Tahap komunikasi pada metode *prototyping* merupakan langkah pertama dalam melakukan pengembangan sistem yang dilakukan dengan berbagi informasi terkait desain dan rancangan awal sistem yang akan dibangun. Pada tahap ini, rancangan dan desain awal dibentuk dan diubah sesuai dengan hasil dari komunikasi ataupun hasil dari pencarian data yang dilakukan oleh peneliti. Target pengguna aplikasi yang akan digunakan, kebutuhan yang diperlukan aplikasi agar dapat berjalan dengan optimal menjadi poin utama dalam komunikasi yang terjadi

antara peneliti sebagai pengembang aplikasi dengan *shareholders* aplikasi yang akan dibuat sebagai capaian dalam penelitian yang berlangsung. Keputusan yang dihasilkan adalah peneliti akan melakukan pengembangan model menggunakan TF – IDF dan algoritma Random Forest sebagai objek utama dalam penelitian ini. Peneliti juga dapat memfokuskan pengembangan aplikasi pada sistem berbasis android dan aplikasi Whatsapp sebagai aplikasi utama yang akan dijadikan sebagai bahan penelitian dalam melakukan filtrasi isi pesan menggunakan model yang dibuat.

3.1.3.2 Tahap Desain Cepat

Tahap desain cepat merupakan tahap di mana peneliti, menggunakan hasil dari tahap komunikasi untuk melakukan rencana pengembangan dan desain yang sesuai dengan apa yang dihasilkan dari komunikasi antara pihak – pihak terkait. Tahap perencanaan dan desain cepat yang dilakukan peneliti meliputi beberapa kegiatan yang terpisah namun berkesinambungan. Kegiatan tersebut diantaranya adalah pengembangan model untuk filtrasi pesan Spam dan *Phising* dan perancangan aplikasi untuk sistem berbasis android.

Pada pengembangan model filtrasi Spam dan *Phising* yang dikembangkan dalam penelitian ini, Peneliti akan menggunakan metode TF – IDF sebagai metode *pre-processing* utama sesuai dengan topik penelitian yang telah ditentukan. Algoritma yang digunakan sebagai algoritma utama dalam melakukan deteksi pesan adalah algoritma klasifikasi Random Forest. Data pertama – tama akan diproses melalui *regular expression* (Regex) agar data *training* menjadi bersih dan seragam. *Pipeline* model filtrasi Spam dan *Phising* yang digunakan adalah *TfidfVectorizer* sebagai *feature extraction* dan *RandomForestClassifier* sebagai algoritma utama model untuk klasifikasi. Kedua library tersebut adalah bentuk implementasi perencanaan pengembangan model menggunakan metode *pre-processing* TF – IDF dan algoritma klasifikasi Random Forest. Untuk mengimplementasikan perencanaan yang telah disusun, Peneliti menggunakan *library module* pada bahasa pemrograman Python, yaitu Scikit-learn.

Pada pengembangan aplikasi android filtrasi Spam dan *Phising*, peneliti menggunakan desain aplikasi Model , View, ViewModel (MVVM) sebagai bentuk desain utama aplikasi yang dikembangkan. MVVM merupakan pola desain

pengembangan aplikasi yang digunakan untuk aplikasi yang bersifat reaktif. Dengan menggunakan pola desain MVVM, maka data yang akan digunakan dan ditampilkan pada layar antarmuka pengguna dapat digunakan secara terus – menerus tanpa perlu melakukan pemanggilan data kembali. Pada kasus di mana aplikasi akan mengalami perpindahan layar, maka data perlu berada lapisan yang dapat menyimpan data secara statis, tidak mengikuti perubahan antarmuka pengguna yang bersifat dinamis sesuai dengan perilaku pengguna terhadap aplikasi.

3.1.3.3 Tahap Konstruksi Prototipe

Tahap konstruksi prototipe merupakan tahap di mana pengembangan model dan aplikasi yang telah di desain sedemikian rupa dilakukan. Pengembangan dilakukan hingga prototipe aplikasi terbentuk. Pada tahap ini, aplikasi prototipe yang dibuat sudah memiliki fungsi utama sesuai dengan yang telah direncanakan pada tahap perencanaan dan desain cepat. Maka, pada tahap ini model yang telah dilatih telah terimplementasi pada aplikasi, dan aplikasi telah memiliki fitur untuk melakukan filtrasi terhadap pesan Whatsapp yang masuk ke dalam perangkat pengguna.

3.1.3.4 Tahap *Deployment* dan *Delivery Feedback*

Tahap terakhir merupakan tahap di mana prototipe yang telah dikembangkan melalui tahapan pengujian dan perubahan sehingga sesuai dengan capaian awal yang telah ditetapkan. Perubahan yang dilakukan pada tahap *Deployment* dan *Delivery Feedback* merupakan bentuk penyesuaian yang dilakukan berdasarkan serangkaian pengujian dengan parameter yang telah ditetapkan secara khusus untuk mengevaluasi aspek – aspek yang krusial pada aplikasi untuk disebut sebagai aplikasi yang siap dan layak pakai. Aspek – aspek yang diuji meliputi performa model saat melakukan klasifikasi dan keterandalan aplikasi dalam melakukan tugasnya melakukan filtrasi terhadap pesan yang masuk ke perangkat pengguna.

3.1.4 Evaluasi Sistem

Pada penelitian untuk mengembangkan Aplikasi Filtrasi Spam dan *Phising*, diperlukan adanya pengujian yang mendalam terhadap prototipe yang telah dibuat guna mendapat data untuk menentukan apakah hasil rancangan prototipe memenuhi tujuan penelitian yang telah ditetapkan saat penelitian berlangsung. Pengujian

dilakukan untuk memastikan bahwa aplikasi yang telah diintegrasikan dengan model yang telah dibuat memiliki keandalan yang tangguh dan kemampuan dalam melakukan tugas yang telah diberikan secara optimal. Adapun pengujian yang dilakukan terhadap prototipe dibagi ke dalam beberapa bagian, yaitu sebagai berikut.

3.1.4.1 Pengujian Model

Pengujian model yang dikembangkan menggunakan metode – metode pengujian yang umum digunakan dalam evaluasi performa model. Seperti yang telah dijelaskan pada tahap evaluasi pengembangan dari model, Penelitian ini akan menguji model yang telah dikembangkan menggunakan alat ukur *Confusion Matrix*.

Confusion Matrix merupakan sebuah matrix dengan susunan N x N, Di mana N merupakan banyaknya kelas yang diprediksi. Pada model dengan 2 kelas yang diprediksi, maka matrix terdiri dari nilai *True Positive*, *True Negative*, *False Positive*, dan *False Negative*. Dengan *Confusion Matrix*, Maka performa dari model dapat diukur berdasarkan Akurasi, Presisi, Recall, dan F1 Score dari hasil. Metrik akurasi merupakan nilai yang didapatkan dari perhitungan total prediksi model yang tepat. Presisi adalah nilai yang didapatkan ketika model dapat memprediksi sesuai dengan *positive cases* (model dapat memprediksi sesuai dengan kejadian di kenyataan). *Recall* atau *Sensitivity* merupakan nilai perbandingan aktual dari *positive cases* yang dapat diprediksi oleh model dengan tepat. F1 Score adalah nilai dari rata – rata antara nilai *Precision* dan *Recall*. Selain metode tersebut, Evaluasi akan dilakukan dengan mengobservasi hasil prediksi saat model sudah terimplementasi atau *deployed* pada aplikasi yang akan dibuat. Pengujian model menggunakan *confusion matrix* akan dibantu library Scikit – learn. Perhitungan nilai akurasi, presisi, *recall* dan F1 – *score* model dapat dilihat pada persamaan 3.1, 3.2, 3.3, dan 3.4.

$$\text{Precision} = \frac{\text{TP}}{\text{FP} + \text{TP}} \quad (1)$$

Persamaan 3. 1 Persamaan Untuk Menghitung Nilai Presisi

$$\text{Recall} = \frac{\text{TP}}{\text{FN} + \text{TP}} \quad (2)$$

Persamaan 3. 2 Persamaan Untuk Menghitung Nilai *Recall*

$$\text{Accuracy} = \frac{TP+TN}{FP+FN+TP+TN} \quad (3)$$

Persamaan 3. 3 Persamaan Untuk Menghitung Nilai Akurasi

$$F1 - Score = \frac{2 \times TP}{2TP+FP+FN} \quad (4)$$

Persamaan 3. 4 Persamaan Untuk Menghitung F1-Score

Selain itu, model akan melalui pengujian yang dilakukan secara langsung saat model dan aplikasi telah diintegrasikan. Pengujian akan dilakukan menggunakan *script* program yang dibuat khusus untuk mengirimkan pesan ke nomor Whatsapp yang digunakan peneliti sebagai tempat uji coba kemampuan filtrasi aplikasi dan model yang telah dibuat.

3.1.4.2 Pengujian Aplikasi

Aplikasi yang telah dibuat akan diuji menggunakan metode pengujian *black box*. Metode pengujian *black box* adalah metode pengujian aplikasi yang bertujuan untuk melihat bagaimana aplikasi menjalankan tugasnya tanpa memperhatikan aspek internal dan teknis. Pada metode pengujian aplikasi *black box*, terdapat teknik yang disebut sebagai *Equivalence partitions*. *Equivalence Partitions* merupakan pengujian yang membandingkan antara dua *case* dengan kontradiksi.

Pada penelitian yang dilakukan Zidan dkk pada tahun 2022, dilakukan dengan membuat beberapa *test case* berdasarkan halaman *web* aplikasi SSO yang diuji. Sebagai contoh, pada halaman *login*, terdapat 3 skenario tes dengan *expected result* yang ditetapkan oleh penguji pada penelitian tersebut. Maka dalam pengujiannya jika pada skenario tes yang telah ditetapkan memiliki hasil yang sama dengan *expected result*, maka pengujian yang dilakukan dapat dikatakan berhasil dan sesuai dengan standar yang telah ditetapkan. Sebaliknya, jika hasil yang diharapkan pada skenario tes tidak terpenuhi, maka pengujian dapat dikatakan gagal atau tidak sesuai dengan standar yang telah ditetapkan dan perlu dibenahi. Hasil dari pengujian antara skenario tes dan hasil yang diharapkan dirangkum menjadi kesimpulan apakah aplikasi tersebut memenuhi standar atau perlu adanya perbaikan.

Tentunya, sesuai dengan prinsip *black box*, maka pengujian tidak akan melibatkan pengujian pada kode sumber program, melainkan mengobservasi

behavior saat aplikasi dijalankan. Pengujian pada aplikasi SSO yang dilakukan Zidan dkk, menyimpulkan dari 19 skenario tes yang dilakukan, 15 diantaranya berhasil memenuhi hasil yang diharapkan oleh peneliti. Sehingga dapat dihasilkan angka kesesuaian sebesar 78.95%.

Selain menguji reliabilitas aplikasi menggunakan metode *black box*, peneliti juga menggunakan metode pengujian menggunakan *script* pengirim pesan Whatsapp untuk menguji integritas antara model dengan aplikasi yang telah dibuat. Peneliti akan mengirimkan pesan yang telah ditandai sebelumnya sebagai pesan yang aman dan pesan yang berbahaya. Peneliti akan melihat apakah pesan – pesan yang dikirim memiliki hasil klasifikasi yang sama dengan yang telah peneliti tetapkan. Jika pesan tersebut ditandai sebagai pesan aman, maka skenario tes memiliki hasil yang diharapkan, sebaliknya jika pesan tersebut tidak diklasifikasikan sesuai dengan apa yang telah ditetapkan peneliti sebelumnya, maka skenario tersebut dinyatakan gagal.

Dengan 2 metode pengujian aplikasi yang peneliti gunakan, diharapkan hasil yang didapatkan dapat digunakan sebagai data yang menunjukkan bahwa model yang dibuat dengan metode *pre-processing* TF-IDF dan algoritma Random Forest dapat bekerja dengan baik. Data ini juga diharapkan dapat merepresentasikan reliabilitas aplikasi dalam melakukan filtrasi terhadap pesan yang masuk ke dalam perangkat pengguna

3.1.5 Penulisan Laporan

Tahap terakhir pada tahap penelitian yang peneliti lakukan adalah menulis laporan. Tahap ini merupakan tahap dengan tujuan agar penelitian yang dilakukan oleh peneliti dapat terdokumentasi dengan baik. Laporan akan dituliskan dalam bentuk skripsi yang mencakup pendahuluan, kajian pustaka, metode penelitian, hasil dan pembahasan, dan simpulan, saran, dan rekomendasi.

3.2 Perangkat Penunjang Penelitian

Selama penelitian berlangsung, peneliti menggunakan perangkat keras berupa *laptop* dengan spesifikasi prosesor Intel Core i5 – 1135G7, RAM 16 GB, dan penyimpanan internal SSD 512 GB. Aplikasi yang digunakan sebagai *tools* dalam pengembangan model *machine learning* yang digunakan pada penelitian ini adalah *code editor* Visual Studio Code. Dalam pengembangan aplikasi untuk sistem

operasi android, android menyediakan *Integrated Development Environment* (IDE) bernama Android Studio. *Github* digunakan selama penelitian berlangsung dengan tujuan sebagai *versioning* selama pengembangan model *machine learning* dan aplikasi android. Perangkat yang digunakan sebagai bahan uji coba untuk fungsi dari aplikasi adalah *smartphone* dengan sistem operasi android versi 13, dengan RAM 8 GB, dan *processor* Snapdragon 680.