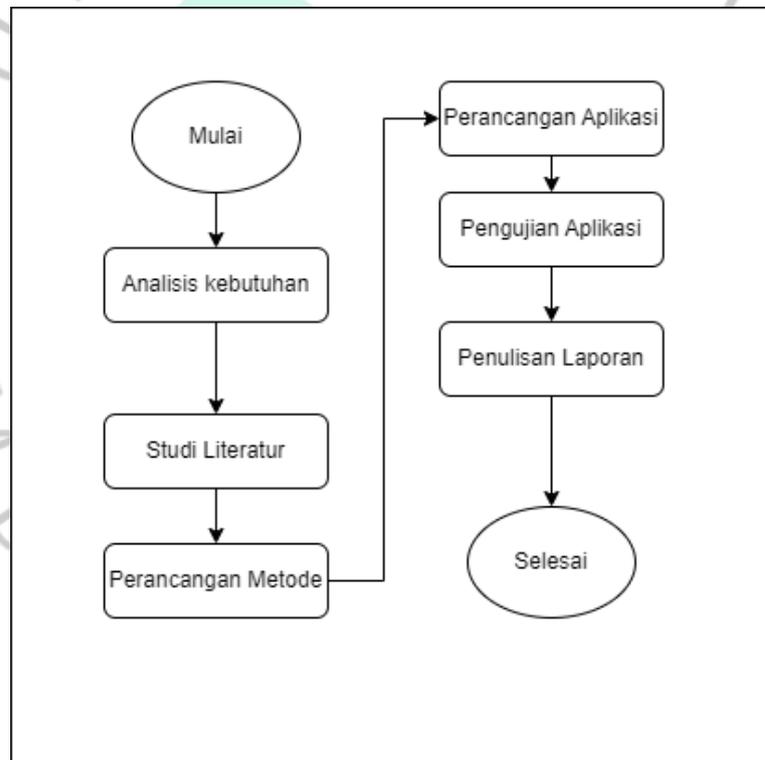


BAB III TAHAPAN PELAKSANAAN

Pada tahapan pelaksanaan membahas proses untuk merancang aplikasi Surat Keterangan Bebas Temuan menggunakan *Algoritma Decision Tree* berbasis web dari awal hingga selesai.

3.1 Langkah-langkah Pelaksanaan

Pada tahap ini dilakukan proses implementasi dalam melaksanakan tugas akhir. Diagram alir digunakan oleh peneliti untuk menggambarkan proses untuk menyelesaikan sebuah proyek, dimulai dari awal sampai dengan penulisan laporan tugas akhir. Berikut adalah rancangan proses pelaksanaan tugas akhir.



Gambar 3.1 Diagram Alir Langkah-langkah Pelaksanaan Tugas Akhir

1) Analisis Kebutuhan

Dalam tahap analisis kebutuhan untuk "Perancangan Aplikasi Surat Keterangan Bebas Temuan Menggunakan *Algoritma Decision Tree* Berbasis Web," mencakup berbagai aspek penting untuk memastikan fungsionalitas dan efisiensi aplikasi. Aplikasi ini harus mampu menerima input data temuan dari berbagai sumber. Implementasi *Algoritma Decision Tree* harus memungkinkan pembagian dataset berdasarkan atribut yang paling signifikan, membentuk pohon keputusan yang dapat mengklasifikasikan data temuan dengan akurat. Hasil klasifikasi harus dapat ditampilkan sebagai evaluasi model dan Surat Keterangan Bebas Temuan dapat diunduh dalam format PDF atau format lainnya. Aplikasi ini juga memerlukan antarmuka pengguna yang intuitif dan responsif, memungkinkan pengguna untuk memasukkan data, menjalankan klasifikasi, dan mengunduh laporan dengan mudah. Aplikasi harus memiliki kinerja yang efisien, mampu memproses data dalam waktu yang cepat bahkan untuk dataset yang besar, serta keandalan yang tinggi dengan minimal downtime. Terakhir, kode sumber aplikasi harus mudah dipahami dan dimodifikasi untuk memudahkan pemeliharaan dan pembaruan di masa mendatang.

2) Studi Literatur

Dalam tahap penentuan kebutuhan studi literatur fokus utamanya adalah memperoleh pemahaman yang komprehensif tentang topik terkait. studi literatur dilakukan untuk memahami secara mendalam konsep dan penerapan *Algoritma Decision Tree* serta prinsip-prinsip desain aplikasi berbasis web. Literatur yang ditelaah meliputi penjelasan tentang *Algoritma Decision Tree*, termasuk struktur, cara kerja, serta berbagai metode implementasinya dalam sistem informasi. Penelitian yang relevan akan dikaji untuk menilai keefektifan algoritma dalam pengambilan keputusan dan bagaimana algoritma ini dapat diintegrasikan dalam aplikasi berbasis web untuk menghasilkan hasil analisis data temuan yang akurat.

3) Perancangan Metode

Aplikasi ini dirancang untuk mendukung penerbitan Surat Keterangan Bebas Temuan dengan memanfaatkan Algoritma Decision Tree. Aplikasi ini harus mampu mengelola dan menganalisis data temuan secara efisien. Pengguna dapat menginput data yang berisi kolom data temuan, yang kemudian diverifikasi untuk memastikan kelengkapan dan validitas, termasuk pengisian data kosong dan encoding data kategorikal. Data yang valid akan dibagi menjadi set pelatihan dan pengujian untuk melatih model. Aplikasi ini juga harus memiliki kemampuan untuk evaluasi model. Selain itu, tabel klasifikasi yang menampilkan data asli, prediksi model, dan atribut input akan memudahkan analisis hasil klasifikasi. Dengan demikian, aplikasi ini bertujuan untuk meningkatkan akurasi dan efisiensi proses penerbitan SKBT.

4) Perancangan Aplikasi

Tahapan perancangan aplikasi dimulai dengan pembuatan *Use Case Diagram* untuk memetakan interaksi antara pengguna dan sistem, mencakup skenario seperti mengunggah dataset, melatih model *Decision Tree*, melihat hasil klasifikasi, dan mengunduh laporan PDF. Selanjutnya, *Activity Diagram* dibuat untuk menggambarkan alur kerja dari pengunggahan file hingga evaluasi model, menunjukkan proses pemrosesan dataset dan visualisasi hasil. *Sequence Diagram* digunakan untuk memetakan komunikasi antara komponen utama aplikasi, termasuk interaksi pengguna dengan antarmuka web dan pemrosesan data oleh backend. Setelah perancangan UML selesai, implementasi dimulai dengan pengembangan antarmuka menggunakan Flask untuk menangani permintaan pengguna. Dataset diproses dan digunakan untuk melatih model *Decision Tree* dengan pustaka *sklearn*, menghasilkan metrik evaluasi seperti *Accuracy*, *Precision*, dan *Recall*. Hasil klasifikasi ditampilkan dalam tabel informatif dan memungkinkan pengguna mengunduh laporan dalam format PDF atau ZIP. Visualisasi pohon keputusan juga disediakan untuk membantu pengguna memahami alur pengambilan keputusan model. Proses diakhiri dengan pengujian fungsionalitas berdasarkan diagram yang telah dirancang, memastikan aplikasi berfungsi

sesuai skenario yang ditetapkan. Aplikasi dinyatakan selesai setelah semua fungsi terintegrasi dengan baik dan memenuhi kebutuhan pengguna.

5) Pengujian Aplikasi

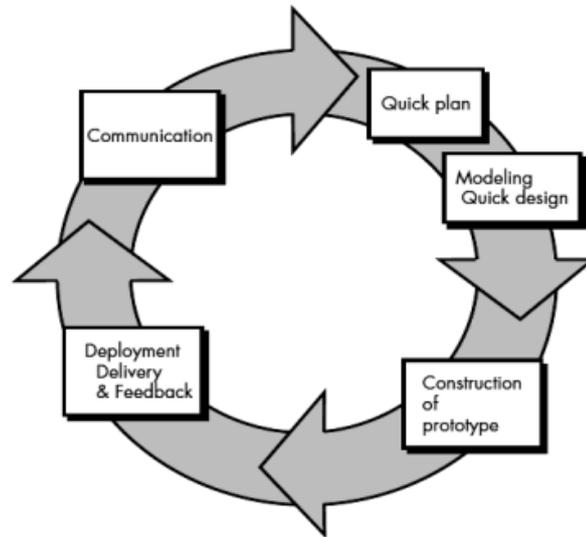
Tahap pengujian aplikasi dimulai dengan pengujian unit, di mana Setiap komponen diuji secara individu untuk memastikan bahwa fungsinya berjalan dengan baik. sesuai dengan spesifikasi. Selanjutnya, pengujian integrasi dilakukan untuk memastikan bahwa setiap komponen dapat bekerja secara harmonis ketika digabungkan, serta untuk memverifikasi alur data dan interaksi antar modul. Pengujian fungsional dilaksanakan untuk mengevaluasi apakah aplikasi beroperasi sesuai dengan fungsionalitas yang diharapkan, termasuk akurasi dalam pengolahan data dan keluaran yang dihasilkan. Selain itu, pengujian *black box* diterapkan untuk menilai fungsionalitas aplikasi berdasarkan input dan output tanpa mempertimbangkan struktur internal, sedangkan pengujian *white box* dilakukan untuk menganalisis logika dan alur kontrol dalam kode sumber, memastikan bahwa semua jalur program telah diuji secara menyeluruh. Dengan demikian, kedua pendekatan ini saling melengkapi dalam menjamin kualitas dan keandalan aplikasi secara keseluruhan.

6) Penulisan Laporan

Tahap penulisan laporan dimulai dengan pengumpulan data dan informasi relevan dari proses pengembangan aplikasi, termasuk analisis kebutuhan, desain, pengujian, dan evaluasi. Struktur laporan disusun dengan bagian utama seperti pendahuluan, tinjauan literatur, metodologi, hasil, diskusi, dan kesimpulan, yang menyajikan informasi secara sistematis dari latar belakang hingga interpretasi hasil pengujian. Penulisan dilakukan dengan rinci pada setiap bagian, diikuti oleh penyuntingan untuk memastikan kejelasan dan akurasi. Setelah disunting, revisi dilakukan berdasarkan umpan balik untuk meningkatkan kualitas akhir. Laporan difinalisasi dengan menyertakan elemen tambahan seperti daftar pustaka dan lampiran, sebelum disampaikan kepada pihak terkait sesuai format dan persyaratan yang ditetapkan. Proses ini menjamin laporan menyajikan gambaran lengkap dan profesional tentang proyek yang telah dilaksanakan.

3.2 Pengembangan aplikasi dengan menggunakan metode prototype

Metode ini adalah strategi antara pengembang dan calon pengguna dalam menciptakan model perangkat lunak yang akan dikembangkan. (Angriani & Saharaeni, 2023). Metode pengembangan prototype dapat dibagi menjadi beberapa tahap yang sistematis. Berikut adalah langkah-langkah yang dapat diikuti :



Gambar 3.2 Diagram metode pengembangan Prototype

Sumber Gambar: Angriani & Saharaeni, 2023

Berdasarkan metode pengembangan prototype yang tercantum pada Gambar 3.2 berikut adalah penjelasan lengkap dari masing-masing alurnya.

1) *Communication*

Tahapan ini merupakan tahapan awal dalam menganalisis kebutuhan yaitu melakukan komunikasi dengan user atau admin tentang program yang akan dikembangkan. Pada bagian ini mencakup berbagai tahapan diantaranya menganalisa sistem administrasi Surat Keterangan Bebas Temuan, dan menggali permasalahan yang muncul dalam proses yang masih dilakukan secara manual.

2) *Quick Plan*

Tahap ini adalah tahap di mana peneliti merancang strategi secara terencana dan cepat untuk memberikan solusi berdasarkan identifikasi awal. Dalam penelitian ini, sebelum membangun sistem, peneliti menyusun sistem usulan dalam bentuk flowchart untuk administrasi Surat Keterangan Bebas Temuan yang dirancang agar lebih efisien dan mudah dipahami.

3) *Modeling Quick Design*

Pada tahap ini dilakukan perancangan desain untuk memberikan gambaran fungsional sistem Surat Keterangan Bebas Temuan. Pengembang membuat rancangan use case untuk menampilkan fungsi bisnis serta penjelasan terkait alur proses bisnis yang akan dijalankan oleh sistem, termasuk interaksi antara proses dan aktor yang terlibat.

4) *Construction Of Prototype*

Tahap ini merupakan tahap *construction of prototype*, yang bertujuan untuk membuat desain antarmuka sesuai dengan susunan rencana tahap sebelumnya. Desain ini hanya berfungsi untuk memastikan kebutuhan pengguna, tanpa diterapkan langsung dalam sistem. Desain antarmuka ini dirancang semata-mata agar pengguna mendapatkan gambaran yang lebih jelas sesuai dengan kebutuhan mereka.

5) *Deployment Delivery & Feedback*

Setelah semua rancangan dinyatakan sesuai, pengembang menyerahkan *prototype* kepada pemangku kepentingan atau instansi terkait untuk dilakukan evaluasi dan mendapatkan masukan terhadap rancangan yang telah dibuat. Hal ini memungkinkan pengembang memperoleh gambaran yang lebih jelas untuk menyempurnakan rancangan sistem di tahap berikutnya.

3.3 Metode Pengujian

Untuk menilai aplikasi, penelitian ini menggunakan metode pengujian kotak hitam, dengan fokus pada temuan output dan input. Selain itu, dengan menguji kode komputer, metode kotak putih digunakan untuk menguji struktur internal aplikasi.

3.3.1 Black Box

Pengujian kualitas program yang berkonsentrasi pada fungsi program dikenal sebagai pengujian "black box". Metode ini berfokus pada evaluasi input dan output suatu program tanpa memeriksa kode sumber atau implementasi internalnya, sehingga hanya perlu memahami spesifikasi dan fungsionalitas yang diharapkan. Tujuan utama dari pengujian kotak hitam adalah untuk mengidentifikasi kesalahan fungsi, masalah antarmuka, kesalahan dalam struktur data, kendala kinerja, serta masalah pada proses inisialisasi dan terminasi. (Setiyani, 2019). Istilah "black box" digunakan untuk menggambarkan sistem yang diuji dan digunakan tanpa operator memahami prosedur internalnya, sehingga metode ini sangat berguna untuk mengidentifikasi kesalahan pada level fungsional, baik dalam perangkat lunak individual maupun dalam sistem yang lebih kompleks. Hal ini juga memungkinkan pengujian untuk tetap obyektif dalam mengevaluasi apakah perangkat lunak memenuhi kebutuhan pengguna dan persyaratan yang ditentukan.

3.3.2 White Box

Menurut Farhan Londjo (2021), whitebox testing merupakan Teknik pengujian yang dirancang dengan memanfaatkan kendali struktur yang telah ditentukan dalam desain komponen, untuk memastikan setiap bagiannya berfungsi sesuai tujuan. Pengujian ini dilakukan dengan menganalisis kode sumber secara langsung untuk memastikan bahwa setiap jalur logika dalam program telah diuji dan berfungsi sebagaimana mestinya. White box digunakan untuk memeriksa organisasi kode, logika pemrograman, dan fungsionalitas program, termasuk validasi alur kontrol, validasi data, serta pengujian jalur eksekusi yang mencakup semua cabang dan loop. Pengujian ini sangat berguna untuk mendeteksi kesalahan tersembunyi yang mungkin tidak terdeteksi melalui pengujian black box, seperti kesalahan dalam algoritma, perhitungan, atau pengelolaan memori.