

BAB III

TAHAPAN PELAKSANAAN

Pada tahap pelaksanaan penelitian ini, akan mencakup tentang bagaimana peneliti menyelesaikan tugas akhirnya. Tahapan tersebut akan terdiri dari serangkaian langkah pelaksanaan dan metode pengujian yang digunakan oleh peneliti.

3.1 Langkah-langkah Pelaksanaan

Berikut ini adalah tahap berurut yang direncanakan untuk dilalui dalam pelaksanaan penelitian ini, yaitu sebagai berikut. Lihat pada Diagram 3.1 Langkah-langkah Pelaksanaan.

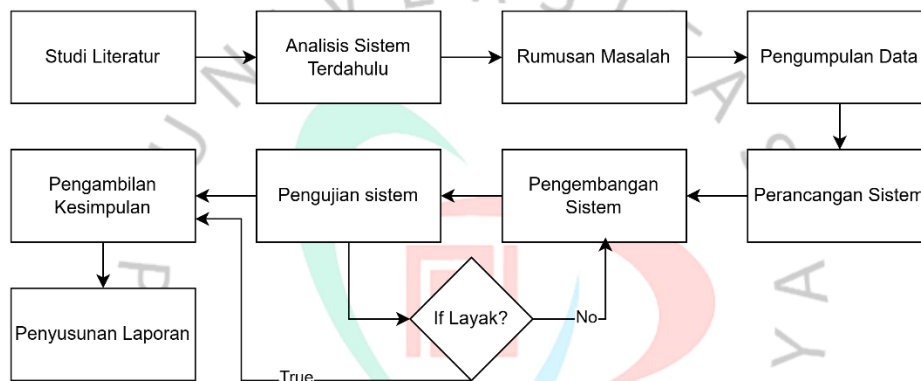


Diagram 3. 1 Langkah-langkah Pelaksanaan

Penjelasan mendalam pada langkah-langkah pelaksanaan antara lain sebagai berikut.

1. Studi Literatur

Dilakukan penelaahan terhadap teori, konsep, dan penelitian sebelumnya yang terkait dengan machine learning, model LSTM, dan analisis kelayakan produk asuransi. Langkah ini bertujuan untuk memahami pendekatan yang telah ada dan mengidentifikasi celah penelitian.

2. Analisis Sistem Terdahulu

Mengevaluasi sistem atau metode yang sudah ada dalam menentukan kelayakan produk asuransi, termasuk pendekatan manual maupun berbasis teknologi. Hasil analisis digunakan sebagai acuan untuk merancang model yang lebih baik.

3. Rumusan Masalah

Merumuskan masalah utama yang hendak diatasi oleh penelitian, seperti keterbatasan metode saat ini dalam menganalisis faktor loss ratio dan variabel pendukung lainnya untuk menentukan kelayakan produk asuransi.

4. Pengumpulan Data

Mengumpulkan data historis yang relevan, seperti data premi, klaim, loss ratio, dan faktor pendukung lainnya dari perusahaan asuransi untuk digunakan sebagai dataset pelatihan dan pengujian model.

5. Perancangan Sistem

Mendesain arsitektur model LSTM, termasuk struktur jaringan, fitur yang digunakan, dan proses input-output sistem. Selain itu, alur kerja sistem yang mencakup pengolahan data, pelatihan model, dan evaluasi juga dirancang pada tahap ini.

6. Pengembangan Sistem

Melakukan implementasi model LSTM menggunakan alat dan bahasa pemrograman yang sesuai (misalnya Python dengan pustaka TensorFlow atau PyTorch). Sistem dikembangkan secara iteratif berdasarkan desain yang telah dibuat.

7. Pengujian sistem

Menguji kinerja model dengan menggunakan data uji untuk mengevaluasi akurasi, presisi, dan kemampuan prediksi sistem dalam menentukan nilai kelayakan produk asuransi. Hasil pengujian digunakan untuk melakukan perbaikan atau optimasi model jika diperlukan.

8. Pengambilan Kesimpulan

Menyimpulkan hasil penelitian berdasarkan performa model dan kontribusinya dalam meningkatkan akurasi dan efisiensi analisis kelayakan produk asuransi.

9. Penyusunan Laporan

Menyusun laporan akhir yang mencakup seluruh tahapan penelitian, hasil pengujian, analisis, dan kesimpulan, sebagai dokumentasi resmi dari hasil penelitian dan pengembangan model machine learning berbasis LSTM.

3.2 Langkah-langkah Penerapan Algoritma

Berikut adalah kumpulan langkah untuk penerapan LSTM dalam pengembangan model *machine learning* pada penelitian ini. Kumpulan langkah tersebut meliputi proses pengumpulan data, pembersihan data, pengolahan data, hingga tahapan lebih lanjut yang meliputi pengujian hasil prediksi model. Lihat pada Diagram 3.2 Tahapan Penerapan Algoritma LSTM.

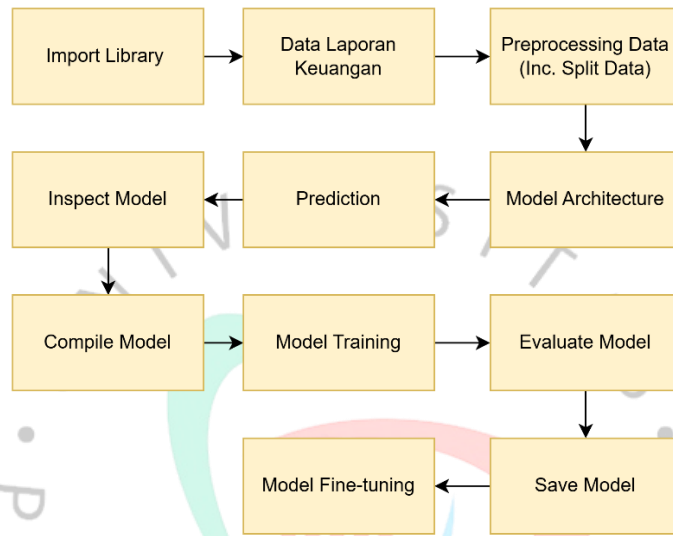


Diagram 3. 2 Tahapan Penerapan Algoritma LSTM

Penjelasan lugas terkait hal-hal yang tercantum pada Diagram 3.2 Tahapan penerapan Algoritma LSTM sebagai berikut.

1. Import Library

Langkah pertama adalah mengimpor pustaka yang diperlukan untuk membangun, melatih, dan mengevaluasi model *machine learning* berbasis LSTM. Pustaka yang digunakan mencakup NumPy dan Pandas untuk manipulasi data, Matplotlib dan Seaborn untuk visualisasi data, TensorFlow atau PyTorch untuk membangun model LSTM, serta Scikit-learn untuk preprocessing data dan evaluasi.

2. Data Laporan Keuangan

Data laporan keuangan yang digunakan mencakup informasi seperti premi, klaim, loss ratio, dan faktor pendukung lainnya dari produk asuransi. Data ini disiapkan sebagai sumber informasi utama untuk melatih dan menguji model.

3. Preprocessing dan Split Data

Proses preprocessing dan pembagian data dilakukan untuk membersihkan dataset, seperti mengisi nilai kosong, normalisasi data, dan encoding variabel kategorikal. Data kemudian dibagi menjadi data pelatihan, validasi, dan pengujian untuk memastikan evaluasi yang adil terhadap performa model.

4. Model Architecture

Arsitektur model dirancang dengan membangun lapisan LSTM yang mampu menganalisis pola temporal dari data sekuensial, dilengkapi dengan lapisan dense untuk menghasilkan output. Struktur jaringan, jumlah neuron, dan fungsi aktivasi dirancang sesuai kebutuhan untuk memastikan kinerja optimal.

5. Prediction

Prediksi dilakukan setelah model selesai dilatih. Data pengujian dimasukkan ke dalam model untuk menghasilkan output yang mewakili nilai kelayakan produk asuransi berdasarkan variabel yang dianalisis.

6. Inspect Model

Inspeksi model dilakukan untuk memeriksa struktur, jumlah parameter, dan distribusi bobot dalam model. Analisis tambahan, seperti visualisasi kurva loss atau analisis error, dilakukan untuk mengevaluasi performa dan potensi perbaikan.

7. Compile Model

Model dikompilasi dengan menetapkan fungsi loss, optimizer, dan metrik evaluasi. Fungsi loss seperti Mean Squared Error (MSE) atau Binary Crossentropy digunakan sesuai dengan tipe output, sementara optimizer seperti Adam dipilih untuk mempercepat konvergensi.

8. Model Training

Model dilatih menggunakan data pelatihan dengan sejumlah iterasi (epochs) dan batch size tertentu. Selama pelatihan, performa model dipantau melalui metrik seperti akurasi atau loss untuk memastikan peningkatan kinerja.

9. Evaluate Model

Evaluasi model dilakukan menggunakan data pengujian untuk mengukur seberapa baik model dapat memprediksi nilai kelayakan. Metrik seperti MSE, akurasi, atau F1-Score digunakan untuk menilai efektivitas model dalam tugas yang diberikan.

10. Save Model

Model yang telah selesai dilatih dan diuji disimpan dalam format yang dapat diakses kembali, seperti HDF5 atau SavedModel, untuk memastikan penggunaannya tanpa perlu melatih ulang di masa mendatang.

11. Model Fine-tuning

Model kemudian disempurnakan melalui proses fine-tuning, dengan menyesuaikan hyperparameter seperti learning rate atau struktur lapisan. Fine-tuning juga dapat melibatkan pelatihan ulang model dengan dataset yang lebih luas atau berkualitas untuk meningkatkan kinerja dan keandalannya.

3.3 Metode Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak merupakan acuan untuk merancang sistem secara sistematis dan terstruktur. Salah satu metode yang digunakan adalah Metode Waterfall, yaitu pendekatan pengembangan perangkat lunak yang bersifat sekuensial, di mana setiap tahapan harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya. Metode ini umum digunakan saat pengembangan aplikasi sudah memiliki kejelasan dan tidak terdapat perubahan signifikan. Penerapan metode Waterfall cocok untuk penelitian dengan lingkup kerja tetap atau yang memiliki standar prosedur umum yang telah terdokumentasi dengan jelas. Metode ini sering diterapkan di berbagai industri, seperti industri bisnis asuransi, karena menciptakan dokumentasi aplikasi yang lebih jelas dan terstruktur pada setiap tahap pengembangan. Setiap tahap, mulai dari analisis kebutuhan hingga pemeliharaan, menghasilkan dokumen mendetail yang menjadi panduan dan referensi, terutama dalam menghadapi perubahan regulasi. Dokumentasi ini mempermudah tim untuk memahami sistem secara menyeluruh dan melakukan penyesuaian secara terorganisir.

Dibandingkan dengan Agile, yang lebih menekankan iterasi cepat dan komunikasi langsung, Waterfall memberikan dasar dokumentasi yang kuat sehingga pengembangan tetap berjalan sesuai kerangka yang dirancang. Dokumentasi yang terstruktur ini memungkinkan tim melacak perubahan dan dampaknya pada keseluruhan sistem, menjadikan pengembangan lebih terkendali, bahkan dalam kondisi yang kompleks. Secara garis besar, mekanisme kerja metode Waterfall dilakukan dengan menyelesaikan tahapan-tahapan berurutan:

perencanaan, analisis kebutuhan, desain aplikasi, penerapan, pengujian, peluncuran, dan pemeliharaan. Lihat Diagram 3.3 Waterfall.

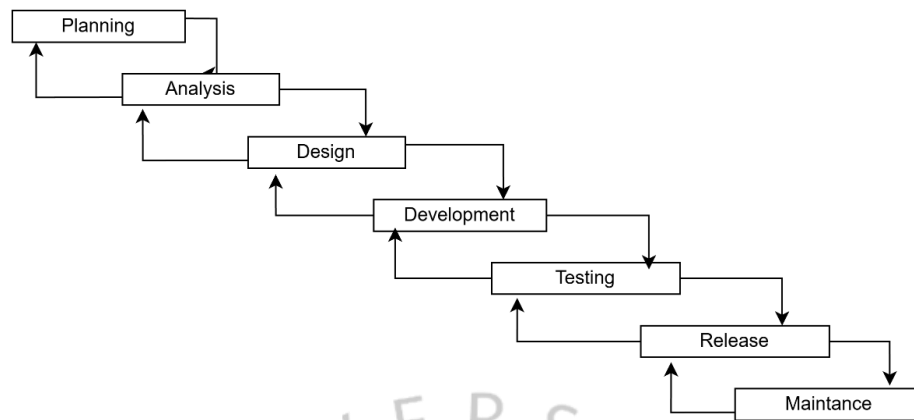


Diagram 3. 3 Metode Waterfall

Penjelasan lugas terkait hal-hal yang tercantum pada Diagram 3.3 Waterfall sebagai berikut.

a) *Planning*

Tahap perencanaan (planning) dimulai dengan menentukan tujuan utama penelitian, yaitu mengembangkan model berbasis LSTM untuk menganalisis kelayakan produk asuransi. Pada tahap ini, ruang lingkup penelitian ditentukan, termasuk data yang akan digunakan, alat dan teknologi yang dibutuhkan, serta risiko yang mungkin muncul selama pengembangan.

b) *Analysis*

Tahap analisis (analysis) dilakukan untuk memahami kebutuhan sistem, seperti data yang relevan (laporan keuangan, premi, klaim, dan loss ratio) dan variabel lain yang memengaruhi kelayakan produk asuransi. Analisis ini juga mencakup evaluasi sistem atau metode yang telah ada untuk memastikan bahwa model LSTM yang dikembangkan dapat memberikan hasil yang lebih baik.

c) *Design*

Tahap desain (design) bertujuan untuk merancang arsitektur model LSTM yang akan digunakan. Ini meliputi perancangan struktur jaringan, pemilihan fitur input, alur kerja sistem, dan parameter model yang akan digunakan. Pada tahap ini, desain diagram dan prototipe model juga dibuat untuk memberikan gambaran teknis yang jelas.

d) Development

Tahap pengembangan (development) melibatkan implementasi desain model ke dalam kode menggunakan alat seperti TensorFlow atau PyTorch. Data yang telah diproses dimasukkan ke dalam sistem, dan model LSTM dibangun untuk mempelajari pola dari data tersebut. Pengkodean dan pengujian unit dilakukan untuk memastikan model berfungsi sesuai dengan spesifikasi.

e) Testing

Tahap pengujian (testing) dilakukan untuk mengevaluasi performa model dalam memprediksi kelayakan produk asuransi. Data pengujian digunakan untuk mengukur akurasi model, sementara metrik seperti MSE atau precision digunakan untuk menilai keefektifan model. Hasil pengujian ini akan menentukan apakah model perlu disempurnakan.

f) Release

Tahap rilis (release) adalah penyebaran model LSTM yang telah diuji ke lingkungan produksi atau simulasi untuk penggunaannya dalam analisis kelayakan produk asuransi. Sebelum rilis, model difinalisasi, didokumentasikan, dan, jika diperlukan, pelatihan pengguna dilakukan agar sistem dapat dioperasikan dengan baik.

3.4 Metode Pengujian

Dalam pengujian model dan sistem yang mampu melakukan identifikasi kelayakan bisnis perusahaan berdasarkan data laporan keuangan menggunakan pendekatan *Research and Development (RnD)* dan *Prototyping*.

3.4.1 Research and Development (RnD)

Pengujian *Research and Development (R&D)* adalah proses evaluasi yang dilakukan untuk menguji kelayakan konsep, model, atau sistem dalam mendukung pengembangan inovasi. Proses ini bertujuan untuk mengidentifikasi kekuatan dan

kelemahan model, menguji fungsionalitas dan performa, serta memastikan relevansi terhadap kebutuhan pengguna atau pasar. Dalam pengujian R&D, data dan analisis digunakan untuk mengevaluasi efektivitas sistem dan memberikan dasar untuk pengembangan lebih lanjut. Keunggulan utama pengujian R&D adalah kemampuannya untuk mengurangi risiko pengembangan, meningkatkan akurasi desain, dan memastikan bahwa inovasi yang dihasilkan sesuai dengan tujuan. Meskipun demikian, metode ini memerlukan sumber daya yang besar dan waktu yang cukup untuk mendapatkan hasil yang optimal.

3.5.2 Prototyping

Pengujian prototyping adalah proses evaluasi model awal atau prototipe sistem untuk memastikan bahwa desain, fungsionalitas, dan kebutuhan pengguna dapat terpenuhi sebelum pengembangan penuh dilakukan. Proses ini melibatkan pembuatan prototipe sederhana yang diuji oleh pengguna untuk mengumpulkan umpan balik, yang kemudian digunakan untuk memperbaiki dan menyempurnakan desain melalui siklus iterasi. Pengujian prototyping bertujuan untuk mengidentifikasi masalah sejak awal, memvalidasi kebutuhan, dan mengurangi risiko pengembangan. Keuntungan utama metode ini adalah memastikan sistem lebih relevan dengan kebutuhan pengguna, meskipun ada kekurangan seperti ketergantungan pada kualitas umpan balik dan lingkup prototipe yang terbatas. Metode ini efektif untuk proyek yang membutuhkan fleksibilitas dan penyesuaian berbasis kebutuhan pengguna.

3.5.2 RMSE

Root Mean Squared Error (RMSE) adalah sebuah indikator dalam mengukur keakuratan model prediksi dalam memreplikasi nilai aktual. Nilai RMSE yang diperoleh digunakan sebagai acuan untuk menilai kualitas model yang dikembangkan. Pengujian Nilai RMSE akan dihitung ketika pengembangan model telah mampu mengidentifikasi kelayakan produk asuransi. RMSE dilakukan bersamaan ketika model *machine learning* sedang berjalan atau tereksekusi bersamaan dengan data yang diolah. RMSE telah umum diterapkan dalam pengujian model lantaran memberikan ukuran kesalahan prediksi yang dikomperasikan dengan nilai aktual, sehingga membantu mengidentifikasi seberapa

berbeda antara selisih nilai aktual dengan nilai prediksi. Dalam proses mengevaluasi nilai RMSE, semakin kecil nilai maka model prediksi mendekati keakuratan yang baik dengan nilai aktual dan sebaliknya. RMSE dilakukan Perhitungan RMSE dapat direpresentasikan dalam bentuk matematis sebagai berikut.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y})^2} \quad (3.1)$$

Keterangan,

- y_i = Nilai aktual
- y = Nilai prediksi dari model
- n = Jumlah data

3.5.2 MAE

Mean Absolute Error (MAE) adalah metrik evaluasi yang digunakan untuk mengukur rata-rata kesalahan nilai absolut antara nilai prediksi dan nilai aktual dalam model *machine learning*. MAE adalah ukuran sederhana dari tingkat kesalahan prediksi yang dihitung dengan rata-rata nilai absolute selisih antara nilai prediksi dengan nilai aktual. MAE memberikan gambaran langsung tentang seberapa besar kesalahan prediksi tanpa memberikan bobot tambahan sehingga mudah untuk diinterpretasikan. MAE umum digunakan untuk mengevaluasi model *machine learning*, ketika semua kesalahan diperlukan secara sama. Dalam penerapan MAE pada model umum digunakan pada proses prediksi dan identifikasi kelayakan produk. MAE dapat direpresentasikan dalam bentuk matematis sebagai berikut.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (3.2)$$

Keterangan,

- y_i = Nilai aktual
- $y_{\hat{i}}$ = Nilai prediksi dari model
- n = Jumlah data

