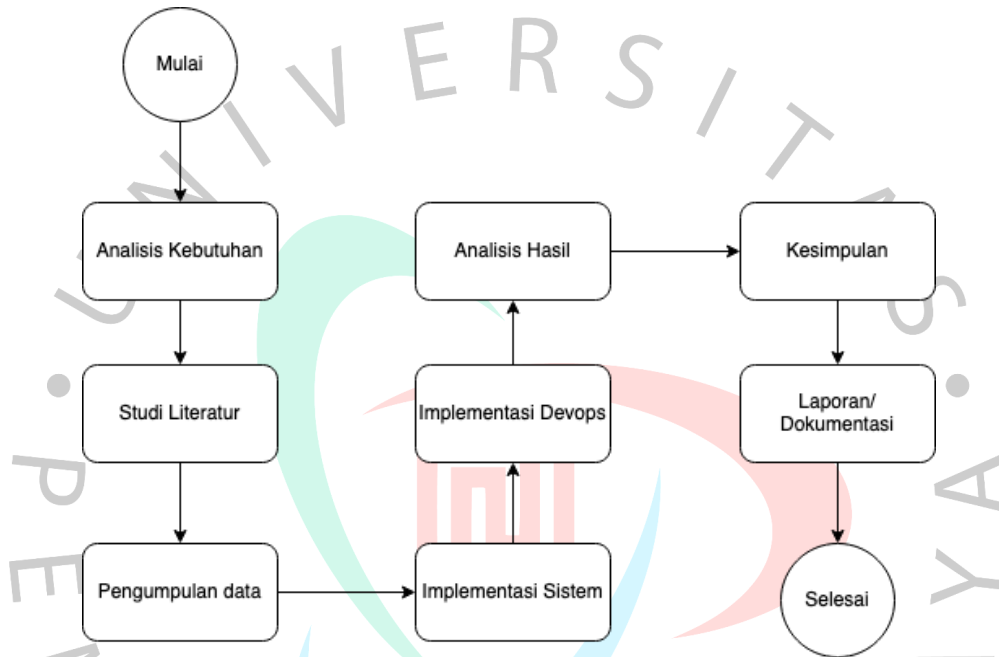


BAB III

TAHAPAN PELAKSANAAN

3.1 Langkah-langkah Pelaksanaan

Dalam kerangka pengembangan sistem automasi DevOps untuk optimalisasi biaya cloud di e-commerce, terdapat serangkaian langkah pelaksanaan yang terstruktur dan sistematis. Berikut adalah detail dari masing-masing tahap pelaksanaan yang dijelaskan pada gambar 3.1



Gambar 3.1 Tahapan Pelaksanaan

Tahap-tahap pelaksanaan ini dirancang untuk mendukung pengembangan sistem yang efektif dan efisien, dimulai dari identifikasi kebutuhan hingga penyusunan laporan akhir. Berikut adalah penjelasan mendalam mengenai masing-masing tahapan yang terdapat dalam gambar tersebut:

1. Analisis kebutuhan

Tahap pertama dalam pelaksanaan sistem adalah analisis kebutuhan. Di sini, identifikasi kebutuhan bisnis, teknis, dan operasional dilakukan untuk memahami permasalahan yang dihadapi dalam pengelolaan sumber daya cloud di lingkungan *e-commerce*. Fokus utama adalah optimalisasi biaya cloud, di mana tantangan pengelolaan infrastruktur dan sumber daya yang efisien menjadi prioritas utama dalam perancangan sistem.

2. Studi literatur

Pada tahap ini, dilakukan pengumpulan pengetahuan terkait teori dan konsep yang relevan dengan penelitian, termasuk DevOps, *cloud computing*, dan teknik *optimisasi biaya* di lingkungan *e-commerce*. Penelitian sebelumnya, jurnal ilmiah, dan literatur lainnya digunakan untuk memberikan dasar yang kuat bagi perancangan dan implementasi sistem yang lebih baik.

3. Pengumpulan data

Tahap pengumpulan data melibatkan pengumpulan data terkait penggunaan sumber daya cloud yang relevan dalam pengelolaan infrastruktur *e-commerce*. Data yang dikumpulkan meliputi informasi mengenai konfigurasi VM, penggunaan sumber daya, serta biaya operasional yang dikeluarkan selama periode tertentu. Data ini digunakan untuk analisis lebih lanjut dalam proses pelatihan model dan optimasi biaya.

4. Implementasi system

Tahap ini melibatkan penerapan praktik DevOps dalam sistem yang telah dikembangkan. Fokus utamanya adalah mengintegrasikan proses otomatisasi, seperti *continuous integration* dan *continuous deployment (CI/CD)*, dengan alur kerja DevOps di lingkungan *e-commerce*. Ini termasuk pengujian berkelanjutan dan penerapan perubahan otomatis ke dalam produksi, sehingga meningkatkan efisiensi dan mengurangi intervensi manual dalam pengelolaan infrastruktur.

5. Implementasi *DevOps*

Tahap ini melibatkan penerapan praktik DevOps dalam sistem yang telah dikembangkan. Fokus utamanya adalah mengintegrasikan proses otomatisasi, seperti *continuous integration* dan *continuous deployment (CI/CD)*, dengan alur kerja DevOps di lingkungan *e-commerce*. Ini termasuk pengujian berkelanjutan dan penerapan perubahan otomatis ke dalam produksi, sehingga meningkatkan efisiensi dan mengurangi intervensi manual dalam pengelolaan infrastruktur.

6. Analisis hasil

Setelah implementasi DevOps, tahap selanjutnya adalah melakukan analisis hasil untuk mengevaluasi efektivitas sistem yang telah diterapkan. Analisis ini bertujuan untuk menilai seberapa baik sistem dapat mengidentifikasi dan mengurangi pemborosan sumber daya cloud serta dampaknya terhadap pengurangan biaya operasional dalam lingkungan *e-commerce*. Hasil analisis ini juga digunakan untuk menyempurnakan sistem lebih lanjut.

7. Kesimpulan

Berdasarkan hasil analisis, ditarik kesimpulan mengenai efisiensi dan efektivitas sistem automasi DevOps yang telah dikembangkan. Kesimpulan ini mencakup sejauh mana sistem berhasil mengoptimalkan penggunaan sumber daya cloud dan mengurangi biaya operasional di sektor *e-commerce*. Temuan ini memberikan dasar yang jelas bagi perusahaan dalam menerapkan sistem serupa untuk kebutuhan mereka.

8. Laporan/Dokumentasi

Tahap akhir melibatkan penyusunan laporan atau dokumentasi yang mencakup seluruh proses pengembangan sistem, mulai dari analisis kebutuhan hingga hasil implementasi dan evaluasi. Laporan ini juga mencakup deskripsi rinci mengenai tahapan pelaksanaan, tantangan yang dihadapi, solusi yang diterapkan, serta hasil dan temuan penting selama penelitian yang dapat digunakan oleh pihak terkait dalam pengelolaan sumber daya cloud secara lebih efisien.

3.2 Metode Pengujian

Pada bagian ini menguraikan tentang metodologi pengujian yang diterapkan dalam penelitian ini, yang mencakup dua pendekatan utama: pengujian *white box* dan metode confusion matrix. Penjelasan terperinci tentang masing-masing metode akan disajikan dalam sub-bab berikut.

3.2.1 Metode A/B Testing

A/B Testing adalah metode pengujian yang membandingkan dua versi sistem atau parameter untuk menentukan mana yang memberikan hasil lebih baik berdasarkan metrik tertentu. Metode ini sering digunakan untuk mengevaluasi perubahan sistem, model, atau konfigurasi, dan pengaruhnya terhadap metrik performa seperti biaya, efisiensi, atau akurasi (Kohavi et al., 2020). Dalam konteks penelitian ini, *A/B Testing* diterapkan untuk membandingkan biaya penggunaan sumber daya cloud sebelum dan sesudah diterapkannya metode optimisasi berbasis *PSO*. Pengujian ini melibatkan langkah-langkah sebagai berikut:

1. Pengumpulan Data
Data biaya sebelum dan sesudah optimisasi dihitung menggunakan data penggunaan sumber daya cloud dan tipe mesin virtual yang direkomendasikan oleh *PSO*.
2. Perhitungan Biaya
Menghitung total biaya untuk kondisi awal (*Cost Before*) dan setelah optimisasi (*Cost After PSO*).
3. Analisis Penghematan
Menghitung penghematan biaya baik dalam nilai absolut (USD) maupun persentase, dengan rumus sebagai berikut:

$$\text{Savings (USD)} = \text{Cost Before} - \text{Cost After PSO}$$

$$\text{Savings (\%)} = \frac{\text{Savings (USD)}}{\text{Cost Before}} \times 100$$

4. Perhitungan *Optimization Recommendation*
Selain penghematan biaya, tahap ini juga mencakup perhitungan *optimization recommendation*. Sistem akan mengevaluasi rekomendasi yang dihasilkan oleh *PSO*, kemudian mengklasifikasikan rekomendasi tersebut sebagai valid atau tidak valid berdasarkan kriteria yang telah ditentukan. Hasil akhirnya akan menunjukkan persentase rekomendasi yang valid dibandingkan dengan total rekomendasi yang dihasilkan oleh sistem.
 - a. Valid Recommendation: Rekomendasi yang menghasilkan penghematan biaya dan memenuhi kriteria performa yang ditetapkan.
 - b. Invalid Recommendation: Rekomendasi yang tidak memberikan penghematan biaya atau tidak sesuai dengan kebutuhan operasional yang sebenarnya. Persentase rekomendasi yang valid dihitung dengan rumus berikut:

$$\text{valid recommendations (\%)} = \frac{\text{Number of valid recommendations}}{\text{Total recommendations}} \times 100$$

Metode ini mendukung pengambilan keputusan berbasis data untuk mengevaluasi efektivitas metode optimisasi yang diterapkan. *A/B Testing* memberikan bukti empiris apakah perubahan konfigurasi sistem melalui PSO benar-benar memberikan dampak positif terhadap biaya operasional. Menurut Kohavi et al. (2020), *A/B Testing* adalah metode yang sangat direkomendasikan untuk mengukur dampak perubahan pada sistem berbasis data karena kemampuannya untuk memberikan hasil yang objektif, valid, dan dapat diandalkan dalam pengujian hipotesis.

3.2.2 Metode *White box*

Pengujian *white box*, sebagaimana dijelaskan oleh Khan, Mohd.E. (2010), berfokus pada analisis struktur internal dan detail teknis perangkat lunak. Metode ini melibatkan pemberian input ke dalam sistem dan mengamati bagaimana sistem tersebut mengolah input untuk menghasilkan keluaran yang diharapkan. Pengujian *white box* memerlukan pemahaman mendalam mengenai kode sumber sistem dan umumnya diterapkan pada tingkat integrasi, unit, serta sistem secara keseluruhan selama proses pengujian.

Dalam konteks penelitian ini, *white box testing* diterapkan untuk *unit testing* pada bagian-bagian tertentu dari sistem automasi DevOps yang telah dikembangkan. Pengujian ini difokuskan untuk memastikan bahwa setiap unit atau modul dalam sistem berfungsi dengan benar sesuai dengan tujuan yang telah ditentukan, terutama dalam pengolahan input yang terkait dengan algoritma dan perhitungan penghematan biaya.

Unit testing dilakukan untuk memverifikasi bahwa setiap bagian dari kode, termasuk komponen sistem yang mengimplementasikan algoritma seperti *PSO* dan *Boosted Decision Tree*, berfungsi sesuai dengan ekspektasi. Ini dilakukan dengan menguji unit-unit kode secara terpisah untuk memverifikasi bahwa mereka memberikan keluaran yang benar ketika diberikan input yang sudah diketahui.

3.2.3 Metode *Confusion Matrix*

Confusion matrix adalah metode evaluasi yang digunakan untuk mengukur kinerja model klasifikasi berdasarkan kombinasi nilai sebenarnya (*actual*) dan nilai prediksi (*prediction*) yang dihasilkan oleh model. *Confusion matrix* sangat berguna dalam masalah klasifikasi dengan dua atau lebih kelas, karena memberikan informasi mengenai hasil prediksi yang benar maupun salah (Putra, Suprpto, & Bukhori, 2022). Matriks ini terdiri dari empat atribut utama yang menjadi dasar perhitungan, yaitu:

1. True Positive (TP): Jumlah data yang diklasifikasikan sebagai positif baik pada kategori prediksi maupun kategori sebenarnya.
2. False Positive (FP): Jumlah data yang diklasifikasikan sebagai positif pada kategori prediksi tetapi sebenarnya bernilai negatif.
3. True Negative (TN): Jumlah data yang diklasifikasikan sebagai negatif baik pada kategori prediksi maupun kategori sebenarnya.
4. False Negative (FN): Jumlah data yang diklasifikasikan sebagai negatif pada kategori prediksi tetapi sebenarnya bernilai positif.

Confusion matrix memungkinkan penghitungan berbagai metrik evaluasi, seperti akurasi, presisi, recall, dan F1-score, yang memberikan pemahaman lebih

mendalam terkait kelebihan dan kelemahan model dalam memprediksi data. Perhitungan dari keempat atribut ini dijelaskan sebagai berikut:

1. Akurasi

Akurasi adalah pendekatan evaluasi yang mengukur seberapa baik prediksi model sesuai dengan nilai aktual. Dengan menghitung jumlah data yang diklasifikasikan dengan benar, tingkat keakuratan dari prediksi tersebut dapat ditentukan.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2. Presisi

Presisi merupakan metode evaluasi yang membandingkan jumlah informasi relevan yang ditemukan oleh sistem dengan total informasi yang diperoleh, termasuk yang relevan maupun yang tidak relevan. Formula untuk menghitung presisi adalah sebagai berikut:

$$precision = \frac{TP}{TP + FP}$$

3. Recall

Recall adalah metode evaluasi yang membandingkan jumlah informasi relevan yang berhasil ditemukan oleh sistem dengan total informasi relevan dalam dataset, baik yang ditemukan maupun yang tidak ditemukan. Formula untuk menghitung recall adalah sebagai berikut:

$$recall = \frac{TP}{TP + FN}$$

4. F1 Score

F1 Score adalah metrik evaluasi yang menghitung rata-rata harmonis antara presisi dan recall, yang bertujuan untuk menemukan keseimbangan antara keduanya. F1 Score sangat bermanfaat terutama dalam kasus di mana terdapat ketidakseimbangan kelas, karena metrik ini memberikan bobot yang sama pada presisi dan recall. Semakin tinggi nilai F1 Score, semakin baik kinerja model dalam mengidentifikasi observasi positif tanpa terlalu banyak menghasilkan kesalahan.

$$F1\ score = 2 \times \frac{precision \times recall}{precision + recall}$$