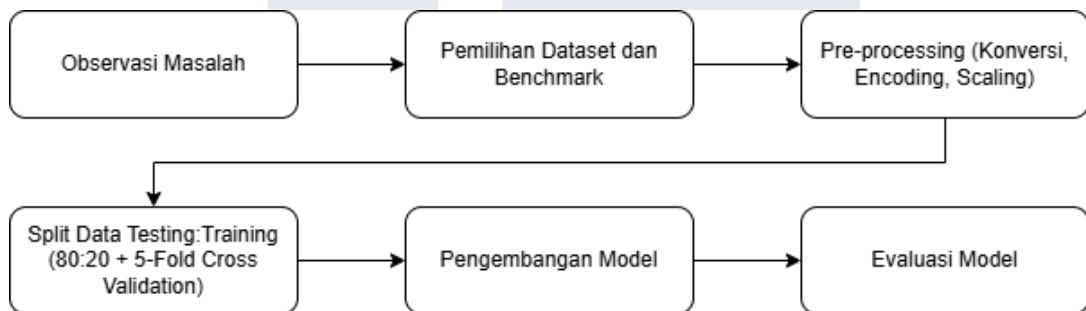


BAB III

METODE PENELITIAN

3.1 Perancangan Solusi

Bab ini menguraikan perancangan solusi yang digunakan untuk mencapai tujuan penelitian. Perancangan solusi disusun dalam beberapa tahapan utama, yaitu observasi masalah, tahap persiapan data, pengembangan model, serta pengembangan skrip prediksi performa. Setiap tahapan dirancang secara sistematis untuk memastikan keterkaitan yang jelas antara permasalahan penelitian, metode yang digunakan, dan solusi yang dihasilkan. Alur penelitian secara keseluruhan ditunjukkan pada Gambar 3.1.



Gambar 3.1 Alur penelitian

Sebagai langkah awal dalam perancangan solusi, dilakukan studi pendahuluan terhadap literatur yang relevan. Studi ini difokuskan pada penelitian-penelitian sebelumnya yang menerapkan pendekatan machine learning untuk memprediksi skor benchmark sistem komputer, khususnya benchmark berbasis SPEC CPU. Analisis literatur dilakukan untuk membandingkan model prediksi yang digunakan, strategi pra-pemrosesan data, serta metode evaluasi yang diterapkan, dengan tujuan mengidentifikasi pendekatan yang mampu memberikan akurasi prediksi yang baik sekaligus memiliki potensi generalisasi terhadap perangkat keras yang belum pernah digunakan pada proses pelatihan. Selain itu, studi literatur juga mencakup

kajian mengenai implementasi model machine learning ke dalam bentuk skrip prediksi yang dapat digunakan secara praktis.

3.1.1. Pemilihan Dataset dan Benchmark

Dataset yang digunakan dalam penelitian ini berasal dari *Standard Performance Evaluation Corporation* (SPEC), SPEC CPU2017. *Benchmark* SPEC dikenal karena sifatnya yang standar, objektif, dan andal dalam mengukur CPU dan memori dibawah beban kerja yang terkontrol secara ketat [2]. Pemilihan dataset ini mengacu pada beberapa penelitian terdahulu. SPEC CPU2017 dipilih karena merupakan versi yang lebih baru dan juga digunakan dalam studi oleh M. Cengiz dkk. [26] serta penelitian oleh A. Tousi dkk. [15] yang menunjukkan keunggulan RF dalam memprediksi kinerja *benchmark* SPEC CPU2017. Jumlah data dari masing-masing dataset dapat dilihat pada tabel 3.1.

Tabel 3.1 Jumlah dataset SPEC CPU

DATASET SPEC CPU 2017	Jumlah data
CINT2017	9963
CFP2017	9878
CINT2017rate	12889
CFP2017rate	12519
TOTAL	45249

Pada tabel 3.1, dataset digunakan pada pengujian RF. Skenario yang akan dilakukan adalah implementasi model RF pada dataset SPEC CPU 2017, dengan metodologi yang mengacu pada penelitian A. Tousi dkk. Dengan skenario ini, diharapkan pengujian model dengan dataset

ini dapat menghasilkan yang tidak hanya mendapatkan model yang optimal, tetapi juga model yang dapat prediksi skor *benchmark* dengan dataset yang berisi spesifikasi baru dan terkini.

3.1.2. Pre-Processing (Konversi, Encoding, Scaling)

- Pre-Processing

Tahapan pra-pemrosesan dan pengembangan model Random Forest (RF) secara langsung mengimplementasikan metodologi yang telah dirangkum pada tabel 2.1. Proses ini mencakup pengembangan dataset, perluasan fitur input, pra-pemrosesan adaptif, transformasi data, feature engineering, seleksi fitur multi-tahap, penskalaan fitur, serta pelatihan dan evaluasi model menggunakan skema validasi silang.

Langkah awal meliputi pemuatan dataset SPEC CPU2017, penghapusan kolom yang tidak relevan, serta pembersihan data yang tidak valid secara definisi SPEC CPU pada variabel target ‘Base Result’ dan ‘Peak Result’, dengan hanya mempertahankan nilai positif. Selanjutnya, transformasi logaritmik menggunakan fungsi `np.log1p` diterapkan pada variabel target dan sebagian besar fitur numerik, termasuk kapasitas memori, ukuran cache L1/L2/L3, frekuensi prosesor, serta jumlah core dan thread. Transformasi ini bertujuan untuk menstabilkan varians dan mengatasi distribusi data yang sangat miring.

- Feature Engineering

Tahap berikutnya adalah *feature engineering* yang mendalam, yang berfokus pada dekomposisi deskripsi teks konfigurasi perangkat keras menjadi fitur terstruktur. Informasi prosesor diurai menjadi Family, Model, dan Iteration, sementara informasi memori dipisahkan menjadi kapasitas total, jumlah modul, dan kecepatan. Fitur kategorikal hasil ekstraksi, seperti sistem operasi dan sistem berkas, ditangani

menggunakan teknik “Target Mean Encoding” yang diterapkan secara terpisah di dalam setiap fold validasi silang untuk mencegah kebocoran informasi.

Strategi seleksi fitur dilakukan melalui beberapa tahap. Pertama, clustering menggunakan algoritma K-Means diterapkan pada fitur numerik yang telah di-encode untuk menghasilkan fitur tambahan berupa `Benchmark_Cluster`, yang merepresentasikan kelompok konfigurasi perangkat keras yang serupa. Selanjutnya, untuk mengurangi redundansi, digunakan kombinasi Korelasi Spearman dan Hierarchical Clustering guna memilih fitur perwakilan dari kelompok fitur yang sangat berkorelasi. Subset fitur hasil penyaringan ini kemudian diproses menggunakan Recursive Feature Elimination with Cross-Validation (RFECV) berbasis model RF dengan metrik Mean Absolute Error (MAE) sebagai skor panduan, sehingga hanya fitur yang paling berkontribusi yang dipertahankan.

Fitur akhir yang terpilih kemudian diskalakan menggunakan `StandardScaler` untuk memastikan kontribusi fitur yang seimbang dalam proses pelatihan model. Evaluasi internal model dilakukan menggunakan skema 5-Fold Cross-Validation dengan pembagian data 80% sebagai data pelatihan dan 20% sebagai data validasi pada setiap fold. Berdasarkan kerangka ini, dibangun empat model regresi Random Forest yang terpisah untuk setiap metrik benchmark, yaitu `CINT2017`, `CFP2017`, `CINT2017rate`, dan `CFP2017rate`. Selain itu, untuk setiap metrik benchmark, model dilatih secara terpisah untuk memprediksi dua jenis skor, yaitu “Base Result” dan “Peak Result”.

Tabel 3.2 Skema Pembagian Data Training dan Testing

Benchmark	Total Data (100%)	Training Aktual (64%)	Validation (16%)	Testing (20%)	Skema Validasi
CINT2017	7,084	4.533	1.133	1.417	5-Fold CV
CFP2017	6,875	4.400	1.100	1.375	5-Fold CV
CINT2017rate	8.724	6.979	1.395	1.745	5-Fold CV
CFP2017rate	8.524	6.819	1.363	1.705	5-Fold CV

Berdasarkan skema pembagian data pada Tabel 3.2, evaluasi performa model dilakukan melalui dua tahap. Tahap pertama adalah evaluasi internal, di mana seluruh dataset terlebih dahulu dipisahkan menggunakan metode *hold-out* menjadi data *training* sebesar 80% dan data *testing* sebesar 20%. Kemudian, data *training* diproses menggunakan skema “5-Fold Cross-Validation”, hingga pada setiap fold sekitar 64% dari total data digunakan untuk pelatihan model dan 16% digunakan sebagai data validasi. Evaluasi pada tahap ini bertujuan untuk mengukur kemampuan model dalam mempelajari pola data dan menjaga stabilitas performa pada data yang masih berasal dari distribusi yang sama.

3.1.3. Pengembangan Model

Pada tahap Pengembangan model, penulis mengimplementasikan model *machine learning* yang populer dan efektif untuk tugas regresi, RF. Pemilihan model ini didasarkan pada kemampuan mereka dalam menangani data kompleks dan memberikan performa prediksi yang tinggi dalam berbagai skenario. Setiap model dikonfigurasi dengan parameter tertentu. Penentuan nilai-nilai spesifik untuk parameter ini merupakan hasil dari proses *tuning* atau optimasi, disesuaikan berdasarkan rekomendasi dari penelitian dan jurnal terkait [15]. Konfigurasi ini dipilih untuk menyeimbangkan antara akurasi prediksi model, kompleksitas model, dan

efisiensi komputasi, guna menghasilkan kinerja yang optimal pada dataset yang digunakan dan mencegah *overfitting*. Model-model ini kemudian dilatih menggunakan dataset pelatihan yang telah melalui proses *pre-processing*.

- Membangun Model RF

Model RF juga dengan parameter yang menyesuaikan dengan penelitian sebelumnya oleh A. Tousi dkk [15]. Parameter yang digunakan pada model RF dapat dilihat di tabel 3.3.

Tabel 3.3 Parameter model RF

Hyperparameter	Nilai	Fungsi
n_estimators	50	Menentukan jumlah pohon keputusan (<i>decision trees</i>) yang dibangun dalam proses <i>bagging</i> . Jumlah pohon yang optimal dapat meningkatkan akurasi model.
max_depth	15	Batas kedalaman maksimum untuk setiap pohon keputusan. Kedalaman yang lebih besar memungkinkan model menangkap pola yang lebih kompleks tetapi meningkatkan risiko

		<i>overfitting</i> .
min_samples_split	2	Jumlah minimum sampel yang dibutuhkan untuk membagi sebuah <i>internal node</i> . Parameter ini membantu mencegah <i>overfitting</i> dengan memastikan bahwa setiap pembagian memiliki jumlah data yang cukup.
min_samples_leaf	5	Jumlah minimum sampel yang dibutuhkan pada setiap <i>leaf node</i> (node daun). Ini juga merupakan parameter yang mengontrol <i>overfitting</i> dengan memastikan bahwa setiap daun memiliki jumlah data yang memadai.
random_state	42	Digunakan untuk memastikan reproduktifitas hasil pelatihan model, sehingga setiap kali kode dijalankan, hasilnya konsisten.
n_jobs	-1	parameter untuk berapa banyak <i>decision tree</i>

		individu dalam hutan yang dilatih secara bersamaan.
max_features	0.5 & sqrt	parameter yang mengontrol jumlah fitur (kolom) yang dipertimbangkan oleh setiap <i>decision tree</i> individual saat mencari <i>split</i> terbaik pada setiap <i>node</i> .

Pada tabel 3.3, parameter model RF ditentukan sesuai penelitian sebelumnya. Pada parameter jumlah *n_estimators*, jumlah yang digunakan adalah 50 karena penelitian sebelumnya menyatakan bahwa jumlah yang lebih dari 50 tidak mengurangi *error* prediksi, tetapi hanya menambahkan waktu untuk prediksi. Kemudian, parameter *max_depth* yang digunakan adalah 15 karena jika tidak dilimit, akan menimbulkan *overfitting* [15].

3.1.4. Evaluasi Performa Model pada Tahap Training dan Validasi

Evaluasi performa model dilakukan pada tahap training dan validasi internal untuk menilai kemampuan model dalam mempelajari pola data serta mendeteksi potensi *overfitting*, sebelum dilakukan pengujian generalisasi pada data uji.

Metrik yang digunakan adalah Mean Absolute Percentage Error (MAPE) dan koefisien determinasi (R^2). MAPE dipilih karena memberikan interpretasi kesalahan prediksi dalam bentuk persentase yang mudah dibandingkan antar benchmark, sedangkan R^2 digunakan

untuk mengukur seberapa besar variasi nilai target yang dapat dijelaskan oleh model. Evaluasi dilakukan menggunakan skema 5-Fold Cross-Validation dengan pembagian data 80% untuk training dan 20% untuk validation pada setiap fold.

3.1.5. Pengujian Data

Pengujian akurasi model dilakukan menggunakan data uji (test set) yang dipisahkan dari dataset SPEC CPU2017 sebelum proses pelatihan model. Pemisahan data ini memastikan bahwa seluruh data uji tidak pernah digunakan pada tahap pelatihan maupun validasi, sehingga evaluasi performa mencerminkan kemampuan generalisasi model secara objektif.

Selain evaluasi kuantitatif pada data uji secara keseluruhan, dilakukan pula analisis perbandingan antara nilai benchmark aktual dan hasil prediksi model pada beberapa konfigurasi CPU dari dataset uji. Evaluasi dilakukan pada metrik benchmark yang bersesuaian, yaitu CINT2017, CFP2017, CINT2017rate, dan CFP2017rate, serta pada skor “Peak Result” dan “Base Result”. Selisih antara nilai aktual dan hasil prediksi digunakan sebagai indikator utama untuk menilai tingkat deviasi prediksi model.

3.2 Metode Pengujian

3.2.1. Evaluasi Model

Pada tahap evaluasi akhir menggunakan data uji, fokus utama evaluasi diarahkan pada akurasi prediksi model, yang didefinisikan sebagai tingkat kedekatan antara nilai hasil prediksi dan nilai benchmark berbasis SPEC CPU2017. Evaluasi dilakukan pada empat benchmark utama, yaitu CINT2017, CFP2017, CINT2017rate, dan CFP2017rate.

- Akurasi Generalisasi (*Ground Truth*): Akurasi model dievaluasi dengan membandingkan hasil prediksi terhadap nilai benchmark aktual (ground truth) yang tersedia di dalam dataset SPEC CPU2017 pada data uji. Selain evaluasi agregat, dilakukan pula analisis perbandingan pada beberapa konfigurasi CPU yang dipilih dari data uji untuk memberikan gambaran kualitatif mengenai performa generalisasi model terhadap variasi perangkat keras.

3.2.2. Lingkungan Pengujian *Ground Truth*

Ground truth pada tahap pengujian diperoleh langsung dari dataset resmi SPEC CPU2017. Spesifikasi perangkat keras dan nilai benchmark yang digunakan sebagai pembanding berasal dari entri data uji yang dipisahkan dari dataset sebelum proses pelatihan model.

Untuk keperluan analisis, dipilih empat konfigurasi CPU dari data uji yang masing-masing merepresentasikan satu metrik benchmark, yaitu CINT2017, CFP2017, CINT2017rate, dan CFP2017rate. Spesifikasi perangkat keras yang digunakan sebagai input model diambil dari kolom spesifikasi yang tersedia dalam dataset, sehingga representasi fitur input dan nilai ground truth berada dalam domain data yang sama.

Pendekatan ini memastikan bahwa evaluasi dilakukan secara konsisten dan objektif, tanpa ketergantungan pada sumber eksternal, serta mencerminkan kemampuan model dalam memprediksi performa perangkat keras berdasarkan data yang tidak digunakan pada tahap pelatihan.