

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1 *Kaizen* dan *Quality Control Circle* (QCC)

Menurut (Fikri Baisalim et al., 2022), penerapan *Quality Control Circle* (QCC) terbukti meningkatkan kesadaran mutu, kerja tim, dan partisipasi karyawan di seluruh jenjang organisasi, sehingga mendukung terciptanya perbaikan berkelanjutan sebagai bagian dari sistem manajemen mutu perusahaan. QCC terdiri atas kelompok kecil karyawan yang berfokus pada identifikasi masalah kualitas, analisis akar penyebab, serta penerapan solusi melalui 8 Step Problem Solving dan penggunaan 7 Tools seperti *Fishbone Diagram*, *Pareto Chart*, *Histogram*, dan *Check Sheet*.

Pada Divisi PBOD PT Toyota Motor Manufacturing Indonesia (TMMIN), penerapan QCC dilakukan dengan dua dimensi utama:

1. **Sumbu X (Teknis/Output)** yang mengukur hasil capaian dan efektivitas penyelesaian masalah.
2. **Sumbu Y (Proses/Metode/Timwork)** yang menilai aspek kolaborasi, kedisiplinan, dan kedalaman metode analisis.

Kombinasi kedua dimensi menghasilkan klasifikasi zona kinerja:

Zona A (Unggul), Zona B (Baik namun perlu perbaikan), Zona C (Cukup, banyak kekurangan), dan Zona D (Rendah, perlu pendampingan intensif). Berdasarkan data internal PBOD TMMIN tahun 2024–2025, distribusi zona menunjukkan ketimpangan: Zona A (4,9%), Zona B (87,2%), Zona C (5,5%), dan Zona D (2,5%). Kondisi ini menandakan sebagian besar tim masih stagnan di zona menengah.

2.1.2 Penilaian Kinerja Tim QCC

Sistem penilaian kinerja pada QCC di PBOD sebelumnya masih dilakukan secara manual dan bersifat deskriptif, sehingga rawan bias antar evaluator serta sulit memberikan deteksi dini terhadap tim berisiko turun zona.

Proses evaluasi dilakukan per individu, sedangkan hasil agregasi tim dilakukan secara manual, menyebabkan keterlambatan dan ketidakkonsistenan hasil. Oleh karena itu, diperlukan sistem otomatis yang mampu:

1. Melakukan perhitungan nilai rata-rata per tim secara real-time,
2. Memberikan prediksi zona kinerja otomatis,
3. Menampilkan insight faktor-faktor dominan yang memengaruhi performa,
4. Menyediakan dukungan keputusan berbasis data bagi manajemen.

2.1.3 Machine Learning dalam Evaluasi Kinerja

Menurut (Géron, 2022), *machine learning* memungkinkan sistem membangun model dari data (*training data*) sehingga mampu melakukan prediksi atau pengambilan keputusan pada data baru tanpa harus ditulis aturan secara eksplisit untuk setiap kasus. Dengan pendekatan ini, model prediksi *machine learning* membantu mengungkap pola dan hubungan antarvariabel yang sulit diidentifikasi melalui analisis manual.

2.1.3.1. Algoritma *Random Forest* (RF)

Random Forest pertama kali diterbitkan secara resmi oleh Leo Breiman pada tahun 2001. Menurut (Géron, 2022) *Random Forest* (RF) merupakan metode *ensemble learning* berbasis *Decision Tree* yang menggabungkan prinsip *bagging* (*bootstrap sampling*) dan pemilihan fitur secara acak pada setiap pemisahan node, sehingga tiap pohon dibangun dari subset data yang berbeda. Hasil akhir model

diperoleh melalui *majority voting* untuk klasifikasi atau rata-rata prediksi untuk regresi. Dalam penelitian (Widyawati et al., 2025), Random Forest memperoleh kinerja tertinggi dalam klasifikasi status gizi balita dengan akurasi 1.00 dan *cross-validation* 99.74%, lebih tinggi daripada Logistic Regression (akurasi 0.87). Hal ini selaras dengan penjelasan mereka bahwa Random Forest efektif untuk pola non-linear, sedangkan Logistic Regression sebagai model linear cenderung kurang optimal pada dataset yang kompleks.

Secara konsep, Random Forest membangun sejumlah pohon keputusan BBB dan untuk kasus klasifikasi prediksi akhir diperoleh melalui voting mayoritas (Géron, 2022).

$$\hat{y} = \text{mode}\{h_1(x), h_2(x), \dots, h_B(x)\}$$

Estimasi probabilitas kelas k dapat dinyatakan sebagai proporsi pohon yang memilih kelas tersebut (Raschka et al., 2022)

$$\hat{P}(Y = k | x) = \frac{1}{B} \sum_{b=1}^B \mathbb{I}(h_b(x) = k)$$

Dalam pembentukan pohon keputusan, pemilihan atribut pemisah (*split*) umumnya didasarkan pada ukuran impurity, seperti ***Gini impurity*** dan ***Entropy*** (Géron, 2022; James et al., 2023):

a. Gini Impurity didefinisikan sebagai:

$$G(t) = 1 - \sum_{k=1}^K p_k(t)^2$$

b. **Entropy** didefinisikan sebagai:

$$H(t) = - \sum_{k=1}^K p_k(t) \log p_k(t)$$

Dengan $p_k(t)$ adalah proporsi kelas k pada node t .

Selain itu, kinerja model *Random Forest* dalam tugas klasifikasi umumnya dievaluasi menggunakan metrik yang diturunkan dari confusion matrix, salah satunya adalah akurasi (James et al., 2023):

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Pada kasus multikelas, akurasi didefinisikan sebagai perbandingan jumlah prediksi yang benar terhadap total jumlah data.

a. Kelebihan RF:

1. Mampu menangani hubungan *non-linear* antar variabel.
2. Tahan terhadap *overfitting*.
3. Dapat mengukur feature importance untuk mengetahui variabel paling berpengaruh terhadap prediksi zona.
4. Cocok untuk data tidak seimbang (*imbalanced dataset*).

a. Kelemahan RF:

1. Kurang interpretatif dibandingkan LR.
2. Membutuhkan waktu komputasi lebih lama.

RF digunakan sebagai **model utama** karena karakteristik data QCC yang non-linear dan tidak seimbang antar zona (B dominan).

2.1.3.2. Algoritma Logistic Regression (LR)

Menurut (James et al., 2023), *Logistic Regression* (LR) adalah metode klasifikasi linear yang memodelkan probabilitas suatu observasi termasuk dalam kelas tertentu berdasarkan fungsi logit:

$$\log \left(\frac{p(x)}{1 - p(x)} \right) = \beta_0 + \beta^T x$$

Sehingga probabilitas suatu observasi termasuk ke dalam kelas positif dapat diperoleh melalui fungsi sigmoid:

$$p(x) = \frac{1}{1 + \exp \left(- (\beta_0 + \beta^T x) \right)}$$

Dalam praktik pembelajaran mesin modern, parameter β diestimasi dengan meminimalkan fungsi *loss* berbasis *log-likelihood* atau *cross-entropy* (Géron, 2022; Raschka et al., 2022):

$$\mathcal{L}(\beta) = - \sum_{i=1}^n \left[y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \right]$$

Untuk kasus multikelas, Logistic Regression dapat diperluas menjadi *multinomial logistic regression* dengan fungsi *softmax* untuk memodelkan probabilitas setiap kelas (James et al., 2023), yang dirumuskan sebagai:

$$P(Y = k \mid x) = \frac{\exp(\beta_{k0} + \beta_k^T x)}{\sum_{j=1}^K \exp(\beta_{j0} + \beta_j^T x)}$$

Logistic Regression banyak digunakan karena menghasilkan estimasi probabilitas dan koefisien model yang bersifat transparan dan mudah

diinterpretasikan, termasuk oleh pengguna non-teknis (Géron, 2022; Raschka et al., 2022).

a. Kelebihan LR:

1. Sederhana dan efisien untuk dataset kecil-menengah.
2. Hasilnya berupa probabilitas yang mudah dijelaskan.
3. Cocok sebagai baseline model interpretatif.

b. Kelemahan LR:

1. Sederhana dan efisien untuk dataset kecil-menengah.
2. Rentan terhadap ketidakseimbangan data (*imbalanced class*).
3. Asumsi hubungan linear antara fitur dan output tidak selalu sesuai untuk kasus nyata.

2.1.4 Pemodelan Sistem UML

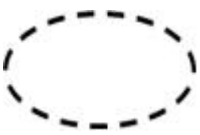
Unified Modeling Language (UML) merupakan bahasa pemodelan visual yang digunakan untuk merancang sistem perangkat lunak berorientasi objek, yang menyediakan berbagai diagram seperti *use case*, *activity*, dan *class diagram* untuk memodelkan kebutuhan dan struktur sistem (Ramdany et al., 2024). (Khairunnisa et al., 2024) juga menyatakan bahwa “Beberapa jenis diagram UML yang sering digunakan dalam pengembangan perangkat lunak meliputi Use Case Diagram dan Activity Diagram. Use Case Diagram menggambarkan fungsionalitas sistem dan interaksi antara aktor dan sistem, sedangkan Activity Diagram menggambarkan alur aktivitas dalam sistem”.

2.1.4.1 Use Case Diagram

Use Case Diagram digunakan untuk menggambarkan aktivitas atau fungsi utama yang dilakukan oleh *user* (pengguna) dalam sistem. Pada Table 2.1 Tipe Relasi pada *Use Case Diagram*, Diagram ini memperlihatkan siapa saja aktor yang terlibat dalam sistem, serta fitur atau layanan apa saja yang bisa diakses oleh masing-masing aktor. Dengan adanya *use case diagram*, kita bisa mengetahui gambaran umum interaksi antara *user* dengan sistem (Pratama & Marjun, 2022).

Tabel 2.1 Penggunaan Simbol pada *Usecase Diagram*

No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan Ketika berinteraksi dengan <i>usecase</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung padanya (elemen yang tidak mandiri).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).

4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya.
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemennya.

10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputer.
----	---	-------------	---

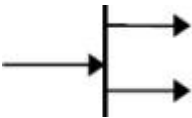
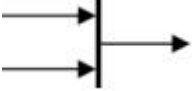
2.1.4.2 Activity Diagram

Menurut (Kurniawan, 2020) “Activity diagram adalah diagram yang memaparkan urutan tindakan dalam sistem yang dibuat. Diagram ini menunjukkan urutan aktivitas yang terjadi, keputusan atau cabang yang diambil, serta hubungan antara aktivitas-aktivitas tersebut”.

Berikut merupakan simbol-simbol utama yang digunakan dalam pembuatan *Activity Diagram* beserta nama dan keterangannya.

Tabel 2.2 Penggunaan Simbol pada *Activity Diagram*

Simbol	Nama	Keterangan
	<i>Start Point</i>	Merepresentasikan dimulainya alur kerja suatu sistem dalam <i>activity diagram</i> .
	<i>End Point</i>	Merepresentasikan berakhirnya alur kerja suatu sistem dalam <i>activity diagram</i> .
	<i>Activity</i>	Merepresentasikan performa dari beberapa tingkah laku didalam alur kerja.
	<i>Control flow</i>	Memperlihatkan urutan eksekusi.

	<i>Fork</i> (Percabangan)	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel.
	<i>Join</i> (Penggabungan)	Digunakan untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Decision</i>	Menentukan kapan alur dalam aktivitas menjadi bercabang.

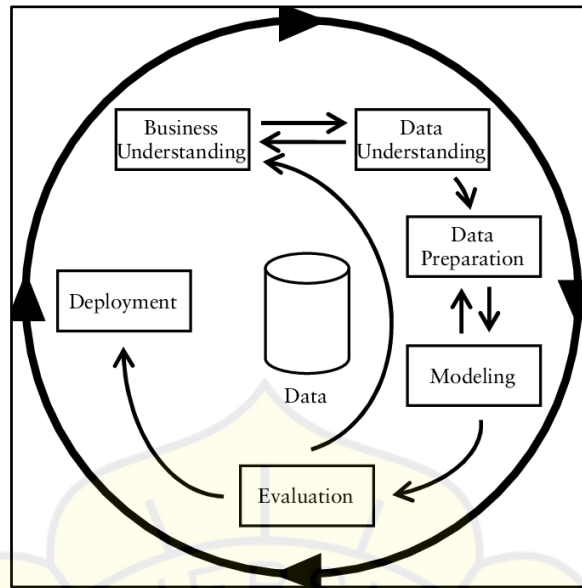
2.1.5 Metode Pengembangan Sistem

Pengembangan sistem dalam penelitian ini menggabungkan dua pendekatan utama, yaitu metodologi CRISP-DM (*Cross Industry Standard Process for Data Mining*) dan pendekatan *Prototype System Development*. Metodologi CRISP-DM digunakan sebagai kerangka utama dalam proses analisis data dan pembentukan model *machine learning* karena menyediakan tahapan yang terstruktur dan sistematis untuk proyek data science, mulai dari pemahaman bisnis hingga tahap deployment (Rianti et al., 2023).

Sementara itu, menurut (Saputri et al., 2024) penerapan Metode Prototype dalam Perancangan Sistem Metode *prototype* digunakan dalam perancangan sistem informasi untuk menjembatani pengembang dan pengguna, serta membantu mendapatkan sistem yang sesuai dengan kebutuhan fungsional dan antarmuka.

Kombinasi kedua metode ini memastikan hasil penelitian tidak hanya valid secara analitis, tetapi juga aplikatif dalam lingkungan kerja nyata Divisi PBOD PT TMMIN.

2.1.5.1 CRISP-DM (Cross Industry Standard Process for Data Mining)



Gambar 2.1 Diagram CRISP-DM

CRISP-DM merupakan metodologi data mining yang bersifat iteratif. (Bošnjak et al., n.d.) menegaskan bahwa “*CRISP-DM is an iterative process model*” serta mencakup tahapan business understanding, data understanding, data preparation, modeling, evaluation, dan deployment.

Implementasi enam tahapan tersebut juga terlihat pada penelitian (Hasanah et al., 2021) yang menjelaskan bahwa CRISP-DM terdiri dari enam tahap utama yang sama, dan pada tahap *modelling* melibatkan penentuan teknik/algoritma data mining untuk membangun model.

2.1.5.1.1 *Business Understanding*

Tahap Business Understanding merupakan fase pertama dalam metodologi CRISP-DM (*Cross-Industry Standard Process for Data Mining*). Tujuannya adalah memahami secara menyeluruh konteks bisnis, permasalahan utama, serta sasaran yang ingin dicapai melalui penerapan *data mining* atau *machine learning*.

Menurut (Hasanah et al., 2021), tahap ini menjadi pondasi bagi seluruh proses analisis karena keputusan yang diambil pada fase-fase berikutnya (*Data Understanding, Data Preparation, Modeling, Evaluation, Deployment*) akan bergantung pada seberapa tepat pemahaman masalah bisnis di awal. Dengan demikian, fase ini tidak hanya bersifat teknis, tetapi juga strategis.

2.1.5.1.2 Data Understanding

Tahap Data Understanding merupakan fase kedua dalam metodologi CRISP-DM (*Cross-Industry Standard Process for Data Mining*) yang berfokus pada proses pengumpulan, penelusuran, serta pemahaman karakteristik data yang relevan dengan tujuan bisnis. Tahap ini bertujuan untuk mengenali kualitas, struktur, dan potensi masalah pada data sebelum dilakukan proses pemodelan, sehingga analisis yang dihasilkan dapat dipercaya dan akurat.

2.1.5.1.3 Data Preparation

Tahap *Data Preparation* bertujuan menyiapkan data yang akan digunakan untuk pemodelan. Proses ini melibatkan *data cleaning* untuk menghapus duplikasi dan menangani missing value, *data selection* untuk memilih atribut relevan, serta *data transformation* agar data siap dianalisis. Tahapan ini juga mencakup imputation dan konversi format data menjadi CSV untuk diolah di Python. Dalam sistem QCC.

2.1.5.1.4 Algoritma Pemodelan

Tahap *Modeling* (Pemodelan) merupakan fase keempat dalam metodologi CRISP-DM (*Cross-Industry Standard Process for Data Mining*). Pada tahap ini, data yang telah melalui proses pembersihan dan transformasi diolah menggunakan algoritma tertentu untuk menghasilkan model prediktif yang dapat memecahkan

permasalahan bisnis yang telah didefinisikan sebelumnya. Tahap pemodelan bertujuan membangun model statistik atau *machine learning* yang mampu mempelajari pola dari data historis dan memberikan hasil prediksi yang optimal terhadap data baru.

2.1.5.1.5 Evaluation

Model diuji menggunakan *cross-validation* dan dievaluasi dengan metrik *accuracy*, *precision*, *recall*, dan *macro F1-score*. Hasil pengujian tersebut yang nantinya akan menunjukkan apakah *Random Forest* memberikan akurasi dan *F1-score* yang lebih tinggi, bersama dengan *Logistic Regression* dalam memberikan keunggulan dalam interpretabilitas model.

2.1.5.1.6 Deployment & Integration

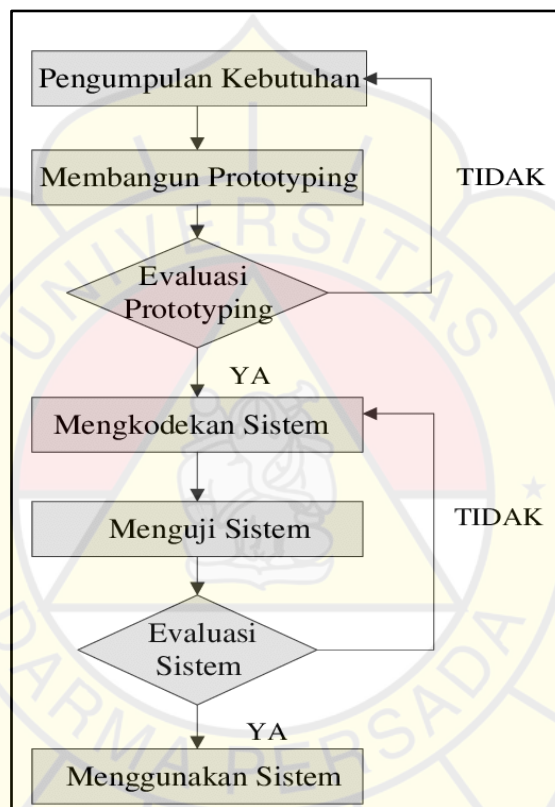
Tahap *Deployment* mencakup pembuatan laporan hasil dan penerapan model untuk menghasilkan informasi yang relevan bagi pengguna akhir. Dalam penelitian QCC, tahapan ini dikembangkan lebih lanjut menjadi *Deployment & Integration*, di mana model terbaik disimpan dalam format pickle (.pkl) menggunakan Joblib, kemudian diintegrasikan dengan dashboard berbasis Flask untuk menampilkan hasil prediksi secara *real-time*.

2.1.5.1.7 Monitoring & Maintenance

Tahap ini memastikan model tetap relevan dengan kondisi data terkini melalui pemantauan performa, pencatatan hasil prediksi, dan proses *retraining* berkala. Dalam konteks penelitian QCC, tahap ini digunakan untuk memperbarui model sesuai perubahan data penilaian dan pola kinerja tim agar sistem tetap akurat dan adaptif.

2.1.5.2 Prototype System Development

Tahap pengembangan prototipe merupakan proses awal dalam merealisasikan rancangan sistem ke dalam bentuk yang dapat diuji dan dievaluasi. Pada tahap ini, rancangan yang telah disusun sebelumnya diimplementasikan menjadi sebuah model sistem yang berfungsi untuk menggambarkan alur kerja dan fitur utama yang akan dikembangkan lebih lanjut.



Gambar 2.2 Diagram Model *Prototype*

Untuk memastikan sistem yang dikembangkan benar-benar sesuai dengan kebutuhan pengguna (*user-centered*), penelitian ini menerapkan pendekatan prototyping pada tahap perancangan dan implementasi sistem dashboard.

Menurut (Bagus Ageng Alfahri et al., 2025), metode *prototyping* merupakan pendekatan pengembangan sistem dalam kerangka *System Development Life Cycle* (SDLC) yang menekankan pembuatan model awal sistem (*prototype*) untuk diuji

dan dievaluasi secara iteratif bersama pengguna. Melalui proses tersebut, pengembang dapat memperoleh umpan balik secara langsung sehingga penyesuaian terhadap kebutuhan dan fungsionalitas sistem dapat dilakukan secara bertahap hingga menghasilkan sistem akhir yang lebih interaktif, mudah digunakan, dan sesuai dengan kebutuhan pengguna.

Pendekatan ini bersifat iteratif, pengguna dapat memberikan umpan balik sejak tahap awal pengembangan hingga sistem final siap digunakan. Melalui kombinasi *CRISP-DM* dan *Prototyping*, sistem *Smart Insight Dashboard* yang dikembangkan tidak hanya berfungsi sebagai alat analisis prediktif, tetapi juga sebagai alat bantu pengambilan keputusan berbasis data yang interaktif, akurat, dan dapat dioperasikan langsung oleh manajemen PBOD tanpa proses teknis tambahan.

Langkah-langkah penerapan metode prototipe adalah sebagai berikut:

2.1.5.2.1 Identifikasi Kebutuhan Pengguna (*Requirement Identification*)

Tahap identifikasi kebutuhan pengguna merupakan langkah awal dalam model pengembangan *prototype* yang bertujuan memahami kebutuhan sistem dari sudut pandang pengguna akhir. Tahapan ini dilakukan melalui teknik seperti wawancara, observasi, dan studi dokumen untuk mengumpulkan informasi mengenai fungsi, tampilan, serta batasan sistem yang diharapkan. Hasil dari tahap ini berupa daftar kebutuhan fungsional dan nonfungsional yang akan menjadi dasar dalam perancangan prototipe.

2.1.5.2.2 Perancangan Prototype Awal (*Initial Prototype Design*)

Tahap perancangan prototipe awal berfokus pada pembuatan rancangan antarmuka dan struktur sistem yang sederhana namun representatif. Desain awal prototipe harus berorientasi pada usability atau kemudahan penggunaan agar

pengguna dapat dengan mudah memahami alur sistem dan memberikan umpan balik terhadap rancangan tersebut. Elemen yang dirancang biasanya meliputi formulir masukan data, tombol aksi, serta tampilan hasil keluaran sementara.

2.1.5.2.3 Implementasi Prototype (*Prototype Implementation*)

Implementasi prototipe merupakan tahap di mana rancangan awal dikonversi menjadi sistem kerja sederhana menggunakan bahasa pemrograman dan teknologi pendukung. Tahap ini memungkinkan pengembang untuk membangun sistem versi awal (*working model*) yang menampilkan fungsi utama sistem tanpa harus mengimplementasikan seluruh komponen. Prototipe ini berfungsi sebagai media interaktif antara pengguna dan pengembang untuk mengevaluasi kesesuaian sistem dengan kebutuhan.

2.1.5.2.4 Uji Coba Pengguna (*User Evaluation*)

Uji coba pengguna adalah tahap di mana prototipe diuji oleh calon pengguna untuk menilai kemudahan penggunaan, kejelasan informasi, dan kesesuaian fungsi terhadap kebutuhan awal. Uji coba pengguna membantu memastikan bahwa sistem memenuhi harapan pengguna serta mengidentifikasi kekurangan atau ketidaksesuaian sebelum pengembangan akhir. Evaluasi dilakukan melalui observasi langsung, wawancara, atau kuesioner terhadap pengalaman pengguna saat mencoba prototipe.

2.1.5.2.5 Penyempurnaan Prototype (*Prototype Refinement*)

Tahap penyempurnaan prototipe dilakukan berdasarkan umpan balik dari hasil uji coba pengguna. Proses penyempurnaan ini bersifat iteratif, di mana pengembang memperbaiki antarmuka, fungsi, maupun struktur sistem sesuai masukan hingga mencapai bentuk akhir yang stabil dan siap dikembangkan secara

penuh. Iterasi berulang ini memungkinkan sistem akhir memiliki tingkat usability dan performa yang lebih baik sesuai dengan kebutuhan pengguna.

2.1.6 Software dan Pemrograman Terkait

Dalam pembuatan aplikasi monitoring Quality Control Circle (QCC) pada Divisi PBOD PT TMMIN, pemilihan perangkat lunak dan bahasa pemrograman yang tepat menjadi faktor kunci dalam menjamin performa serta keandalan sistem. Komponen perangkat lunak berfungsi sebagai sarana utama dalam membangun logika aplikasi, mengelola basis data, serta menyajikan antarmuka yang interaktif bagi pengguna. Oleh karena itu, kombinasi antara bahasa pemrograman, *framework*, dan sistem manajemen basis data perlu dirancang secara terintegrasi agar proses pengolahan data dan visualisasi hasil monitoring dapat berjalan efisien dan sesuai kebutuhan pengguna.

2.1.6.1 Software

Berdasarkan (Gede et al., 2022) dalam *Jurnal Ekonomi Manajemen Sistem Informasi (JEMSI)*, perangkat lunak (*software*) merupakan sekumpulan program atau instruksi yang digunakan untuk mengarahkan komputer agar dapat menjalankan fungsi tertentu sesuai kebutuhan pengguna. Perangkat lunak berperan mengatur proses kerja komputer, mendukung pengolahan data, serta menjadi penghubung antara pengguna dan perangkat keras sehingga perintah pengguna dapat diterjemahkan dan dieksekusi oleh sistem komputer.

2.1.6.1.1 Editor Visual Studio Code (VS Code)

Menurut (Ananda et al., 2024), Visual Studio Code (VS Code) adalah editor kode sumber yang ringan dan kaya fitur, serta mendukung penggunaan bahasa pemrograman web seperti HTML, CSS, dan JavaScript. Selain itu, VS Code

memiliki fitur seperti *syntax highlighting*, *auto-completion*, *live server*, *debugging*, dan dukungan ekstensi, sehingga membantu pengguna menulis kode lebih efisien dan mempercepat proses pembelajaran maupun pengembangan aplikasi berbasis web.

2.1.6.1.2 Integrated Development Environment (IDE)

Integrated Development Environment (IDE) merupakan lingkungan pengembangan yang menyediakan berbagai *tools* seperti *code editor*, *compiler*, dan *debugger* untuk mendukung proses pengembangan perangkat lunak secara terintegrasi (Piao et al., 2021). Pada umumnya IDE digunakan secara lokal pada laptop atau komputer, namun (Slamet & Anistyasari, 2021) menunjukkan bahwa IDE juga dapat dikembangkan secara daring melalui aplikasi *NiceRoom* sebagai online IDE berbasis website dan aplikasi mobile untuk mendukung pembelajaran pemrograman Python.

2.1.6.2 Bahasa Pemrograman

Bahasa pemrograman digunakan sebagai sarana untuk mengimplementasikan rancangan sistem ke dalam bentuk aplikasi yang dapat dijalankan. Hal ini sejalan dengan penelitian (Roselyn Charity Nebore et al., 2024) yang menyatakan bahwa “implementasi website menggunakan HTML, CSS, dan PHP sebagai bahasa pemrograman untuk mengimplementasikan desain sistem informasi ke dalam bentuk website agar dapat diakses secara online”.

2.1.6.2.1 Python

Menurut (Maesaroh et al., 2024), Python merupakan bahasa pemrograman tingkat tinggi yang mendukung berbagai kebutuhan (multiguna) dan menekankan keterbacaan kode melalui sintaks yang sederhana, sehingga lebih mudah dipahami

dan dipelajari baik oleh pemula maupun pengguna yang telah menguasai bahasa pemrograman lain. Python dapat dijalankan pada berbagai sistem operasi seperti Windows, macOS, dan Linux, serta banyak digunakan dalam pengembangan perangkat lunak, analisis data, dan pengembangan kecerdasan buatan. Bahasa ini dikembangkan oleh Guido van Rossum dan pertama kali diperkenalkan pada tahun 1991. Keunggulan Python terletak pada kemudahan implementasi dan dukungan ekosistem yang luas, sehingga pengembang dapat lebih fokus pada penyelesaian masalah dan pengembangan aplikasi tanpa harus menulis kode yang terlalu kompleks.

2.1.6.2.2 Javascript

Menurut (Santoso, 2025), JavaScript merupakan bahasa pemrograman berbasis skrip yang bekerja di sisi klien (*client-side*) dan berfungsi untuk membuat halaman web menjadi lebih interaktif. Dalam pengembangan web, JavaScript dapat digunakan untuk menampilkan elemen dinamis dan interaksi pengguna seperti menampilkan pesan/notifikasi, melakukan validasi input, serta mengatur perilaku komponen antarmuka melalui manipulasi elemen pada halaman web.

2.1.6.2.3 Hyper Text Markup Language (HTML)

Menurut (Noviantoro et al., 2022), HTML (*Hyper Text Markup Language*) merupakan bahasa markup yang digunakan untuk menyusun dan mendeskripsikan struktur sebuah halaman web. HTML memanfaatkan elemen-elemen (*tag*) untuk membentuk komponen halaman seperti judul, paragraf, gambar, tautan, dan bagian-bagian lainnya sehingga dokumen dapat ditampilkan dan dipublikasikan secara daring. Dengan demikian, HTML menjadi komponen dasar dalam pembuatan

website karena menentukan susunan dan struktur informasi yang disajikan kepada pengguna.

2.1.6.2.4 *Cascading Style Sheets* (CSS)

Dalam pengembangan aplikasi berbasis web, pengaturan tampilan antarmuka menjadi salah satu aspek penting untuk meningkatkan kenyamanan dan pengalaman pengguna. Salah satu teknologi yang digunakan untuk mengatur tampilan tersebut adalah CSS (*Cascading Style Sheets*). Menurut (Orisa et al., 2023), CSS merupakan teknologi yang digunakan untuk membuat tampilan halaman web lebih menarik serta mengontrol format tampilan pada halaman HTML dan XHTML. CSS juga digunakan untuk memisahkan konten utama dengan tampilan dokumen, seperti pengaturan layout, warna, dan font, sehingga tampilan website menjadi lebih terstruktur dan mudah dikelola. Selain itu, CSS disimpan dalam bentuk *style sheet* berupa file teks sederhana dengan ekstensi .css.

2.1.6.2.5 Bootstrap

Dalam pengembangan antarmuka web modern, penggunaan framework *front-end* menjadi penting untuk mempercepat proses desain serta memastikan tampilan yang responsif pada berbagai perangkat. Salah satu framework yang banyak digunakan adalah Bootstrap. Menurut (Perdana et al., 2024), Bootstrap 5 digunakan dalam perancangan *front-end* website untuk membantu pengembang membangun tampilan antarmuka secara lebih cepat dan terstruktur. Dengan memanfaatkan komponen antarmuka yang telah tersedia, pengembangan elemen seperti tata letak halaman, navigasi, dan tombol dapat dilakukan lebih efisien. Selain itu, Bootstrap mendukung desain responsif, sehingga tampilan website dapat

menyesuaikan ukuran layar pada komputer maupun perangkat seluler, yang pada akhirnya memudahkan proses desain dan meningkatkan kenyamanan pengguna.

2.1.6.3 Library dan Framework Terkait

Beberapa *library* utama yang digunakan dalam penelitian ini adalah:

Tabel 2.3 *Library* yang digunakan

Library	Fungsi Utama	Keterangan
Pandas	Manipulasi dan analisis data tabular	Untuk membaca dan membersihkan dataset QCC
numpy	Operasi numerik dan komputasi cepat	Mendukung perhitungan statistik pada model
matplotlib & seaborn	Visualisasi data	Membuat grafik distribusi zona dan hasil model
scikit-learn	Pustaka utama machine learning	Membuat model Logistic Regression, Random Forest, evaluasi akurasi, F1-score, dan confusion matrix
joblib	Serialisasi model ML	Menyimpan model terlatih dalam format .pkl

Flask	Framework web ringan	Mengintegrasikan model ML ke dashboard interaktif
Chart.js	Visualisasi pada dashboard	Membuat grafik dinamis pada antarmuka web
MySQL	Database	Menyimpan data hasil penilaian dan prediksi zona

2.1.6.3.1 Flask

Menurut penelitian (Larasati & Susetyo, 2024), Flask merupakan *framework* web berbasis bahasa pemrograman Python yang digunakan untuk membangun aplikasi web secara cepat dan efisien. Flask termasuk *microframework* karena bersifat ringan dan tidak membutuhkan banyak dependensi, sehingga pengembang dapat fokus pada logika aplikasi utama tanpa kerumitan konfigurasi alat yang berat. Karena karakteristiknya yang sederhana dan fleksibel, Flask cocok digunakan dalam pengembangan aplikasi web yang memerlukan integrasi dengan berbagai komponen seperti antarmuka pengguna dan API backend.

2.1.7 Database dan DBMS

Menurut (Rouf & Marasabessy, 2024), database atau basis data merupakan sekumpulan data yang saling berhubungan dan disimpan secara sistematis dalam perangkat komputer agar dapat diolah serta dimanfaatkan menjadi informasi yang berguna bagi pengguna. Basis data berperan sebagai komponen utama dalam sistem informasi, karena seluruh data terorganisir dan dapat diakses secara efisien untuk

mendukung pengambilan keputusan dalam organisasi, termasuk lembaga pendidikan.

(Rouf & Marasabessy, 2024) juga mengatakan, *Database Management System* (DBMS) merupakan perangkat lunak untuk membuat, mengelola, dan mengatur basis data agar dapat diakses oleh banyak pengguna secara bersamaan dengan tetap menjaga keamanan serta konsistensi data. DBMS berfungsi sebagai penghubung antara pengguna dan basis data, yang mengendalikan proses penyimpanan, pembaruan, dan pengambilan data agar data dapat dimanfaatkan secara optimal dalam sistem informasi.

2.1.7.1 Laragon

Menurut penelitian (Chandra & Setyaningsih, 2025), Laragon merupakan salah satu lingkungan pengembangan lokal (local development environment) yang banyak digunakan dalam pengembangan aplikasi web modern karena kemampuannya menyediakan fasilitas server lokal secara cepat dan efisien untuk pengujian serta pengembangan. Laragon dibandingkan dengan lingkungan lain seperti XAMPP dan MAMP dalam hal kinerja dan performa, serta menawarkan konfigurasi yang lebih ringan dan responsif dalam mengelola proyek pengembangan web berbasis PHP dan stack server lainnya

2.1.7.2 MySQL

Menurut (Olta et al., 2025), MySQL merupakan sistem manajemen basis data relasional yang umum digunakan bersama PHP dalam pengembangan aplikasi web berbasis data. Kombinasi PHP dan MySQL bekerja melalui perintah SQL untuk melakukan proses penyimpanan, pengambilan, dan pembaruan data agar aplikasi dapat berjalan dinamis dan interaktif.

2.2 Kajian Pustaka Terdahulu

Kajian penelitian terdahulu dilakukan untuk mengetahui posisi penelitian ini terhadap studi-studi yang relevan sebelumnya. Enam penelitian dijadikan acuan utama dalam pengembangan sistem prediksi zona kinerja QCC berbasis Machine Learning di PT TMMIN. Tabel di bawah ini menunjukkan resume jurnal dari penelitian sebelumnya dan penelitian saat ini.

2.2.1 Paper Penelitian 1

Bagian ini memaparkan hasil penelitian terdahulu yang relevan sebagai dasar dan pembanding dalam pengembangan sistem. Salah satunya adalah penelitian yang dilakukan oleh (Yuliawan et al., 2025) yang dirangkum pada Tabel 2.4 berikut.

Tabel 2.4 Resume Penelitian Pertama

Judul	Penilaian Kinerja Pegawai dengan Metode TOPSIS dan Backpropagation Neural Network
Penulis	Audi, B. Y., Hariyadi, M. A., Kusumawati, R., Crysdian, C., & Nugroho, F.
Tahun Publikasi	2025
Jenis Akreditasi Jurnal	JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika) - SINTA 3
Tujuan Penelitian	Mengintegrasikan metode TOPSIS dan BPNN untuk meningkatkan akurasi penilaian kinerja pegawai.
Metode Penelitian	Pembobotan kriteria dengan TOPSIS, kemudian hasilnya dijadikan input bagi model

	Backpropagation Neural Network untuk prediksi kinerja.
Temuan Utama	Akurasi meningkat hingga 80,41%, lebih tinggi dari Decision Tree (79,8%).
Kesimpulan Penelitian	Integrasi multi-metode meningkatkan akurasi penilaian. Relevan karena sama-sama menilai performa berbasis data, meskipun fokusnya masih individu, bukan tim QCC.

2.2.2 Paper Penelitian 2

Penelitian lain yang memiliki relevansi dengan topik ini adalah penelitian yang dilakukan oleh (Pahlevi et al., 2023), membahas tentang penerapan algoritma *Random Forest* dalam penilaian kelayakan kredit. Ringkasan penelitian tersebut disajikan pada Tabel 2.5 berikut.

Tabel 2.5 Resume Penelitian Kedua

Judul	Implementasi Algoritma Random Forest untuk Penilaian Kelayakan Kredit
Penulis	Pahlevi, O., Amrin, & Handrianto, Y.
Tahun Publikasi	2023
Jenis Akreditasi Jurnal	Jurnal Infortech - SINTA 4
Tujuan Penelitian	Menilai kelayakan kredit kendaraan bermotor menggunakan algoritma Random Forest.
Metode Penelitian	Menerapkan Random Forest pada 481 data nasabah dan mengevaluasi performa dengan metrik accuracy dan AUC.

Temuan Utama	Akurasi 78,6%, AUC 0,907 menunjukkan model sangat baik untuk klasifikasi biner.
Kesimpulan Penelitian	Random Forest efektif untuk data kompleks dan non-linear. Relevan dengan penelitian ini karena model RF juga digunakan sebagai model utama prediksi zona kinerja QCC.

2.2.3 Paper Penelitian 3

Penelitian selanjutnya dilakukan oleh (Yulianti et al., 2024), yang membahas perbandingan performa *algoritma Logistic Regression* dan *Random Forest* dengan penanganan data tidak seimbang menggunakan teknik *SMOTE*. Ringkasan penelitian tersebut disajikan pada Tabel 2.6 berikut.

Tabel 2.6 Resume Penelitian Ketiga

Judul	Penilaian Pembayaran Kredit dengan Logistic Regression dan Random Forest pada Home Credit
Penulis	Yulianti, T., Cahyana, A. H., Komarudin, M., Mulyani, Y., & Septama, H. D.
Tahun Publikasi	2024
Jenis Akreditasi Jurnal	Jurnal Pseudocode - SINTA 4
Tujuan Penelitian	Membandingkan performa Logistic Regression dan Random Forest dengan penanganan data tidak seimbang menggunakan SMOTE.

Metode Penelitian	Eksperimen klasifikasi pembayaran kredit dengan kombinasi RF+SMOTE dan LR
Temuan Utama	Kombinasi RF+SMOTE menghasilkan akurasi 90% dan AUC 0,97, melampaui Logistic Regression.
Kesimpulan Penelitian	Kombinasi RF dan LR efektif untuk data tidak seimbang. Penelitian ini menjadi dasar penggunaan Soft Voting Ensemble (RF+LR) dalam prediksi zona QCC.

2.2.4 Paper Penelitian 4

Penelitian berikutnya oleh (Zumarniansyah et al., 2021)berfokus pada penerapan metode *Simple Additive Weighting* (SAW) dalam sistem pendukung keputusan untuk penilaian karyawan terbaik. Ringkasan hasil penelitian tersebut ditampilkan pada Tabel 2.7 berikut.

Tabel 2.7 Resume Penelitian Keempat

Judul	Penerapan Sistem Pendukung Keputusan Penilaian Karyawan Terbaik dengan Metode Simple Additive Weighting (SAW).
Penulis	Zumarniansyah, A., Ardianto, R., Alkhalifi, Y., & Azizah, Q. N.
Tahun Publikasi	2021

Jenis Akreditasi Jurnal	Jurnal Sistem Informasi STMIK Antar Bangsa - SINTA 3
Tujuan Penelitian	Menentukan karyawan terbaik secara terstruktur menggunakan metode SAW.
Metode Penelitian	Pembobotan manual berdasarkan empat kriteria: kehadiran, loyalitas, kinerja, dan kedisiplinan.
Temuan Utama	Sistem efektif dalam memberi ranking, namun masih subjektif karena bobot ditentukan manual.
Kesimpulan Penelitian	Pendekatan SAW terstruktur tetapi tidak prediktif. Penelitian ini memperbaikinya dengan pendekatan Machine Learning yang objektif dan berbasis data historis.

2.2.5 Paper Penelitian 5

Penelitian yang dilakukan oleh (Maharani et al., 2025) menilai kinerja karyawan menggunakan metode *Behaviorally Anchored Rating Scales* (BARS) dan *Management by Objectives* (MBO) berdasarkan perilaku kerja dan pencapaian tujuan organisasi. Ringkasan penelitian tersebut dapat dilihat pada Tabel 2.8 berikut.

Tabel 2.8 Resume Penelitian Kelima

Judul	Penilaian Kinerja Karyawan Menggunakan Metode Behaviorally Anchor Rating Scale
--------------	--

	(BARS) dan Management by Objectives (MBO) di Apotek X.
Penulis	Anggun, M., Umam, M. I. H., Kusumanto, I., Nazaruddin, & Suherman.
Tahun Publikasi	2025
Jenis Akreditasi Jurnal	Ekonomis: Journal of Economics and Business - SINTA 3
Tujuan Penelitian	Menilai kinerja berdasarkan perilaku dan pencapaian tujuan organisasi menggunakan metode BARS dan MBO.
Metode Penelitian	Evaluasi berbasis kuesioner dengan lima faktor utama (kehadiran, perilaku kerja, kualitas, keandalan, manajerial).
Temuan Utama	Faktor kepribadian memiliki bobot tertinggi dalam memengaruhi hasil penilaian.
Kesimpulan Penelitian	Sistem masih manual dan berbasis kuesioner. Penelitian ini mengembangkan sistem otomatis prediktif dengan dashboard berbasis Machine Learning.

2.2.6 Paper Penelitian 6

Penelitian oleh (Sufina & Wati, 2025) mengimplementasikan algoritma *Random Forest* dalam sistem evaluasi kinerja dosen berdasarkan Sasaran Kinerja Pegawai (SKP). Ringkasan penelitian ini ditampilkan pada Tabel 2.9 berikut.

Tabel 2.9 Resume Penelitian Keenam

Judul	Implementation of Random Forest Algorithm in Lecturer Performance Evaluation System
Penulis	Sufina & Wati, L.
Tahun Publikasi	2025
Jenis Akreditasi Jurnal	Journal of Artificial Intelligence and Software Engineering - SINTA 4
Tujuan Penelitian	Mengimplementasikan algoritma Random Forest dalam sistem evaluasi kinerja dosen berdasarkan SKP.
Metode Penelitian	Random Forest digunakan untuk klasifikasi kategori performa dosen menggunakan data multi-kriteria.
Temuan Utama	Akurasi model mencapai 72,73% untuk kinerja utama dan 81,82% untuk perilaku kerja.
Kesimpulan Penelitian	Random Forest efektif untuk analisis evaluasi kinerja. Relevan dengan penelitian ini karena sama-sama berbasis data multikriteria, namun penelitian ini menambahkan LR untuk

	interpretasi probabilistik dan integrasi ke dashboard prediktif.
--	--

2.2.7 Paper Penelitian 7

Penelitian tambahan yang relevan dengan topik evaluasi kinerja berbasis data adalah penelitian yang dilakukan oleh (Jaya et al., 2025) yang menganalisis pengaruh beban kerja terhadap hasil evaluasi kinerja karyawan menggunakan pendekatan machine learning. Ringkasan penelitian tersebut disajikan pada Tabel 2.10 berikut.

Tabel 2.10 Resume Penelitian Ketujuh

Judul	Computational Modeling Using Linear Regression and Random Forest to Analyze the Impact of Workload on Employee Performance Evaluations
Penulis	Safitri Jaya, Aries Yulianto, & Edi Purwanto
Tahun Publikasi	2025
Jenis Akreditasi Jurnal	Jurnal internasional (penerbit IIETA – IJCMEM) - Q3
Tujuan Penelitian	Pemodelan komputasional memakai dua model: Linear Regression untuk interpretasi hubungan variabel dan Random Forest untuk menangkap pola non-linear dari data beban kerja.

Metode Penelitian	Linear Regression menghasilkan akurasi 81,32% (precision 83%), sedangkan Random Forest menghasilkan akurasi 93% (precision 90%)
Temuan Utama	Akurasi model mencapai 72,73% untuk kinerja utama dan 81,82% untuk perilaku kerja.
Kesimpulan Penelitian	Random Forest lebih unggul pada data kompleks, sedangkan model linear lebih mudah dijelaskan. Hal ini mendukung pemilihan Random Forest sebagai model utama prediksi zona.

