BAB 2
LANDASAN TEORI

2.1 Penelitian Terdahulu

Kajian terdahulu yang relevan dengan prediksi keterlambatan dan waktu resolusi *issue* dapat dikelompokkan menjadi empat area, yaitu penyediaan *dataset* Jira publik, prediksi waktu resolusi *issue*, prediksi keterlambatan dinamis, dan penjelasan prediksi. Ringkasan penelitian terdahulu ditunjukkan pada Tabel 2.1.

Tabel 2.1. Perbandingan penelitian terdahulu dan landasan pendukung metodologi

Referensi	**Fokus**	**Data**	**Metode**	**Posisi terhadap metodologi**
Montgomery dkk. (2022) [3]	*Dataset* Jira publik	16 repositori Jira publik lintas proyek	Publikasi *dataset*	Menjadi sumber data utama, tetapi tidak membahas model prediksi.
Kula dkk. (2023) [2]	Prediksi keterlambatan dinamis	Data proyek industri	Pola keterlambatan dan Bayesian	Menguatkan pentingnya prediksi dinamis, tetapi tidak spesifik pada *issue*-level Jira publik.
Madaraboina dkk. (2024) [5]	Prediksi prioritas dan waktu resolusi *bug*	Spring Jira Bug Dataset	*Multi-target classification*	Relevan untuk prediksi *bug* Jira, tetapi belum berfokus pada *event log* dan fitur proses.
Özkan dkk. (2024) [6]	Prediksi waktu resolusi *bug*	Data Jira proyek jaringan	kNN, Naive Bayes, dan *neural network*	Menjadi pembanding fitur statis dan *assignee*, tetapi target masih berbasis *fast/slow*.
Qiao dkk. (2025) [7]	Prediksi waktu resolusi *issue* OSS	*Snapshot dataset* 3h, 1d, 7d, 30d	RF, XGBoost, dan model pembanding	Sangat relevan untuk konsep *snapshot*, tetapi sebagian fitur tidak selalu tersedia pada *dump* Jira publik.
van der Aalst dan Carmona (2022) [11]	Landasan *process mining*	*Event log* dan proses bisnis	Kerangka konseptual *process mining*	Memperkuat dasar pembentukan *event log*, fitur proses, dan pemantauan proses berjalan.
De Weerdt dan Wynn (2022) [12]	Struktur data kejadian proses	*Process event data*	Konsep *case*, aktivitas, dan *timestamp*	Menjadi dasar teknis transformasi riwayat status Jira menjadi *event log*.
Tomura dan Dam (2024) [9]	Penjelasan prediksi keterlambatan	Data proyek perangkat lunak	*Explainable AI*	Menguatkan kebutuhan interpretasi model, tetapi bukan fokus utama pada fitur proses Jira.

Kajian terdahulu menunjukkan bahwa masalah prediksi pada data *issue tracker* telah berkembang dari penyediaan *dataset* publik, prediksi prioritas *bug*, prediksi waktu resolusi, hingga prediksi keterlambatan yang bersifat dinamis. Namun, belum tampak satu rancangan metodologi yang secara serentak

memadukan empat kebutuhan berikut: penggunaan *dataset* Jira publik lintas proyek yang besar, pembentukan fitur proses berbasis *event log* perubahan status, evaluasi pada beberapa *snapshot* observasi yang konsisten, dan perbandingan eksplisit terhadap metode penelitian terdahulu pada protokol evaluasi yang sebanding. Oleh karena itu, penguatan kontribusi metodologis tidak hanya bergantung pada akurasi model, tetapi juga pada rancangan eksperimen yang mampu menunjukkan apakah metode yang digunakan benar-benar memberikan peningkatan dibandingkan dengan pendekatan terdahulu.

Berdasarkan kajian tersebut, dua peluang penelitian utama masih terbuka. Pertama, metadata *issue*, fitur proses dari riwayat perubahan status, dan histori *assignee* belum banyak diintegrasikan dalam satu rancangan *early warning* multi-horizon. Kedua, kontribusi fitur proses perlu dievaluasi pada sumber data, target, pembagian temporal, algoritma, dan prosedur evaluasi yang sama. Penjelasan lokal digunakan sebagai keluaran pendukung untuk membantu membaca prediksi, bukan sebagai masalah penelitian yang terpisah.

2.2 Jira dan Data *Issue Tracking*

Jira merupakan sistem pelacakan pekerjaan yang digunakan untuk mengelola *issue*, *backlog*, alur kerja, dan aktivitas kolaborasi tim perangkat lunak. Satu *issue* dapat mewakili *bug*, *task*, *improvement*, *new feature*, atau bentuk pekerjaan lain. Atribut yang umum tersedia pada *issue* meliputi proyek, tipe *issue*, prioritas, status, tanggal pembuatan, tanggal resolusi, komentar, tautan antar-*issue*, serta *assignee*. Struktur semacam ini membuat Jira relevan untuk penelitian berbasis *issue tracking*, terutama ketika data historis dan riwayat perubahan tersedia dalam skala besar [3, 4].

The Public Jira Dataset menjadi sumber data penting karena menyediakan repositori publik dalam skala besar dan tetap mempertahankan struktur historis Jira. *Dataset* tersebut memuat perubahan status, komentar, dan tautan antar-*issue* dalam bentuk MongoDB *dump*, serta telah dianonimisasi dengan menjaga keunikan entitas pengguna melalui masker UUID [4]. Karakteristik ini membuat *dataset* sesuai untuk eksperimen prediksi risiko keterlambatan, ekstraksi fitur proses, dan evaluasi model lintas waktu.

2.3 Keterlambatan *Issue* dan Prediksi Waktu Resolusi

Keterlambatan pada level *issue* dapat didefinisikan berdasarkan *due date*, target *sprint*, target rilis, atau durasi penyelesaian yang tidak wajar dibandingkan riwayat proyek. Pada data publik, *due date* sering tidak tersedia secara merata. Oleh karena itu, durasi penyelesaian historis dapat digunakan sebagai pendekatan operasional untuk mengidentifikasi *issue* berdurasi tinggi. Pendekatan ini selaras dengan studi prediksi waktu resolusi yang menggunakan atribut *bug* atau *issue* untuk memperkirakan apakah suatu pekerjaan akan selesai cepat atau lambat [5, 6, 7].

Selain itu, studi industri terhadap *lead time* penyelesaian *issue* Jira menunjukkan bahwa durasi penyelesaian dapat dipengaruhi oleh berbagai faktor proyek, sehingga prediksi risiko durasi tinggi perlu dipahami sebagai masalah yang bergantung pada konteks dan distribusi data [13].

Prediksi waktu resolusi memiliki nilai praktis karena dapat membantu perencanaan *sprint*, pengendalian *backlog*, dan penentuan prioritas mitigasi. Apabila risiko durasi tinggi dapat diketahui pada fase awal, manajer proyek dapat melakukan intervensi lebih cepat, misalnya dengan meninjau ulang prioritas, memecah pekerjaan, mengurangi hambatan koordinasi, atau memindahkan sumber daya.

Penelitian terbaru juga menunjukkan bahwa estimasi waktu resolusi *issue* dapat diperkuat dengan memanfaatkan metadata seperti prioritas, komponen, label, dan histori *assignee*, serta tetap membutuhkan interpretasi agar hasil prediksi dapat dipahami oleh pengambil keputusan [14].

2.4 *Early Warning* Dinamis

Early warning dinamis merujuk pada mekanisme prediksi yang tidak hanya dilakukan sekali pada saat *issue* dibuat, tetapi diperbarui pada titik observasi tertentu selama *issue* masih berjalan. Kula dkk. menunjukkan bahwa model prediksi keterlambatan yang memanfaatkan dinamika proyek dapat mengungguli pendekatan statis [2]. Prinsip ini penting karena risiko keterlambatan dapat berubah setelah terjadi perpindahan status, penambahan tautan, perubahan *assignee*, atau munculnya pola *rework*.

Dalam prediksi dinamis, skor risiko tidak harus meningkat secara monoton dari satu horizon ke horizon berikutnya. Skor dapat meningkat atau menurun

ketika tersedia informasi baru mengenai status, transisi, tautan, *assignee*, atau pola *rework*. Kula dkk. [2] menunjukkan bahwa risiko keterlambatan perlu dinilai ulang sepanjang pelaksanaan proyek karena kondisi proyek dapat berubah. Oleh karena itu, perbedaan skor antara Day-1, Day-7, dan Day-30 dipahami sebagai pembaruan penilaian berdasarkan informasi yang tersedia pada masing-masing horizon, bukan sebagai inkonsistensi data.

Pada konteks *issue* Jira, titik observasi dapat dibentuk sebagai *snapshot* Day-1, Day-7, dan Day-30. Day-1 menggambarkan peringatan paling awal, Day-7 menggambarkan sinyal utama setelah pekerjaan berjalan satu minggu, sedangkan Day-30 menggambarkan tahap eskalasi untuk *issue* yang tetap belum selesai setelah satu bulan. Pendekatan ini memungkinkan skor risiko diperbarui setelah tersedia informasi baru pada Day-1, Day-7, dan Day-30, sedangkan model statis hanya menghasilkan satu prediksi.

2.5 *Process Mining* dan *Event Log*

Process mining adalah pendekatan untuk menganalisis proses nyata berdasarkan data kejadian yang terekam pada sistem informasi. Pendekatan ini digunakan untuk memahami, memantau, dan mengevaluasi proses aktual berdasarkan jejak peristiwa yang terjadi selama proses berjalan [11]. Dalam *process mining*, *event log* menjadi struktur dasar yang minimal memuat *case identifier*, aktivitas, dan *timestamp* [12].

Pada data Jira, satu *issue* dapat diperlakukan sebagai satu *case*. Perubahan status dapat diperlakukan sebagai aktivitas, sedangkan waktu perubahan dapat menjadi *timestamp*. Dengan struktur tersebut, riwayat perubahan status Jira dapat direpresentasikan sebagai *event log* yang menggambarkan pergerakan *issue* dari satu status ke status lain.

Transformasi riwayat perubahan Jira menjadi *event log* memungkinkan terbentuknya fitur proses, seperti jumlah transisi status, jumlah status unik, waktu tunggu antar-status, waktu berada pada status saat ini, indikasi *rework*, indikasi *reopen*, dan kecepatan perpindahan status. Fitur-fitur tersebut menggambarkan dinamika kerja yang tidak dapat dilihat hanya dari atribut statis. Pemanfaatan *event log* untuk prediksi keadaan proses berjalan juga sejalan dengan konsep *predictive process monitoring*, yaitu penggunaan data proses historis untuk memprediksi keluaran dari proses yang masih berjalan [15].

PM4Py menyediakan dukungan perangkat lunak untuk *process mining*

berbasis Python, termasuk dukungan berbagai format *event log* dan integrasi dengan ekosistem analitik data [16]. Dengan demikian, PM4Py relevan sebagai landasan teknis dalam pembentukan representasi proses dan validasi struktur *event log* pada data Jira.

Pada metodologi yang dikembangkan, *process mining* digunakan untuk membentuk *event log* dan fitur prediktif dari riwayat perubahan status Jira, bukan untuk melakukan *process discovery* secara penuh.

2.6 Fitur Histori *Assignee*

Durasi penyelesaian *issue* tidak hanya dipengaruhi oleh karakteristik *issue*, tetapi juga oleh pola kerja historis *assignee*. Özkan dkk. menunjukkan bahwa *assignee* dan *reporter* dapat berperan penting dalam proses penyelesaian *bug* [6]. Akan tetapi, penggunaan identitas *assignee* mentah dapat membuat model terlalu bergantung pada individu tertentu sehingga portabilitas model menurun.

Pendekatan yang lebih portabel adalah menggunakan statistik historis *assignee*, seperti jumlah *issue* yang pernah ditangani sebelumnya, rata-rata durasi penyelesaian sebelumnya, dan rata-rata durasi penyelesaian pada sejumlah *issue* terakhir. Fitur tersebut menangkap konteks pengalaman historis, volume *issue* yang pernah ditangani, dan pola durasi penyelesaian tanpa menjadikan identitas personal sebagai fitur utama. Fitur ini tidak merepresentasikan beban kerja aktif karena tidak menggunakan jumlah *issue* yang sedang dikerjakan secara bersamaan, jadwal kerja, atau ketersediaan pegawai. Pendekatan ini juga sejalan dengan karakter *dataset* publik yang menggunakan anonimisasi identitas pengguna [4].

2.7 CatBoost dan Data Tabular Campuran

CatBoost adalah algoritma *gradient boosting* berbasis pohon keputusan yang mendukung fitur kategorikal dan numerik. Karakteristik ini sesuai dengan data Jira karena fitur yang digunakan terdiri atas kategori, seperti proyek, tipe *issue*, prioritas, dan status, serta fitur numerik, seperti jumlah transisi, jumlah tautan, dan durasi historis. Dokumentasi CatBoost menyediakan *CatBoostClassifier* untuk klasifikasi, dukungan parameter pembobotan kelas, serta fungsi penyimpanan dan pemuatan model [17].

Penerapan CatBoost pada data rekayasa perangkat lunak juga telah ditunjukkan oleh De Rosa dan Liguori [8]. Penelitian tersebut menggunakan

CatBoost bersama beberapa model *supervised* lain untuk memprediksi *residual defect* berdasarkan metrik produk, proses, statistik, dan aktivitas pengembang. Meskipun berbeda dari prediksi keterlambatan *issue*, penelitian tersebut menunjukkan relevansi CatBoost untuk klasifikasi berbasis metrik perangkat lunak.

Penggunaan CatBoost juga relevan karena model dapat menghasilkan kontribusi fitur melalui *ShapValues*. CatBoost mendefinisikan *ShapValues* sebagai vektor kontribusi setiap fitur terhadap prediksi untuk setiap objek, ditambah nilai ekspektasi model [10]. Kemampuan ini penting untuk menghasilkan penjelasan lokal, misalnya faktor utama yang mendorong skor risiko menjadi lebih tinggi atau lebih rendah pada satu *issue* tertentu.

2.8 Perbandingan Pemilihan Algoritma

Beberapa algoritma dapat digunakan untuk klasifikasi data tabular. Perbandingan karakteristik algoritma yang relevan untuk klasifikasi data tabular Jira ditunjukkan pada Tabel 2.2.

Tabel 2.2. Perbandingan algoritma untuk data tabular Jira

Algoritma	**Kelebihan**	**Keterbatasan**
Logistic Regression	Sederhana dan mudah diinterpretasikan.	Terbatas dalam menangkap hubungan nonlinier dan interaksi fitur yang kompleks.
Random Forest	Mampu menangkap hubungan nonlinier pada data tabular.	Fitur kategorikal umumnya memerlukan transformasi sebelum pelatihan.
XGBoost	Memiliki kinerja yang kuat pada berbagai tugas data tabular.	Penanganan fitur kategorikal memerlukan prapemrosesan atau konfigurasi tambahan.
CatBoost	Mendukung fitur kategorikal dan numerik secara langsung serta menyediakan kontribusi fitur untuk interpretasi prediksi.	Tetap memerlukan penanganan ketimpangan kelas dan penentuan ambang keputusan.

Berdasarkan karakteristik tersebut, CatBoost dipilih karena data Jira memuat fitur kategorikal dan numerik dalam jumlah besar. Selain itu, CatBoost menyediakan pembobotan kelas, keluaran probabilitas, dan nilai kontribusi fitur yang mendukung kebutuhan prediksi serta interpretasi model dalam penelitian ini.

2.9 Penentuan Ambang Keputusan dan Evaluasi Model

Model klasifikasi umumnya tidak langsung menghasilkan keputusan akhir berupa kelas positif atau negatif. Model terlebih dahulu menghasilkan skor probabilitas, yaitu nilai antara 0 sampai 1 yang menunjukkan seberapa besar kemungkinan suatu data masuk ke kelas tertentu. Agar skor tersebut dapat diubah menjadi label, diperlukan ambang keputusan. Ambang keputusan adalah batas nilai probabilitas yang digunakan untuk menentukan apakah suatu *issue* dikategorikan berisiko atau tidak. Dokumentasi scikit-learn menjelaskan bahwa klasifikasi dapat dipisahkan menjadi dua masalah, yaitu masalah statistik untuk menghasilkan skor atau probabilitas dan masalah keputusan untuk menentukan tindakan berdasarkan skor tersebut [18]. Oleh karena itu, ambang keputusan tidak selalu harus bernilai 0,5, terutama pada skenario *early warning* yang menekankan *recall* terhadap kelas berisiko.

Evaluasi model klasifikasi didasarkan pada *confusion matrix*, yaitu komponen *true positive* (TP), *true negative* (TN), *false positive* (FP), dan *false negative* (FN). Metrik yang digunakan adalah *accuracy*, *precision*, *recall*, *F1-score*, ROC-AUC, dan *Average Precision* (AP). *Accuracy* mengukur proporsi prediksi benar terhadap seluruh data. *Precision* mengukur ketepatan prediksi positif, sedangkan *recall* mengukur kemampuan model menemukan seluruh kasus positif. *F1-score* menggabungkan *precision* dan *recall*. ROC-AUC digunakan untuk mengevaluasi kemampuan pemisahan kelas pada berbagai ambang. AP merangkum kurva *precision-recall* dengan pembobotan kenaikan *recall* pada setiap ambang keputusan [19]. AP penting pada data tidak seimbang karena fokus pada kualitas peringkat terhadap kelas positif.

Perhitungan *accuracy*, *precision*, *recall*, dan *F1-score* masing-masing ditunjukkan pada Persamaan 2.1, Persamaan 2.2, Persamaan 2.3, dan Persamaan 2.4.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.3}$$

$$F1\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.4}$$

Tabel 2.3. Interpretasi metrik evaluasi klasifikasi

Metrik	Nilai tinggi menunjukkan	Nilai rendah menunjukkan
Accuracy	Sebagian besar prediksi benar terhadap seluruh data.	Banyak prediksi salah terhadap seluruh data. Pada data tidak seimbang, nilai ini dapat kurang informatif.
Precision	Sebagian besar *issue* yang diprediksi berisiko benar-benar termasuk kelas berisiko.	Banyak *false positive*, sehingga model sering memberi peringatan pada *issue* yang sebenarnya tidak berisiko.
Recall	Sebagian besar *issue* berisiko berhasil ditemukan oleh model.	Banyak *false negative*, sehingga banyak *issue* berisiko tidak terdeteksi.
F1-score	Nilai harmonik *precision* dan *recall* tinggi.	Salah satu atau kedua nilai *precision* dan *recall* rendah.
ROC-AUC	Model mampu memisahkan kelas positif dan negatif pada berbagai ambang keputusan.	Kemampuan pemisahan kelas masih lemah dan skor model sulit membedakan kelas positif dari kelas negatif.
Average Precision	Model mampu memberi peringkat tinggi pada kelas positif, terutama pada data tidak seimbang.	Kualitas peringkat terhadap kelas positif masih rendah, sehingga kasus berisiko tidak konsisten berada di skor tertinggi.

Berdasarkan Tabel 2.3, pada skenario *early warning*, *precision* dan *recall* perlu dibaca sesuai kebutuhan operasional. *Recall* yang tinggi penting apabila tujuan utama adalah menemukan sebanyak mungkin *issue* berisiko. Sebaliknya, *precision* yang tinggi penting apabila organisasi ingin mengurangi jumlah peringatan palsu. *F1-score* digunakan sebagai ukuran ringkas ketika *precision* dan *recall* perlu dipertimbangkan secara seimbang.

2.10 Research Gap

Berdasarkan kajian teori dan penelitian terdahulu, celah penelitian dirumuskan sebagai berikut.

1. Belum banyak penelitian yang mengintegrasikan metadata *issue*, fitur proses dari *event log*, dan histori *assignee* dalam satu rancangan *early warning* multi-horizon pada data Jira publik.

2. Kontribusi fitur proses terhadap prediksi risiko durasi tinggi dan estimasi waktu tersisa belum banyak diuji melalui perbandingan pada sumber data,

target, pembagian temporal, algoritma, dan prosedur evaluasi yang sama.

Atas dasar celah tersebut, penelitian ini membangun model CatBoost pada horizon Day-1, Day-7, dan Day-30 serta membandingkan beberapa kelompok fitur dalam satu protokol evaluasi temporal. Penjelasan lokal digunakan sebagai keluaran pendukung untuk membantu membaca prediksi model dan tidak diposisikan sebagai masalah penelitian yang terpisah.