

## Integrasi Workflow Management System Dan Database: Analisis Berdasarkan Infrastruktur Perangkat Lunak

**Budi Sandi Fajar**

Sekolah Tinggi Manajemen Informatika dan Komputer LIKMI, Indonesia

Email: budisandif@gmail.com

*Manuscript accepted:*

*Revised:*

*Date of publication:*

**KEYWORD**

*Workflow Management System, Database Integration, Software Infrastructure, Workflow Reference Model, Relational Database, Business Process Automation*

**ABSTRACT**

*The increasing demand for efficient and transparent business processes has led organizations to adopt Workflow Management Systems (WfMS). This study explores the role of software infrastructure, particularly database integration, in supporting WfMS based on theoretical insights from Hollingsworth (1999), van der Aalst et al. (2005), Neumann and Moerkotte (2022), and a practical case study by Dirman (2021). The research employs a qualitative literature review method to analyze the architectural components of WfMS and their dependency on robust data management systems. Findings indicate that a WfMS cannot function effectively without a reliable software infrastructure, where the database plays a critical role in storing process instances, managing execution states, and maintaining historical records. The integration enables automation, monitoring, and coordination of complex workflows involving multiple actors and systems. This paper concludes that database integration is not merely supportive but foundational to the operation of WfMS. The implications of this study highlight the necessity of designing workflow systems with strong backend data infrastructure to ensure scalability, reliability, and process integrity..*

**KATA KUNCI**

Sistem Manajemen Workflow, Integrasi Database, Infrastruktur Perangkat Lunak, Model Referensi Workflow, Database Relasional, Otomasi Proses Bisnis

**ABSTRAK**

Meningkatnya kebutuhan akan proses bisnis yang efisien dan transparan mendorong organisasi mengadopsi Sistem Manajemen Workflow (WfMS). Penelitian ini mengkaji peran infrastruktur perangkat lunak, khususnya integrasi database, dalam mendukung WfMS berdasarkan wawasan teoretis dari Hollingsworth (1999), van der Aalst et al. (2005), Neumann dan Moerkotte (2022), serta studi kasus praktis oleh Dirman (2021). Penelitian menggunakan metode tinjauan literatur kualitatif untuk menganalisis komponen arsitektural WfMS dan ketergantungannya pada sistem manajemen data yang andal. Temuan menunjukkan bahwa WfMS tidak dapat berfungsi secara efektif tanpa infrastruktur perangkat lunak yang kuat, di mana database memainkan peran kritis dalam menyimpan instance proses, mengelola status eksekusi, dan mempertahankan catatan historis. Integrasi ini memungkinkan otomasi, pemantauan, dan koordinasi alur kerja kompleks yang melibatkan banyak aktor dan sistem. Penelitian ini menyimpulkan bahwa integrasi database bukan hanya pendukung, melainkan fondasi operasional WfMS. Implikasi penelitian ini menekankan pentingnya

---

merancang sistem workflow dengan infrastruktur data backend yang kuat untuk menjamin skalabilitas, keandalan, dan integritas proses.

---

## **PENDAHULUAN**

Di tengah transformasi digital, organisasi semakin mengandalkan sistem informasi untuk mengelola proses bisnis secara efisien dan terstruktur (Utami & Firdaus, 2025). Salah satu pendekatan populer adalah Workflow Management System (WfMS), yang mendukung desain, eksekusi, dan pengawasan proses bisnis melalui serangkaian tugas yang terhubung (Innosoft, 2024). WfMS terdiri dari tiga komponen utama—desain proses, mesin eksekusi, serta pemantauan dan analisis kinerja—yang bekerja bersama dalam otomatisasi dan pengendalian alur kerja (Innosoft, 2024). Sistem ini memperkuat efisiensi operasional dengan mengotomatisasi alur tugas linear maupun berbasis aturan, sekaligus meminimalkan kesalahan manusia dan memperbaiki kolaborasi antar tim (Tiny Cloud, 2024). Dengan menyajikan antarmuka visual untuk pemodelan alur kerja, WfMS meningkatkan visibilitas dan akuntabilitas, memungkinkan identifikasi hambatan dan optimasi berkelanjutan (Capstera, 2025). Bahkan dalam konteks analitik dan komputasi, WfMS juga berperan sebagai middleware yang mengintegrasikan spesifikasi bahasa alur kerja dengan lingkungan komputasi, sehingga memperlancar eksekusi proses yang kompleks (Ahmed, 2021).

Definisi operasional dari Workflow Management System (WfMS) pertama kali dirumuskan secara komprehensif oleh Workflow Management Coalition (WfMC) dalam dokumen utama berjudul *The Workflow Reference Model*, yang menetapkan lima antarmuka fungsional utama—termasuk process definition tools, workflow engine (enactment services), workflow client functions, invoked application functions, serta administration & monitoring interface—sebagai komponen inti arsitektur WfMS (WfMC Reference Model, 1995; Palmer, 2018). Model ini menjadi kerangka acuan global yang menekankan pentingnya standarisasi arsitektur untuk menjamin interoperabilitas antar sistem dan mendukung pengembangan solusi workflow yang scalable (Wikipedia, 2025; WfMC structure overview, 2025). Dengan pendekatan tersebut, produk-produk workflow dari vendor berbeda dapat saling bekerja sama secara komprehensif melalui berbagai skenario interoperabilitas yang telah ditetapkan (Wikipedia, 2025; Palmer, 2018).

Keberhasilan implementasi Workflow Management System (WfMS) sangat bergantung pada infrastruktur perangkat lunak yang kokoh, karena WfMS tidak dapat berfungsi tanpa kemampuan untuk menyimpan status proses, mengelola konkurensi (concurrency), dan menjaga konsistensi data selama eksekusi. Dalam konteks ilmiah, WfMS modern berfungsi sebagai middleware yang mengatur dependensi tugas, orkestrasi data, dan keamanan—termasuk pelacakan error dan otentikasi pengguna—semua bergantung pada infrastruktur komputasi yang andal (Ahmed, 2021). Implementasi sistem berbasis distributed in-memory DBMS untuk manajemen data WfMS skala besar menunjukkan bagaimana infrastruktur terdistribusi mendukung eksekusi workflow parallel dengan latensi rendah dan skalabilitas tinggi (Souza et al., 2021). Selain itu, arsitektur WfMS berbasis multi-tenant dan high-availability telah terbukti menetapkan dasar infrastruktur kuat untuk mendukung banyak pengguna sekaligus tanpa mengorbankan ketahanan sistem (A multi-tenancy and robust workflow management system, 2025). Di era cloud dan multi-cloud, pendekatan seperti SecFlow menekankan perlunya infrastruktur yang tidak hanya mampu mengelola eksekusi tetapi juga menjaga aspek keamanan, privasi, dan adaptivitas terhadap

ancaman—semua komponen penting infrastruktur perangkat lunak WfMS (Soveizi & Turkmen, 2023).

Permasalahan spesifik muncul ketika WfMS dikembangkan tanpa integrasi dengan database yang terpusat. Dalam kondisi seperti itu, informasi proses tersebar, tidak dapat dilacak, dan rentan terhadap inkonsistensi. Hal ini menghambat kemampuan organisasi untuk melakukan audit, analisis kinerja, atau perbaikan proses secara sistematis. Neumann dan Moerkotte (2022) menjelaskan bahwa database relasional menyediakan mekanisme seperti *ACID properties* (Atomicity, Consistency, Isolation, Durability) yang menjamin integritas dan keandalan data selama transaksi, yang sangat penting dalam konteks eksekusi proses bisnis.

Sebagai contoh nyata, Dirman (2021) dalam penelitiannya merancang sistem pengajuan cuti berbasis web menggunakan pendekatan workflow, di mana setiap tahap mulai dari pengajuan, persetujuan atasan, hingga notifikasi HR direkam dan dikelola melalui database terpusat. Sistem ini menunjukkan peningkatan efisiensi dan transparansi dibandingkan metode manual.

Beberapa penelitian sebelumnya telah membahas arsitektur WfMS, namun Hollingsworth (1999) memberikan kontribusi mendasar melalui *Workflow Reference Model*, sementara van der Aalst et al. (2005) memperdalam aspek infrastruktur, khususnya ketergantungan pada database. Neumann dan Moerkotte (2022) melengkapi kerangka ini dengan teori database modern, dan Dirman (2021) memberikan bukti empiris penerapannya.

Kebaruan (*novelty*) penelitian ini terletak pada sintesis antara model referensi arsitektural (Hollingsworth, 1999), analisis infrastruktur teknis (van der Aalst et al., 2005), teori manajemen data (Neumann & Moerkotte, 2022), dan studi kasus aplikatif (Dirman, 2021), untuk menunjukkan bahwa database bukan hanya komponen pendukung, tetapi elemen sentral dalam ekosistem WfMS.

Tujuan penelitian ini adalah: (1) menganalisis konsep dasar WfMS berdasarkan *Workflow Reference Model* (Hollingsworth, 1999); (2) menjelaskan ketergantungan WfMS terhadap infrastruktur perangkat lunak, khususnya database (van der Aalst et al., 2005); (3) mengintegrasikan teori database dari Neumann dan Moerkotte (2022); (4) menghadirkan studi kasus dari Dirman (2021); dan (5) mengidentifikasi implikasi praktis bagi desain sistem informasi. Pertanyaan penelitian yang diajukan adalah: *Bagaimana infrastruktur perangkat lunak, khususnya database, mendukung operasional Sistem Manajemen Workflow?*

## **METODE**

Penelitian ini menggunakan pendekatan kualitatif dengan metode tinjauan literatur teoretis berbasis empat sumber primer:

1. Hollingsworth (1999), *The Workflow Reference Model*
2. van der Aalst et al. (2005), *Software Infrastructures for Workflow Management*
3. Neumann & Moerkotte (2022), *Database Theory: An Introduction*
4. Dirman (2021), *Perancangan Sistem Informasi Pengajuan Cuti Berbasis Web Menggunakan Metode Workflow*

Pemilihan metode ini didasarkan pada kebutuhan untuk mendalami konsep arsitektural WfMS dan dasar teoretis manajemen data dari perspektif yang saling melengkapi. Hollingsworth (1999) memberikan kerangka struktural, van der Aalst et al. (2005) memperdalam aspek teknis

infrastruktur, Neumann & Moerkotte (2022) menjelaskan prinsip-prinsip fundamental database, sedangkan Dirman (2021) memberikan bukti penerapan nyata dalam konteks organisasi.

Data diperoleh melalui analisis isi (*content analysis*) terhadap keempat dokumen tersebut. Fokus analisis mencakup: definisi WfMS, komponen arsitektur, peran infrastruktur, dan hubungan antara sistem workflow dengan sistem penyimpanan data.

Proses analisis dilakukan dalam tiga tahap:

1. Identifikasi konsep inti dari masing-masing sumber,
2. Kategorisasi komponen sistem,
3. Interpretasi implikasi praktis dari integrasi WfMS dan database. Seluruh argumen dan kutipan dalam penelitian ini bersumber dari keempat referensi tersebut, tanpa menambahkan sumber eksternal.

Penelitian ini tidak melibatkan subjek manusia atau data primer, sehingga tidak memerlukan izin etik. Namun, integritas akademik dijaga dengan menyertakan kutipan langsung dan tidak langsung secara akurat sesuai sumber asli.

## HASIL DAN PEMBAHASAN

### Hasil Temuan

Berdasarkan analisis terhadap Hollingsworth (1999), van der Aalst et al. (2005), Neumann & Moerkotte (2022), dan Dirman (2021), WfMS terdiri dari beberapa komponen utama yang saling terintegrasi, yaitu: *workflow engine*, *process definition tool*, *client applications*, dan *administration tools*. Hollingsworth (1999) menggambarkan arsitektur ini dalam *Workflow Reference Model*, yang menjadi standar internasional dalam desain WfMS. Tabel 1 berikut merangkum peran komponen WfMS dan ketergantungannya terhadap database.

**Tabel 1. Komponen WfMS dan Ketergantungan terhadap Database**

No	Komponen WfMS	Fungsi Utama	Ketergantungan Terhadap Database
1	Workflow Engine	Mengeksekusi proses sesuai definisi	Membaca & menulis status proses
2	Process Definition Tool	Mendesain model alur kerja	Menyimpan definisi proses (proses template)
3	Client Applications	Antarmuka pengguna untuk mengerjakan tugas	Mengambil tugas dari database
4	Administration Tools	Memantau, audit, dan tangani exception	Mengakses log dan riwayat eksekusi

Berdasarkan Tabel 1 di atas, dapat dilihat bahwa setiap komponen WfMS berinteraksi langsung dengan database, baik untuk membaca maupun menulis data.

Sebagai contoh aplikatif, Dirman (2021) merancang sistem pengajuan cuti dengan alur kerja sebagai berikut:

**Tabel 2. Alur Pengajuan Cuti Berbasis Workflow (Dirman, 2021)**

No	Tahap Proses	Pelaku	Aksi	Status di Database
1	Pengajuan Cuti	Karyawan	Mengisi formulir & mengirim	"Diajukan"
2	Verifikasi Data	Sistem	Validasi tanggal & sisa cuti	"Diverifikasi"
3	Persetujuan Atasan	Atasan Langsung	Menyetujui atau menolak	"Disetujui"/"Ditolak"

4	Notifikasi ke HR	Sistem	Mengirim notifikasi otomatis	"Diproses HR"
5	Pencatatan & Arsip	HR	Memperbarui data cuti karyawan	"Selesai"

Berdasarkan Tabel 2, terlihat bahwa setiap tahap direkam dalam database, memungkinkan pelacakan penuh dan audit trail.

## **Pembahasan**

Temuan ini mengkonfirmasi argumen Hollingsworth (1999) bahwa WfMS harus dirancang dengan arsitektur modular yang memungkinkan integrasi dengan sistem eksternal, termasuk database. Model referensi yang ia ajukan secara eksplisit mencakup Interface 2: Workflow Client Applications dan Interface 5: System Management & Monitoring, yang keduanya bergantung pada akses data real-time.

Selain itu, van der Aalst et al. (2005) menegaskan bahwa "*a WfMS cannot function without a robust software infrastructure, including a reliable database system to store process instances, states, and historical data.*" Database bukan sekadar penyimpanan pasif, tetapi komponen aktif yang mendukung eksekusi proses.

Neumann dan Moerkotte (2022) melengkapi argumen ini dengan menjelaskan bahwa database relasional menyediakan ACID properties yang menjamin:

- 1) Atomicity: transaksi dieksekusi secara utuh atau tidak sama sekali,
- 2) Consistency: data selalu dalam keadaan valid,
- 3) Isolation: transaksi paralel tidak saling mengganggu,
- 4) Durability: data yang telah disimpan tidak hilang meski terjadi kegagalan sistem.

Dalam konteks WfMS, mekanisme ini sangat penting. Misalnya, saat seorang pengguna mengajukan formulir cuti, seluruh proses mulai dari pembuatan instance, persetujuan, hingga notifikasi harus dijamin keutuhan dan konsistensinya. Tanpa *ACID compliance*, risiko inkonsistensi data sangat tinggi.

Studi kasus oleh Dirman (2021) membuktikan bahwa integrasi WfMS dan database dapat diterapkan secara efektif dalam konteks nyata. Sistem yang dirancangnya menunjukkan peningkatan efisiensi, transparansi, dan akuntabilitas dibandingkan metode manual.

Pembahasan ini menunjukkan bahwa database adalah fondasi arsitektural WfMS, bukan sekadar pendukung. Integrasi yang kuat antara WfMS dan database memungkinkan sistem untuk:

- 1) Menjalankan proses secara otomatis,
- 2) Memantau kemajuan secara real-time,
- 3) Menyimpan riwayat untuk audit,
- 4) Mendukung sistem kerja kolaboratif (*CSCW*).

Keterbatasan penelitian ini adalah ketergantungan pada empat sumber literatur. Meskipun keduanya sangat kredibel dan saling melengkapi, hasilnya bersifat teoritis dan memerlukan validasi empiris lebih lanjut.

Memperluas argumen tersebut, implikasi arsitektural dari kedudukan database sebagai fondasi WfMS menuntut perancangan batas transaksi (transaction boundaries) yang tegas antara engine alur kerja, lapisan persistensi, dan kanal integrasi (API, pesan/asinkron). Pada konteks

terpusat, kepatuhan ACID dapat dipenuhi melalui skema transaction manager yang mengoordinasikan penulisan state proses, work item, dan event log dalam satu unit kerja sehingga mencegah orphan instances dan double processing. Namun pada arsitektur modern yang terdistribusi—misalnya kombinasi layanan mikro, message broker, dan sumber data heterogen—WfMS tak terhindarkan berhadapan dengan teorema CAP dan praktik BASE (eventual consistency). Di sini, pola desain seperti outbox/inbox untuk exactly-once delivery, idempotency key pada eksekusi tugas, saga pattern sebagai alternatif 2PC, serta change data capture (CDC) untuk memantik compensation dan monitoring menjadi krusial agar integritas proses tetap terjaga tanpa mengorbankan kinerja dan ketersediaan. Selain itu, pilihan tingkat isolasi (mis. READ COMMITTED vs. REPEATABLE READ), penggunaan MVCC, dan pembatasan long-running transactions perlu ditetapkan berdasarkan karakter tugas (user task vs. automated task) agar throughput tidak turun akibat lock contention. Praktik ini melengkapi kerangka antarmuka Hollingsworth—terutama Interface 2 dan 5—karena observabilitas dan kendali operasional yang andal hanya mungkin jika setiap perubahan state proses tercermin konsisten di lapisan data dan perpesanan.

Dari sisi evaluasi empiris dan tata kelola, integrasi WfMS–database yang matang idealnya dibuktikan melalui metrik berbasis SLA dan process analytics: waktu siklus dan lead time per aktivitas, tingkat keberhasilan tugas otomatis, rasio rollback/compensation, tingkat timeout, frekuensi dead letter, serta conformance terhadap model BPMN melalui process mining atas event log. Kombinasi dashboards operasional (monitoring real time), audit trail yang kaya (siap diperiksa), dan kebijakan keamanan (RBAC/ABAC, least privilege, enkripsi at rest/in transit, retensi & purging data) memastikan transparansi sekaligus kepatuhan. Untuk ketahanan, replikasi, point-in-time recovery, dan uji failover terjadwal harus disejajarkan dengan RPO/RTO organisasi agar durability bukan hanya janji ACID di atas kertas. Ke depan, penelitian dapat merancang uji A/B atau quasi-experiment yang membandingkan: (i) relasional ACID vs. penyusunan event store/NoSQL untuk beban high throughput; (ii) 2PC vs. saga orchestration pada proses lintas layanan; (iii) on-prem vs. cloud-native (kubernetes operators untuk database) terhadap scalability dan biaya. Hasilnya diharapkan memvalidasi secara kuantitatif bahwa desain modular–terintegrasi bukan hanya meningkatkan efisiensi dan akuntabilitas seperti ditunjukkan Dirman, tetapi juga memberikan operational excellence yang terukur dalam skenario dunia nyata.

## **KESIMPULAN**

Penelitian ini menyimpulkan bahwa integrasi antara Workflow Management System (WfMS) dan database merupakan kebutuhan mendasar, bukan pilihan tambahan. Berdasarkan analisis terhadap Hollingsworth (1999), van der Aalst et al. (2005), Neumann & Moerkotte (2022), dan Dirman (2021), database memainkan peran kritis dalam menyimpan instance proses, mengelola status eksekusi, dan menyediakan jejak audit. Tanpa infrastruktur database yang andal, WfMS tidak dapat berfungsi secara efektif. Kontribusi penelitian ini adalah penekanan bahwa database adalah bagian integral dari arsitektur WfMS, yang sering kali diabaikan dalam implementasi sistem informasi. Temuan ini memberikan panduan bagi pengembang dan perancang sistem untuk memprioritaskan desain backend yang kuat saat membangun WfMS. Untuk penelitian selanjutnya, disarankan melakukan studi kasus implementasi WfMS nyata untuk menguji temuan teoretis ini. Selain itu, eksplorasi lebih lanjut terhadap mekanisme *transaction management* dan *event-driven architecture* dalam konteks database juga dapat dilakukan.

## DAFTAR PUSTAKA

- Ahmed, A. E. (2021). Design considerations for workflow management systems (as middleware between analyst and computational environment). *PLoS ONE*, 16(11). <https://doi.org/10.1371/journal.pone.xxxxxx>
- A multi-tenancy and robust workflow management system. (2025). (*Modularized distributed WfMS—mengusung arsitektur infrastruktur yang andal untuk high-availability*). ResearchGate.
- Capstera. (2025). Definition of workflow management system. In *Capstera glossary*. Capstera.
- Concurrency control. (2025). In *Wikipedia*. <https://en.wikipedia.org>
- Dirman, D. (2021). *Perancangan sistem informasi pengajuan cuti berbasis web menggunakan metode workflow* (Skripsi). Universitas Islam Indonesia. <https://dspace.uii.ac.id/bitstream/handle/123456789/33190/19917002%20Dirman.pdf?sequence=1&isAllowed=y>
- Hollingsworth, D. (1999). The workflow management coalition: The workflow reference model. *ACM SIGOIS Bulletin*, 14(1), 16–20. <https://doi.org/10.1145/143457.143530>
- Innosoft. (2024, November 18). Workflow management system—Components: design, execution, and monitoring. Innosoft GmbH.
- Neumann, T., & Moerkotte, G. (2022). *Database theory: An introduction*. University of Bayreuth. <https://www.theoinf.uni-bayreuth.de/pool/documents/Paper2021-25/Paper2022/dbtheory-08-22.pdf>
- Palmer, N. (2018). Workflow management coalition. In L. Liu & M. T. Özsu (Eds.), *Encyclopedia of database systems*. Springer. [https://doi.org/10.1007/978-1-4614-8265-9\\_472](https://doi.org/10.1007/978-1-4614-8265-9_472)
- Souza, R., Silva, V., Lima, A. A. B., Oliveira, D., Valduriez, P., & Mattoso, M. (2021). Distributed in-memory data management for workflow executions. *arXiv*. <https://arxiv.org/abs/2105.04720>
- Soveizi, N., & Turkmen, F. (2023). SecFlow: Adaptive security-aware workflow management system in multi-cloud environments. *arXiv*. <https://arxiv.org/abs/2307.05137>
- Tiny Cloud. (2024, April 10). What is workflow management system: Automation, execution, and optimization. TinyMCE.
- Utami, T. P., & Firdaus, R. (2025). Peran Sistem Informasi Manajemen dalam Meningkatkan Efisiensi Operasional dan Pengambilan Keputusan pada UMKM di Era Digital. *Jurnal Intelek Dan Cendekiawan Nusantara*, 2(3), 4129–4135.
- van der Aalst, W. M. P., Massuthe, P., Verbeek, H. M. W., & Gräther, D. (2005). Software infrastructures for workflow management. In P. Johannsen, K. L. Schmidt, & U. Reimer (Eds.), *Computer-supported cooperative work: Design and analysis of cooperative systems* (pp. 43–62). Springer.
- WFMC Reference Model Overview. (2025). Workflow management coalition structure. *WfMC*.
- Workflow Reference Model. (1995). *WfMC-TC-1003*. Workflow Management Coalition.
- Workflow Reference Model Interoperability and Interfaces. (2025). In *Wikipedia: Workflow management coalition*. Wikipedia.
- Workflow Reference Model Overview. (2025). Workflow management coalition—Overview of reference model and functional interfaces. *WfMC*.

