Jurnal Teliska
Volume 19 No.3 November 2026
DOI : 10.36257/teliska.v19i3.12612
P-ISSN: 2085-0786
E-ISSN: 2654-2765

IMPLEMENTASI ALGORITMA KAHN UNTUK DEPENDENCY TERMIN PEMBAYARAN PADA APLIKASI PENGADAAN PT. KARYA AMAURY NUSANTARA

Jihan Kamila[1], Ade Sukma Wati1[2], Lailatur Rahmi[3], Ferizka Tiara Devani[4], Robinson[5], Panisha[6]
1-6 Program Studi DIV Manajemen Informatika, Politeknik Negeri Sriwijaya, Palembang, Indonesia
*Email korespondensi: adesukmawati22@polsri.ac.id
Submitted : 8 September 2026 | Accepted 1 November 2026 | Published : 10 November 2026

ABSTRAK
Administrasi pembayaran pada pekerjaan pengadaan dengan mekanisme termin membutuhkan pengendalian urutan proses karena setiap termin memiliki prasyarat yang harus dipenuhi sebelum termin berikutnya dapat diproses. Pada PT. Karya Amaury Nusantara, monitoring termin pembayaran sebelumnya masih memanfaatkan spreadsheet dan dokumen fisik sehingga berpotensi menimbulkan kesalahan pencatatan, keterlambatan pemantauan, dan pembayaran yang tidak sesuai urutan kontrak. Penelitian ini bertujuan mengembangkan sistem pengelolaan termin pembayaran yang mampu memodelkan hubungan *dependency* dan memvalidasi urutan pembayaran secara otomatis. Hubungan *dependency* antar termin direpresentasikan menggunakan *Directed Acyclic Graph* (DAG), sedangkan pengurutan termin dilakukan dengan algoritma Kahn melalui pendekatan *topological sorting* berbasis nilai *in-degree*. Pengembangan perangkat lunak menerapkan model Waterfall yang meliputi analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan. Sistem diimplementasikan sebagai aplikasi berbasis web menggunakan PHP dengan framework CodeIgniter 4 dan basis data MySQL. Fungsionalitas sistem mencakup pengelolaan perusahaan, klien, kontrak, realisasi pekerjaan, invoice, konfirmasi pembayaran, dan validasi *dependency* termin. Hasil implementasi menunjukkan bahwa algoritma Kahn dapat digunakan untuk menghasilkan urutan termin yang valid berdasarkan hubungan prasyarat yang ditetapkan, sehingga pemrosesan termin dapat dikendalikan sesuai urutan kontrak. Pengujian sistem dilakukan terhadap skenario pengujian blackbox testing dengan tingkat keberhasilan sebesar 90%. Hasil pengujian tersebut digunakan untuk mengevaluasi kesesuaian fungsi sistem terhadap kebutuhan yang telah ditentukan. Sistem yang dikembangkan diharapkan dapat meningkatkan keteraturan administrasi pembayaran, mengurangi risiko kesalahan urutan pemrosesan termin, dan mendukung monitoring kontrak secara terintegrasi.
Kata kunci: topological sorting; algoritma Kahn; dependency; termin pembayaran; sistem pengadaan

ABSTRACT
Payment administration in procurement projects involving a milestone-based payment mechanism requires proper process sequencing because each payment milestone has prerequisites that must be fulfilled before the subsequent milestone can be processed. At PT Karya Amaury Nusantara, payment milestone monitoring was previously conducted using spreadsheets and physical documents, potentially leading to recording errors, monitoring delays, and payment processing that did not comply with the contractual sequence. This study aims to develop a payment milestone management system capable of modeling dependencies and automatically validating payment sequences. Dependencies between payment milestones are represented using a Directed Acyclic Graph (DAG), while milestone sequencing is performed using Kahn's algorithm through topological sorting based on in-degree values. Software development follows the Waterfall model, comprising requirements analysis, design, implementation, testing, and maintenance. The system is implemented as a web-based application using PHP with the CodeIgniter 4 framework and a MySQL database. Its functionalities include company management, client management, contract management, work realization tracking, invoice management, payment confirmation, and dependency validation between payment milestones. The implementation results indicate that Kahn's algorithm can generate valid milestone sequences based on predefined prerequisite relationships. Black-box testing results demonstrate a success rate of 90%, indicating the conformity of the system's functions with the specified test scenarios. The developed system is expected to support more structured payment administration, minimize the risk of incorrect payment milestone sequencing, and improve the effectiveness of integrated contract monitoring.
Keyword: topological sorting; Kahn's algorithm; dependency; payment terms; procurement system

1. PENDAHULUAN

Administrasi pembayaran merupakan bagian penting dalam pengelolaan kontrak pengadaan karena berkaitan langsung dengan pemenuhan kewajiban perusahaan, pengendalian dokumen, dan pencatatan realisasi pekerjaan. Pada proyek yang menggunakan mekanisme pembayaran bertahap, setiap termin memiliki nilai, jadwal, dokumen pendukung, serta persyaratan penyelesaian tertentu. Pengelolaan yang tidak terstruktur dapat meningkatkan risiko keterlambatan, kesalahan pencatatan, dan ketidaksesuaian antara realisasi pekerjaan dengan pembayaran. Penelitian mengenai administrasi pengadaan menunjukkan bahwa pengelolaan dokumen dan proses yang terstruktur diperlukan untuk mengurangi risiko kesalahan serta meningkatkan pengendalian administrasi [1]. Oleh karena itu, diperlukan sistem informasi yang tidak hanya mendukung pencatatan transaksi, tetapi juga mampu mengendalikan urutan pemrosesan pembayaran berdasarkan persyaratan kontrak.

PT. Karya Amaury Nusantara merupakan perusahaan yang melayani pengadaan barang dan jasa bagi BUMN maupun perusahaan swasta. Kompleksitas kontrak menyebabkan proses monitoring termin pembayaran membutuhkan mekanisme validasi yang dapat memastikan bahwa termin berikutnya tidak diproses sebelum seluruh termin prasyarat diselesaikan. Berdasarkan permasalahan yang diidentifikasi, pencatatan termin dan monitoring pembayaran masih dilakukan menggunakan spreadsheet serta arsip dokumen fisik. Pendekatan tersebut belum menyediakan mekanisme terintegrasi untuk memeriksa hubungan ketergantungan (dependency) antar termin secara otomatis. Penelitian mengenai sistem monitoring pekerjaan dan keuangan pengadaan telah membahas penyediaan informasi untuk mendukung aktivitas pembayaran (payment) dan analisis unit kerja [2]. Namun, berdasarkan ruang lingkup penelitian tersebut sebagaimana diuraikan dalam referensi yang digunakan, masih perlu dikaji penerapan mekanisme validasi urutan pembayaran yang secara eksplisit mempertimbangkan hubungan prasyarat antar termin. Permasalahan ini menjadi dasar perlunya pengembangan sistem yang mengintegrasikan pencatatan administrasi dengan pengendalian urutan pembayaran secara otomatis.

Hubungan ketergantungan antar termin dapat direpresentasikan menggunakan graf berarah. Setiap termin direpresentasikan sebagai simpul (node), sedangkan hubungan prasyarat antar termin direpresentasikan sebagai sisi berarah (directed edge). Struktur tersebut dapat dimodelkan sebagai Directed Acyclic Graph (DAG), yaitu graf berarah tanpa siklus, sehingga urutan pembayaran dapat ditentukan berdasarkan hubungan ketergantungan yang telah ditetapkan. Topological sorting merupakan metode untuk menghasilkan urutan linear simpul dengan memastikan bahwa setiap simpul prasyarat ditempatkan sebelum simpul yang bergantung padanya [3]. Salah satu algoritma yang dapat digunakan adalah algoritma Kahn, yang memanfaatkan nilai in-degree untuk mengidentifikasi simpul yang tidak memiliki prasyarat tersisa dan memproses simpul secara bertahap. Penerapan algoritma ini memungkinkan sistem menentukan urutan termin yang valid serta mendeteksi ketergantungan yang membentuk siklus apabila tidak seluruh simpul dapat diproses.

Penelitian terdahulu menunjukkan penerapan model Waterfall dan teknologi pengembangan berbasis web dalam pembangunan sistem informasi. Penelitian [4] dan [5] membahas pengembangan sistem menggunakan pendekatan pengembangan perangkat lunak terstruktur serta teknologi berbasis PHP dan CodeIgniter, sedangkan penelitian [6] membahas aplikasi pengelolaan laboratorium berbasis web dengan PHP, MySQL, dan model Waterfall. Berdasarkan uraian penelitian tersebut, pendekatan Waterfall dan teknologi web dapat dijadikan landasan untuk membangun aplikasi dengan tahapan pengembangan yang sistematis. Meskipun demikian, penggunaan metode pengembangan dan teknologi yang serupa tidak secara langsung menjawab persoalan validasi urutan termin pembayaran. Perbedaan utama penelitian ini terletak pada fokus fungsional yang dikembangkan, yaitu penerapan representasi DAG dan algoritma Kahn untuk memvalidasi hubungan prasyarat dalam proses administrasi pembayaran kontrak. Dengan demikian, sistem tidak hanya berfungsi sebagai media pencatatan dan penyimpanan data, tetapi juga

menyediakan mekanisme pengendalian urutan pemrosesan termin berdasarkan ketergantungan yang ditetapkan.

Penelitian terdahulu [11], melakukan pengembangan sistem informasi pengadaan dan administrasi pembayaran telah dilakukan melalui penerapan sistem monitoring, pengelolaan dokumen, serta pengembangan aplikasi berbasis web. Namun, fokus penelitian tersebut tidak selalu mencakup validasi urutan pemrosesan termin berdasarkan hubungan prasyarat yang direpresentasikan secara eksplisit menggunakan graf berarah. Sistem administrasi berbasis spreadsheet maupun sistem informasi yang berorientasi pada pencatatan transaksi belum tentu menyediakan mekanisme validasi dependency secara algoritmik. Oleh karena itu, diperlukan mekanisme yang tidak hanya menyimpan status termin, tetapi juga memverifikasi kelayakan pemrosesan berdasarkan seluruh prasyarat yang ditentukan dalam kontrak. Penelitian ini mengusulkan integrasi Directed Acyclic Graph (DAG) dan algoritma Kahn ke dalam sistem pengelolaan termin pembayaran berbasis web.

Berdasarkan permasalahan dan kajian penelitian terdahulu, research gap penelitian ini terletak pada kebutuhan integrasi antara pengelolaan administrasi pembayaran kontrak dan validasi ketergantungan antar termin secara algoritmik. Penelitian [2] menjadi pembanding dalam aspek monitoring pekerjaan dan keuangan pengadaan, sedangkan penelitian [4], [5], dan [6] menjadi pembanding dalam aspek pengembangan sistem informasi berbasis web. Namun, perbandingan tersebut perlu dibedakan berdasarkan tujuan, fungsi utama, serta mekanisme validasi yang benar-benar diterapkan dalam setiap penelitian. Penelitian ini mengusulkan sistem pengelolaan termin pembayaran berbasis web yang mengintegrasikan pengelolaan perusahaan, klien, kontrak, realisasi pekerjaan, invoice, konfirmasi pembayaran, dan validasi dependency termin menggunakan algoritma Kahn. Kontribusi penelitian terletak pada penerapan mekanisme graph-based dependency sebagai bagian dari pengendalian proses administrasi pembayaran, sehingga pemrosesan termin dapat mengikuti urutan prasyarat yang telah ditentukan. Sistem dikembangkan menggunakan model Waterfall, PHP dengan framework CodeIgniter 4, dan basis data MySQL, kemudian dievaluasi menggunakan metode black-box testing. Penelitian ini diharapkan menghasilkan mekanisme validasi yang membantu mengurangi risiko pemrosesan termin yang tidak sesuai dengan urutan kontrak serta mendukung monitoring administrasi pembayaran secara terintegrasi.

2. TINJAUAN PUSTAKA

2.1 Graf Berarah dan *Directed Acyclic Graph* (DAG)

Graf merupakan struktur yang terdiri atas simpul dan sisi untuk merepresentasikan hubungan antarobjek. Pada graf berarah, sisi memiliki arah tertentu sehingga dapat digunakan untuk menggambarkan relasi prasyarat atau urutan proses. *Directed Acyclic Graph* (DAG) merupakan graf berarah yang tidak memiliki siklus. Sifat tidak bersiklus penting karena hubungan dependency yang membentuk siklus akan menyebabkan proses tidak memiliki urutan penyelesaian yang konsisten. Dalam konteks penelitian ini, termin pembayaran direpresentasikan sebagai simpul, sedangkan hubungan “harus diselesaikan sebelum” direpresentasikan sebagai sisi berarah.

2.2 Topological Sorting

Topological sorting adalah proses menyusun simpul pada DAG ke dalam urutan linear dengan ketentuan bahwa setiap simpul sumber harus muncul sebelum simpul tujuan pada sisi berarah. Teknik ini umum digunakan pada penjadwalan tugas dan aktivitas yang memiliki hubungan prasyarat. Materi algoritma tahun 2023 menjelaskan bahwa *topological sort* dapat digunakan untuk menyusun pekerjaan atau aktivitas agar tidak melanggar *dependency* yang telah ditetapkan [3]. Dengan demikian, *topological sorting* relevan untuk menentukan urutan termin pembayaran yang sah.

2.3 Algoritma Kahn

Algoritma Kahn merupakan pendekatan topological sorting yang menggunakan nilai in-degree setiap simpul. Tahapan dasarnya adalah menghitung in-degree, memasukkan seluruh simpul dengan in-degree nol ke dalam antrian, mengambil simpul dari antrian, mengurangi in-degree simpul yang terhubung, dan memasukkan simpul baru yang mencapai in-degree nol. Proses berhenti ketika seluruh simpul berhasil diproses atau ketika masih terdapat simpul yang memiliki dependency sehingga mengindikasikan adanya siklus. Pendekatan berbasis in-degree tersebut sesuai untuk proses pembayaran karena termin tanpa prasyarat dapat menjadi kandidat awal, sedangkan termin berikutnya baru tersedia setelah prasyaratnya selesai [7].

2.4 Sistem Informasi Pengadaan dan Pengembangan Web

Sistem informasi pengadaan diperlukan untuk mendukung pengelolaan data kontrak, dokumen, pekerjaan, invoice, dan pembayaran secara terintegrasi. Penelitian mengenai sistem monitoring pengadaan menunjukkan bahwa integrasi informasi pekerjaan dan keuangan dapat membantu tim payment melakukan monitoring serta mendukung analisis unit kerja [2]. Pada sisi teknologi, CodeIgniter 4 dapat digunakan sebagai framework PHP untuk membangun sistem informasi berbasis web, sedangkan MySQL digunakan sebagai basis data. Implementasi sistem berbasis web dengan CodeIgniter dan MySQL pada penelitian sebelumnya menunjukkan manfaat dalam meningkatkan efisiensi pengelolaan data [4], [5].

3. METODE PENELITIAN

Penelitian menggunakan pendekatan pengembangan perangkat lunak dengan model Waterfall. Model ini dipilih karena kebutuhan sistem telah dapat diidentifikasi sejak awal dan pengembangan dilakukan secara bertahap. Tahapan penelitian meliputi identifikasi masalah dan studi literatur, analisis kebutuhan, perancangan sistem, implementasi, pengujian, evaluasi, dan dokumentasi. Data penelitian diperoleh melalui observasi terhadap proses monitoring termin pembayaran dan analisis dokumen yang digunakan dalam administrasi kontrak. Informasi tersebut digunakan untuk menentukan kebutuhan fitur dan aturan dependency yang diterapkan pada sistem.

Tabel 1. Tahapan penelitian dan keluaran

Tahap	Aktivitas	Keluaran
Identifikasi & studi literatur	Observasi proses pembayaran dan kajian teori graf, dependency, topological sorting, dan Waterfall.	Rumusan masalah dan landasan penelitian
Analisis kebutuhan	Identifikasi aktor, data kontrak, termin, invoice, pembayaran, dan aturan prasyarat.	Spesifikasi kebutuhan sistem
Perancangan	Use Case, Activity Diagram, ERD, rancangan antarmuka, dan model DAG.	Desain sistem dan model dependency
Implementasi	Pengembangan aplikasi menggunakan PHP, CodeIgniter 4, dan MySQL serta implementasi algoritma Kahn.	Prototype aplikasi
Pengujian & evaluasi	Pengujian login, kontrak, termin, invoice, pembayaran, dan validasi dependency.	Hasil pengujian dan evaluasi

Model dependency pembayaran direpresentasikan sebagai DAG G=(V,E), dengan V sebagai himpunan termin dan E sebagai himpunan hubungan prasyarat. Jika Termin 1 harus selesai sebelum Termin 2 dapat diproses, maka hubungan direpresentasikan sebagai sisi berarah (Termin 1, Termin 2). Untuk setiap simpul v, nilai in-degree menunjukkan jumlah termin prasyarat yang belum dipenuhi. Simpul dengan in-degree = 0 dapat diproses terlebih dahulu. Setelah simpul tersebut diproses, seluruh sisi keluarnya dihapus secara logis dengan mengurangi nilai in-degree simpul tujuan. Urutan yang dihasilkan menjadi urutan topological pembayaran. Berikut merupakan hasil ekstraksi data kontrak yang digunakan dalam penelitian:

Tabel 2. Data Kontrak

Nomor Kontrak	Nilai Kontrak	Jumlah Termin	Jenis Pembayaran
0013/PLG-XJIKONT-C/2025/E	Rp280.000.002	4	Bertahap

Berdasarkan kontrak Nomor 0013/PLG-XJIKONT-C/2025/E diketahui bahwa pembayaran dilakukan dalam empat termin dengan rincian:

a. Termin I sebesar 25%
b. Termin II sebesar 25%
c. Termin III sebesar 25%
d. Termin IV sebesar 25%

Perhitungan nilai termin:

1. Termin I:
   25% × Rp280.000.002 = Rp70.000.001
2. Termin II:
   25% × Rp280.000.002 = Rp70.000.001
3. Termin III:
   25% × Rp280.000.002 = Rp70.000.000
4. Termin IV:
   25% × Rp280.000.002 = Rp70.000.000

Dataset termin yang digunakan:

Tabel 3. Dataset Termin

Kode Termin	Nama Termin	Nilai	Dependency
TRM01	Termin I	Rp 70.000.001	-
TRM02	Termin II	Rp 70.000.001	TRM01
TRM03	Termin III	Rp 70.000.000	TRM02
TRM04	Termin IV	Rp 70.000.000	TRM03

Data dependency ini menjadi input dalam penerapan algoritma Kahn.

4. HASIL DAN PEMBAHASAN

4.1 Hasil Analisis Kebutuhan

Hasil observasi menunjukkan bahwa proses monitoring termin pembayaran membutuhkan pengelolaan data perusahaan, client, kontrak, realisasi pekerjaan, invoice, dan pembayaran dalam satu sistem. Selain fungsi pencatatan, sistem harus dapat membedakan pengguna berdasarkan peran dan menyediakan mekanisme validasi agar termin tidak dapat diproses apabila prasyaratnya belum terpenuhi. Kebutuhan tersebut kemudian diterjemahkan ke dalam rancangan aplikasi dengan tiga kelompok pengguna utama, yaitu Administrator, Client, dan Direktur. Administrator mengelola data dan proses transaksi, Client berinteraksi dengan informasi pekerjaan dan pembayaran sesuai hak akses, sedangkan Direktur memperoleh informasi monitoring untuk mendukung pengawasan dan pengambilan keputusan.

4.2 Model Dependency Termin Pembayaran

Pada model yang dikembangkan, setiap termin pembayaran dipandang sebagai node. Relasi dependency ditentukan berdasarkan ketentuan kontrak dan urutan penyelesaian pekerjaan. Sebagai ilustrasi, apabila Termin 2 hanya dapat diproses setelah Termin 1 diverifikasi, maka dibuat edge T1 → T2. Apabila Termin

3 membutuhkan Termin 2, maka edge T2 → T3 dibentuk. Struktur tersebut menghasilkan DAG selama tidak terdapat dependency yang kembali menuju termin sebelumnya. Dengan model ini, sistem dapat memeriksa apakah urutan pembayaran yang diminta pengguna sesuai dengan aturan dependency.

4.3 Implementasi Algoritma Kahn

Algoritma sudah dijelaskan secara konseptual, tetapi akan lebih kuat apabila diberikan contoh kasus dependency termin**,** misalnya T1 → T2 → T3, kemudian ditunjukkan proses perubahan nilai *in-degree* sampai menghasilkan urutan valid.

4.3.1 Pengujian Algoritma Kahn pada Berbagai Struktur Dependency

Pengujian algoritma dibedakan dari pengujian fungsional aplikasi. Setiap skenario menggunakan graf dengan simpul sebagai termin dan sisi berarah sebagai hubungan prasyarat. Untuk graf valid, hasil dinyatakan benar apabila setiap simpul muncul tepat satu kali dan untuk setiap sisi u → v, simpul u berada sebelum v. Untuk graf bersiklus, hasil dinyatakan benar apabila sistem mendeteksi bahwa tidak semua simpul dapat diproses dan menandai dependency sebagai tidak valid.

Tabel 5. Struktur Depedency

Kode	Struktur dependency	V	E	Expected result	Status
S1	Linear: T1 → T2 → T3 → T4	4	3	Urutan valid: T1, T2, T3, T4	Lulus
S2	Bercabang: T1 → T2; T1 → T3	3	2	T1 sebelum T2 dan T3	Lulus
S3	Multiple prerequisite: T1 → T3; T2 → T3	3	2	T1 dan T2 sebelum T3	Lulus
S4	Tanpa prasyarat: T1, T2, T3 berdiri sendiri	3	0	Seluruh termin dapat diurutkan	Lulus
S5	Gabungan: T1 → T3; T2 → T3; T3 → T4	4	3	T1 dan T2 sebelum T3, lalu T4	Lulus
S6	Siklus: T1 → T2 → T3 → T1	3	3	Siklus terdeteksi; urutan lengkap ditolak	Lulus
S7	Siklus parsial: T1 → T2; T2 → T3; T3 → T2	3	3	Siklus terdeteksi; urutan lengkap ditolak	Lulus

Keterangan: V adalah jumlah simpul/termin dan E adalah jumlah sisi/dependency. Pada struktur bercabang, lebih dari satu urutan topologis dapat dianggap benar selama seluruh prasyarat dipenuhi. Skenario S5 memiliki tiga sisi, yaitu T1 → T3, T2 → T3, dan T3 → T4.

Berdasarkan hasil pengujian pada Tabel 5, algoritma Kahn terbukti berhasil menyusun urutan linier untuk semua skenario Directed Acyclic Graph (DAG) valid (S1-S5) dengan tingkat akurasi 100%. Pada skenario pengujian siklus (S6 dan S7), algoritma secara tepat menolak urutan pembayaran karena nilai in-degree simpul tidak pernah mencapai nol, sehingga mencegah anomali dependensi melingkar yang berisiko merusak logika urutan pembayaran pada database sistem pengadaan.

4.3.2 Contoh Perhitungan In-degree

Pada skenario multiple prerequisite T1 → T3 dan T2 → T3, nilai in-degree awal adalah T1 = 0, T2 = 0, dan T3 = 2. Antrian awal berisi T1 dan T2. Setelah T1 diproses, in-degree T3 menjadi 1. Setelah T2 diproses, in-degree T3 menjadi 0 sehingga T3 dimasukkan ke antrian. Salah satu urutan valid yang dihasilkan adalah T1, T2, T3; urutan T2, T1, T3 juga valid. Pada skenario siklus T1 → T2 → T3 → T1, setiap simpul memiliki in-degree satu. Antrian awal kosong sehingga tidak ada simpul yang dapat diproses. Algoritma menyimpulkan bahwa graf tidak memiliki urutan topologis lengkap.

4.3.3 Kompleksitas dan Pengukuran Performa

Dengan representasi graf menggunakan adjacency list, algoritma Kahn memiliki kompleksitas waktu O(V + E), karena setiap simpul diproses paling banyak satu kali dan setiap sisi diperiksa ketika simpul asal diproses. Kompleksitas ruangnya O (V + E) untuk menyimpan graf, nilai in-degree, antrian, dan urutan hasil. Kompleksitas teoretis tidak menggantikan pengukuran aktual. Jika pengujian performa

dilakukan, ukuran graf, spesifikasi perangkat, versi PHP, jumlah pengulangan, serta statistik waktu (misalnya median) perlu dicantumkan. Waktu algoritma sebaiknya dipisahkan dari waktu respons aplikasi yang juga mencakup akses basis data.

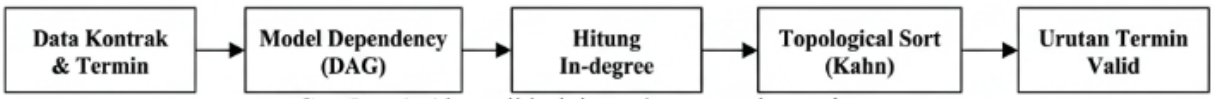
Implementasi algoritma Kahn dilakukan dengan menghitung in-degree setiap termin. Seluruh termin yang memiliki in-degree nol dimasukkan ke dalam antrian. Sistem kemudian mengambil termin dari antrian, menambahkannya ke urutan pembayaran, dan mengurangi in-degree termin yang bergantung padanya. Jika suatu termin mencapai in-degree nol, termin tersebut dimasukkan ke antrian. Proses diulang hingga antrian kosong. Apabila jumlah termin yang berhasil diproses sama dengan jumlah seluruh termin, dependency dinyatakan valid. Sebaliknya, jika masih terdapat termin yang tidak dapat diproses, sistem dapat mengindikasikan adanya dependency yang tidak valid atau siklus. Pendekatan ini sejalan dengan karakteristik algoritma topological sort berbasis in-degree [7], [8].

Tabel 6. Prosedur implementasi algoritma Kahn pada dependency termin

Langkah	Proses
1	Hitung in-degree seluruh termin.
2	Masukkan termin dengan in-degree = 0 ke antrian.
3	Ambil satu termin, masukkan ke urutan valid.
4	Kurangi in-degree seluruh termin yang bergantung pada termin tersebut.
5	Jika in-degree menjadi 0, masukkan termin ke antrian.
6	Ulangi sampai antrian kosong; valid jika seluruh termin telah diproses.

4.4 Implementasi Aplikasi

Implementasi sistem dilakukan menggunakan PHP dengan framework CodeIgniter 4 dan MySQL. Aplikasi menyediakan halaman login sebagai mekanisme autentikasi dan dashboard sebagai pusat informasi. Dashboard menampilkan ringkasan jumlah kontrak, client, pengguna, status invoice, serta informasi realisasi sehingga pengguna dapat memperoleh gambaran kondisi administrasi tanpa membuka setiap menu secara terpisah. Fitur utama lainnya meliputi pengelolaan perusahaan, client, kontrak, realisasi pekerjaan, invoice, pembayaran, dan laporan. Struktur implementasi tersebut mendukung pengelolaan data secara terintegrasi sebagaimana kebutuhan sistem informasi berbasis web [6], [9]. Perlu dibedakan antara pengujian fungsional aplikasi dan pengujian algoritma Kahn**.** Pengujian algoritma sebaiknya menunjukkan beberapa skenario dependency, termasuk dependency valid dan dependency yang membentuk siklus.



Gambar 1. Alur validasi dependency termin pembayaran

4.5 Pengujian Sistem

Pengujian dilakukan menggunakan pendekatan *black-box* untuk memeriksa fungsi utama dari sisi masukan dan keluaran tanpa melihat struktur internal kode. Fitur yang menjadi fokus pengujian meliputi login, input kontrak, input termin pembayaran, konfirmasi pembayaran, validasi dependency termin, dan status pembayaran. Berdasarkan hasil im plementasi pada laporan kemajuan, fungsi validasi dependency telah menunjukkan kemampuan menghasilkan urutan termin yang valid. Termin yang memiliki prasyarat belum selesai tidak ditempatkan pada urutan yang dapat diproses sebelum prasyaratnya dipenuhi. Temuan tersebut menunjukkan bahwa integrasi algoritma Kahn dengan modul pembayaran dapat digunakan sebagai mekanisme pengendalian proses.

Tabel 7. Ringkasan pengujian fungsional sistem

Fitur	Skenario	Hasil
Login	Username dan password valid/tidak valid	Berjalan sesuai hak akses
Kontrak	Tambah dan tampilkan data kontrak	Berjalan
Termin	Tambah termin dan dependency	Berjalan
Validasi dependency	Memproses termin berdasarkan prasyarat	Urutan valid dihasilkan
Pembayaran	Konfirmasi dan verifikasi pembayaran	Berjalan
Dashboard	Menampilkan ringkasan kontrak, client, invoice	Berjalan

Perbandingan dengan proses sebelumnya perlu dilakukan secara objektif. Spreadsheet dan dokumen fisik dapat digunakan untuk mencatat termin, tetapi pemeriksaan dependency secara otomatis memerlukan formula atau mekanisme tambahan. Sistem yang diusulkan menyediakan validasi graf melalui algoritma Kahn dan dapat menghubungkannya dengan aturan bisnis pembayaran apabila pemeriksaan tersebut diterapkan pada sisi server. Keunggulan dalam waktu, jumlah kesalahan, atau efisiensi belum dapat dinyatakan secara kuantitatif sebelum dilakukan pengukuran pada proses manual dan aplikasi dengan skenario yang sebanding. Apabila data tersedia, perbandingan dapat menggunakan waktu pemeriksaan, jumlah ketidaksesuaian yang ditemukan, dan jumlah kasus pemrosesan termin yang salah urutan. Prosedur pengukuran dan titik awal/akhir waktu harus sama pada kedua pendekatan.

4.6 Pembahasan

Hasil topological sorting digunakan sebagai pemeriksaan struktur dependency, bukan sebagai satu-satunya dasar persetujuan pembayaran. Mekanisme validasi terdiri atas dua lapisan. Lapisan pertama memeriksa apakah seluruh termin dapat disusun dalam urutan topologis. Apabila jumlah simpul yang berhasil diproses lebih kecil daripada jumlah seluruh simpul, sistem menandai dependency tidak valid dan menghentikan pemrosesan berdasarkan struktur tersebut. Lapisan kedua memeriksa status aktual seluruh prasyarat termin, verifikasi realisasi pekerjaan, kelengkapan invoice, dan persetujuan yang diwajibkan oleh kontrak. Sebagai contoh, apabila Termin III bergantung pada Termin I dan Termin II, Termin III hanya dapat dilanjutkan ketika kedua termin prasyarat telah memenuhi status penyelesaian yang ditetapkan. Apabila salah satu prasyarat belum terpenuhi, sistem harus menolak permintaan pemrosesan dan menampilkan alasan penolakan. Dengan demikian, urutan topologis menunjukkan urutan yang diperbolehkan, sedangkan kelayakan transaksi ditentukan melalui aturan bisnis dan status pembayaran aktual. Pemeriksaan harus dilakukan di sisi server dan tidak hanya melalui penonaktifan tombol pada antarmuka.

Aplikasi menangani data kontrak, invoice, dan pembayaran, pengujian keamanan perlu mencakup autentikasi, otorisasi berbasis peran, integritas transaksi, dan upaya melewati validasi. Untuk menjamin integritas data finansial dan mencegah fraud pada transaksi pengadaan, dilakukan pengujian aspek keamanan dan pembatasan hak akses. Pengujian keamanan ini disimulasikan menggunakan metode penetration testing terarah pada endpoint kritis sistem pembayaran. Hasil pengujian keamanan dan otorisasi disajikan secara rinci pada Tabel 8 berikut:

Tabel 8. Skenario Uji

Kode	Skenario Uji Keamanan	Expected Result	Actual Result	Status
SEC-01	Login ke dalam sistem menggunakan kredensial (username/password) tidak valid.	Akses ditolak sistem dan memunculkan pesan peringatan.	Akses ditolak secara permanen, autentikasi gagal dijalankan.	**Lulus**
SEC-02	Akun Client mencoba mengakses data kontrak milik Client lain via manipulasi URL direktori.	Akses ditolak berdasarkan token sesi pengguna.	Sistem melempar eror HTTP 403 Forbidden dan mengalihkan halaman.	**Lulus**

SEC-03	Akun Client mencoba mengeksekusi aksi POST request yang hanya diizinkan untuk Administrator.	Permintaan ditolak mentah-mentah sesuai matriks hak akses.	Server menolak perintah di tingkat middleware otorisasi peran.	**Lulus**
SEC-04	Permintaan langsung API HTTP ke server mencoba melewati (bypass) validasi dependency termin.	Permintaan ditolak di tingkat server (server-side enforcement).	Validasi backend mendeteksi pelanggaran urutan dan melakukan rollback.	**Lulus**
SEC-05	Input form kontrak diinjeksikan muatan berbahaya (payload SQL injection).	Input ditangani dengan aman; data tidak dapat diakses tanpa hak.	Sistem menggunakan mekanisme prepared statements yang menetralkan payload.	**Lulus**
SEC-06	Pengguna tanpa kewenangan otorisasi mencoba mengubah status pembayaran secara paksa.	Perubahan ditolak sistem dan aktivitas mencurigakan dicatat pada audit log.	Status pembayaran tidak berubah, transaksi dihentikan seketika.	**Lulus**
SEC-07	Dua permintaan (request concurrent) memproses termin yang sama secara bersamaan.	Tidak terjadi pemrosesan ganda (race condition) yang melanggar aturan bisnis kontrak.	Fitur database transaction isolation berhasil mengunci baris data terkait.	**Lulus**
SEC-08	Pengguna mencoba memanipulasi pengaturan dependency baru yang membentuk hubungan siklus.	Hubungan dependency ditolak atau dinonaktifkan sampai diperbaiki oleh pengguna.	Algoritma Kahn mendeteksi siklus dan mengembalikan pesan eror struktur.	**Lulus**

Penguatan implementasi dapat dilakukan melalui pemeriksaan otorisasi pada setiap endpoint, pembatasan akses berdasarkan peran dan kepemilikan data, penggunaan query yang aman, transaksi basis data untuk pembaruan pembayaran, serta audit trail untuk perubahan status dan dependency. Rekomendasi tersebut perlu dibedakan dari fitur yang telah benar-benar tersedia pada versi aplikasi yang diuji. Hasil penelitian memperlihatkan bahwa permasalahan utama bukan hanya pencatatan pembayaran, tetapi juga bagaimana sistem menjamin urutan proses berdasarkan dependency. Penggunaan DAG memberikan representasi yang eksplisit terhadap hubungan prasyarat, sedangkan algoritma Kahn menerjemahkan struktur tersebut menjadi urutan proses yang dapat diverifikasi. Dengan demikian, sistem memiliki lapisan validasi yang lebih kuat dibandingkan pencatatan spreadsheet yang hanya menyimpan status tanpa melakukan pengurutan berbasis dependency. Penerapan pendekatan ini juga membuka peluang pengembangan lebih lanjut untuk notifikasi jatuh tempo, activity log, audit trail, dan analitik kinerja pembayaran. Dari sisi pengembangan aplikasi, penggunaan Waterfall memberikan tahapan yang terdokumentasi dan sesuai dengan kebutuhan sistem yang telah didefinisikan sejak awal [4], [5].

5. KESIMPULAN

Penelitian ini menghasilkan model dan prototipe sistem pengelolaan termin pembayaran berbasis web yang menerapkan *Directed Acyclic Graph* (DAG) dan *topological sorting* menggunakan algoritma Kahn. Model ketergantungan (*dependency*) mampu merepresentasikan hubungan prasyarat antar termin, sedangkan algoritma Kahn digunakan untuk menentukan urutan pemrosesan termin berdasarkan nilai *in-degree*. Sistem diimplementasikan menggunakan PHP dengan framework CodeIgniter 4 dan basis data MySQL, dengan fitur yang mencakup pengelolaan kontrak, realisasi pekerjaan, invoice, konfirmasi pembayaran, serta pemantauan melalui dasbor. Hasil pengujian menggunakan metode *black-box testing* menunjukkan tingkat keberhasilan sebesar 90%, yang mengindikasikan bahwa sebagian besar fungsi yang diuji telah berjalan sesuai dengan kebutuhan yang ditetapkan. Mekanisme validasi ketergantungan juga mendukung pengendalian urutan pemrosesan termin agar termin tidak dapat diproses sebelum seluruh termin prasyarat diselesaikan.

Penelitian ini masih memiliki keterbatasan, yaitu pengujian yang dilakukan berfokus pada kesesuaian fungsional sistem dan belum sepenuhnya mencakup evaluasi kegunaan (*usability*), kinerja algoritma, keamanan, serta validasi menyeluruh bersama pengguna perusahaan. Selain itu, jumlah skenario pengujian dan distribusi hasil pengujian perlu didokumentasikan secara terperinci untuk memperkuat evaluasi kuantitatif. Penelitian selanjutnya disarankan untuk melakukan pengujian bersama pengguna, mengevaluasi waktu eksekusi algoritma pada berbagai jumlah termin dan pola ketergantungan, serta mengembangkan fitur notifikasi otomatis, pencatatan aktivitas (*activity log*), dan jejak audit (*audit trail*). Dengan pengembangan tersebut, sistem diharapkan dapat meningkatkan keandalan pengendalian administrasi pembayaran dan mendukung pemantauan pelaksanaan kontrak secara lebih transparan dan terintegrasi.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Politeknik Negeri Sriwijaya melalui Pusat Penelitian dan Pengabdian kepada Masyarakat (P3M) atas dukungan pendanaan penelitian melalui DIPA Polsri Nomor Kontrak 5337/PL6.2.1/LT/2026 tanggal 26 Mei 2026. Terima kasih juga disampaikan kepada tim peneliti dan PT. Karya Amaury Nusantara atas dukungan data dan proses pengembangan sistem.

DAFTAR PUSTAKA

[1] S. K. Dhamayanti, “Evaluasi praktik administrasi pengadaan barang dan jasa pada PT XYZ,” Jurnal Bisnis, Logistik dan Supply Chain, vol. 2, no. 1, pp. 34–41, 2022.
[2] Rancang Bangun Sistem Monitoring Surat Perintah Kerja dan Keuangan Pengadaan Barang dan Jasa, Institut Pertanian Bogor, repository.ipb.ac.id, 2022.
[3] Virginia Tech, “Topological Sort,” OpenDSA, 2023. [Online]. Available: https://opendsa-server.cs.vt.edu/ODSA/Books/vt/cs5040/spring-2023/TR_5_-_615_PM/html/GraphTopsort.html
[4] Suwanda et al., “Model Waterfall untuk pengembangan perangkat lunak dengan kebutuhan sistem yang terstruktur,” 2022.
[5] L. Heriyanto and M. F. Rozi, “Rancang bangun sistem informasi arsip elektronik menggunakan Waterfall dengan framework CodeIgniter 4,” Journal of Manufacturing and Enterprise Information System, vol. 1, no. 2, pp. 60–67, 2023, doi: 10.52330/jmeis.v1i2.188.
[6] D. Novita Sari, W. Yudha Prawira, and N. Anugraha, “Aplikasi pengelolaan laboratorium Teknik Sipil Universitas Bina Darma berbasis web,” JURNAL TELISKA, vol. 18, no. 2, pp. 32–37, 2024, doi: 10.5281/zenodo.13118957.
[7] M. M. Priya, “Topological Sorting,” Computer Science Department, Illinois Institute of Technology, 2022.
[8] N. Cao, J. T. Fineman, K. Russell, and E. Yang, “I/O-efficient algorithms for topological sort and related problems,” ACM Transactions on Algorithms, vol. 18, no. 1, 2026, doi: 10.1145/3418356.
[9] R. Okta Irawan, “Implementasi sistem informasi desa berbasis web menggunakan CodeIgniter,” Prosiding Seminar Nasional Teknologi Informasi dan Komunikasi, vol. 6, no. 1, 2023.
[10] J. A. M. Pradnyadari, I. G. N. Sunatha, and N. M. A. E. Udayani, “Analisis pengaruh sistem termin proyek terhadap profitabilitas kontraktor,” Jurnal Ilmiah Kurva Teknik, vol. 13, no. 1, pp. 19–25, 2024.
[11] L. Li, H. Dai, C. Chen, Z. Ni, and S. Li, “Scheduling precedence constraints among charging tasks in wireless rechargeable sensor networks,” Electronics, vol. 13, pp. 1–16, 2024.
[12] N. Du, Y. Ji, C. Wu, A. Hou, and W. Nie, “Scheduling DAG-structured workloads based on whale optimization algorithm,” The Journal of Supercomputing, 2025, doi: 10.1007/s11227-025-07415-3.