E-ISSN : 2988-1986
https://ejournal.cahayailmubangsa.institute/index.php/kohesi

Kohesi: Jurnal Multidisiplin Saintek
Volume 10 No 7 Tahun 2025

PENTINGNYA SERIALIZABILITY DALAM MANAJEMEN TRANSAKSI: TINJAUAN MEKANISME KONTROL KONKURENSI UNTUK MENJAMIN KONSISTENSI DATA

Ridha Adjie Eriyadi[1] , Rahma Aulia Dini[2] , Widya Novianti[3] , Sabili Putra Kirana[4], Ihsan Kamil[5]
[1-5] Informatika, Universitas Teknologi Digital, Bandung, Indonesia
Email: [1]ridhaadjie@digitechuniversity.ac.id, [2]rahma20124033@digitechuniversity.ac.id, [3]noviantiwidya16@gmail.com, [4]sabiliputra6@gmail.com , [5]ihsankamil1357@gmail.com

ABSTRAK

Perkembangan sistem basis data modern menuntut kemampuan untuk menangani transaksi secara konkuren tanpa mengorbankan konsistensi dan keutuhan data. Eksekusi transaksi paralel berpotensi menimbulkan berbagai anomali, seperti lost update, dirty read, dan write skew, apabila tidak dikelola dengan mekanisme kontrol yang tepat. Serializability merupakan prinsip fundamental dalam manajemen transaksi yang menjamin bahwa hasil eksekusi transaksi konkuren setara dengan eksekusi secara serial. Penelitian ini bertujuan untuk mengkaji pentingnya serializability dalam manajemen transaksi basis data serta meninjau berbagai mekanisme kontrol konkurensi yang digunakan untuk menjamin konsistensi data. Metode penelitian yang digunakan adalah studi literatur sistematis dengan menganalisis jurnal nasional dan internasional yang relevan. Hasil kajian menunjukkan bahwa meskipun banyak sistem basis data modern mengadopsi tingkat isolasi yang lebih rendah demi meningkatkan kinerja, pendekatan tersebut masih menyisakan risiko terjadinya anomali transaksi. Berbagai mekanisme kontrol konkurensi, baik tradisional seperti Two-Phase Locking maupun pendekatan modern seperti Snapshot Isolation, Serializable Snapshot Isolation, dan metode hybrid, dikembangkan untuk menyeimbangkan antara konsistensi data dan efisiensi sistem. Kesimpulan dari penelitian ini menegaskan bahwa serializability tetap menjadi fondasi utama dalam menjaga integritas data, dan penerapannya perlu disesuaikan dengan karakteristik sistem serta kebutuhan kinerja pada lingkungan basis data modern.

Kata Kunci: Serializability, Manajemen Transaksi, Kontrol Konkurensi, Konsistensi Data, Basis Data

ABSTRACT

The advancement of modern database systems requires the ability to handle concurrent transactions without compromising data consistency and integrity. Parallel transaction execution may lead to various anomalies, such as lost updates, dirty reads, and write skew, if not properly managed through appropriate concurrency control mechanisms. Serializability is a fundamental principle in transaction management that ensures the outcome of concurrent transactions is equivalent to a serial execution. This study aims to examine the importance of serializability in database transaction management and to

Article History
Received: November 2025
Reviewed:Desember 2025
Published:Desember 2025

Plagiarism Checker No 235
Prefix DOI :
10.8734/Kohesi.v1i2.365
Copyright : Author
Publish by : Kohesi

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License

review various concurrency control mechanisms used to guarantee data consistency. The research method employed is a systematic literature review by analyzing relevant national and international journals. The results indicate that although many modern database systems adopt lower isolation levels to improve performance, such approaches still expose systems to transaction anomalies. Consequently, various concurrency control mechanisms, ranging from traditional methods such as Two-Phase Locking to modern approaches like Snapshot Isolation, Serializable Snapshot Isolation, and hybrid techniques, have been developed to balance data consistency and system efficiency. This study concludes that serializability remains a crucial foundation for maintaining data integrity, and its implementation must be adapted to system characteristics and performance requirements in modern database environments.
Keywords: *Serializability, Transaction Management, Concurrency Control, Data Consistency, Database Systems*

PENDAHULUAN

Kemajuan teknologi basis data saat ini menuntut sistem mampu menangani sejumlah transaksi secara bersamaan tanpa mengorbankan keakuratan dan keutuhan data. Dalam sistem yang digunakan oleh banyak pengguna secara serempak, transaksi yang berjalan paralel dapat memicu konflik, anomali data, hingga inkonsistensi pada basis data apabila tidak dikelola dengan baik [1], [2]. Oleh sebab itu, diperlukan mekanisme pengaturan transaksi yang dapat menjamin bahwa eksekusi transaksi konkuren tetap menghasilkan keadaan akhir yang konsisten sebagaimana jika transaksi tersebut diproses secara berurutan.

Serializability menjadi salah satu prinsip fundamental dalam menjaga konsistensi transaksi, karena menjamin bahwa hasil akhir dari transaksi paralel tetap serupa dengan hasil eksekusi serial [3]. Tanpa penerapan serializability, dapat terjadi beberapa masalah seperti hilangnya pembaruan (lost updates), pembacaan data sementara (dirty read), hasil pembacaan yang tidak konsisten, serta data yang tidak dapat direproduksi kembali pada transaksi selanjutnya [4], [5]. Kondisi tersebut tentu berkaitan erat dengan risiko menurunnya integritas informasi pada basis data.

Untuk memastikan serializability, sejumlah metode kontrol konkurensi telah dikembangkan, antara lain teknik penguncian dua fase atau Two-Phase Locking (2PL), Timestamp Ordering, dan Optimistic Concurrency Control (OCC) [6], [7]. Masing-masing metode mempunyai karakteristik berbeda dalam menangani konflik transaksi, mengatasi deadlock, serta menjaga performa sistem terutama pada skala yang lebih besar [8]. Beberapa penelitian menegaskan bahwa pemilihan metode kontrol konkurensi yang sesuai dapat meningkatkan kinerja manajemen transaksi sekaligus mempertahankan tingkat konsistensi data sistem [9], [10].

Dengan melihat tantangan dalam pengelolaan transaksi paralel, penelitian ini mengkaji secara mendalam mekanisme-mekanisme kontrol konkurensi dalam basis data sekaligus menyoroti urgensi serializability sebagai fondasi utama dalam manajemen transaksi masa kini. Kajian ini diharapkan dapat memberikan kontribusi berupa pemahaman yang lebih komprehensif mengenai manfaat dan peran penting serializability dalam menjaga integritas dan akurasi data pada berbagai sistem berbasis database.

METODE PENELITIAN

Penelitian ini menggunakan desain penelitian berbasis studi literatur yang dipadukan dengan metode adopsi terhadap model, konsep, atau algoritma dari penelitian sebelumnya. desain ini dipilih karena memungkinkan peneliti memperoleh dasar teoritis yang kuat melalui pengkajian berbagai sumber ilmiah yang relevan. prosedur penelitian dimulai dengan mengumpulkan literatur dari databases seperti Google Scholar, Garuda, dan jurnal terindeks nasional maupun internasional. [7] Setiap literatur kemudian diseleksi berdasarkan kepuasan topik dan kualitas metodologinya. setelah itu, peneliti melakukan proses adopsi, yaitu mengambil model, kerangka teori, algoritma, atau prosedur yang telah digunakan dan di validasi dalam penelitian terdahulu, kemudian menyesuaikannya dengan kebutuhan penelitian saat ini. Proses adopsi dapat mencakup langkah-langkah teknis, termasuk penyusunan ulang algoritma atau pseudcode berdasarkan referensi yang digunakan, serta penyesuaian langkah kerja agar sesuai dengan konteks penelitian.

Akuisisi data dalam penelitian ini berasal dari data sekunder, yaitu informasi yang diperoleh dari literatur ilmiah yang telah dipublikasikan. Setiap sumber dianalisis untuk mengekstasi data konseptual, data numerik atau detail algoritmik yang relevan dengan tujuan penelitian. Pengujian dilakukan melalui analisis logis dan komparatif, yaitu membandingkan konsistensi model atau konsep yang diadopsi dengan teori-teori lain serta temuan penelitian sebelumnya. validasi pendekatan ini diperkuat melalui penggunaan referensi ilmiah yang kredibel, sehingga setiap tahapan metode, mulai dari desain, prosedur hingga pengujian dan analisis, dapat diterima secara akademik dan memenuhi standar penelitian ilmiah[8].

HASIL DAN PEMBAHASAN

Matriks Sistematik Literatur (*Systematic Literature Review Matrix*)

Tabel 1 menampilkan data yang diekstraksi secara komprehensif dari 13 karya literatur terpilih, yang menggambarkan konteks sistem, metode yang diterapkan, dan hasil utama dari setiap studi.

Tabel 1. Matriks Tinjauan Literatur Sistematis

No.	Studi (Penulis/Tahun)	Konteks & Platform	Fokus Masalah	Mekanisme Utama *(Methods)*	Temuan Kunci *(Key Findings)*
1.	Alif Finandhita [19]	Teori Dasar	Pemahaman fundamental manajemen transaksi dan properti ACID.	ACID, Schedule, Serializability.	Mendefinisikan standar konsistensi data yang harus diadopsi oleh sistem manajemen basis data (DBMS).
2.	Flé & Roucairol (1985) [18]	Teori Komputasi	Kompleksitas sinkronisasi pada transaksi berulang (*iterated transactions*).	*Finite Automaton*, *Regular Language*.	Membuktikan bahwa *serializability* transaksi kompleks dapat dimodelkan dan diperkirakan menggunakan model matematis.
3.	Rahayu (2007) [7]	Oracle 10g (RDBMS)	Inkonsistensi data pada implementasi sistem komersial (Oracle).	*Pessimistic Locking* + *Row Versioning*.	Locking dan versioning telah menjadi kombinasi yang efektif, tetapi fitur *Flashback* diperlukan untuk mengakses data yang sama di masa lalu.

4.	Wang et al. (2017) [10]	In-Memory Database	Risiko anomali pada isolasi rendah tanpa mekanisme *locking* berat.	*Serial Safety Net (SSN)* (Certifier).	SSN dapat digabungkan dengan RC/SI sehingga dapat saling bergantung secara berurutan dengan ketergantungan siklus yang efektif.
5.	Ren et al. (2019) [14]	Geo-Replicated DB	Latensi tinggi (waktu respons lama) pada transaksi lintas wilayah.	*SLOG* (Deterministic Ordering).	Memastikan pemanfaatan maksimal dari data locality dan determinisme untuk menghindari keterlambatan dalam *Two-Phase Commit* (2PC).
6.	Helt et al. (2021) [15]	Distributed Systems	*Overhead* kinerja akibat sinkronisasi waktu *real-time*.	*Regular Sequential Serializability (RSS)*.	RSS menawarkan jaminan invarian yang setara dengan Strict *Serializability* dan kinerja yang lebih baik karena persyaratan temporal yang lebih longgar.
7.	Burke etal. (2023) [12]	High-Contention DB	Tingginya tingkat *abort* pada OCC saat terjadi konflik tinggi.	*Morty* (Partial Re-execution).	Daripada membatalkan transaksi, lebih efektif untuk melakukan re-eksekusi parsial (perbaikan) untuk mempertahankan *throughput*.
8.	Lu et al. (2023) [8]	Cloud Datastores	Masalah *blocking* dan anomali *timestamp-inversion*.	*NCC* (*Non-blocking* + RTC).	Uji *natural arrival order*, dan mencapai latensi RTT 1 tanpa kemungkinan terjadinya anomali pembalikan waktu.
9.	Ketsman et al. (2023) [16]	Teori Database	Kelemahan isolasi terkait ketahanan aplikasi (*robustness*)	Kerangka Teoritis (*Robustness Framework*).	Menyediakan alat formal untuk menentukan kemampuan aplikasi untuk menjalankan operasi dengan aman dalam konteks *Read Committed*.
10.	Vandevoort et al. (2023) [17]	MVRC Environment	Deteksi otomatis kerentanan aplikasi terhadap anomali.	Analisis Statis (*Static Analysis*).	Algoritma berbasis graf mampu mendeteksi kemungkinan anomali dalam program transaksi sebelum eksekusi.
11.	Kalay (2025) [9]	PostgreSQL (Disk-based)	Komparasi *trade-off* performa antar level isolasi standar.	SSI vs SI vs RC Benchmark.	*Serializable Snapshot Isolation* (SSI) di PostgreSQL secepat SI saat bekerja dengan dataset besar, akan tetapi lebih aman.
12.	Hong et al. (2025) [11]	Main-Memory DB	Keterbatasan metode tunggal (hanya Calvin atau hanya OCC).	*HDCC* (Hybrid Calvin + OCC).	Strategi *hybird* adaptif dapat meningkatkan kinerja hingga 3,1 kali dengan mengalokasikan transaksi berdampak tinggi ke Calvin dan yang berdampak rendah ke OCC.

13.	Ghasemirad et al. (2025) [13]	Geo-Distributed DB	Validasi dan verifikasi formal protokol yang kompleks.	*Eiger-PORT+* (Verifikasi Formal).	Secara matematis membuktikan bahwa konsistensi kausal transaksional (TCCv) dapat dicapai tanpa mengorbankan kinerja maksimum.

Analisis Kerentanan Anomali pada Berbagai Level Isolasi

Dari analisis literatur yang diuraikan dalam Tabel 1, teridentifikasi bahwa tantangan utama dalam sistem basis data terletak pada keseimbangan antara kinerja (*throughput*) dan akurasi data. Tingkat isolasi di bawah *Serializability* secara luas digunakan dalam sebagian besar sistem komersial untuk menghindari penurunan kinerja, meskipun metode ini menimbulkan kerentanan terhadap anomali data dari segala jenis. Dalam Tabel 2, ringkasan kerentanan anomali pada tingkat isolasi yang berbeda ditampilkan.

Tabel 2. Perbandingan Anomali pada Berbagai Level Isolasi

Level Isolasi	Dirty Read	Non-Repeatable Read	Phantom Read	Serialization Anomaly (Write Skew)
Read Committed (RC)	Tidak	Ya	Ya	Ya
Snapshot Isolation	Tidak	Tidak	Tidak	Ya
Serializable Snapshot Isolation (SSI)	Tidak	Tidak	Tidak	Tidak
Strict Serializability	Tidak	Tidak	Tidak	Tidak

Sumber: *Diolah dari Kalay [9] dan Wang et al. [10]*

Tabel 2 menunjukkan temuan penting bahwa Snapshot Isolation (SI) sangat diminati dalam sistem modern karena dapat menangani pembacaan tanpa memblokirnya, namun gagal menghindari anomali Write Skew. Seperti yang dijelaskan oleh Kalay [9] dalam evaluasi kinerjanya terhadap PostgreSQL, transaksi bersamaan dengan SI memiliki kemampuan untuk memperbarui data menggunakan snapshot yang sudah usang, sehingga melanggar persyaratan integritas bisnis.

Wang et al. [10] mengusulkan solusi berupa Serial Safety Net (SSN), yaitu sertifikat yang beroperasi di atas SI/RC untuk mendeteksi siklus ketergantungan berbahaya. Hal ini membuktikan bahwa untuk mencapai konsistensi ideal tanpa kehilangan kinerja yang signifikan, mekanisme tidak boleh hanya bergantung pada isolasi standar, tetapi harus ditingkatkan dengan mekanisme lain seperti SSI atau SSN.

Tabel 2 juga menjadikan Strict Serializability sebagai standar pertama yang sepenuhnya bebas dari berbagai bentuk anomali, meskipun dalam praktiknya hal ini hanya dapat dilakukan dengan mengorbankan kinerja, seperti yang akan dijelaskan lebih lanjut dalam 3.3.

Komparasi Mekanisme Kontrol Konkurensi: Tradisional vs Modern

Tinjauan literatur menunjukkan pergeseran paradigma besar dari mekanisme penguncian pesimistis ke pendekatan optimistis dan deterministik. Perubahan ini dipicu oleh kebutuhan akan sistem terdistribusi untuk mengurangi latensi komunikasi. Tabel 3 menyajikan perbandingan fitur antara mekanisme-mekanisme tersebut.

Tabel 3. Perbandingan Karakteristik Mekanisme Kontrol Konkurensi

Fitur / Mekanisme	Two-Phase Locking (2PL)	Optimistic Concurrency Control (OCC)	Hybrid / Deterministic (HDCC/Calvin)
Paradigma	Pesimistik (Mencegah konflik di awal)	Optimistik (Validasi di akhir)	Deterministik (Terjadwal di awal)
Penanganan Konflik	Menunggu (*Blocking*) pada Lock	*Abort* dan *Restart* transaksi	Antrian eksekusi (*Ordering*)

Kelebihan	Menjamin keamanan ketat, Mencegah konflik sejak dini.	Kinerja sangat tinggi pada beban kerja ringan (*low contention*).	Menghilangkan *overhead* commit dua fase (2PC) pada sistem terdistribusi.
Kekurangan	Rentan *Deadlock*; Menurunkan *throughput* secara signifikan.	Tingkat *Abort* tinggi pada beban kerja padat (*high contention*).	Kurang fleksibel untuk transaksi interaktif, Perlu deklarasi *read/write sets* di awal.

Dari Tabel 3, metode 2PL tradisional seperti yang digunakan dalam sistem basis data lama sering menjadi hambatan utama dalam lingkungan saat ini yang memerlukan tingkat koncurrency yang tinggi. Burke et al. [12] menekankan bahwa dalam pendekatan OCC, biaya pembatalan transaksi saat terjadi konflik cukup tinggi.

Sebagai alternatif, studi-studi terbaru beralih ke pendekatan Hybrid. Hong et al. [11] melalui protokol HDCC menunjukkan bahwa menggabungkan determinisme (Calvin) untuk transaksi dengan konflik tinggi dan optimisme (OCC) untuk transaksi dengan konflik rendah dapat meningkatkan *throughput* hingga 3,1 kali lipat dibandingkan dengan pendekatan tunggal. Hal ini menegaskan bahwa kemampuan algoritma untuk beradaptasi merupakan faktor penting dalam efisiensi di masa depan.

Implementasi Serializability pada Sistem Basis Data Nyata

Teori *serializability* dan peng-aplikasiannya dalam dunia nyata berbeda-beda tergantung pada arsitektur sistem (berbasis disk, berbasis memori, atau terdistribusi secara geografis). Tabel 4 merangkum strategi implementasi dalam sistem yang diteliti.

Tabel 4. Strategi Implementasi Serializability pada Sistem Basis Data

Sistem Basis Data	Metode Utama	Strategi Jaminan Serializability	Temuan Kunci
Oracle 10g [7]	Pesimistik + MVCC	Menggunakan *Row-Level Locking* untuk update dan fitur *Flashback* untuk konsistensi baca. Menggunakan Row-Level Locking, mengunci baris yang sedang diubah dan mendukung fitur *Flashback* untuk konsistensi pembacaan.	Efektif mencegah inkonsistensi namun berpotensi *blocking* saat modifikasi data konkuren.
PostgreSQL [9]	SSI (*Serializable Snapshot Isolation*)	Mengidentifikasi *rw-antidependency* dalam graf serialisasi.	Memberikan jaminan *Serializable*dengan performa mendekati SI, namun rentan terhadap positif palsu (*false aborts*).
NCC (Protokol Baru) [8]	Non-blocking + Timestamp	Mencegahanomali *timestamp-inversion* dengan *Response Timing Control* (RTC).	Berhasil mencapai *Strict Serializability* dengan latensi minimal (1 round-trip) tanpa memerlukan *lock* global.

Tabel 4 menunjukkan evolusi dari mekanisme berbasis kunci (Oracle) ke mekanisme deteksi siklus (PostgreSQL) dan akhirnya ke protokol *non-blocking* (NCC). Inovasi NCC oleh Lu et al. [8] sangat signifikan karena berhasil membantah asumsi bahwa konsistensi ketat harus lambat. Dengan menggunakan urutan kedatangan alami dan memisahkan eksekusi dan respons, NCC menunjukkan bahwa konsistensi ketat dapat dicapai dengan latensi yang sangat rendah, hal yang menjadi perhatian kritis dalam sistem basis data awan saat ini.

KESIMPULAN

Serializability merupakan konsep utama dalam manajemen transaksi basis data yang bertujuan menjamin konsistensi dan keutuhan data ketika transaksi dijalankan secara bersamaan. Prinsip ini memastikan bahwa hasil eksekusi transaksi paralel tetap setara dengan eksekusi secara berurutan.

Hasil kajian menunjukkan bahwa banyak sistem basis data modern memilih level isolasi yang lebih rendah demi meningkatkan kinerja, namun pilihan tersebut masih menyisakan risiko terjadinya anomali transaksi. Berbagai mekanisme kontrol konkurensi, baik tradisional maupun modern, dikembangkan untuk mengatasi permasalahan ini, mulai dari pendekatan penguncian pesimistis hingga metode optimistis, deterministik, dan hybrid yang lebih adaptif terhadap kebutuhan sistem terdistribusi dan beban kerja yang dinamis.

Secara keseluruhan, penelitian ini menegaskan bahwa serializability tetap menjadi fondasi penting dalam manajemen transaksi basis data, dan pencapaiannya perlu disesuaikan dengan karakteristik sistem serta kebutuhan kinerja. Pemilihan mekanisme kontrol konkurensi yang tepat dan adaptif menjadi kunci dalam menjaga keseimbangan antara konsistensi data dan efisiensi sistem pada lingkungan basis data modern.

DAFTAR PUSTAKA

[1] A. Silberschatz, H. Korth, and S. Sudarshan, *Database System Concepts*, 7th ed. McGraw-Hill,
[2] R. Elmasri and S. Navathe, *Fundamentals of Database Systems*, 7th ed. Pearson, 2021.
[3] P. A. Bernstein, V. Hadzilacos, and N. Goodman, *Concurrency Control and Recovery in Database Systems*. Addison-Wesley, 1987.
[4] S. Yoon and K. Shim, “Conflict-free concurrency control in distributed DBMS,” *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 9, pp. 1721–1735, 2020
[5] Z. Li et al., “Detecting and preventing update anomalies in dynamic transaction systems,” *J. Inf. Syst.*, vol. 45, pp. 14–27, 2020.
[6] M. T. Özsu and P. Valduriez, *Principles of Distributed Database Systems*, 4th ed. Springer, 2020.
[7] Nur Wijayaning Rahayu, *“IMPLEMENTASI KENDALI KONKURENSI PADA BASIS DATA ORACLE 10g”*,2007
[8] Haonan Lu, Shuai Mu , Siddhartha Sen , Wyatt Lloyd, *“Natural Concurrency Control for Strictly Serializable Datastores by Avoiding the Timestamp-Inversion Pitfall”*, 2023
[9] M. U. Kalay, "Concurrency Challenges in Database Systems: A Focus on PostgreSQL," *Bilgisayar Bilimleri ve Teknolojileri Dergisi*, vol. 6, no. 1, pp. 1-16, 2025.
[10] T. Wang, R. Johnson, A. Fekete, and I. Pandis, "Efficiently making (almost) any concurrency control mechanism serializable," *The VLDB Journal*, vol. 26, pp. 537-562, 2017.
[11] Y. Hong, H. Zhao, W. Lu, X. Du, Y. Chen, A. Pan, and L. Zheng, "A Hybrid Approach to Integrating Deterministic and Non-deterministic Concurrency Control in Database Systems," *Proceedings of the VLDB Endowment*, vol. 18, no. 5, pp. 1376-1389, 2025.
[12] M. Burke, F. Suri-Payer, J. Helt, L. Alvisi, and N. Crooks, "Morty: Scaling Concurrency Control with Re-Execution," in *EuroSys '23*, Rome, Italy, 2023, pp. 1-16.
[13] S. Ghasemirad, C. Sprenger, S. Liu, L. Multazzu, and D. Basin, "Pushing the Limit: Verified Performance-Optimal Causally-Consistent Database Transactions," *arXiv preprint arXiv:2411.07049v2*, 2025.
[14] K. Ren, D. Li, and D. J. Abadi, "SLOG: Serializable, Low-latency, Geo-replicated Transactions," *Proceedings of the VLDB Endowment*, vol. 12, no. 11, pp. 1747-1761, 2019.
[15] J. Helt, M. Burke, A. Levy, and W. Lloyd, "Regular Sequential Serializability and Regular Sequential Consistency," in ACM SIGOPS 28th Symposium on Operating Systems Principles (SOSP ’21), Virtual Event, Germany, 2021, pp. 163-179.
[16] B. Ketsman, C. Koch, F. Neven, and B. Vandevoort, "Concurrency control for database theorists," SIGMOD Record, vol. 51, no. 4, pp. 6-17, 2023.

[17] B. Vandevoort, B. Ketsman, C. Koch, and F. Neven, "Detecting Robustness against MVRC for Transaction Programs with Predicate Reads," arXiv preprint arXiv:2302.08789, 2023.
[18] M. P. Flé and G. Roucairol, "Maximal Serializability of Iterated Transactions," Theoretical Computer Science, vol. 38, pp. 1-16, 1985
[19] A. Finandhita, "Manajemen Transaksi," Modul Kuliah Sistem Basis Data, Universitas Komputer Indonesia.