



Implementation of an Integrated Database System in the Bank Tabungan Negara Replication Monitoring Dashboard Application Using the Prototype Method

Implementasi Sistem Basis Data Terintegrasi pada Aplikasi Dashboard Monitoring Replikasi Bank Tabungan Negara Metode Prototype

Hendra Yunianto Ardhi^{1*}, Asep Muhidin², Tri Ngudi Wiyatno³

^{1,2,3}Program Studi Teknik Informatika, Fakultas Teknik,
Universitas Pelita Bangsa, Indonesia

E-Mail: ¹hendra.312210534@mhs.pelitabangsa.ac.id,
²asep.muhammad@pelitabangsa.ac.id, ³tringudi@pelitabangsa.ac.id

Received May 14th 2025; Revised Jul 01st 2025; Accepted Jul 11 th 2025; Available Online Jul 31th 2025, Published Jul 31th 2025
Corresponding Author: Hendra Yunianto Ardhi
Copyright © 2025 by Authors, Published by Institut Riset dan Publikasi Indonesia (IRPI)

Abstract

In the banking industry, the availability and protection of data are crucial. Bank Tabungan Negara (Bank BTN), as one of Indonesia's prominent financial institutions, requires a reliable system to monitor data replication in real time. This study aims to design and implement an integrated dashboard application for data replication monitoring using the prototype method. The system provides real-time status, performance, and alerts for replication processes. The method used includes requirement analysis, system design using flowcharts and Unified Modeling Language (UML) diagrams, and development through iterative prototyping involving users. The system is tested functionally using User Acceptance Testing (UAT) involving system analysts and quality control teams. Results show high acceptance rates with average scores between 4.3–4.7 on a 5-point Likert scale. The dashboard enables IT teams to detect replication delays, errors, and system anomalies more effectively. It also improves response time during emergencies and simplifies reporting. The conclusion shows that the implementation of the monitoring dashboard increases operational efficiency, reduces risks of data loss, and supports strategic decision-making. However, further development is suggested to include automated alert systems and interactive visualizations for better responsiveness.

Keyword: Dashboard Monitoring, Data Center, Data Replication, Database System

Abstrak

Dalam industri perbankan, ketersediaan dan perlindungan data menjadi aspek krusial. Bank Tabungan Negara (BTN) memerlukan sistem yang mampu memantau proses replikasi data secara *real-time*. Penelitian ini bertujuan untuk merancang dan mengimplementasikan Aplikasi *Dashboard Monitoring* Replikasi data menggunakan metode *prototyping*. Sistem ini menyediakan informasi status replikasi, kinerja, dan peringatan kesalahan secara terintegrasi. Metodologi yang digunakan meliputi analisis kebutuhan, perancangan sistem menggunakan *flowchart* dan diagram Unified Modeling Language (UML), serta pengembangan sistem secara iteratif dengan melibatkan pengguna. Hasil pengujian sistem dilakukan melalui User Acceptance Testing (UAT) yang melibatkan analis sistem dan tim *quality control*. Nilai rata-rata pada skala *likert* 1–5 menunjukkan skor tinggi antara 4,3 hingga 4,7. Aplikasi membantu tim IT mendeteksi keterlambatan replikasi, kesalahan sistem, serta mempercepat waktu tanggap dalam situasi darurat. Kesimpulannya, implementasi *dashboard* ini mampu meningkatkan efisiensi operasional, mengurangi risiko kehilangan data, dan mendukung pengambilan keputusan strategis. Namun, sistem ini masih dapat dikembangkan lebih lanjut dengan fitur notifikasi otomatis dan visualisasi interaktif.

Kata Kunci: *Dasboard Monitoring*, Data Center, Replikasi, Sistem Basis Data

1. PENDAHULUAN

Dalam industri perbankan yang semakin terintegrasi dan berbasis teknologi, kemampuan untuk mengelola dan mereplikasi data menjadi faktor krusial dalam menjaga kesinambungan layanan. Bank



Tabungan Negara (BTN), sebagai salah satu institusi keuangan besar di Indonesia, melalui Unit Data *Center* bertanggung jawab dalam memastikan ketersediaan dan keamanan data, termasuk saat terjadi gangguan atau kegagalan sistem. Proses ini sejalan dengan amanat regulasi, sebagaimana dinyatakan dalam Peraturan Otoritas Jasa Keuangan (OJK) yang mewajibkan setiap bank memiliki rencana pemulihan bencana (*disaster recovery plan*) dan memastikan implementasinya secara efektif [1]. Hal ini menjadi penting karena ketersediaan aplikasi dan sistem informasi perbankan tidak dapat dikompromikan, terutama ketika risiko kehilangan data dapat berdampak sistemik terhadap layanan dan reputasi institusi keuangan.

Proses *monitoring* replikasi data di Bank BTN masih menghadapi sejumlah kendala teknis yang signifikan dalam menjaga ketersediaan layanan adalah melalui proses replikasi data, yang memastikan bahwa salinan data tersedia secara *real-time* di berbagai pusat data. Adapun permasalahan, Pertama, pemantauan replikasi saat ini masih dilakukan secara manual melalui *command line*, yang mengharuskan tim teknis mengakses masing-masing server secara terpisah. Hal ini menyebabkan keterbatasan dalam pemantauan secara menyeluruh dan *real-time*. Kedua, karena tidak terhubung secara otomatis, sistem tidak dapat memberikan pembaruan status replikasi secara langsung. Akibatnya, jika terjadi gangguan atau keterlambatan (*lag time*), permasalahan tersebut sulit terdeteksi secara dini. Ketiga, pelaporan hasil replikasi yang dilakukan secara harian masih disusun manual menggunakan Microsoft Excel, yang selain memakan waktu, juga berisiko menurunkan akurasi data. Terakhir, tidak adanya pencatatan riwayat (*log history*) replikasi menyebabkan kesulitan dalam melakukan pelacakan waktu terakhir sinkronisasi berhasil dilakukan, sehingga menyulitkan proses analisis dan mitigasi saat terjadi anomali atau kegagalan system [2].

Untuk mengatasi permasalahan tersebut, penerapan teknologi *Dashboard Monitoring* Replikasi terhadap proses replikasi menjadi solusi yang inovatif dan efisien. Aktivitas *monitoring* penting dalam sistem informasi modern karena berfungsi untuk menjamin konsistensi data antar server dan mendeteksi potensi anomali. Dalam praktiknya, *dashboard monitoring* telah menjadi alat bantu yang efektif karena mampu menyajikan visualisasi status, performa sistem, log kesalahan, serta indikator keberhasilan replikasi secara *real-time* dan terpusat. Keunggulan ini memungkinkan tim teknis memberikan respons yang lebih cepat, efisien, dan tepat sasaran terhadap gangguan yang terjadi. Manfaat *dashboard* dalam menyajikan KPI secara *real-time* dan meningkatkan efisien *monitoring* dibandingkan metode manual [3]. Hal ini sangat penting untuk mendukung pengambilan keputusan strategis yang dapat mempengaruhi keberlangsungan operasional dalam bisnis Perusahaan.

Berbagai studi sebelumnya telah membuktikan pentingnya penerapan *dashboard* dalam konteks bisnis. Willana (2018) merancang *dashboard monitoring* untuk keperluan evaluasi akademik di FILKOM UB [4], sedangkan Sulistiani dan Sulistiawati (2018) mengembangkan *dashboard* interaktif untuk memantau penjualan [5]. Penelitian oleh Apriyanto (2021) dalam pengembangan sistem informasi perpustakaan digital [6], dan Suhaidir & Sensuse (2010) menyoroti peran *dashboard* dalam mendukung keputusan berbasis analisis sensitivitas kinerja [7]. Meskipun seluruh penelitian tersebut menunjukkan manfaat *dashboard* dalam pengambilan keputusan dan efisiensi pemantauan, fokusnya terbatas pada lingkup fungsional tertentu, seperti manajemen akademik, penjualan, atau keuangan perusahaan. Selain itu, *dashboard* yang dikembangkan belum menyertakan integrasi sistem replikasi database secara langsung maupun komprehensif, serta belum dirancang untuk kebutuhan lingkungan yang bersifat kritikal seperti sistem perbankan. Tidak ada pula yang secara spesifik membahas penggabungan berbagai teknologi replikasi seperti *IBM DB2 HADR*, *SQL Server Replication*, dan *Oracle Data Guard* dalam satu antarmuka terpadu. Dengan demikian, penelitian ini memiliki perbedaan mendasar, yaitu mengusulkan desain dan implementasi *dashboard monitoring* yang secara khusus ditujukan untuk sistem replikasi data di sektor perbankan, dengan tingkat integrasi dan *real-time monitoring* yang lebih tinggi serta dirancang untuk mendukung keberlangsungan layanan di lingkungan produksi yang bersifat critical seperti Bank BTN.

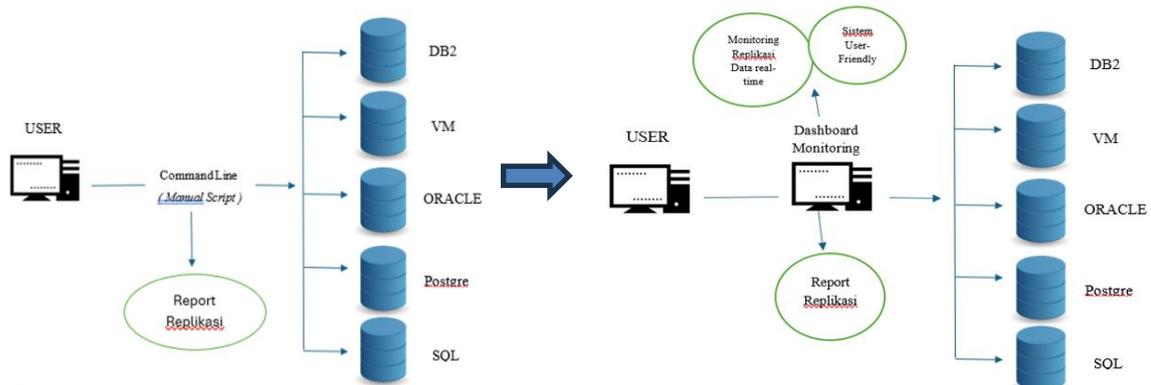
Penelitian ini bertujuan untuk merancang dan mengimplementasikan Aplikasi *Dashboard Monitoring* Replikasi terintegrasi yang dirancang khusus untuk kebutuhan sistem informasi perbankan di Bank BTN. Aplikasi ini akan menyajikan data replikasi secara *real-time*, memantau kondisi sinkronisasi antar server, serta menyediakan informasi log historis secara otomatis. Selain itu, sistem akan mendukung pelaporan otomatis untuk mengurangi ketergantungan terhadap proses manual, serta meningkatkan akurasi dan efisiensi kerja tim teknis. Evaluasi penelitian akan difokuskan pada performa teknis aplikasi, akurasi data yang ditampilkan [8], pengalaman pengguna (*user experience*), serta keandalan sistem dalam mendukung operasional Unit Data *Center*. Dengan pendekatan ini, diharapkan aplikasi yang dikembangkan dapat memperkuat ketahanan data, meningkatkan efektivitas *monitoring*, serta mendukung keberlangsungan layanan perbankan secara menyeluruh. Melalui penelitian ini, kami berharap hasil penelitian dapat memberikan wawasan nyata, memberikan kontribusi yang berarti dalam pengembangan sistem, memberikan manfaat peningkatan kualitas pelayanan dan menjadi acuan atau referensi bagi Perusahaan lain yang menghadapi tantangan serupa.

2. BAHAN DAN METODE

2.1. Metode yang Digunakan

Dalam penyusunan penelitian ini penulis melakukan dengan mengambil *object* penelitian pada sebuah perusahaan keuangan Bank BTN Yang berlokasi di Jakarta. Dalam melakukan metodologi perancangan yang digunakan pada penelitian ini dilakukan dengan beberapa tahapan, pertama peneliti melakukan penentuan masalah, kemudian penelitian menentukan tinjauan pustaka yang akan digunakan sebagai referensi dalam penelitian. Tahapan metodologi selanjutnya adalah pengumpulan data yang akan dibutuhkan untuk penelitian, mencari informasi metode replikasi yang digunakan sebelumnya dan yang akan diusulkan, pengembangan sistem dengan menggunakan metode *prototype*, membuat *flowchart Dashboard Monitoring Replikasi* dan *use case diagram*, dan yang terakhir adalah pengujian sistem.

2.2. Sistem yang Berjalan dan yang Diusulkan



Gambar 1. Sistem yang Berjalan dan yang Diusulkan

Gambar 1 menggambarkan perbandingan antara arsitektur sistem replikasi data yang berjalan saat ini dengan sistem *monitoring* replikasi yang diusulkan. Pada sistem *eksisting* (sebelah kiri), proses pemantauan replikasi masih bersifat manual, di mana pengguna (*user*) menjalankan *script* melalui *command line interface* (CLI) untuk setiap jenis *database* seperti *DB2*, *VM*, *Oracle*, *Postgre*, dan *SQL*. Proses ini tidak hanya memerlukan waktu dan pemahaman teknis dari pengguna, tetapi juga rawan terhadap kesalahan input dan keterlambatan dalam pengambilan keputusan karena tidak bersifat *real-time*.

Sebagai solusi, sistem yang diusulkan (sebelah kanan) menerapkan *dashboard monitoring* yang dapat mengakses seluruh sumber data secara terpusat dan menampilkan status replikasi secara *real-time*. Sistem ini memiliki antarmuka grafis yang lebih *user-friendly*, memungkinkan pengguna dari berbagai latar belakang teknis untuk memantau replikasi tanpa harus menjalankan perintah manual. *Dashboard* juga mampu menyajikan laporan replikasi secara otomatis dan terintegrasi, sehingga mendukung proses audit, pelaporan, dan pengambilan keputusan secara cepat dan akurat [9].

2.3. Sistem dengan Metode *Prototype*

Metode pengembangan sistem yang digunakan untuk proses pengembangan perangkat lunak adalah dengan menggunakan metode *prototype*. *Prototype* merupakan suatu metode dalam pengembangan sistem yang menggunakan pendekatan untuk membuat sesuatu program dengan cepat dan bertahap sehingga segera dapat dievaluasi oleh pemakai. Selain itu *prototype* juga membuat proses pengembangan sistem informasi menjadi lebih cepat dan lebih mudah.

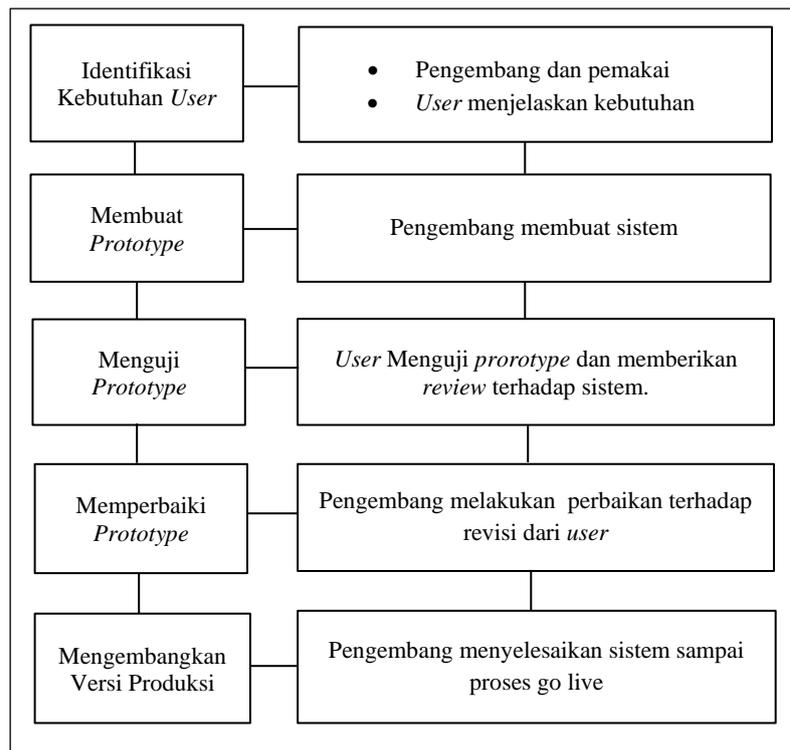
Metodologi *prototyping* merupakan salah satu pendekatan dalam pengembangan sistem informasi yang bersifat iteratif dan melibatkan interaksi langsung antara pengguna dengan pengembang. Proses ini diawali dengan tahap pengumpulan kebutuhan, di mana pengguna dan pengembang bersama-sama mendefinisikan secara rinci seluruh kebutuhan sistem yang akan dibangun. Pendekatan ini bertujuan untuk memperoleh pemahaman yang mendalam terhadap harapan dan kebutuhan pengguna. [10]

Tahap berikutnya adalah membangun *prototype*, yaitu pembuatan rancangan sistem sementara yang menitikberatkan pada aspek antarmuka dan alur fungsional utama. Tujuannya adalah untuk memberikan gambaran awal kepada pengguna mengenai bentuk dan fungsi sistem. Setelah *prototype* awal selesai, dilakukan tahap penggunaan sistem, di mana pengguna mencoba sistem tersebut dan memberikan umpan balik mengenai kesesuaiannya dengan kebutuhan yang telah disampaikan. Proses ini memungkinkan pengembang melakukan penyesuaian secara dini sebelum sistem dikembangkan secara penuh. Gambar 2 merupakan sistem metode *prototype*.

Jika *prototype* telah disetujui, maka proses dilanjutkan ke tahap pengkodean sistem, yaitu penerjemahan desain sistem ke dalam bahasa pemrograman yang sesuai. Pengembang akan membangun

sistem secara komprehensif berdasarkan struktur dan logika yang telah disepakati. Setelah pengkodean selesai, sistem akan masuk ke tahap pengujian, dengan menggunakan metode seperti *functional testing* untuk memastikan bahwa setiap fungsi utama berjalan sesuai dengan spesifikasi dan dapat memberikan *output* yang akurat.

Setelah pengujian selesai, dapat dilakukan evaluasi sistem oleh pengguna untuk menilai sejauh mana sistem yang telah dikembangkan memenuhi harapan mereka, baik dari segi fungsionalitas, kemudahan penggunaan, maupun stabilitas. Tahap akhir adalah evaluasi dan finalisasi *prototype*, di mana sistem yang telah melalui proses revisi dan pengujian secara menyeluruh dinyatakan siap untuk diimplementasikan ke dalam lingkungan operasional. Dengan demikian, pendekatan *prototyping* tidak hanya mempercepat proses pengembangan sistem, tetapi juga memastikan bahwa solusi yang dihasilkan benar-benar sesuai dengan kebutuhan pengguna.



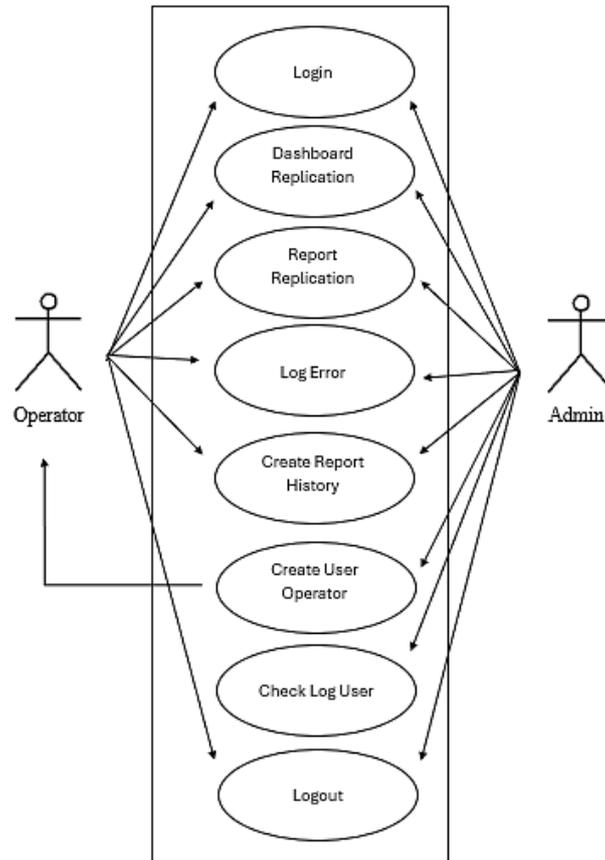
Gambar 2. Sistem Metode *Prototype*

2.4. Perancangan Sistem

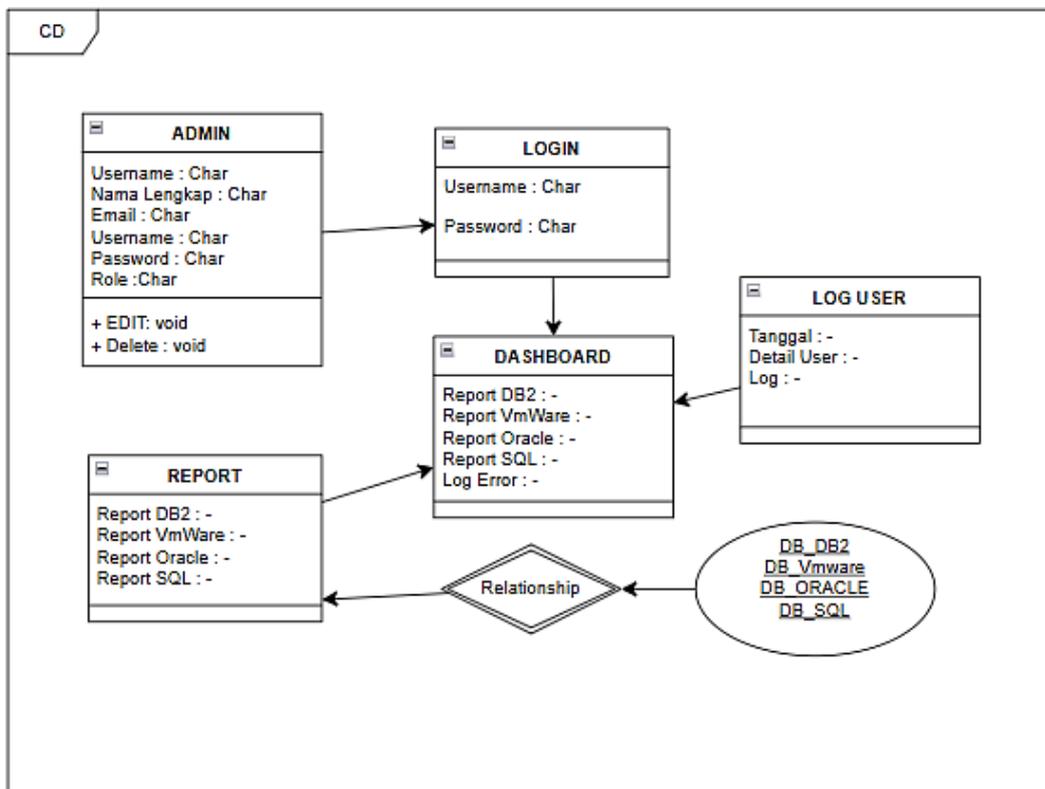
Use case diagram untuk sistem *Dashboard Monitoring Replikasi*, yang merepresentasikan interaksi antara dua aktor utama, yaitu operator dan *admin*, dengan berbagai fungsi sistem. Diagram ini dirancang berdasarkan prinsip *Unified Modeling Language (UML)*, yang secara luas digunakan dalam pengembangan sistem berbasis *object-oriented* untuk menggambarkan hubungan antara aktor eksternal dan layanan yang disediakan oleh system [11]. *Use case diagram* memberikan gambaran umum tentang fasilitas-fasilitas yang dibutuhkan program yang akan dibuat, menjabarkan fasilitas-fasilitas yang disediakan sistem untuk membantu *user* dalam melakukan pekerjaannya [12]. *Use case diagram* dapat dilihat pada Gambar 3.

Class diagram merupakan sebuah *class* yang menggambarkan struktur dan penjelasan *class*, paket, dan objek serta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi dan lain-lain [11]. *Class diagram* juga menjelaskan hubungan antar *class* dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar saling berkolaborasi untuk mencapai sebuah tujuan. Berdasarkan hasil analisis kebutuhan yang dilakukan, *class diagram* yang terbentuk dari sistem yang akan dibuat dapat dilihat pada Gambar 4.

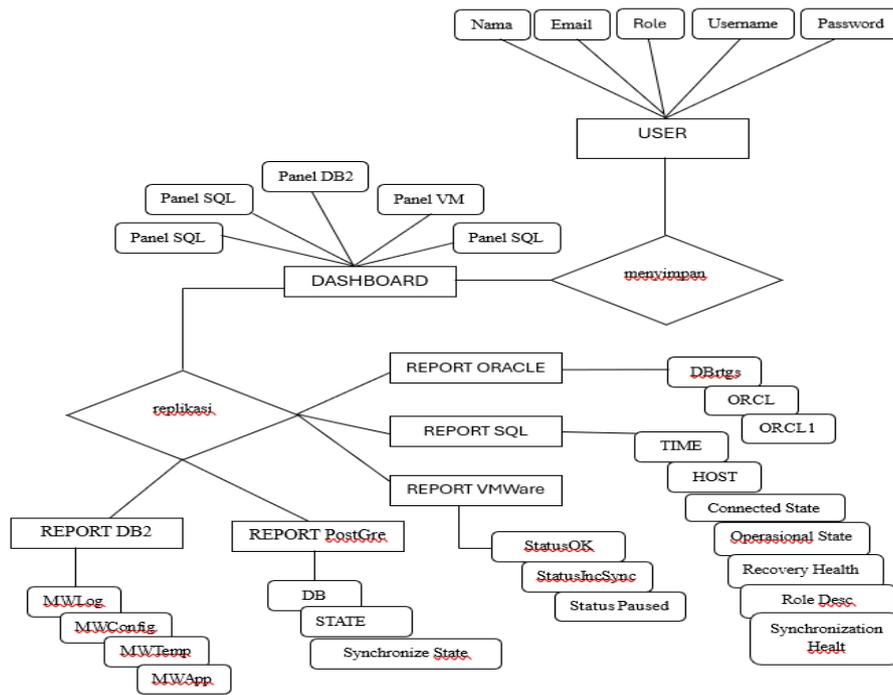
Entity Relationship Diagram (ERD) merupakan representasi konseptual dari hubungan antar entitas yang terdapat dalam sistem *Dashboard Monitoring Replikasi*. Diagram ini berfungsi untuk menggambarkan struktur data dan interaksi antar komponen yang digunakan dalam pengembangan aplikasi *dashboard* yang terintegrasi untuk memantau status replikasi di lingkungan Bank BTN [13]. Rancangan *ERD* dapat dilihat pada Gambar 5.



Gambar 3. Use Case Diagram



Gambar 4. Class Diagram



Gambar 5. ERD Dashboard Replikasi

3. HASIL DAN PEMBAHASAN

3.1. Kebutuhan Fungsional

Kebutuhan fungsional pengoperasian sistem aplikasi *dashboard monitoring* berbasis web pada Bank BTN merupakan tahap yang berfungsi menjelaskan secara rinci setiap fungsi yang digunakan untuk memantau dan *me-monitoring* replikasi yang ada pada setiap aplikasi dan server yang terhubung. Fungsi-fungsi yang digunakan oleh sistem dapat dilihat pada Tabel 1.

Tabel 1. Akses Menu

Akses	User	
	Admin	Operator
Menu Login	√	√
Menu Dashboard	√	√
Menu Report	√	√
Menu Notification / Log Error	√	√
Menu User	√	-
Menu Log User	√	-

Pada Tabel 1 menunjukkan pembagian hak akses menu dalam sistem *monitoring* replikasi berdasarkan peran pengguna, yaitu *admin* dan *operator*. Setiap jenis pengguna memiliki akses yang berbeda terhadap menu-menu yang tersedia, sesuai dengan tanggung jawab dan wewenangnya dalam sistem. Pengguna dengan peran *admin* memiliki akses penuh terhadap seluruh menu dalam sistem, termasuk *Menu Login*, *Dashboard*, *Report*, *Notification/Log Error*, *User*, dan *Log User*. Hal ini bertujuan agar *admin* dapat melakukan pengawasan, pengelolaan pengguna, serta analisis log secara menyeluruh untuk keperluan pemeliharaan sistem.

Sementara itu, pengguna dengan peran *operator* memiliki akses yang lebih terbatas. *Operator* hanya dapat mengakses *Menu Login*, *Dashboard*, *Report*, dan *Notification/Log Error*. Tanpa diberikan hak akses terhadap *Menu User* dan *Menu Log User*. Pembatasan ini bertujuan untuk menjaga integritas dan keamanan sistem, di mana aktivitas manajemen *user* dan pemantauan log historis hanya dilakukan oleh *admin*. Dengan pembagian hak akses seperti ini, sistem dapat berjalan dengan prinsip *Role-Based Access Control (RBAC)*, yang memastikan bahwa setiap pengguna hanya dapat mengakses fitur yang sesuai dengan tugas dan kewenangannya.

3.2. Kebutuhan Non-Fungsional

Kebutuhan non fungsional bertujuan untuk memenuhi kebutuhan fungsional. Berikut ini adalah kebutuhan non Fungsional dari sistem Aplikasi *Dashboard Monitoring* Replikasi untuk perangkat pendukung. dalam pengembangan dan implementasi sistem *monitoring* replikasi ini, diperlukan perangkat

keras dan lunak pendukung yang memadai guna memastikan sistem dapat berjalan secara optimal, khususnya dalam menangani proses replikasi data yang bersifat kritikal dan *real-time*. Adapun spesifikasi perangkat pendukung yang digunakan dapat dilihat pada Tabel 2.

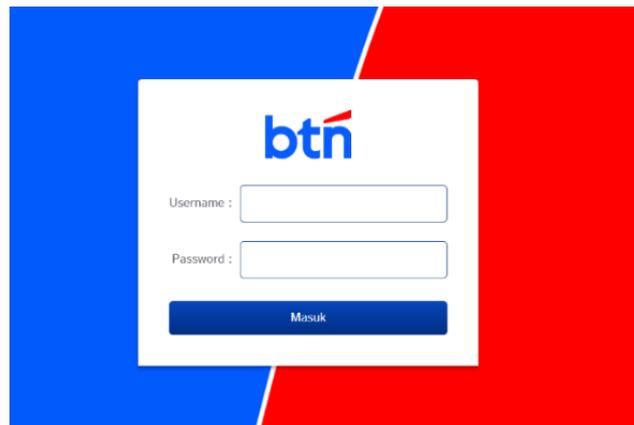
Tabel 2. Perangkat Pendukung

Spesifikasi	
Server	<i>Virtual Machine (VM)</i>
Operating Sistem	Ubuntu 20
Spesifikasi	
RAM	32 GB
CPU	16 Core
Disk / Storage	320 GB

Tabel 2 menunjukkan bahwa perangkat yang digunakan adalah *Virtual Machine (VM)* yang menjalankan sistem operasi Ubuntu versi 20, yang dikenal stabil dan cocok untuk kebutuhan server berbasis Linux. *VM* ini dikonfigurasi dengan RAM sebesar 32 GB, CPU sebanyak 16 core, serta kapasitas penyimpanan sebesar 320 GB. Spesifikasi ini telah disesuaikan untuk mendukung beban kerja proses replikasi data dari berbagai sumber aplikasi dan memastikan performa sistem tetap responsif. Penggunaan infrastruktur virtual juga memberikan fleksibilitas dalam pengelolaan sumber daya serta kemudahan dalam proses *backup* dan *recovery* apabila terjadi kegagalan Sistem.

3.3. Sistem Interface Dashboard Replikasi

3.3.1. Tampilan Login



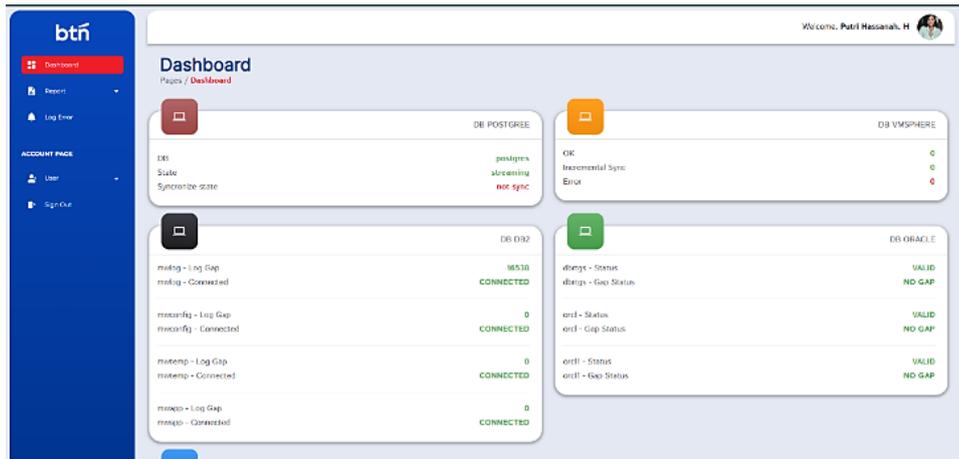
Gambar 6. Tampilan Login

Tampilan *login* di atas merupakan halaman awal yang ditampilkan ketika pengguna pertama kali mengakses Aplikasi *Dashboard Monitoring* Replikasi. Halaman ini bertujuan untuk membatasi akses hanya kepada pengguna yang telah memiliki akun dan otorisasi yang sah seperti yang ditunjukkan pada Gambar 6, tampilan *login* dirancang dengan antarmuka yang sederhana namun profesional. *Form login* terdiri atas dua *input* utama, yaitu kolom *Username* dan *Password*. Pengguna diwajibkan untuk mengisi kedua kolom ini dengan informasi yang benar sebelum dapat mengakses fitur-fitur utama dalam sistem. Setelah data diisi, pengguna dapat menekan tombol "Masuk" untuk melanjutkan proses autentikasi.

Desain visual dibuat dengan perpaduan warna merah dan biru sebagai ciri khas perusahaan, yang juga membantu menciptakan kesan tegas dan modern. Penggunaan warna dan *layout* yang responsif juga ditujukan untuk meningkatkan kenyamanan pengguna saat berinteraksi dengan sistem, Fungsi *login* ini sangat penting untuk menjaga keamanan data dan memastikan bahwa setiap aktivitas dalam sistem dapat ditelusuri berdasarkan identitas pengguna yang sah. Selain itu, halaman ini juga menjadi titik awal untuk mencatat aktivitas *audit trail* dari setiap sesi *login*, yang berguna dalam proses pengawasan dan evaluasi sistem secara berkala.

3.3.2. Tampilan Menu Dashboard

Gambar 7 merupakan tampilan menu *dashboard*, dimana menu ini menampilkan data replikasi secara keseluruhan dalam bentuk *card view*. Untuk data yang ditampilkan replikasi *DB2*, *VMWare*, *ORACLE*, dan *SQL*. Dimana terlihat untuk menu lainya seperti menu *report*, menu *log error*, menu *user* dan *log user*.



Gambar 7. Tampilan Dashboard

3.4. Pengujian Sistem

Pengujian sistem merupakan tahapan krusial dalam proses pengembangan perangkat lunak, yang bertujuan untuk memverifikasi dan memvalidasi bahwa sistem yang dibangun telah sesuai dengan kebutuhan pengguna serta bekerja secara optimal dalam kondisi operasional yang diharapkan. Pada penelitian ini, proses pengujian difokuskan pada Aplikasi *Dashboard Monitoring Replikasi*, yang berfungsi untuk memantau aktivitas replikasi data secara *real-time* pada lingkungan sistem perbankan yang bersifat *critical* [14].

Pengujian dilakukan dengan pendekatan metodologi pengujian fungsional menggunakan *Black Box Testing*, serta pengujian berbasis persepsi pengguna melalui *User Acceptance Testing (UAT)*. Metode ini dipilih karena sesuai untuk menilai sistem berbasis antarmuka pengguna (*dashboard*) dan dapat mengukur keberterimaan sistem terhadap kebutuhan nyata dari *user* akhir.[15]

3.4.1. Metode pengujian *Black Box Testing*

Black Box Testing digunakan untuk menguji fungsionalitas dari masing-masing fitur tanpa melihat kode program. Fokus utama pengujian ini adalah apakah sistem dapat menerima *input* yang benar, memproses data dengan benar, dan menghasilkan *output* yang sesuai[16]. Berdasarkan hasil pengujian *blackbox*, seluruh fitur berhasil dijalankan sesuai dengan skenario yang telah ditentukan[17]. Tabel di bawah ini menunjukkan hasil uji coba :

Tabel 3. Pengujian *Blackbox*

No	Fitur yang Diuji	Input	Ekspektasi Output	Hasil Aktual
1	Login	Username dan password valid	Admin berhasil login	Sesuai
2	Tampilan Status Replikasi	Menampilkan status <i>synchronized</i> , lag, atau <i>failed</i> pada setiap server	Status <i>real-time</i> tampil akurat di <i>dashboard</i> utama	Sesuai
3	Monitoring Real-time	Sistem menampilkan data per 5 detik secara otomatis tanpa perlu refresh	<i>Dashboard</i> memperbarui informasi secara otomatis	Sesuai
4	Riwayat Log Replikasi	Pengguna dapat melihat histori replikasi 7 hari terakhir	Data log tampil lengkap dan kronologis	Sesuai
5	Laporan Harian Otomatis	Sistem menghasilkan laporan replikasi harian dalam format Excel dan PDF	File laporan ter-generate otomatis dan dapat diunduh	Sesuai
6	Autentikasi dan Hak Akses	Pengguna dengan role "viewer" tidak bisa menghapus atau mengubah data	Hak akses diterapkan sesuai level pengguna	Sesuai
7	Logout	Klik tombol <i>logout</i>	Sistem kembali ke halaman <i>login</i>	Sesuai

Berdasarkan hasil 7 kriteria pengujian yang telah dilakukan didapatkan hasil dari pengujian telah sesuai dan lulus tes.

Tabel 4. Hasil *Blackbox*

Kriteria Pengujian Sistem	Berhasil	Tidak Berhasil
Login	√	
Tampilan Status Replikasi	√	

Kriteria Pengujian Sistem	Berhasil	Tidak Berhasil
<i>Monitoring Real-time</i>	√	
Riwayat Log Replikasi	√	
Laporan Harian Otomatis	√	
Autentikasi dan Hak Akses	√	
<i>Logout</i>	√	

3.4.2. Metode User Acceptance Test (UAT)

UAT dilakukan oleh pengguna akhir (tim teknis Data Center) untuk mengevaluasi apakah sistem yang dikembangkan sudah sesuai dengan kebutuhan pengguna dan dapat diterima secara fungsional [18]. Untuk mengukur sejauh mana Aplikasi *Dashboard Monitoring* Replikasi diterima oleh pengguna, dilakukan pengujian menggunakan metode UAT melalui penyebaran kuesioner kepada 10 responden yang merupakan calon pengguna dari aplikasi ini. Kuesioner terdiri dari 10 pernyataan yang merepresentasikan fitur-fitur utama dalam sistem, seperti kemudahan penggunaan, ketepatan data, stabilitas sistem, dan kepuasan pengguna secara keseluruhan. Kuesioner UAT dapat dilihat pada Tabel 5.

Tabel 5. Kuesioner UAT

No	Pernyataan	Skor (1-5)
1	Tampilan <i>dashboard</i> mudah dipahami	
2	Data replikasi ditampilkan dengan akurat	
3	Fitur filter tanggal bekerja dengan baik	
4	<i>Export</i> data berjalan lancar	
5	Notifikasi gangguan muncul tepat waktu	
6	Proses <i>login</i> dan <i>logout</i> tanpa kendala	
7	Respons sistem cepat dan stabil	
8	Fitur update status server mudah digunakan	
9	<i>Dashboard</i> membantu saya memantau sistem	
10	Saya puas menggunakan <i>dashboard monitoring</i> ini	

Masing-masing pernyataan dinilai menggunakan skala *Likert* dengan rentang nilai 1 (sangat tidak setuju) hingga 5 (sangat setuju). Hasil pengujian menunjukkan bahwa semua responden memberikan skor yang tinggi pada hampir seluruh pernyataan, dengan nilai rata-rata berkisar antara 4,3 hingga 4,7.

Tabel 6. Hasil Kuesioner UAT

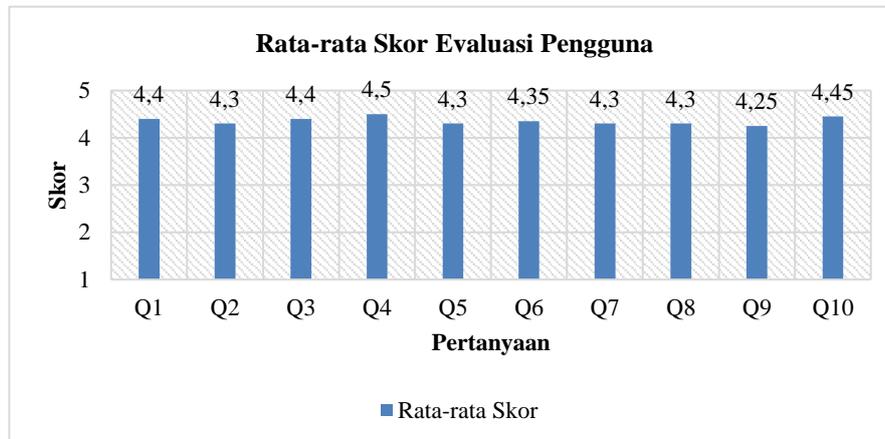
Responden	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Nilai Rata-rata	Keterangan
1	4	5	5	4	5	4	4	5	4	4	4,4	Diterima
2	5	4	5	4	4	5	5	5	4	4	4,5	Diterima
3	4	5	4	5	4	4	4	5	4	5	4,4	Diterima
4	4	5	4	4	5	4	5	4	5	5	4,5	Diterima
5	4	4	5	5	5	4	4	4	5	4	4,4	Diterima
6	5	4	4	4	5	5	4	4	4	4	4,3	Diterima
7	4	5	5	5	5	4	5	4	4	4	4,5	Diterima
8	4	5	4	4	4	5	5	4	4	4	4,3	Diterima
9	4	4	5	5	4	4	4	4	5	5	4,4	Diterima
10	5	5	4	5	5	5	4	4	5	5	4,7	Diterima

Pada Tabel 6 Rata-rata nilai UAT sebesar 4,5 menunjukkan bahwa pengguna merasa puas terhadap tampilan dan kinerja sistem. Berdasarkan distribusi jawaban, sebagian besar responden memberikan skor 5 terhadap fitur *dashboard* dan *login*. Ini menunjukkan sistem telah memenuhi ekspektasi pengguna, meskipun beberapa fitur seperti visualisasi data masih perlu pengembangan. Hasil UAT menunjukkan bahwa sistem dinilai layak dan siap digunakan dalam lingkungan produksi. Pengguna merasa aplikasi ini memberikan visibilitas yang lebih baik terhadap status replikasi dibanding metode manual sebelumnya [19]. Grafik hasil pengguna dapat dilihat pada Gambar 8.

3.5. Diskusi

Pada bagian ini, penulis akan menginterpretasikan hasil yang diperoleh dari implementasi sistem *Dashboard Monitoring* Replikasi yang dikembangkan. Secara umum penggunaan *dashboard* terintegrasi memudahkan pengguna dalam mengakses informasi replikasi secara *real-time* melalui tampilan visual yang terstruktur, menggantikan metode lama yang masih berbasis *command line* dan pelaporan manual. Beberapa fitur seperti pemantauan aktivitas replikasi dari berbagai sistem *database* seperti *Oracle*, *SQL Server*, *DB2*, dan *PostgreSQL* yang disajikan dalam bentuk indikator warna dan status teks, terbukti dapat meningkatkan efisiensi tim IT dalam mendeteksi dan merespon anomali atau keterlambatan replikasi dengan lebih cepat.

Keberhasilan ini menunjukkan bahwa visualisasi data melalui *dashboard* merupakan strategi efektif untuk mendukung pengambilan keputusan teknis secara cepat dan akurat, sebagaimana juga didukung dalam penelitian oleh Rusdianto et al [20].



Gambar 8. Grafik Hasil Pengguna

Namun demikian, implementasi sistem ini masih menyisakan beberapa catatan penting yang perlu diperhatikan dalam pengembangan selanjutnya. Salah satunya adalah tampilan antarmuka pengguna (*user interface*) yang dinilai oleh sebagian pengguna masih cukup kompleks dan membutuhkan waktu adaptasi, terutama bagi pengguna yang tidak terbiasa dengan sistem *monitoring* berbasis web. Hal ini menunjukkan perlunya penyederhanaan navigasi dan peningkatan aspek keterbacaan serta interaksi dalam desain antarmuka, agar sistem dapat lebih inklusif bagi berbagai tingkatan pengguna.

Kendala lainnya adalah belum terintegrasinya sistem log histori replikasi secara penuh untuk semua jenis server, sehingga rekam jejak status sinkronisasi yang dibutuhkan dalam proses audit atau *troubleshooting* belum terdokumentasi dengan optimal. Beberapa responden juga mencatat bahwa sistem belum memberikan notifikasi otomatis saat terjadi lag time atau status gagal sinkronisasi, yang dapat menghambat proses mitigasi dini terhadap gangguan operasional.

Secara keseluruhan, sistem *Dashboard Monitoring* Replikasi yang dikembangkan telah menunjukkan kinerja yang baik dalam memantau proses replikasi data secara *real-time*, mengurangi beban kerja manual, dan meningkatkan kecepatan respons teknis. Namun demikian, ruang perbaikan masih terbuka, khususnya pada aspek kemudahan penggunaan (*usability*), dokumentasi log, dan perluasan fitur notifikasi otomatis. Penambahan fitur-fitur tersebut diharapkan dapat meningkatkan kualitas sistem menjadi lebih andal, adaptif, dan responsif terhadap kebutuhan pengguna di lingkungan perbankan yang dinamis dan kompleks.

4. KESIMPULAN

Berdasarkan hasil analisa dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa Aplikasi *Dashboard Monitoring* Replikasi sangat diperlukan untuk mempermudah proses pengawasan dan pengolahan data replikasi dalam mendukung operasional sistem. Aplikasi ini mampu menyajikan informasi status replikasi secara *real-time* serta menghasilkan laporan yang akurat sesuai dengan kebutuhan. Dengan adanya sistem ini, pengguna dapat lebih cepat mengidentifikasi kendala replikasi, sehingga dapat segera dilakukan penanganan. Kesimpulan ini menjawab tujuan utama penelitian, yaitu merancang dan mengimplementasikan sistem *monitoring* yang dapat meningkatkan efisiensi pemantauan serta akurasi laporan proses replikasi data. Permasalahan terkait kurangnya visibilitas dan keterlambatan dalam proses pemantauan replikasi dapat diatasi melalui *dashboard* ini.

Namun, penelitian ini memiliki beberapa keterbatasan. Salah satu kelemahan adalah belum terintegrasinya sistem dengan notifikasi otomatis berbasis *threshold* atau *alert*, serta keterbatasan dalam hal fleksibilitas visualisasi data yang masih bersifat statis. Saran untuk pengembangan selanjutnya, disarankan agar sistem ini dilengkapi dengan fitur notifikasi otomatis (misalnya melalui email atau pesan instan) apabila terjadi anomali atau kegagalan replikasi. Selain itu, peningkatan pada sisi tampilan visualisasi interaktif dan integrasi dengan sistem log audit juga perlu dilakukan untuk mendukung analisa yang lebih mendalam dan akuntabel. Diharapkan pengembangan lebih lanjut dapat memberikan nilai tambah dalam hal kecepatan pengambilan keputusan serta pengendalian risiko dalam pengelolaan data perusahaan.

REFERENSI

- [1] OJK, "Salinan Peraturan Otoritas Jasa Keuangan Republik Indonesia Nomor 11 /POJK.03/2022 Tentang Penyelenggaraan Teknologi Informasi Oleh Bank Umum," *Peratur. Otoritas Jasa Keuang.,*

- p. 76, 2022, [Online]. Available: [https://ojk.go.id/id/regulasi/Documents/Pages/Penyelenggaraan-Teknologi-Informasi-Oleh-Bank-Umum/POJK 11 - 03 - 2022.pdf](https://ojk.go.id/id/regulasi/Documents/Pages/Penyelenggaraan-Teknologi-Informasi-Oleh-Bank-Umum/POJK%2011-03-2022.pdf)
- [2] S. Niedermaier, F. Koetter, A. Freymann, and S. Wagner, "On Observability and Monitoring of Distributed Systems – An Industry Interview Study," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11895 LNCS, pp. 36–52, 2019, doi: 10.1007/978-3-030-33702-5_3.
 - [3] N. Mutiah, "Sistem Informasi Pengukuran Dan Pemantauan Kinerja Menggunakan Metode Performance Dashboard Performance Measurement and Monitoring Information System Using Performance Dashboard Method," *CESS (Journal Comput. Eng. Syst. Sci.)*, vol. 7, no. January, pp. 238–252, 2022, [Online]. Available: www.jurnal.unimed.ac.id
 - [4] W. W. Sihombing, H. Aryadita, and D. S. Rusdianto³, *Perancangan Dashboard Untuk Monitoring Dan Evaluasi (Studi Kasus : FILKOM UB)*, vol. 3, no. 1. 2019. [Online]. Available: <http://j-ptiik.ub.ac.id>
 - [5] H. Sulistiani, "Perancangan Dashboard Interaktif Penjualan (Studi Kasus : Pt Jaya Bakery)," *J. Tekno Kompak*, vol. 12, no. 1, p. 15, 2018, doi: 10.33365/jtk.v12i1.61.
 - [6] D. Aprianto and K. R. Wardani, "Penggunaan metode prototyping dalam pengembangan sistem informasi perpustakaan digital," *J. Inf. Syst. Comput. Sci.*, vol. 1, no. 2, pp. 75–88, 2021, [Online]. Available: [http://eprints.binadarma.ac.id/10884/%0Ahttp://eprints.binadarma.ac.id/10884/1/Penggunaan Metode Prototyping Dalam Pengembangan Sistem Informasi Perpustakaan Digital.pdf](http://eprints.binadarma.ac.id/10884/%0Ahttp://eprints.binadarma.ac.id/10884/1/Penggunaan%20Metode%20Prototyping%20Dalam%20Pengembangan%20Sistem%20Informasi%20Perpustakaan%20Digital.pdf)
 - [7] W. Suhaidir and I. Sensuse, "Perancangan Digital Dashboard System Untuk Menyajikan Sensitivity Analysis Kinerja Keuangan Perusahaan Studi Kasus: PT XYZ," *J. Inf. Syst.*, vol. 6, no. 2, pp. 94–107, 2010.
 - [8] S. A. N. Maulana, E. Wijayanti, and ..., "Penggunaan Barcode dalam Sistem Inventory Modern untuk Meningkatkan Akurasi dan Kecepatan Operasional: Utilization of Barcode Technology in Modern ...," *Mach. Learn. ...*, vol. 5, no. July, pp. 807–818, 2025, [Online]. Available: <https://www.journal.irpi.or.id/index.php/malcom/article/view/1943>
 - [9] U. Rahardja, Q. Aini, and N. Enay, "Optimalisasi Dashboard pada Sistem Penilaian Ujian Mahasiswa Sebagai Media Informasi Perguruan Tinggi Optimizing Dashboard on Assessment System As An Information Media in Higher Education," *J. Ilm. SISFOTENIKA*, vol. 7, no. 2, pp. 168–176, 2017.
 - [10] Hery, J. Renaldo Luih, C. Alencia Haryani, and A. E. Widjaja, "Penerapan Teknologi Qr Code Berbasis Web pada Sistem Manajemen Inventaris di Gudang PT XYZ," *Technomedia J.*, vol. 7, no. 2, pp. 202–215, 2022, doi: 10.33050/tmj.v7i2.1903.
 - [11] A. Helsalia, H. Pratama, M. Kristiani, and Y. B. Marpaung, "Perancangan Aplikasi Pemesanan Obat di Apotek Dengan Analisis Design UML Yang Menerapkan GIS dan LBS," *J. Tek. Inform.*, vol. 1, no. 1, pp. 1–20, 2021.
 - [12] I. T. Kusnadi, W. Kusnadi, A. Supiandi, and R. Nugraha, "Pengembangan Sistem Informasi Penjualan Berbasis Web Menggunakan Metode Usecase Driven," *J. Inform.*, vol. 60, no. 1, pp. 53–77, 2020.
 - [13] S. Sentosa, R. E. Indrajit, and E. Dazki, "Enterprise Architecture of the Basic Banking Feature for a New Challenger of Digital Banking in Indonesia," *Sinkron*, vol. 8, no. 4, pp. 2197–2211, 2024, doi: 10.33395/sinkron.v8i4.14116.
 - [14] O. Deshmukh and M. Kaushik, "A Overview of Software Verification & Validation and Selection Process," *Int. J. Comput. Trends Technol.*, vol. 4, no. 2, pp. 176–182, 2013, [Online]. Available: <http://www.internationaljournalssrg.org>
 - [15] K. Kurniati, "Penerapan Metode Prototype Pada Perancangan Sistem Pengarsipan Dokumen Kantor Kecamatan Lais," *J. Softw. Eng. Ampera*, vol. 2, no. 1, pp. 16–27, 2021, doi: 10.51519/journalsea.v2i1.89.
 - [16] I. Wahyudi and F. Alameka, "Analisis Blackbox Testing Dan User Acceptance Testing Terhadap Sistem Informasi," vol. 04, no. 01, pp. 1–9, 2023.
 - [17] A. Amalia, S. Wanda, P. Hamidah, and T. Kristanto, "Pengujian Black Box Menggunakan Teknik Equivalence Partitions Pada Aplikasi E-Learning Berbasis Web," vol. 3, no. 3, pp. 269–274, 2021, doi: 10.47065/bits.v3i3.1062.
 - [18] A. Firdhayanti, T. Taufik, and B. Bachry, "User Acceptance Testing through Blackbox Evaluation for Corn Distribution Information System," vol. 6, no. 2, 2023, doi: 10.32877/bt.v6i2.1065.
 - [19] G. S. Mahendra and I. K. A. Asmarajaya, "Evaluation Using Black Box Testing and System Usability Scale in the Kidung Sekar Madya Application," *Sinkron*, vol. 7, no. 4, pp. 2292–2302, 2022, doi: 10.33395/sinkron.v7i4.11755.
 - [20] Siti hidayatul Khoirun Nisa' and R. Y. Rusdianto, "Pemanfaatan Visualisasi Data dalam Meningkatkan Pengambilan Keputusan Bisnis," *J. Informasi, Sains dan Teknol.*, vol. 7, no. 2, pp. 200–208, 2024, doi: 10.55606/isaintek.v7i2.290.