

Implementasi *Dashboard* Monitoring dan Evaluasi Pengguna pada Sistem *Marketplace* Berbasis Web

David Wahyu Widodo¹, Emmy Wahyuningtyas²

Email: ¹dww070@mhs.uwks.ac.id, ²emmy@uwks.ac.id

Abstrak

Perkembangan sistem *marketplace* berbasis web mendorong kebutuhan akan penyajian informasi yang lebih terstruktur dan mudah dipahami oleh pengguna. Pada sistem *marketplace* toko bunga, proses monitoring aktivitas pengguna dan evaluasi performa sistem masih sering dilakukan secara manual sehingga informasi terkait aktivitas transaksi, interaksi pengguna, serta performa penjualan sulit dipantau secara menyeluruh. Kondisi tersebut menyebabkan proses evaluasi pengguna menjadi kurang efektif dan penyajian informasi sistem belum optimal.

Penelitian ini bertujuan untuk mengimplementasikan *dashboard* monitoring dan evaluasi pengguna pada sistem *marketplace* toko bunga berbasis web. Sistem dikembangkan menggunakan metode *prototype* dengan arsitektur *client-server* berbasis *REST API*. *Frontend* sistem dibangun menggunakan *React.js*, *backend* menggunakan *Node.js*, dan *PostgreSQL* digunakan sebagai basis data. *Dashboard* dirancang untuk menampilkan informasi aktivitas pengguna, statistik transaksi, data review dan rating, serta visualisasi data dalam bentuk grafik dan tabel interaktif guna mendukung proses monitoring dan evaluasi pengguna secara lebih terstruktur.

Hasil penelitian menunjukkan bahwa *dashboard* yang dikembangkan mampu membantu proses monitoring aktivitas pengguna dan penyajian informasi sistem secara lebih efektif. Informasi yang ditampilkan pada *dashboard* mempermudah proses evaluasi pengguna melalui visualisasi data yang lebih interaktif dan mudah dipahami. Berdasarkan hasil pengujian fungsional menggunakan metode *blackbox testing*, seluruh fitur utama sistem dapat berjalan sesuai dengan kebutuhan pengguna. Dengan demikian, implementasi *dashboard* monitoring pada *marketplace* toko bunga berbasis web diharapkan dapat meningkatkan efektivitas pengelolaan sistem serta mendukung proses evaluasi pengguna secara lebih optimal.

Kata kunci: *Dashboard*, *Marketplace*, Monitoring Pengguna, *React.js*, Sistem Informasi

Abstract

The development of web-based marketplace systems has increased the need for structured and easily accessible information presentation for users. In flower shop marketplace systems, the process of monitoring user activities and evaluating system performance is still often carried out manually, making information related to transaction activities, user interactions, and sales performance difficult to monitor comprehensively. This condition causes the user evaluation process to become less effective and the presentation of system information to be less optimal.

This study aims to implement a monitoring and user evaluation dashboard on a web-based flower shop marketplace system. The system was developed using the prototype method with a REST API-based client-server architecture. The Frontend was developed using React.js, the backend using Node.js, and PostgreSQL was used as the database management system. The dashboard was designed to display user activity information, transaction statistics, review and rating data, as well as data visualizations in the form of interactive charts and tables to support a more structured monitoring and user evaluation process.

The results of this study indicate that the developed dashboard is able to support the monitoring of user activities and provide system information more effectively. The information displayed on the dashboard simplifies the user evaluation process through interactive and easy-to-understand data visualizations. Based on functional testing using the blackbox testing method, all main system features were able to run according to user requirements. Therefore, the implementation of the monitoring dashboard in the web-based flower shop marketplace system is expected to improve system management effectiveness and support a more optimal user evaluation process.

Keywords: *Dashboard*, *Marketplace*, Monitoring User, *React.js*, Information System

I. PENDAHULUAN

Perkembangan teknologi informasi telah mendorong transformasi digital pada berbagai sektor bisnis, termasuk perdagangan berbasis *marketplace*. *Marketplace* berbasis web menjadi salah satu media yang banyak digunakan untuk mendukung proses transaksi secara lebih cepat, praktis, dan efisien. Salah satu bidang usaha yang mulai memanfaatkan teknologi *marketplace* adalah toko bunga. Melalui

marketplace toko bunga, pengguna dapat melakukan pemesanan produk bunga secara daring tanpa harus datang langsung ke toko, sehingga proses penjualan dan pelayanan dapat dilakukan dengan lebih fleksibel.

Dalam proses pengelolaan *marketplace*, penyajian informasi yang terstruktur menjadi aspek penting untuk mendukung aktivitas pengguna dan pengelola sistem. Informasi seperti aktivitas transaksi, data pengguna, performa penjualan, serta interaksi pengguna perlu ditampilkan secara jelas agar dapat membantu proses monitoring dan evaluasi sistem. Namun, pada beberapa *marketplace* berbasis web, proses monitoring aktivitas pengguna masih dilakukan secara manual sehingga pengelolaan informasi menjadi kurang efektif dan sulit dipantau secara menyeluruh[1].

Pada sistem *marketplace* toko bunga, proses evaluasi pengguna dan monitoring aktivitas memiliki peran penting dalam membantu pengelola memahami kondisi sistem secara *real-time*. Monitoring aktivitas pengguna dapat digunakan untuk mengetahui aktivitas transaksi, interaksi pengguna, serta perkembangan penggunaan sistem. Selain itu, evaluasi pengguna juga diperlukan untuk membantu pengelola memahami performa sistem berdasarkan data aktivitas yang tersedia. Tanpa adanya *dashboard* yang terintegrasi, proses penyajian informasi menjadi kurang optimal dan data yang tersedia sulit dianalisis secara cepat.

Dashboard merupakan salah satu komponen dalam sistem informasi yang digunakan untuk menampilkan data dalam bentuk visual yang lebih mudah dipahami. *Dashboard* mampu menyajikan informasi dalam bentuk grafik, tabel, maupun statistik interaktif sehingga mempermudah proses monitoring dan pengambilan keputusan. Dengan adanya *dashboard* monitoring, pengelola sistem dapat memahami kondisi *marketplace* secara lebih terstruktur dan efisien. Selain itu, *dashboard* juga membantu proses evaluasi pengguna melalui penyajian data aktivitas yang lebih informatif dan mudah dianalisis[2].

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk mengimplementasikan *dashboard* monitoring dan evaluasi pengguna pada sistem *marketplace* toko bunga berbasis web. Sistem dikembangkan menggunakan *React.js* sebagai *Frontend*, *Node.js* sebagai *backend*, dan *PostgreSQL* sebagai basis data dengan pendekatan *client-server* berbasis *REST API*. *Dashboard* yang dibangun diharapkan mampu membantu proses monitoring aktivitas pengguna, penyajian informasi sistem, serta mendukung evaluasi pengguna secara lebih efektif dan terstruktur[3].

II. METODE PENELITIAN

Metode penelitian merupakan tahapan yang digunakan untuk mendukung proses pengembangan sistem agar berjalan secara terstruktur dan sesuai dengan tujuan penelitian. Pada penelitian ini, metode penelitian difokuskan pada pengembangan *dashboard* monitoring dan evaluasi pengguna pada sistem *marketplace* toko bunga berbasis web. Proses penelitian dilakukan mulai dari tahap identifikasi kebutuhan sistem hingga tahap pengujian sistem untuk memastikan seluruh fitur dapat berjalan dengan baik.

Penelitian ini menggunakan pendekatan *prototype* karena mampu mendukung proses pengembangan sistem secara bertahap dan fleksibel. Pendekatan *prototype* memungkinkan proses evaluasi dan perbaikan sistem dilakukan secara berulang sehingga fitur *dashboard* yang dikembangkan dapat disesuaikan dengan kebutuhan pengguna. Selain itu, metode ini juga memudahkan proses pengembangan antarmuka *dashboard* yang bersifat interaktif dan dinamis.

Sistem dikembangkan menggunakan arsitektur *client-server* berbasis *REST API*. *Frontend* sistem dibangun menggunakan *React.js* untuk menghasilkan tampilan *dashboard* yang responsif, sedangkan *backend* dikembangkan menggunakan *Node.js* untuk mengelola proses bisnis dan komunikasi data. *PostgreSQL* digunakan sebagai basis data untuk menyimpan data pengguna, transaksi, review, rating, dan aktivitas sistem secara terstruktur[4].

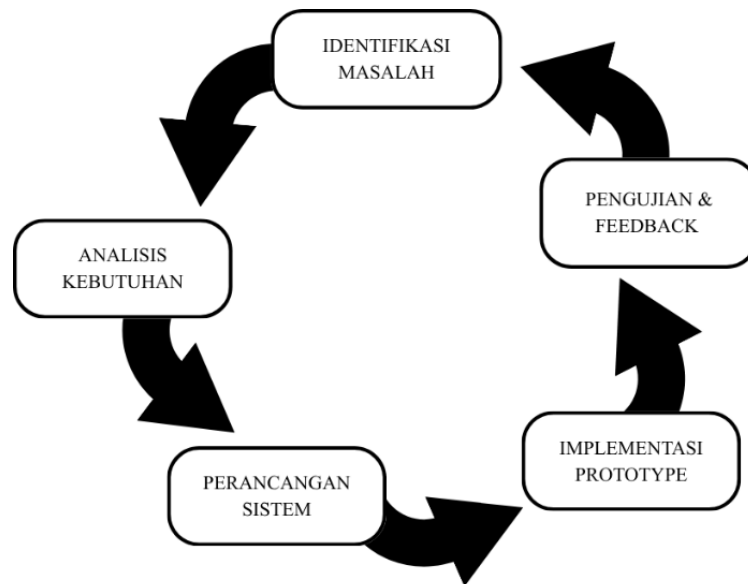
2.1. Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan pada penelitian ini adalah metode *prototype*. Metode *prototype* dipilih karena mampu mendukung proses pengembangan sistem secara fleksibel melalui tahapan evaluasi dan penyempurnaan yang dilakukan secara bertahap. Dengan pendekatan ini, sistem yang dikembangkan dapat disesuaikan dengan kebutuhan pengguna selama proses pengembangan berlangsung.

Tahapan metode *prototype* dimulai dari proses identifikasi kebutuhan sistem, dilanjutkan dengan proses perancangan awal sistem, implementasi *prototype*, evaluasi sistem, dan penyempurnaan fitur. Setiap tahapan dilakukan secara berulang hingga sistem yang dikembangkan sesuai dengan kebutuhan monitoring dan evaluasi pengguna pada *marketplace* berbasis web. Pendekatan ini mempermudah proses pengembangan *dashboard* karena perubahan sistem dapat dilakukan secara lebih cepat dan terstruktur.

Model *prototype* pada penelitian ini digunakan untuk membantu proses pengembangan *dashboard* monitoring yang mampu menampilkan informasi aktivitas pengguna secara interaktif dan *real-time*.

Dengan metode ini, proses pengembangan sistem dapat dilakukan secara lebih efektif sehingga *dashboard* yang dihasilkan mampu mendukung proses monitoring dan evaluasi pengguna secara optimal seperti pada gambar 1.



Gambar 1. Metode Prototype

2.2. Analisis Kebutuhan Sistem

Tahap analisis kebutuhan sistem dilakukan untuk menentukan kebutuhan fungsional dan nonfungsional pada *dashboard* monitoring dan evaluasi pengguna. Analisis kebutuhan dilakukan dengan mengidentifikasi informasi yang diperlukan untuk mendukung proses monitoring aktivitas pengguna dan pengelolaan sistem *marketplace* toko bunga berbasis web.

Kebutuhan fungsional sistem meliputi pengelolaan data pengguna, transaksi, review, rating, dan aktivitas sistem yang akan ditampilkan pada *dashboard*. Selain itu, sistem juga dirancang untuk mampu menampilkan visualisasi data dalam bentuk grafik, tabel, dan statistik interaktif agar informasi dapat dipahami dengan lebih mudah oleh pengguna maupun pengelola sistem.

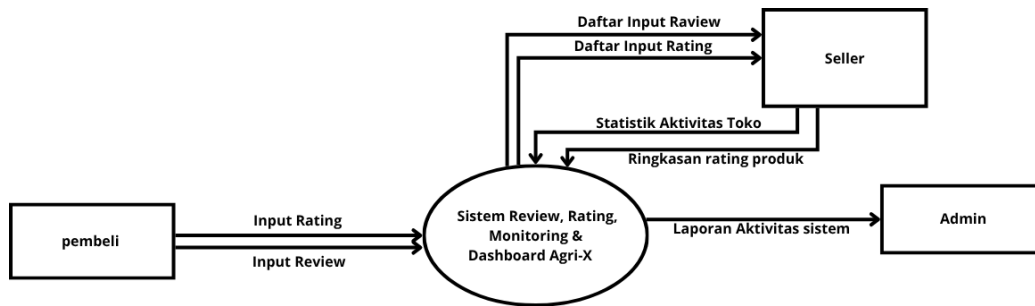
Kebutuhan nonfungsional sistem meliputi kemampuan sistem dalam menampilkan data secara *real-time*, kemudahan penggunaan antarmuka *dashboard*, serta integrasi antara *Frontend* dan *backend* melalui *REST API*. Hasil dari tahap analisis kebutuhan ini digunakan sebagai dasar dalam proses perancangan dan implementasi *dashboard* monitoring pada sistem *marketplace* berbasis web[5].

2.3. Perancangan Sistem

Tahap perancangan sistem dilakukan untuk menentukan struktur sistem dan alur kerja *dashboard* sebelum proses implementasi dilakukan. Pada penelitian ini, sistem dirancang menggunakan arsitektur *client-server* berbasis *REST API* untuk mendukung komunikasi data antara *Frontend* dan *backend* secara terintegrasi.

Frontend sistem dibangun menggunakan *React.js* untuk menghasilkan tampilan *dashboard* yang dinamis dan responsif, sedangkan *backend* dikembangkan menggunakan *Node.js* untuk mengelola proses bisnis dan pengolahan data sistem. *PostgreSQL* digunakan sebagai basis data utama untuk menyimpan data pengguna, transaksi, review, rating, dan aktivitas sistem secara terstruktur. *Dashboard* dirancang untuk mampu menampilkan informasi monitoring pengguna dalam bentuk visualisasi data interaktif.

Selain perancangan arsitektur sistem, pada tahap ini juga dilakukan perancangan alur data menggunakan *DFD Level 0*. Diagram tersebut digunakan untuk menggambarkan hubungan antara pengguna dengan sistem serta proses pengolahan data yang terjadi pada *dashboard* monitoring dan evaluasi pengguna. *DFD Level 0* membantu memberikan gambaran umum mengenai proses *input*, pengolahan, dan *output* data pada sistem *marketplace* berbasis web seperti pada gambar 2.



Gambar 2. DFD Level 0

2.4. Implementasi Sistem

Tahap implementasi sistem dilakukan setelah proses perancangan selesai dilakukan. Pada tahap ini dilakukan pembangunan *dashboard* monitoring dan evaluasi pengguna berbasis web menggunakan *React.js* sebagai *Frontend* dan *Node.js* sebagai *backend*. Sistem kemudian diintegrasikan dengan *PostgreSQL* sebagai basis data utama melalui *REST API*.

Dashboard yang dikembangkan mampu menampilkan informasi aktivitas pengguna secara *real-time* dalam bentuk grafik, tabel, dan statistik interaktif. Informasi yang ditampilkan meliputi data transaksi, aktivitas pengguna, review, rating, serta statistik penggunaan sistem yang digunakan untuk mendukung proses monitoring dan evaluasi pengguna pada *marketplace* toko bunga berbasis web.

Selain implementasi *dashboard*, pada tahap ini juga dilakukan proses integrasi antara *Frontend* dan *backend* untuk memastikan proses pengambilan dan pengolahan data dapat berjalan dengan baik. Hasil implementasi sistem kemudian digunakan sebagai dasar pada tahap pengujian untuk memastikan seluruh fitur sistem dapat berjalan sesuai kebutuhan pengguna.

2.5. Pengujian Sistem

Tahap pengujian sistem dilakukan untuk memastikan seluruh fitur yang dikembangkan dapat berjalan sesuai dengan kebutuhan pengguna. Pengujian dilakukan menggunakan metode *blackbox testing*, yaitu pengujian yang berfokus pada fungsi sistem tanpa melihat struktur kode program secara langsung.

Proses pengujian meliputi pengujian tampilan *dashboard*, pengambilan data dari basis data, proses visualisasi data, monitoring aktivitas pengguna, serta integrasi *Frontend* dan *backend*. Setiap fitur diuji berdasarkan *input* dan *output* yang dihasilkan untuk memastikan sistem dapat berjalan secara optimal dan sesuai dengan rancangan yang telah dibuat.

Hasil pengujian digunakan untuk mengetahui tingkat keberhasilan implementasi *dashboard* monitoring dan evaluasi pengguna pada *marketplace* berbasis web. Dengan dilakukannya proses pengujian, sistem yang dikembangkan diharapkan mampu mendukung proses monitoring aktivitas pengguna dan pengelolaan informasi sistem secara lebih efektif dan terstruktur.

III. HASIL DAN PEMBAHASAN

Hasil dan pembahasan merupakan tahapan yang dilakukan untuk menjelaskan hasil implementasi *dashboard* monitoring dan evaluasi pengguna pada *marketplace* toko bunga berbasis web. Pada tahap ini dilakukan pembahasan mengenai implementasi antarmuka sistem, *dashboard* monitoring, integrasi basis data, serta pengujian sistem yang telah dikembangkan. Sistem dibangun menggunakan *React.js* sebagai *Frontend*, *Node.js* sebagai *backend*, dan *Supabase PostgreSQL* sebagai basis data utama[6].

Dashboard dikembangkan untuk membantu proses monitoring aktivitas pengguna dan penyajian informasi sistem secara lebih terstruktur. Informasi yang ditampilkan pada *dashboard* meliputi data transaksi, aktivitas pengguna, review, rating, dan statistik penggunaan sistem dalam bentuk visualisasi data interaktif. Implementasi sistem dilakukan menggunakan arsitektur *client-server* berbasis *REST API* sehingga proses pengambilan data dapat dilakukan secara *real-time*.

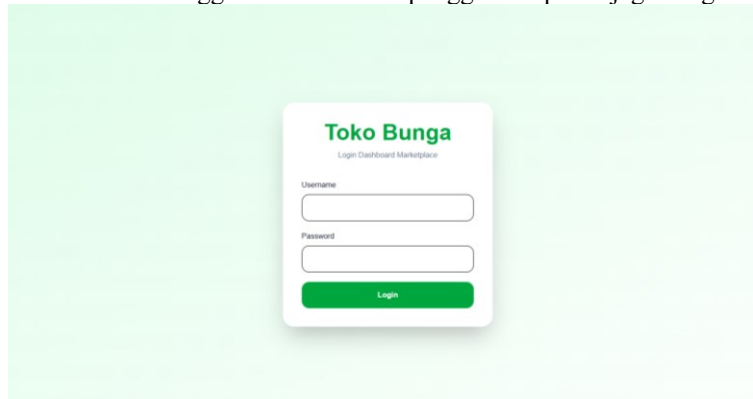
Pada bab ini juga dilakukan pembahasan mengenai hasil implementasi antarmuka *login*, *dashboard admin*, *dashboard Seller*, serta struktur basis data yang digunakan pada sistem. Selain itu, dilakukan pengujian sistem menggunakan metode *blackbox testing* untuk memastikan seluruh fitur *dashboard* dapat berjalan sesuai dengan kebutuhan pengguna.

3.1. Implementasi Halaman Login

Halaman *login* merupakan halaman awal yang digunakan pengguna untuk mengakses sistem *dashboard marketplace* berbasis web. Halaman ini berfungsi untuk melakukan proses autentikasi pengguna sebelum masuk ke dalam sistem *dashboard* monitoring dan evaluasi pengguna.

Implementasi halaman *login* dilakukan menggunakan *React.js* sebagai *Frontend* dan *Node.js* sebagai *backend* yang terhubung dengan sistem autentikasi pada *Supabase PostgreSQL*. Pengguna diharuskan memasukkan email dan password untuk melakukan proses *login* ke dalam sistem. Setelah proses autentikasi berhasil dilakukan, pengguna akan diarahkan menuju halaman *dashboard* sesuai hak akses yang dimiliki.

Antarmuka halaman *login* dirancang dengan tampilan sederhana dan responsif untuk mempermudah pengguna dalam mengakses sistem. Selain itu, proses autentikasi dilakukan secara terintegrasi dengan basis data sehingga keamanan data pengguna dapat terjaga dengan baik[7].



Gambar 3. Halaman Login Sistem

3.2. Implementasi Dashboard Seller

Dashboard Seller digunakan untuk menampilkan informasi aktivitas dan performa pengguna pada *marketplace* toko bunga berbasis web. *Dashboard* ini menampilkan berbagai informasi seperti jumlah transaksi, data review dan rating, statistik penjualan, serta aktivitas pengguna dalam bentuk visualisasi data interaktif.

Implementasi *dashboard Seller* dilakukan menggunakan *React.js* untuk menghasilkan tampilan antarmuka yang dinamis dan responsif. Data yang ditampilkan pada *dashboard* diambil dari *Supabase PostgreSQL* melalui *REST API* yang dikembangkan menggunakan *Node.js*. Dengan pendekatan tersebut, informasi pada *dashboard* dapat diperbarui secara *real-time* sesuai dengan aktivitas pengguna pada sistem *marketplace*[8].

Dashboard Seller dirancang untuk membantu pengguna dalam memahami performa penggunaan sistem melalui tampilan data yang lebih terstruktur dan mudah dipahami. Visualisasi data pada *dashboard* juga membantu proses monitoring aktivitas pengguna sehingga informasi sistem dapat dianalisis dengan lebih efektif.



Gambar 5. Dashboard Seller

3.3. Implementasi Dashboard Admin

Dashboard admin digunakan untuk membantu proses monitoring aktivitas pengguna dan pengelolaan sistem *marketplace* secara menyeluruh. *Dashboard* ini menampilkan informasi aktivitas pengguna, statistik transaksi, data review dan rating, serta ringkasan penggunaan sistem dalam bentuk grafik dan tabel interaktif.

Implementasi *dashboard admin* dilakukan menggunakan *React.js* sebagai *Frontend* dan *Node.js* sebagai *backend* yang terhubung dengan *Supabase PostgreSQL*. Informasi yang ditampilkan pada *dashboard* diambil secara *real-time* melalui *REST API* sehingga *admin* dapat memantau kondisi sistem secara lebih cepat dan terstruktur.

Dashboard admin dirancang untuk mendukung proses monitoring sistem dan evaluasi pengguna melalui penyajian data aktivitas yang lebih informatif. Dengan adanya *dashboard admin*, proses pengawasan sistem *marketplace* dapat dilakukan secara lebih efektif dan efisien.

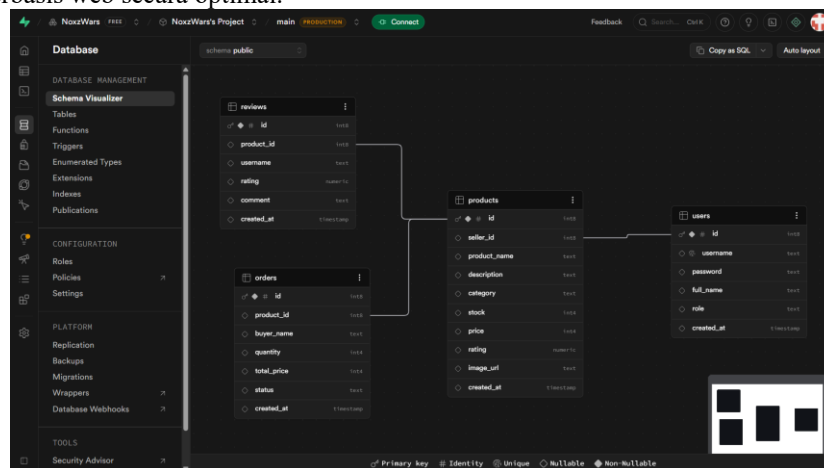


Gambar 4. Dashboard Admin

3.4. Implementasi Basis Data Supabase PostgreSQL

Basis data pada sistem menggunakan *Supabase PostgreSQL* sebagai media penyimpanan data pengguna, transaksi, review, rating, dan aktivitas sistem. Penggunaan *Supabase PostgreSQL* mendukung proses pengelolaan data secara *cloud-based* sehingga data dapat diakses dan dikelola secara *real-time*[9].

Struktur basis data dirancang untuk mendukung kebutuhan *dashboard* monitoring dan evaluasi pengguna pada *marketplace* toko bunga berbasis web. Data yang tersimpan pada basis data kemudian diproses oleh *backend Node.js* sebelum ditampilkan pada *dashboard* menggunakan *React.js* melalui *REST API*. Implementasi basis data dilakukan dengan memperhatikan hubungan antar tabel agar proses pengambilan data dapat dilakukan secara lebih efisien. Selain itu, *Supabase* juga mendukung proses autentikasi pengguna dan integrasi sistem secara lebih mudah sehingga mendukung pengembangan *dashboard* berbasis web secara optimal.



Gambar 6. Struktur Database Supabase PostgreSQL

3.5. Pengujian Sistem

Pengujian sistem dilakukan menggunakan metode *blackbox testing* untuk memastikan seluruh fitur pada *dashboard* monitoring dan evaluasi pengguna dapat berjalan sesuai dengan kebutuhan sistem. Metode *blackbox testing* digunakan karena pengujian difokuskan pada fungsi sistem berdasarkan *input* dan *output* yang dihasilkan tanpa melihat struktur kode program secara langsung.

Proses pengujian dilakukan pada beberapa fitur utama sistem seperti halaman *login*, *dashboard Seller*, *dashboard admin*, pengambilan data dari *Supabase PostgreSQL*, visualisasi data monitoring, serta integrasi *Frontend* dan *backend* melalui *REST API*. Pengujian dilakukan untuk memastikan setiap fitur mampu berjalan dengan baik dan data dapat ditampilkan secara benar pada *dashboard* monitoring berbasis web[10].

Hasil pengujian menunjukkan bahwa seluruh fitur utama sistem dapat berjalan sesuai dengan kebutuhan pengguna. *Dashboard* mampu menampilkan informasi aktivitas pengguna, transaksi, review, dan rating secara *real-time* sehingga proses monitoring dan evaluasi pengguna dapat dilakukan dengan lebih efektif dan terstruktur.

Tabel 1. Pengujian *Black-box*

No	Skenario Pengujian	Input	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Pengujian <i>login</i> pengguna	Email dan password benar	Sistem berhasil masuk ke <i>dashboard</i>	<i>Dashboard</i> berhasil ditampilkan	Berhasil
2	Pengujian <i>login</i> gagal	Email atau password salah	Sistem menampilkan pesan kesalahan <i>login</i>	Pesan error berhasil ditampilkan	Berhasil
3	Menampilkan <i>dashboard Seller</i>	Pengguna memilih menu <i>dashboard Seller</i>	Sistem menampilkan data <i>dashboard Seller</i>	Data <i>dashboard Seller</i> berhasil tampil	Berhasil
4	Menampilkan <i>dashboard admin</i>	Pengguna memilih menu <i>dashboard admin</i>	Sistem menampilkan data <i>dashboard admin</i>	Data <i>dashboard admin</i> berhasil tampil	Berhasil
5	Menampilkan data transaksi	Sistem mengambil data transaksi dari database	Data transaksi tampil pada <i>dashboard</i>	Data transaksi berhasil tampil	Berhasil
6	Menampilkan data review dan rating	Sistem mengambil data review dan rating	Review dan rating tampil pada <i>dashboard</i>	Data review dan rating berhasil tampil	Berhasil
7	Menampilkan grafik monitoring	Sistem memproses data aktivitas pengguna	Grafik monitoring tampil dengan benar	Grafik berhasil ditampilkan	Berhasil
8	Pengambilan data dari <i>Supabase PostgreSQL</i>	Request data dari database	Sistem berhasil mengambil data	Data berhasil ditampilkan pada <i>dashboard</i>	Berhasil
9	Integrasi <i>Frontend</i> dan <i>backend</i>	Request API dari <i>Frontend</i>	<i>Backend</i> merespon data dengan benar	Data API berhasil diproses	Berhasil
10	Logout pengguna	Pengguna menekan tombol logout	Sistem keluar dari <i>dashboard</i>	Pengguna berhasil logout	Berhasil

Pengujian sistem dilakukan menggunakan metode *blackbox testing* untuk memastikan seluruh fitur pada *dashboard* monitoring dan evaluasi pengguna dapat berjalan sesuai dengan kebutuhan sistem. Metode *blackbox testing* digunakan karena pengujian difokuskan pada fungsi sistem berdasarkan *input* dan *output* yang dihasilkan tanpa melihat struktur kode program secara langsung.

Proses pengujian dilakukan pada beberapa fitur utama sistem seperti halaman *login*, *dashboard Seller*, *dashboard admin*, pengambilan data dari *Supabase PostgreSQL*, visualisasi data monitoring, serta integrasi *Frontend* dan *backend* melalui *REST API*. Pengujian dilakukan untuk memastikan setiap fitur mampu berjalan dengan baik dan data dapat ditampilkan secara benar pada *dashboard* monitoring berbasis web. Hasil pengujian menunjukkan bahwa seluruh fitur utama sistem dapat berjalan sesuai dengan kebutuhan pengguna. *Dashboard* mampu menampilkan informasi aktivitas pengguna, transaksi, review, dan rating secara *real-time* sehingga proses monitoring dan evaluasi pengguna dapat dilakukan dengan lebih efektif dan terstruktur.

IV. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, *dashboard* monitoring dan evaluasi pengguna pada sistem *marketplace* toko bunga berbasis web berhasil diimplementasikan menggunakan *React.js* sebagai *Frontend*, *Node.js* sebagai *backend*, dan *Supabase PostgreSQL* sebagai basis data utama. Sistem dikembangkan menggunakan metode *prototype* dengan arsitektur *client-server* berbasis *REST API* sehingga proses pengelolaan dan penyajian data dapat dilakukan secara lebih terstruktur dan *real-time*.

Dashboard yang dikembangkan mampu menampilkan informasi aktivitas pengguna, data transaksi, review, rating, serta statistik penggunaan sistem dalam bentuk visualisasi data interaktif. Implementasi *dashboard* membantu proses monitoring aktivitas pengguna dan penyajian informasi sistem menjadi lebih efektif serta mempermudah proses evaluasi pengguna pada *marketplace* berbasis web. Selain itu, penggunaan *Supabase PostgreSQL* mendukung proses pengelolaan data secara *cloud-based* sehingga integrasi sistem dapat berjalan dengan baik.

Berdasarkan hasil pengujian menggunakan metode *blackbox testing*, seluruh fitur utama sistem dapat berjalan sesuai dengan kebutuhan pengguna. Sistem mampu menampilkan data *dashboard* secara stabil dan mendukung proses monitoring serta evaluasi pengguna secara lebih optimal. Dengan demikian, implementasi *dashboard* monitoring dan evaluasi pengguna pada *marketplace* toko bunga berbasis web diharapkan dapat membantu proses pengelolaan informasi sistem dan meningkatkan efektivitas monitoring aktivitas pengguna.

REFERENSI

- [1] Raya Adinda Jayadi Ahmad And Saprudin, "Rancang Bangun Aplikasi Dashboard Monitoring Stok Barang Berbasis Web Menggunakan Metode Waterfall Pada Gudang Produksi Cv Kurnia Jaya Industri," *Jurnal Penelitian Terapan Ilmu Multidisiplin*, Jul. 2025, Accessed: May 12, 2026. [Online]. Available: <https://jurnal.portalpublikasi.id/index.php/jptim/article/view/1773/1436>
- [2] M. Zhuang, D. Concannon, And E. Manley, "A Framework For Evaluating Dashboards In Healthcare," Feb. 2022, Doi: 10.1109/Tvcg.2022.3147154.
- [3] Niraj Ramnani, Dr. Vishal Shrivastava, And Dr. Akhil Pandey, "Enrichment In Web Development Using Mern Stack," Nov. 2025, Accessed: May 12, 2026. [Online]. Available: <https://www.ijrpa.com/myimgup/246300373ijrpa%20550.pdf>
- [4] U. Thorat, U. Mudgade, P. Bhende, And K. Patil, "Campus Mart-Student Marketplace," 2026. [Online]. Available: www.ijrti.org
- [5] F. P. E. Putra, R. W. Efendi, A. B. Tamam, And W. A. Pramadi, "Trends And Best Practices In Api-Based Web Development Using Laravel And React," *Brilliance: Research Of Artificial Intelligence*, Vol. 5, No. 1, Pp. 223–233, Jun. 2025, Doi: 10.47709/Brilliance.V5i1.5971.
- [6] F. Akmal, "Penerapan Reactjs Dalam Pengembangan Front-End Aplikasi Komunitas Zeromaterial," *Journal Of Digital Business And Technology Innovation (Dbesti)*, Vol. 2, No. 2, Pp. 198–205, 2025, [Online]. Available: <https://journal.nurulfikri.ac.id/index.php/dbesti>
- [7] R. Makhfuddin Akbar, S. Zahara, Y. Nur Sukmaningtyas, A. Syaifuddin, And H. Fianto Putra, "Implementasi React Js Pada Marketplace Desa Kalikatur Mojokerto," *Jurnal Informatika Dan Teknik Elektro Terapan*, Vol. 14, No. 1, Jan. 2026, Doi: 10.23960/Jitet.V14i1.8887.

-
- [8] S. Banerjee, C. E. Fullerton, S. S. Gaharwar, And E. J. Jaselskis, “Strategic Web-Based Data Dashboards As Monitoring Tools For Promoting Organizational Innovation,” *Buildings*, Vol. 15, No. 13, Jul. 2025, Doi: 10.3390/Buildings15132204.
 - [9] S. Sauda And M. Barokah, “Penerapan Nodejs Dan Postgresql Sebagai Backend Pada Aplikasi Ecommerce Localla”, Doi: 10.31949/Infotech.V8i2.2944.
 - [10] D. Gusdianto And A. M. Thantawi, “Rancang Bangun Dashboard Berbasis Website Untuk Monitoring Penjualan (Studi Kasus: Toko Kopi Gans Jakarta Timur)”, Doi: 10.37817/Ikraith-Informatika.V8i3.