



Analisis Implementasi Dashboard Monitoring Produktivitas Developer Menggunakan Integrasi GitLab API dan n8n Berbasis Web dengan Metode Prototype

Ulung Wira Yuda¹, Wydyanto²

Fakultas Sain Teknologi Universitas Bina Darma^{1,2}

ulungwirayuda4@gmail.com¹, wydyanto@binadarma.ac.id²

Abstract. *The rapid development of information technology has encouraged organizations to implement monitoring systems capable of providing fast, accurate, and integrated information. In software development processes, developer activities recorded in GitLab repositories can be utilized to measure team productivity and monitor project progress continuously. However, monitoring activities directly through GitLab repositories is often less effective because information is not presented in a centralized and real-time manner. This study aims to analyze the implementation of a web-based developer productivity monitoring dashboard using GitLab API and n8n integration with the Prototype method. The Prototype method consists of communication, quick design, build prototype, user evaluation, and refinement prototype stages to ensure the system meets user requirements. GitLab API serves as the primary data source using Personal Access Token authentication, while n8n is utilized as a workflow automation platform to process repository and commit activity data automatically. The dashboard is developed using HTML, CSS, and JavaScript and is integrated with webhooks to enable real-time data synchronization. The results indicate that the system successfully integrates GitLab API with the monitoring dashboard, enabling the display of repository information, commit activities, developer productivity metrics, data visualization charts, and developer rankings based on commit contributions. The system also provides automatic notifications through Telegram, WhatsApp, and email whenever new repository activities occur. Testing results demonstrate that all system features function according to user requirements and improve the effectiveness of monitoring software development activities in a faster, more structured, and real-time manner.*

Keywords: *Monitoring Dashboard, GitLab API, n8n, Developer Productivity, Prototype Method.*

Abstrak. Perkembangan teknologi informasi mendorong organisasi untuk menerapkan sistem monitoring yang mampu menyajikan informasi secara cepat, akurat, dan terintegrasi. Dalam proses pengembangan perangkat lunak, aktivitas developer yang tercatat pada repository GitLab dapat dimanfaatkan untuk mengukur produktivitas tim pengembang serta memantau perkembangan proyek secara berkelanjutan. Namun, proses monitoring yang dilakukan secara langsung melalui repository GitLab sering kali kurang efektif karena informasi belum tersaji secara terpusat dan sulit dianalisis secara real-time. Penelitian ini bertujuan untuk menganalisis implementasi dashboard monitoring produktivitas developer menggunakan integrasi GitLab API dan n8n berbasis web dengan metode Prototype. Metode Prototype digunakan melalui tahapan communication, quick design, build prototype, user evaluation, dan refinement prototype untuk menghasilkan sistem yang sesuai dengan kebutuhan pengguna. GitLab

API dimanfaatkan sebagai sumber data utama dengan autentikasi menggunakan Personal Access Token, sedangkan n8n digunakan sebagai workflow automation untuk mengolah data aktivitas repository dan commit secara otomatis. Dashboard dikembangkan menggunakan teknologi web berupa HTML, CSS, dan JavaScript serta dilengkapi dengan mekanisme webhook untuk sinkronisasi data secara real-time. Hasil penelitian menunjukkan bahwa sistem berhasil mengintegrasikan GitLab API dengan dashboard monitoring sehingga mampu menampilkan informasi repository, aktivitas commit, produktivitas developer, grafik visualisasi data, serta peringkat developer berdasarkan jumlah commit. Sistem juga berhasil mengimplementasikan notifikasi otomatis melalui Telegram, WhatsApp, dan email ketika terjadi aktivitas baru pada repository. Berdasarkan hasil pengujian, seluruh fitur sistem berjalan sesuai dengan kebutuhan pengguna dan mampu meningkatkan efektivitas monitoring aktivitas pengembangan perangkat lunak secara lebih cepat, terstruktur, dan real-time.

Kata Kunci: Dashboard Monitoring, GitLab API, n8n, Produktivitas Developer, Prototype.

1. PENDAHULUAN

Perkembangan teknologi informasi telah mendorong organisasi untuk menerapkan berbagai sistem digital guna meningkatkan efektivitas dan efisiensi proses bisnis. Dalam bidang pengembangan perangkat lunak, penggunaan platform manajemen repository menjadi salah satu kebutuhan utama untuk mendukung kolaborasi antar developer. Salah satu platform yang banyak digunakan adalah GitLab yang menyediakan layanan pengelolaan source code, version control, issue tracking, serta pencatatan aktivitas pengembangan perangkat lunak secara terintegrasi. Melalui GitLab, aktivitas developer seperti commit, branch, merge request, dan kontribusi repository dapat terdokumentasi dengan baik sehingga memudahkan proses pengelolaan proyek perangkat lunak. Dalam proses pengembangan perangkat lunak, GitLab digunakan sebagai media pengelolaan repository dan kolaborasi tim pengembang. Namun, proses monitoring aktivitas developer masih dilakukan dengan mengakses repository GitLab secara langsung sehingga informasi mengenai produktivitas developer, aktivitas commit, serta perkembangan repository belum tersaji secara terpusat dan mudah dipahami. Kondisi tersebut menyebabkan proses monitoring membutuhkan waktu yang lebih lama dan menyulitkan dalam memperoleh informasi secara real-time.

Beberapa penelitian terdahulu menunjukkan bahwa dashboard monitoring mampu membantu proses penyajian informasi secara visual sehingga memudahkan pengguna dalam melakukan pemantauan dan pengambilan keputusan. Penelitian yang dilakukan oleh Kurniawan dan Pratama (2022) menunjukkan bahwa dashboard monitoring berbasis web dapat meningkatkan efektivitas penyajian data secara real-time. Penelitian lain oleh Rahman dan Setiawan (2023) menjelaskan bahwa integrasi Application Programming Interface (API) dan sistem notifikasi otomatis mampu meningkatkan kecepatan distribusi informasi pada sistem monitoring. Selain itu, workflow automation menggunakan n8n telah banyak digunakan untuk menghubungkan berbagai layanan digital secara otomatis dan efisien. Meskipun demikian, sebagian besar penelitian sebelumnya masih berfokus pada monitoring data operasional, akademik, maupun penjualan. Penelitian yang mengintegrasikan

GitLab API, workflow automation menggunakan n8n, visualisasi data produktivitas developer, serta mekanisme notifikasi real-time dalam satu dashboard monitoring masih relatif terbatas. Selain itu, belum banyak penelitian yang menerapkan integrasi webhook untuk melakukan sinkronisasi data aktivitas developer secara otomatis ketika terjadi perubahan pada repository GitLab. Berdasarkan permasalahan tersebut, penelitian ini menawarkan sebuah dashboard monitoring produktivitas developer berbasis web yang mengintegrasikan GitLab API, n8n, dan webhook real-time.

Sistem yang dikembangkan mampu mengumpulkan data aktivitas developer secara otomatis menggunakan Personal Access Token (PAT), melakukan analisis produktivitas berdasarkan aktivitas commit, menampilkan visualisasi data dalam bentuk grafik dan tabel, serta mengirimkan notifikasi otomatis melalui Telegram, WhatsApp, dan email ketika terjadi aktivitas baru pada repository.

Tujuan penelitian ini adalah menganalisis implementasi dashboard monitoring produktivitas developer menggunakan integrasi GitLab API dan n8n berbasis web dengan metode Prototype. Sistem yang dikembangkan diharapkan mampu membantu proses monitoring aktivitas pengembangan perangkat lunak secara lebih efektif, terstruktur, dan real-time sehingga dapat mendukung proses evaluasi produktivitas developer dan pengambilan keputusan dalam pengelolaan proyek perangkat lunak.

2. KAJIAN LITERATUR

2.1 Dashboard Monitoring

Dashboard monitoring merupakan media visual yang digunakan untuk menampilkan informasi penting secara ringkas dan terintegrasi sehingga memudahkan pengguna dalam melakukan pemantauan dan pengambilan keputusan. Dashboard dapat menyajikan data dalam bentuk grafik, tabel, indikator, maupun statistik yang diperbarui secara berkala. Menurut Few (2021), dashboard berfungsi sebagai sarana penyajian informasi yang memungkinkan pengguna memperoleh gambaran kondisi sistem secara cepat dan akurat.

2.2 Produktivitas Developer

Produktivitas developer merupakan ukuran yang digunakan untuk menilai tingkat kontribusi pengembang perangkat lunak dalam suatu proyek. Menurut Forsgren, Humble dan Kim (2021), produktivitas tim pengembang dapat diukur melalui berbagai indikator seperti aktivitas commit, deployment frequency, lead time, dan kualitas perangkat lunak yang dihasilkan. Pengukuran produktivitas membantu organisasi dalam mengevaluasi kinerja tim pengembang secara objektif.

2.3 Application Programming Interface (API)

Application Programming Interface (API) merupakan sekumpulan aturan dan protokol yang memungkinkan suatu aplikasi berkomunikasi dengan aplikasi lain secara otomatis. Menurut Richardson dan Amundsen (2022), API berperan sebagai penghubung antar sistem yang memungkinkan pertukaran data dilakukan secara efisien tanpa memerlukan interaksi langsung dari pengguna.

2.4 GitLab API

GitLab API merupakan layanan REST API yang disediakan oleh GitLab untuk mengakses data repository, commit, branch, merge request, dan contributor secara otomatis. Menurut GitLab Inc. (2025), GitLab API memungkinkan integrasi dengan berbagai aplikasi eksternal sehingga data aktivitas pengembangan perangkat lunak

dapat dimanfaatkan untuk kebutuhan monitoring dan analisis.

2.5 Workflow Automation Menggunakan n8n

Workflow automation merupakan pendekatan yang digunakan untuk mengotomatisasi proses bisnis sehingga aktivitas tertentu dapat berjalan tanpa intervensi manual. Menurut Putra dan Wijaya (2024), n8n merupakan platform workflow automation open-source yang mendukung integrasi berbagai layanan melalui node-node otomatisasi yang fleksibel dan mudah dikembangkan.

2.6 Sistem Notifikasi Real-Time

Sistem notifikasi real-time merupakan mekanisme penyampaian informasi secara langsung ketika terjadi suatu aktivitas atau perubahan data dalam sistem. Menurut Saputra, Pratama dan Wijayanti (2023), sistem notifikasi real-time dapat meningkatkan efektivitas komunikasi dan mempercepat proses penyampaian informasi kepada pengguna.

2.7 Metode Prototype

Metode Prototype merupakan metode pengembangan perangkat lunak yang dilakukan melalui proses pembuatan model awal sistem sebelum dikembangkan menjadi sistem final. Menurut Pressman dan Maxim (2020), metode Prototype terdiri atas tahap communication, quick design, build prototype, user evaluation, dan refinement prototype.

2.8 Monitoring Aktivitas Repository dan Commit

Monitoring aktivitas repository merupakan proses pemantauan terhadap aktivitas yang terjadi pada repository perangkat lunak, seperti commit, perubahan branch, dan kontribusi developer. Menurut Kerzner (2022), proses monitoring yang dilakukan secara berkelanjutan dapat membantu organisasi dalam mengendalikan risiko proyek serta meningkatkan produktivitas tim pengembang.

2.9 Kerangka Pemikiran

Penelitian ini didasarkan pada kebutuhan monitoring produktivitas developer secara real-time. Data aktivitas developer diperoleh melalui GitLab API menggunakan Personal Access Token dan diproses menggunakan workflow automation n8n. Data yang telah diproses kemudian ditampilkan pada dashboard monitoring dalam bentuk grafik, statistik commit, top developer, dan informasi repository. Selain itu, sistem juga mengirimkan notifikasi aktivitas terbaru melalui Telegram, WhatsApp, dan Gmail sehingga pengguna dapat memperoleh informasi secara cepat dan terintegrasi.

METODE PENELITIAN

Penelitian ini menggunakan metode Prototype sebagai metode pengembangan sistem dan studi literatur sebagai metode pengumpulan data. Metode Prototype dipilih karena memungkinkan pengembangan sistem dilakukan secara bertahap melalui proses evaluasi dan penyempurnaan berdasarkan kebutuhan pengguna. Studi literatur dilakukan untuk memperoleh landasan teori mengenai dashboard monitoring, GitLab API, workflow automation menggunakan n8n, sistem notifikasi

real-time, dan metode Prototype.

3.1 Studi Literatur

Studi literatur dilakukan dengan mengumpulkan berbagai referensi yang relevan dari jurnal ilmiah, buku, dokumentasi resmi GitLab API, dokumentasi n8n, serta penelitian terdahulu yang diterbitkan dalam rentang tahun 2021–2026. Tahap ini bertujuan untuk memperoleh pemahaman mengenai konsep monitoring produktivitas developer, dashboard monitoring, integrasi API, workflow automation, dan sistem notifikasi real-time yang digunakan sebagai dasar pengembangan sistem.

3.2 Perangkat dan Teknologi Penelitian

Penelitian ini menggunakan beberapa perangkat dan teknologi untuk mendukung proses pengembangan sistem.

Tabel 1. Perangkat dan Teknologi Penelitian

Komponen	Teknologi
Editor Kode	Visual Studio Code
Frontend	HTML, CSS, JavaScript
Backend	Ngrok, JS
Workflow Automation	n8n
Sumber Data	GitLab API
Autentikasi	Personal Access Token
Notifikasi	WhatsApp, Telegram, Gmail
Browser Pengujian	Google Chrome, Microsoft Edge
Version Control	GitLab
Metode Integrasi	Webhook

Sumber: Dokumentasi Penelitian (2026)

3.4 Tahapan Penelitian

Penerapan metode Prototype dalam penelitian ini dilaksanakan selama delapan minggu pengembangan.

Tabel 2. Tahapan Penelitian

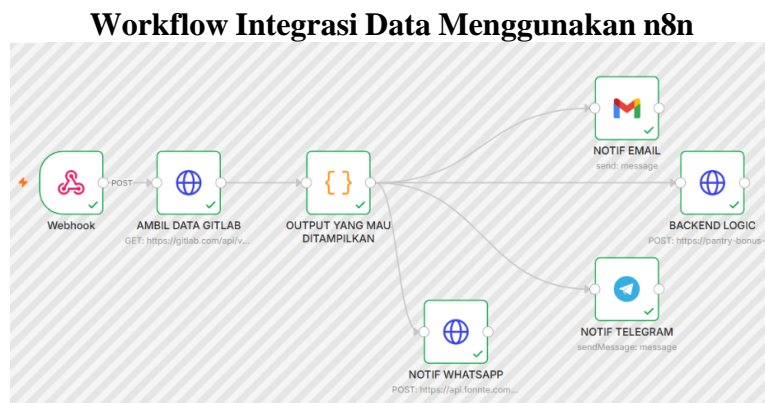
Minggu	Tahapan	Kegiatan
Week 1	Planning & Research	Memahami kebutuhan proyek, mempelajari GitLab API, n8n, dan konsep dashboard monitoring
Week 2	API Integration & Database Design	Integrasi GitLab API dan perancangan struktur data sistem
Week 3	Frontend Design & n8n Workflow	Pengembangan dashboard monitoring dan workflow otomatisasi
Week 4	Metrics & Advanced Data Analysis	Implementasi metrik produktivitas developer dan analisis data commit

Week 5	Real-Time Sync & Backend Logic	Pengembangan sinkronisasi data menggunakan webhook dan backend processing
Week 6	Notification System	Implementasi notifikasi otomatis WhatsApp, Telegram, dan Gmail
Week 7	Testing & Optimization	Pengujian sistem dan optimasi performa dashboard
Week 8	Final Deployment & Documentation	Finalisasi sistem dan penyusunan dokumentasi penelitian

Sumber: Data Dari Pembimbing Mitra (2026)

4. PEMBAHASAN

4.1 Implementasi Workflow n8n

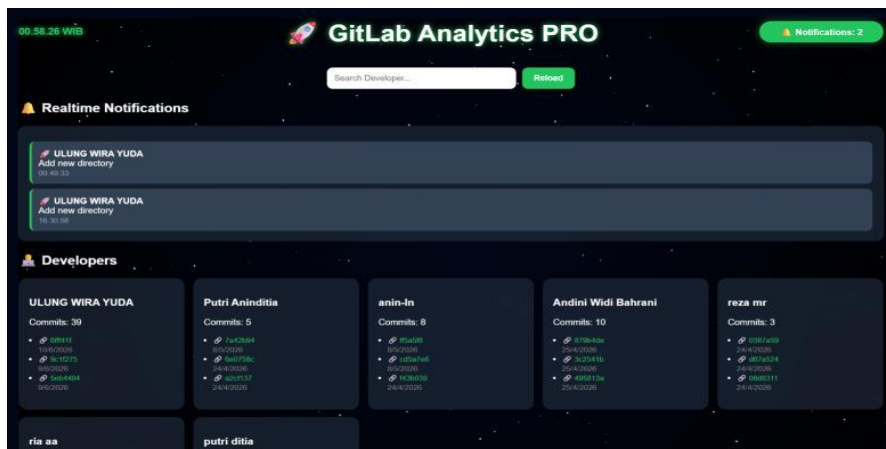


Gambar 1 Workflow Integrasi Data Menggunakan n8n
Sumber: Hasil Implementasi Sistem (2026)

Dalam penelitian ini, workflow n8n digunakan sebagai media integrasi antara GitLab API, dashboard monitoring, dan sistem notifikasi. Workflow dimulai dari proses penerimaan data melalui Webhook Node yang menerima event commit dari GitLab. Selanjutnya data diproses menggunakan HTTP Request Node untuk mengambil informasi repository dan commit terbaru melalui GitLab API. Data yang diperoleh kemudian diproses menggunakan JavaScript Node untuk menghasilkan informasi produktivitas developer, statistik commit, dan aktivitas repository. Selanjutnya data didistribusikan ke dashboard monitoring serta dikirimkan ke Telegram, WhatsApp, dan Gmail melalui node notifikasi yang tersedia.

4.2 Implementasi Dashboard Monitoring Produktivitas Developer

Tampilan Dashboard Monitoring Produktivitas Developer

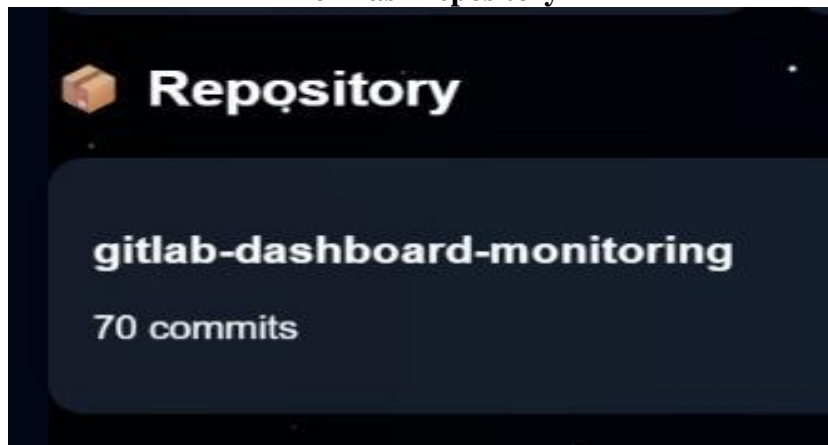


Gambar 2 Tampilan Dashboard Monitoring Produktivitas Developer
Sumber: Hasil Implementasi Sistem (2026)

Dashboard monitoring produktivitas developer dikembangkan menggunakan HTML, CSS, dan JavaScript, dan Ngrok dengan bantuan Visual Studio Code sebagai editor kode. Dashboard dirancang untuk menampilkan informasi aktivitas developer yang diperoleh dari GitLab API melalui workflow automation n8n. Dashboard mampu menampilkan informasi developer aktif, jumlah commit, aktivitas repository, top developer, grafik produktivitas, dan notifikasi real-time secara terintegrasi dalam satu halaman.

4.3 Analisis Statistik Repository

Informasi Repository



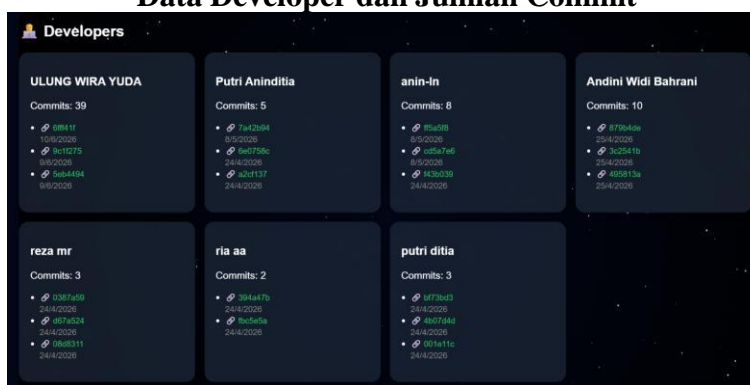
Gambar 3. Informasi Repository
Sumber: Hasil Implementasi Sistem (2026)

Berdasarkan hasil implementasi sistem, dashboard berhasil menampilkan informasi repository yang diperoleh secara otomatis melalui GitLab API. Repository

yang dimonitor dalam penelitian ini adalah gitlab-dashboard- monitoring dengan total 70 commit yang dilakukan oleh beberapa developer selama proses pengembangan sistem berlangsung. Informasi repository menjadi indikator utama dalam monitoring aktivitas pengembangan perangkat lunak karena menunjukkan tingkat aktivitas pengembang pada proyek yang sedang berjalan. Dengan adanya integrasi GitLab API, data repository dapat diperoleh secara otomatis tanpa memerlukan proses input manual sehingga informasi yang ditampilkan selalu sesuai dengan kondisi repository terbaru. Selain menampilkan jumlah commit, sistem juga dapat digunakan untuk mengidentifikasi tingkat aktivitas repository berdasarkan data commit yang masuk secara berkala. Informasi tersebut dapat digunakan sebagai bahan evaluasi perkembangan proyek oleh pengelola maupun pembimbing proyek.

4.4 Analisis Produktivitas Developer

Data Developer dan Jumlah Commit



Gambar 4. Data Developer dan Jumlah Commit

Sumber: Hasil Implementasi Sistem (2026)

Dashboard monitoring berhasil menampilkan data produktivitas developer berdasarkan jumlah commit yang diperoleh dari GitLab API. Data tersebut kemudian diproses menggunakan workflow automation n8n untuk menghasilkan informasi produktivitas masing-masing developer.

Berdasarkan hasil implementasi sistem diperoleh data kontribusi developer sebagai berikut:

Tabel 3. Kontribusi Developer

No	Nama Developer	Total Commit
	Ulung Wira Yuda	39
	Andini Widi Bahrani	10
	anin-ln	8
	Putri Aninditia	5
	Reza MR	3
	Putri ditia	3
	Ria aa	2

Sumber: Data GitLab API (2026)

Hasil analisis menunjukkan bahwa Ulung Wira Yuda merupakan developer dengan jumlah commit tertinggi yaitu sebanyak 39 commit. Hal ini menunjukkan

tingkat kontribusi yang lebih tinggi dibandingkan developer lainnya selama proses pengembangan sistem berlangsung. Melalui dashboard monitoring, pengguna dapat mengetahui kontribusi masing-masing developer secara cepat dan objektif sehingga membantu proses evaluasi produktivitas tim pengembang.

4.6 Analisis Top Developer dan Top Developer Hari Ini

Grafik Top Developer

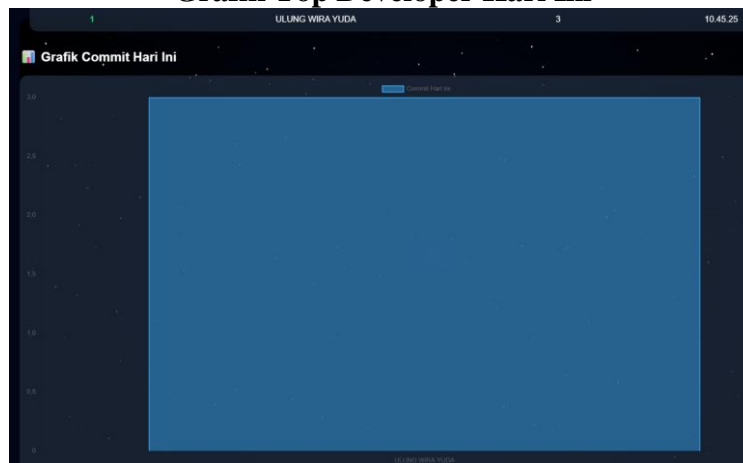


Gambar 6. Grafik Top Developer

Sumber: Hasil Implementasi Sistem (2026)

Fitur Top Developer digunakan untuk menampilkan peringkat developer berdasarkan jumlah commit yang telah dilakukan pada repository. Data tersebut diperoleh dari GitLab API kemudian dihitung menggunakan workflow n8n dan JavaScript sebelum ditampilkan pada dashboard. Hasil analisis menunjukkan bahwa lima developer dengan kontribusi tertinggi berhasil ditampilkan dalam bentuk grafik batang sehingga memudahkan pengguna dalam membandingkan tingkat produktivitas masing-masing developer. Visualisasi tersebut membantu pengelola proyek dalam melakukan evaluasi kontribusi anggota tim serta mempermudah identifikasi developer yang memiliki tingkat aktivitas tertinggi selama proses pengembangan berlangsung.

Grafik Top Developer Hari Ini



Gambar 7. Grafik Top Developer Hari Ini

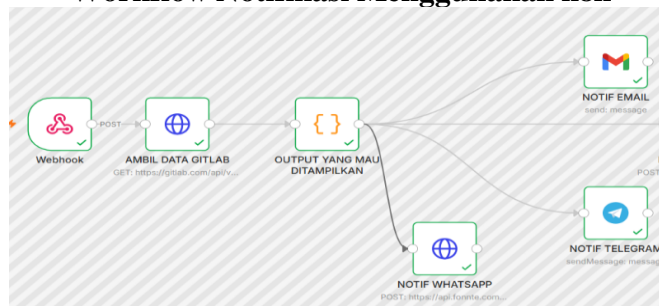
Sumber: Hasil Implementasi Sistem (2026)

Fitur Top Developer Hari Ini dikembangkan untuk menampilkan developer dengan jumlah commit terbanyak pada hari berjalan. Data diperoleh dari GitLab API kemudian difilter berdasarkan tanggal commit menggunakan workflow automation n8n dan JavaScript. Fitur ini memungkinkan pengguna mengetahui developer yang paling aktif dalam melakukan pengembangan pada hari tertentu. Informasi tersebut membantu proses monitoring aktivitas harian dan memberikan gambaran mengenai tingkat kontribusi developer secara real-time.

Berdasarkan hasil implementasi, sistem berhasil menghitung jumlah commit harian dan menampilkan developer dengan aktivitas tertinggi pada dashboard secara otomatis.

4.7 Analisis Implementasi Notifikasi Real-Time

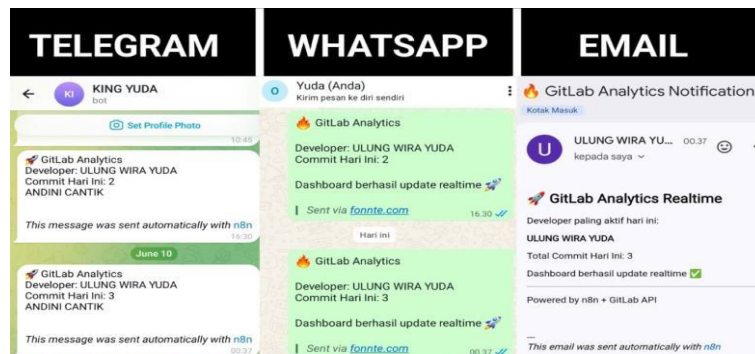
Workflow Notifikasi Menggunakan n8n



Gambar 8. Workflow Notifikasi Menggunakan n8n
 Sumber: Hasil Implementasi Sistem (2026)

Sistem notifikasi real-time dikembangkan menggunakan workflow automation n8n yang terintegrasi dengan GitLab melalui webhook. Setiap kali terjadi aktivitas commit baru pada repository, GitLab akan mengirimkan data ke endpoint webhook yang telah dikonfigurasi. Data yang diterima kemudian diproses menggunakan JavaScript Node untuk menghasilkan informasi yang relevan sebelum dikirimkan ke berbagai platform notifikasi. Dalam penelitian ini, notifikasi berhasil dikirimkan melalui Telegram, WhatsApp, dan email secara otomatis.

Notifikasi Berhasil Masuk Secara Otomatis Tiap Commit Hari Ini



Gambar 9. Notifikasi Berhasil Masuk Secara Otomatis Tiap Commit Hari Ini

Sumber: Hasil Implementasi Sistem (2026)

Implementasi notifikasi real-time memberikan keuntungan karena pengguna dapat memperoleh informasi aktivitas terbaru tanpa harus membuka dashboard secara terus-menerus. Dengan demikian, proses monitoring menjadi lebih efektif dan responsif terhadap perubahan yang terjadi pada repository.

4.8 Analisis Implementasi Metode Prototype

Metode Prototype digunakan karena memungkinkan sistem dikembangkan secara bertahap sesuai kebutuhan pengguna. Selama proses pengembangan, dilakukan beberapa tahap penyempurnaan berdasarkan hasil evaluasi dan pengujian sistem. Pada tahap awal, prototype sederhana dibuat untuk menampilkan data repository dan commit dari GitLab API. Selanjutnya dilakukan pengembangan fitur visualisasi data, analisis produktivitas developer, implementasi webhook real-time, dan sistem notifikasi otomatis. Penerapan metode Prototype terbukti membantu proses pengembangan karena pengguna dapat memberikan masukan secara langsung terhadap fitur yang dikembangkan. Dengan demikian, sistem yang dihasilkan lebih sesuai dengan kebutuhan monitoring produktivitas developer.

4.9 Pengujian Sistem

Pengujian sistem dilakukan menggunakan metode Black Box Testing untuk memastikan seluruh fungsi sistem berjalan sesuai dengan kebutuhan pengguna.

Tabel 4. Hasil Pengujian Black Box

No	Fitur yang Diuji	Hasil
1	Integrasi GitLab API	Berhasil
2	Monitoring Repository	Berhasil
3	Monitoring Commit	Berhasil
4	Analisis Produktivitas Developer	Berhasil
5	Grafik Commit Harian	Berhasil
6	Top Developer Chart	Berhasil
7	Webhook Real-Time	Berhasil
8	Notifikasi Telegram	Berhasil
9	Notifikasi WhatsApp	Berhasil
10	Notifikasi Email	Berhasil

Sumber: Hasil Pengujian Sistem (2026)

Berdasarkan hasil pengujian, seluruh fitur sistem dapat berjalan sesuai dengan kebutuhan pengguna. Dashboard berhasil menampilkan data repository, aktivitas commit, statistik produktivitas developer, dan informasi real-time yang diperoleh dari GitLab API. Workflow n8n juga berhasil menjalankan proses otomatisasi data dan pengiriman notifikasi tanpa kendala yang berarti.

KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa Dashboard Monitoring Produktivitas Developer Menggunakan Integrasi GitLab API dan n8n Berbasis Web dengan Metode Prototype berhasil dikembangkan dan diimplementasikan sesuai dengan kebutuhan sistem. Integrasi GitLab API menggunakan Personal Access Token mampu mengambil data repository, commit, dan aktivitas developer secara otomatis sehingga informasi yang diperoleh lebih akurat dan terstruktur. Penerapan workflow automation menggunakan n8n berhasil mendukung proses pengumpulan, pengolahan, dan distribusi data aktivitas developer secara otomatis. Sistem mampu memantau produktivitas developer berdasarkan aktivitas commit, menampilkan statistik repository, mengidentifikasi aktivitas repository yang berjalan, serta menyajikan informasi dalam bentuk dashboard berbasis web dengan visualisasi data secara real-time. Implementasi webhook pada GitLab memungkinkan sinkronisasi data dilakukan secara otomatis ketika terjadi aktivitas baru pada repository. Selain itu, sistem berhasil mengirimkan notifikasi harian melalui Telegram, WhatsApp, dan email sehingga pengguna dapat memperoleh informasi perkembangan proyek tanpa harus mengakses dashboard secara langsung. Berdasarkan hasil pengujian Black Box Testing, seluruh fitur utama sistem berjalan sesuai dengan kebutuhan pengguna, meliputi integrasi GitLab API, monitoring repository, monitoring commit, analisis produktivitas developer, visualisasi data, sinkronisasi real-time, dan sistem notifikasi. Dengan demikian, sistem yang dikembangkan mampu membantu proses monitoring aktivitas pengembangan perangkat lunak secara lebih efektif, cepat, dan terintegrasi serta mendukung proses evaluasi produktivitas developer.

UCAPAN TERIMA KASIH

Penulis Ulung Wira Yuda mengucapkan puji dan syukur kepada Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga penelitian dan penyusunan jurnal ini dapat diselesaikan dengan baik. Ucapan terima kasih juga disampaikan kepada Bapak Wydyanto, M.M., M.Kom. selaku dosen pembimbing yang telah memberikan arahan, bimbingan, serta masukan selama proses penelitian dan penyusunan jurnal ini. Penulis juga mengucapkan terima kasih kepada Bapak Deni Malyono selaku pembimbing lapangan yang telah memberikan bimbingan, dukungan, serta kesempatan untuk terlibat langsung dalam pengembangan proyek Dashboard Monitoring Produktivitas Developer berbasis GitLab API dan n8n. Selain itu, penulis mengucapkan terima kasih kepada seluruh rekan kerja serta civitas akademika Universitas Bina Darma yang telah memberikan bantuan, dukungan, dan motivasi selama pelaksanaan kegiatan magang hingga penyelesaian penelitian ini. Penulis menyadari bahwa penelitian ini masih memiliki keterbatasan. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan untuk pengembangan penelitian di masa yang akan datang. Semoga penelitian ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan.

Daftar Pustaka

- [1] S. Few, *Information Dashboard Design: Displaying Data for At-a-Glance Monitoring*. Burlingame, CA, USA: Analytics Press, 2021.
- [2] N. Forsgren, J. Humble, and G. Kim, *Accelerate: The Science of Lean Software and DevOps*. Portland, OR, USA: IT Revolution Press, 2021.
- [3] GitLab Inc., "REST API Documentation," 2025. [Online]. Available:

- <https://docs.gitlab.com/api/rest/>. [Accessed: Jun. 10, 2026].
- [4] GitLab Inc., "GitLab API Documentation," 2025. [Online]. Available: <https://docs.gitlab.com/api/>. [Accessed: Jun. 10, 2026].
- [5] GitLab Inc., "Repositories API Documentation," 2025. [Online]. Available: <https://docs.gitlab.com/api/repositories/>. [Accessed: Jun. 10, 2026].
- [6] A. Kurniawan and R. Pratama, "Implementasi Dashboard Monitoring Berbasis Web untuk Visualisasi Data Real-Time," *Jurnal Teknologi Informasi*, vol. 14, no. 2, pp. 101–109, 2022.
- [7] n8n GmbH, "n8n Documentation," 2025. [Online]. Available: <https://docs.n8n.io>. [Accessed: Jun. 10, 2026].
- [8] n8n GmbH, "Workflow Documentation," 2025. [Online]. Available: <https://docs.n8n.io/workflows/>. [Accessed: Jun. 10, 2026].
- [9] n8n GmbH, "n8n Workflow Automation Platform," 2025. [Online]. Available: <https://n8n.io>. [Accessed: Jun. 10, 2026].
- [10] M. Putra and D. Wijaya, "Implementasi Workflow Automation Menggunakan n8n pada Integrasi Sistem Informasi," *Jurnal Informatika Terapan*, vol. 8, no. 1, pp. 55–64, 2024.
- [11] M. Rahman and D. Setiawan, "Implementasi Webhook pada Sistem Integrasi Data Real-Time," *Jurnal Sistem Informasi dan Komputer*, vol. 8, no. 1, pp. 45–53, 2023.
- [12] L. Richardson and M. Amundsen, *RESTful API Design*. Sebastopol, CA, USA: O'Reilly Media, 2022.
- [13] A. S. Rosa and M. Shalahuddin, *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung, Indonesia: Informatika, 2021.
- [14] R. Saputra, H. Pratama, and S. Wijayanti, "Implementasi Sistem Notifikasi Real-Time Berbasis Webhook pada Aplikasi Monitoring," *Jurnal Teknologi Komputer*, vol. 12, no. 1, pp. 21–29, 2023.
- [15] I. Sommerville, *Software Engineering*, 11th ed. Boston, MA, USA: Pearson Education, 2021.