Pengembangan Sistem Penjaminan Mutu Internal Elektronik dengan Metode Devops di IAIN Palangka Raya

Slamet Riyadi*[1], Mohammad Jamaludin[2]
[1,2]Institut Agama Islam Negeri Palangka Raya
E-mail: *[1]**slamet.riau@iain-palangkaraya.ac.id** , [2]mohammad.jamaludin@iain-palangkaraya.ac.id

Abstrak

Sistem Penjaminan Mutu Pendidikan Tinggi terbagi menjadi dua bagian utama, yakni Sistem Penjaminan Mutu Eksternal (SPME) dan Sistem Penjaminan Mutu Internal (SPMI). Sistem Penjaminan Mutu Internal merupakan kegiatan penjaminan mutu Perguruan Tinggi yang dilakukan PT secara sistemik, mandiri dan otonom yang tujuannya untuk melakukan kendali mutu dan melakukan peningkatan kualitas penyelenggaraan PT secara terencana dan berkelanjutan. Terdapat beberapa kendala yang dialami selama pelaksanaan SPMI secara manual dan semi manual terutama terkait pelaksanaan, pendokumentasian, dan kontrol. Permasalahan tersebut dapat diatasi dengan membangun sebuah sistem pengolahan data yang berbasis teknologi informasi. Pada pengembangan dengan cara tradisional, setelah proses pengujian selesai, sering didapati adanya konflik akibat environment yang berbeda, yang memperlambat proses rilis. Keterlambatan ini membuat klien kecewa dan memberikan efek yang tidak baik bagi perusahaan. Continuous Integration / Continuous Delivery (CI/CD) yang ada di DevOps dapat dijadikan solusi untuk memecahkan masalah ini. Penggunaan metode DevOps dalam pengembangan E-Monev telah terbukti memberikan manfaat besar dalam hal efisiensi, kualitas, dan responsivitas. Namun, penerapan DevOps tidak hanya sekadar memilih alat-alat otomatisasi, tetapi juga melibatkan perubahan budaya dan kolaborasi yang lebih erat antara tim pengembang dan tim operasi. Penerapan DevOps dengan benar dalam konteks E-Monev dapat menjadi faktor kunci kesuksesan dalam pengembangan dan pengelolaan aplikasi tersebut.

Kata Kunci—SPMI, E-Monev, DevOps

1. PENDAHULUAN

Pasal 50 ayat (6) UU Sisdiknas UU No. 20 Tahun 2003 Tentang Sistem Pendidikan Nasional telah memperkenalkan tentang konsep Kemandirian Perguruan Tinggi atau Otonomi Perguruan Tinggi dalam mengelola Lembaga mereka sendiri. Budaya mutu PT adalah target utama dalam pengimplementasian SPM Dikti (Sistem Penjaminan Mutu Pendidikan Tinggi)[1]. Menurut Pasal 53 UU Dikti No. 12 Tahun 2012[2], Sistem Penjaminan Mutu Pendidikan Tinggi terbagi menjadi dua bagian utama, yakni Sistem Penjaminan Mutu Eksternal (SPME) dan Sistem Penjaminan Mutu Internal (SPMI). Sistem Penjaminan Mutu Internal merupakan kegiatan penjaminan mutu Perguruan Tinggi yang dilakukan PT secara sistemik, mandiri dan otonom yang tujuannya untuk melakukan kendali mutu dan melakukan peningkatan kualitas penyelenggaraan PT secara terencana dan berkelanjutan.

Terdapat beberapa kendala yang dialami selama pelaksanaan SPMI secara manual dan semi manual terutama terkait pelaksanaan, pendokumentasian, dan kontrol. Misalnya, salah satu kegiatan SPMI yaitu Monitoring Evaluasi (monev), Komite Penjaminan Mutu mengalami kesulitan untuk melakukan tracing atau merekap data sesuai dengan spesifikasi tertentu. Belum lagi jika dibutuhkan laporan riwayat monev dari beberapa semester, maka perlu mencari dan membuka satu per satu file-file tersebut. Dokumen hasil Monev juga seharusnya dapat terintegrasi dengan sistem lainnya (SIMAK, BKD, Survey, Data mahasiswa di Mikwa Institut), sehingga saat

penyusunan laporan evaluasi, penyusunan dokumen AMI, hingga penyusunan borang akreditasi Prodi dan PT, dokumen dapat diakses secara langsung tanpa melalui proses yang memakan waktu lama.

Permasalahan diatas dapat diatasi dengan membangun sebuah sistem pengolahan data yang berbasis teknologi informasi. Keberhasilan pengembangan sistem informasi dipengaruhi oleh faktor endogen seperti: kompleksitas, manajemen proyek, teknologi, metode pengembangan, keterlibatan pengguna, keterampilan dan pengalaman profesional, dan kualitas data [3]. Pada penelitian Rustiarni, dkk [4] salah satu metode *Agile* yaitu SCRUM berhasil digunakan untuk mengembangkan Sistem Informasi Yayasan Amfibi Reptil Indonesia Berbasis Website yang membantu pihak Yayasan dalam mengolah data tentang kegiatan pada Yayasan Amfibi dan Reptil Indonesia. Penggunaan metode *agile* lainnya yaitu *extreme programming* juga dapat digunakan untuk pengembangan Marketplace "Nufish" Berbasis web untuk meningkatkan pemasaran hasil perikanan[5].

Pada kebanyakan organisasi, secara tradisi memiliki tim pengembang dan tim operasi yang terpisah. Hal ini sering menimbulkan gap karena perbedaan tujuan(Feijter et al. 2017). Pendekatan pengembangan perangkat lunak yang menggunakan metode tradisional, termasuk di dalamnya metode Agile. DevOps merupakan tren baru yang muncul (tahun 2009) dari tumbukan antara dua tren lama yaitu 'agile system' dan 'agile operations'. Hal ini merupakan kolaborasi antara pengembangan dan staf operasi di setiap tahap Siklus hidup DevOps. 'Dev' berarti semua pengembang yang terlibat dalam tahap produksi[6].

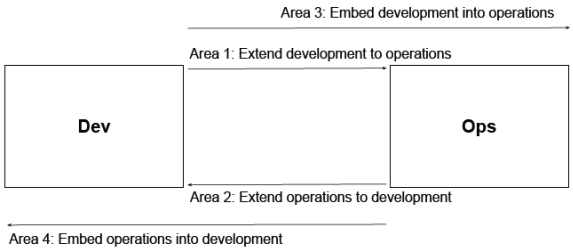
Pada pengembangan dengan cara tradisional, setelah proses pengujian selesai, sering didapati adanya konflik akibat environment yang berbeda, yang memperlambat proses rilis. Keterlambatan ini membuat klien kecewa dan memberikan efek yang tidak baik bagi perusahaan. Continuous Integration / Continuous Delivery (CI/CD) yang ada di DevOps dapat dijadikan solusi untuk memecahkan masalah ini[7].

2. METODE PENELITIAN

Metode yang digunakan pada penelitian adalah menggunakan metode Research & Development (R&D), Research and Development (R&D) merupakan metode penelitian yang digunakan untuk menghasilkan produk tertentu dan menguji keefektifan produk tersebut. Metode ini dipilih karena penelitian ini fokus pada pengembangan dan implementasi E-Monev dengan metode DevOps di IAIN Palangka Raya.

Pada banyak organisasi memiliki kebiasaan dimana kegiatan pengembangan dan operasi perangkat lunak dilakukan secara terspisah yang memiliki tujuan masing-masing. Dimana bagian pengembang ditugaskan untuk membuat, menambah, mengubah menjadi sebuah sistem baru. Sedangkan bagian operasi ditugaskan untuk menjaga jalannya stabilitas sistem. Hal ini umumnya menimbulkan konflik ketika akan diterapkan sistem baru, karena bagian operasi cenderung tidak ingin ada perubahan karena kekhawatiran akan tidak stabilnya sistem (H¨uttermann, 2012).

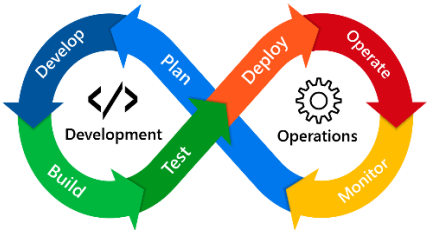
Dengan demikian, kerjasama antara bagian pengembang dengan bagian operasi sangat penting untuk menciptakan keselarasan antara pengembangan sistem dengan implementasi serta maintence-nya. H¨uttermann (2012) membuat ilustrasi keselarasan antara pihak-pihak tersebut di atas jelas dengan menghadirkan empat area DevOps dalam apa yang disebut DevOpsAreaMatrix, yang dapat diadopsi oleh organisasi untuk mengembangkan sistem seperti pada gambar 1.



Gambar 1. Area DevOps Menurut H¨uttermann

Budaya DevOps menurut sejumlah peneliti, membuat batas antara peran *development* dan peran *operations* menjadi kabur hingga pada akhirnya berdampak pada hilangnya perbedaan antara keduanya. Sikap tanggung jawab bersama merupakan salah satu aspek budaya DevOps yang mendorong kolaborasi yang lebih erat. Sehingga tidak ada batas antara *development* dan *operations*. Pengadopsian DevOps pada sejumlah organisasi dan peruhsaan menunjukkan adanya manfaat nyata bagi pengembangan produk-produk aplikasi mereka, sesuai dengan penelitian yang dilakukan Ellen.

Adapun siklus Pengembangan Perangkat Lunak Berdasarkan Paradigma DevOps seperti pada gambar 2.



Gambar 2. Siklus Pengembangan DevOps

Area *Development* meliputi tahap *Plan*, *Code*, *Build*, *Test*. Sedangkan area *Operation* meliputi *Deploy*, *Operate*, *Monitor*, *Release*. Area *Development* seterusnya disingkat Dev, sedangkan area *Operation* disingkat dengan Ops. Jika kedua kata tersebut dipadukan maka menjadi sebuah brand yaitu “DevOps”.

3. HASIL DAN PEMBAHASAN

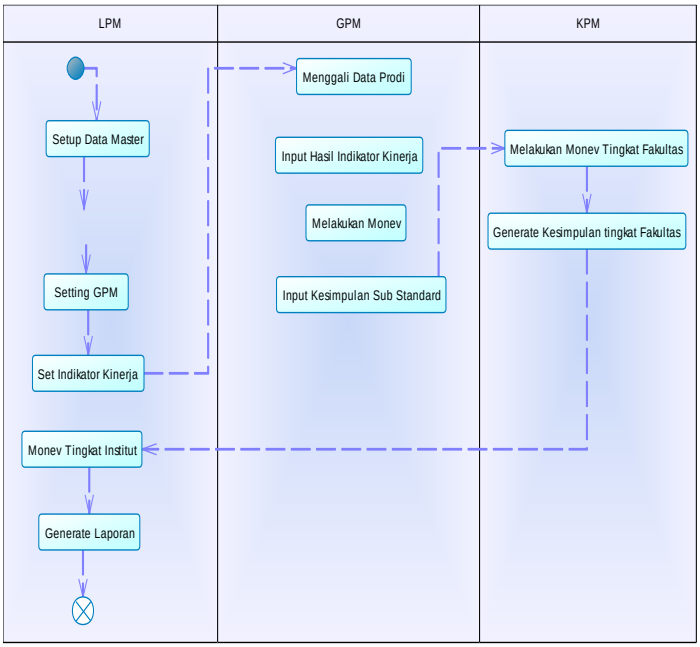
3.1. Perencanaan (Plan)

Tahap pertama dalam motede DevOps yaitu tahap Perencanaan yang secara detail dalam Pengembangan Sistem e-Monev ini yaitu:

1. Memahami Kebutuhan Bisnis. Melakukan pertemuan dengan pemangku kepentingan (stakeholders) yang terdiri dari GPM, KPM, serta LPM untuk memahami tujuan utama sistem e-Monev. Mengidentifikasi fitur-fitur utama yang diperlukan, termasuk input data, pelaporan data, serta visualisasi dan analisis data.
2. Penentuan Tim DevOps. Membentuk tim DevOps yang terdiri dari pengembang, tester, administrator sistem, dan ahli keamanan jika diperlukan pada fase Development. Menentukan peran dan tanggung jawab masing-masing anggota tim.

3. Rencana Alur Kerja (Workflow) DevOps. Menentukan tahapan-tahapan dalam alur kerja, seperti pengembangan kode, pengujian, integrasi, dan penerapan.
4. Penentuan Teknologi dan Platform. Memilih teknologi yang paling sesuai untuk membangun sistem e-Monev, termasuk bahasa pemrograman, basis data, dan framework. Adapun bahasa pemrograman yang dipilih yaitu PHP serbagai server side, serta jquery dan serta bootstrap sebagai sisi client. Untuk database yang akan digunakan yaitu MySQL. Sedangkan framework yang digunakan yaitu CodeIgniter. Pemilihan bahasa, database, dan framework tersebut berdasarkan kemudahan dan banyaknya referensi yang tersedia
5. Strategi Version Control. Menggunakan strategi version control untuk mengelola perubahan kode, dalam hal ini menggunakan Git. Buat branch terpisah untuk pengembangan fitur baru dan perbaikan dengan menggunakan Git.
6. Rencana Pengujian. Identifikasi jenis-jenis pengujian yang akan dilakukan, termasuk pengujian unit, integrasi, fungsional, dan uji beban. Pengujian yang digunakan nantinya yaitu metode blackbox, metode yang dapat digunakan untuk menguji fungsionalitas, integritas, serta penerimaan pengguna akhir.
7. Penentuan Alat DevOps. Memilih tools DevOps yang akan digunakan untuk otomatisasi deployment dan integrasi berkelanjutan. Tools yang digunakan antara lain HeidiSQL untuk manajemen basis data, Notepad++ untuk manajemen source code/project, serta Git untuk version Control

Berdasarkan tahapan memahami proses bisnis pada tahap perencanaan ini dihasilkan desain sistem awal. Adapaun desain dari e-monev ini dapat digambarkan dalam bentuk activity diagram berikut ini.



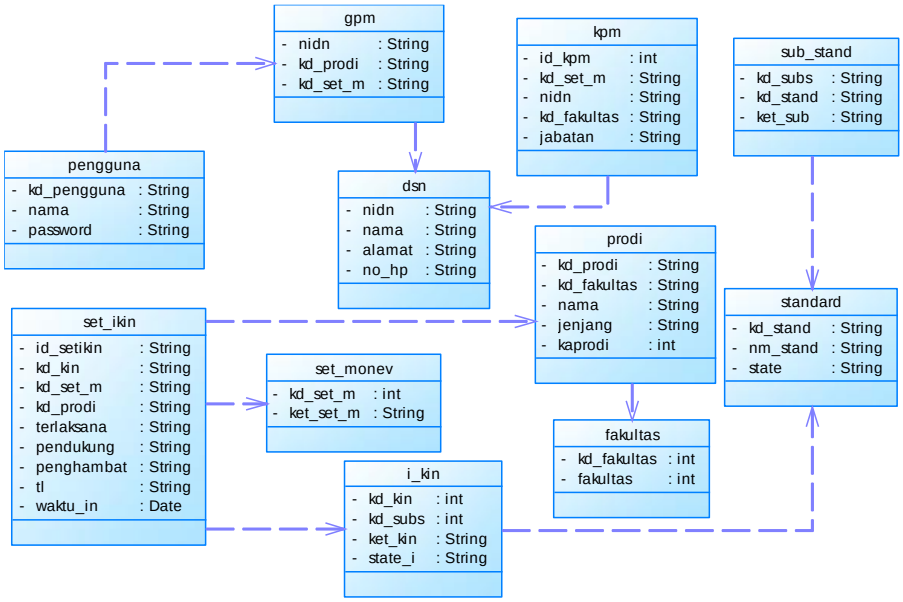
Gambar 3. Activity Diagram E-Monev.

3.2. Pengkodean (Develop)

Langkah-langkah pada tahap Pengkodean Infrastruktur melibatkan pembuatan kode komputer. Berikut adalah langkah-langkahnya:

1. Pemilihan Alat(tools) pengkodean: Tools yang dipilih sesuai dengan yang telah direncanakan pada tahap perencanaan.
2. Pemahaman Kebutuhan Infrastruktur: Sebelum menulis kode, pahami dengan baik kebutuhan infrastruktur. Identifikasi sumber daya apa yang perlu dibuat, seperti server, jaringan, basis data, atau layanan cloud tertentu.
3. Inisiasi Proyek: Buat direktori atau repositori khusus untuk proyek dalam sistem manajemen sumber kode seperti Git. Inisialisasi proyek dengan tools yang dipilih.
4. Penentuan Lingkungan Tujuan (Target Environment): Tentukan lingkungan tujuan di mana infrastruktur akan dibuat atau dikelola. Ini dapat berupa lingkungan pengembangan, pengujian, atau produksi. Kode harus fleksibel sehingga dapat diterapkan di berbagai lingkungan, baik sistem operasi maupun hardware pendukung.
5. Pembuatan dan Inisialisasi Basis Data. Tool Heidi digunakan untuk membuat dan menginisialisasi basis data sesuai dengan skema yang telah direncanakan. Hal ini termasuk pembuatan tabel, indeks, dan pengaturan awal lainnya.
6. Penulisan Kode: Mulai menulis kode sesuai dengan kebutuhan infrastruktur. Kode akan berisi deskripsi sumber daya, konfigurasi, dan hubungan antar sumber daya. Pastikan kode mematuhi sintaks dan standar yang ditetapkan oleh tools yang digunakan.
7. Pengujian Kode: Uji kode secara menyeluruh. Pastikan bahwa kode tersebut tidak hanya berjalan tanpa error, tetapi juga menghasilkan infrastruktur yang sesuai dengan harapan. Lakukan pengujian secara otomatis jika memungkinkan.
8. Simpan Kode di Repositori: Simpan kode Anda di repositori Git atau alat manajemen sumber kode lainnya. Gunakan kontrol versi untuk melacak perubahan kode.
9. Integrasi dengan CI/CD: Integrasikan kode IaC ke dalam alur kerja Continuous Integration/Continuous Deployment (CI/CD).
10. Dokumentasi Kode: Buat dokumentasi yang baik. Jelaskan tujuan, penggunaan, dan konfigurasi yang dapat diubah. Dokumentasi membantu pengguna lain memahami dan bekerja dengan kode tersebut.
11. Manajemen Perubahan: Ketika ada perubahan kebutuhan infrastruktur, ubah kode dan simpan perubahan tersebut di repositori. Pastikan untuk mengikuti praktik manajemen perubahan yang baik.

Adapun hasil dari Inisialisasi Basis Data yang akan digunakan dapat digambarkan dalam class diagram berikut ini.



Gambar 4. Class Diagram Aplikasi E-Monev

3.3. Integrasi Kode (Build)

Tahap Pengembangan dan Integrasi Kode yang Lebih Detail dalam Pengembangan Sistem e-Monev dengan Metode DevOps:

1. Penggunaan Version Control. Setiap perubahan kode harus dilakukan di dalam branch terpisah sesuai fitur atau perbaikan yang sedang dikerjakan.
2. Review Kode. Makukan review kode oleh sesama anggota tim untuk memastikan kualitas kode dan kesesuaian dengan pedoman penulisan kode. Diskusikan perubahan dan berikan umpan balik yang konstruktif.
3. Kode. Makukan penggabungan (merge) kode dari berbagai branch ke branch utama, seperti "develop" atau "main". Pastikan bahwa integrasi kode tidak merusak fungsionalitas yang sudah ada.
4. Pengujian Otomatis. Setiap kali terjadi integrasi kode, otomatisasi pengujian (continuous integration) harus berjalan. Jalankan pengujian unit dan pengujian integrasi untuk memverifikasi perilaku aplikasi.
5. Penanganan Konflik. Jika terjadi konflik saat penggabungan kode, segera tangani. Pastikan bahwa kode yang digabungkan masih mempertahankan integritasnya

3.4. Pengujian (Test)

Tahap akhir dari proses development yaitu pengujian. Tahap Pengujian secara detail dalam pengembangan sistem e-Monev yaitu:

1. Perencanaan Pengujian. Menentukan jenis-jenis pengujian yang akan dilakukan, termasuk pengujian unit (function/procedure), integrasi, fungsional, keamanan, dan uji beban. Identifikasi skenario pengujian untuk setiap jenis pengujian.
2. Pengujian Unit. Penguji fokus menguji setiap unit atau komponen secara mandiri. Pastikan setiap fungsi atau metode bekerja sesuai harapan.

3. Pengujian Integrasi. Uji interaksi antara berbagai komponen atau modul aplikasi. Memastikan bahwa komponen-komponen bekerja harmonis saat diintegrasikan.
4. Pengujian Fungsional. Uji fungsionalitas aplikasi berdasarkan spesifikasi yang telah ditentukan. Pastikan aplikasi dapat melakukan tugas-tugas yang diharapkan dengan benar.
5. Pengujian Keamanan. Melakukan pengujian keamanan untuk mengidentifikasi potensi kerentanan atau ancaman terhadap sistem. Uji kerentanan umum seperti SQL injection, cross-site scripting (XSS), dan lainnya.
6. Pengujian Uji Beban. Simulasikan beban tinggi pada aplikasi untuk mengukur kinerja, respons, dan skalabilitas. Identifikasi batas kapasitas dan performa sistem.
7. Pengujian Uji Penetrasi (Opsional). Uji penetrasi dilakukan untuk mencoba menembus keamanan aplikasi dan menemukan potensi celah.
8. Pengujian Pengguna (User Acceptance Testing, UAT). Melibatkan pengguna akhir dalam pengujian untuk memastikan aplikasi sesuai dengan kebutuhan mereka. Mendapatkan umpan balik dan perbaikan dari pengguna.
9. Pengujian Kompatibilitas. Uji aplikasi pada berbagai platform, perangkat, dan browser yang berbeda untuk memastikan kompatibilitas yang luas.

Hasil Pengujian dari aplikasi E-Monev ini yaitu sebagaimana tertuang dalam tabel 1 berikut ini.

Tabel 1. Hasil Pengujian Aplikasi E-Monev

No.	Fitur Pengujian	Deskripsi Pengujian	Hasil	Status
1	Login Pengguna	Memastikan pengguna dapat masuk	Lulus	Diterima
2	CRUD Data Fakultas	Menguji manajemen data Fakultas	Lulus	Diterima
3	CRUD Data Prodi	Menguji manajemen data Prodi	Lulus	Diterima
4	CRUD Data Prodi	Menguji manajemen data Prodi	Lulus	Diterima
5	CRUD Data Standard	Menguji manajemen data Standard	Lulus	Diterima
6	CRUD Data Indikator Kinerja	Menguji manajemen data Indikator Kinerja	Lulus	Diterima
7	Trs. Monev	Menguji kemampuan Trs. Monev dan pengisian kesimpulan	Lulus	Diterima
8	Generate Laporan	Menguji kemampuan sistem dalam men-generate laporan monev	Lulus	Diterima
9	Keamanan	Menguji keamanan sistem	Lulus	Diterima
10	Kinerja	Menguji kinerja aplikasi	Lulus	Diterima
11	Kompatibilitas Browser	Memastikan aplikasi berjalan di berbagai Browser	Lulus	Diterima

3.5. Penerapan (Deployment)

Tahapan Penerapan (Deployment) dalam Metode DevOps merupakan langkah kritis dalam siklus pengembangan perangkat lunak yang mengacu pada proses pengiriman dan pelaksanaan kode yang telah dikembangkan ke lingkungan produksi simulasi produksi. Berikut langkah-langkah detail mengenai tahapan Deployment sistem e-Monev ini:

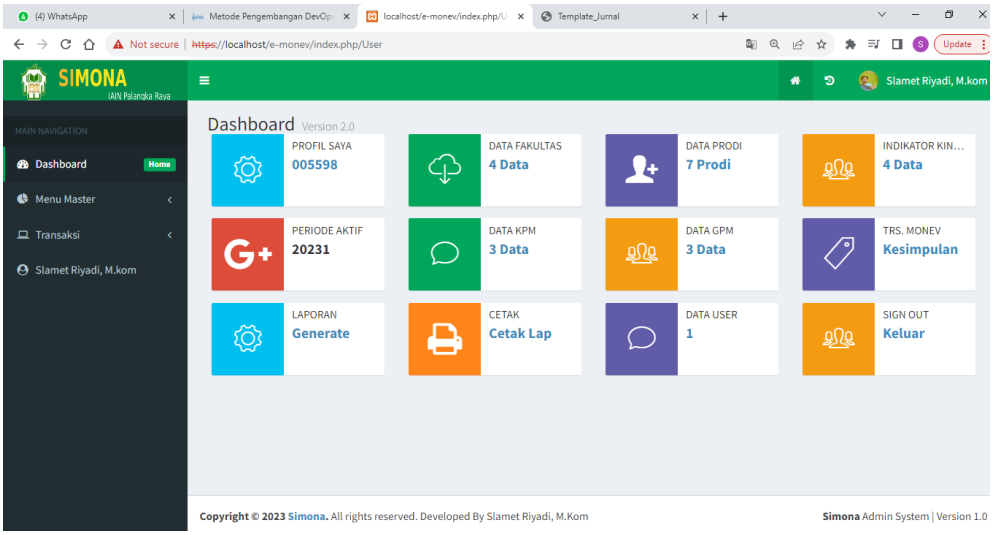
1. Penyiapan Lingkungan. Memastikan lingkungan produksi atau uji coba produksi telah siap untuk menerima perubahan. Ini termasuk infrastruktur perangkat keras dan perangkat lunak yang diperlukan.

2. Penyiapan Infrastruktur. Selama tahap ini, infrastruktur yang diperlukan untuk aplikasi, termasuk server, jaringan, basis data, dan layanan lainnya, disiapkan secara sesuai dengan spesifikasi yang ditetapkan.
3. Pengiriman Kode. Setelah lingkungan siap, kode yang telah diuji dan diverifikasi dalam tahap pengujian akan dikirim ke lingkungan produksi atau uji coba produksi.
4. Pengelolaan Versi. Selama penerapan, pastikan bahwa versi kode yang akan diterapkan diidentifikasi dengan jelas. Ini dapat dicapai dengan menggunakan tools manajemen versi seperti Git.
5. Uji Pascapenerapan (Post-Deployment Testing). Lakukan pengujian pascapenerapan (post-deployment testing) untuk memastikan bahwa perubahan kode tidak menyebabkan masalah di lingkungan produksi.

3.6. Monitoring dan Operasi

Tahapan Pemantauan dan Operasi dalam metode DevOps melibatkan pengawasan dan pengoperasian berkelanjutan terhadap sistem atau aplikasi yang telah diterapkan ke lingkungan produksi. Berikut langkah-langkah lebih detail mengenai tahapan Pemantauan dan Operasi sistem e-Monev ini:

1. Pemantauan Kinerja (Performance Monitoring). Gunakan alat pemantauan untuk mengukur kinerja aplikasi dan infrastruktur di lingkungan produksi. Pantau metrik kunci seperti waktu respons, throughput, penggunaan CPU, `penggunaan memori, dan utilitas jaringan.
2. Pemantauan Ketersediaan (Availability Monitoring). Pastikan bahwa sistem tersedia dan berfungsi dengan benar. Monitor ketersediaan layanan dan infrastruktur untuk memastikan bahwa pengguna dapat mengakses aplikasi tanpa gangguan.
3. Pendeteksian Insiden (Incident Detection). Gunakan alat pemantauan dan peringatan otomatis untuk mendeteksi insiden atau masalah potensial sejak dini.
4. Respons Terhadap Insiden (Incident Response). Jika terjadi masalah atau insiden, tim harus memiliki prosedur respons yang jelas dan siap untuk dijalankan. Identifikasi penyebab akar masalah dan lakukan tindakan perbaikan.
5. Pemantauan Keamanan (Security Monitoring). Monitor aktivitas keamanan untuk mendeteksi aktivitas mencurigakan atau ancaman keamanan. Gunakan alat pemantauan keamanan untuk menganalisis dan merespons potensi pelanggaran keamanan.
6. Analisis Log (Log Analysis). Pemantauan log aktivitas sistem dan aplikasi untuk mengidentifikasi masalah, melacak jejak, dan menganalisis perilaku aplikasi. Gunakan log untuk membantu dalam investigasi insiden dan pemecahan masalah.
7. Pemeliharaan Rutin (Routine Maintenance). Lakukan pemeliharaan rutin untuk menjaga kinerja sistem, termasuk pembaruan perangkat lunak, pemantauan kapasitas, dan pembersihan data.
8. Optimisasi Kinerja (Performance Optimization). Berdasarkan hasil pemantauan, identifikasi area yang memerlukan peningkatan kinerja. Lakukan tindakan optimisasi, seperti pengaturan kapasitas, tuning, atau pengoptimalan kode.
9. Pemulihan Darurat (Disaster Recovery). Pastikan rencana pemulihan darurat telah disiapkan dan diuji, sehingga sistem dapat pulih dari kegagalan besar dengan cepat.
10. Umpan Balik dan Peningkatan Lanjutan. Terima umpan balik dari pengguna dan pemangku kepentingan, dan gunakan informasi ini untuk terus meningkatkan kualitas dan kinerja sistem.
11. Pencatatan dan Pelaporan (Logging and Reporting). Catat aktivitas dan hasil pemantauan secara rinci. Buat laporan berkala untuk melacak tren kinerja dan memenuhi kebutuhan pelaporan.



Gambar 5. Halaman Utama Aplikasi E-Monev

Tahapan Pemantauan dan Operasi dalam DevOps memastikan bahwa sistem tetap berjalan dengan baik, aman, dan efisien setelah penerapan. Pemantauan aktif dan tindakan operasional yang tepat dapat membantu mengurangi waktu henti, meningkatkan kualitas layanan, dan memberikan pengalaman yang lebih baik kepada pengguna.

4. KESIMPULAN

Penerapan metode DevOps dalam pengembangan E-Monev menunjukkan dampak yang signifikan dalam meningkatkan efisiensi dan efektivitas seluruh siklus pengembangan perangkat lunak. Hal tersebut dapat digambarkan berdasarkan kesimpulan-kesimpulan berikut:

1. Kepuasan Pengguna: Dengan pengujian yang lebih komprehensif dan integrasi pengguna akhir dalam UAT, hasil akhir yang dihasilkan oleh pengembang memiliki tingkat kepuasan pengguna yang lebih tinggi. Hal ini meningkatkan adopsi aplikasi E-Monev dan memastikan bahwa pengguna merasa aplikasi tersebut memenuhi kebutuhan mereka dengan baik.
2. Pemantauan Kinerja: DevOps memungkinkan pemantauan kinerja yang kontinu dan real-time. Ini membantu mendeteksi masalah kinerja sejak dini, memungkinkan tindakan korektif yang cepat untuk menjaga aplikasi E-Monev tetap responsif dan berkinerja tinggi.
3. Perbaikan Berkelanjutan: Dalam budaya DevOps, perbaikan berkelanjutan adalah prinsip utama. Tim pengembang selalu mencari cara untuk meningkatkan proses pengembangan, pengujian, dan penyebaran. Hal ini membantu mengurangi pemborosan dan meningkatkan produktivitas.
4. Skalabilitas dan Fleksibilitas: DevOps memungkinkan skalabilitas yang lebih baik dalam menghadapi pertumbuhan permintaan atau beban kerja. Aplikasi E-Monev dapat dengan mudah diubah ukurannya sesuai kebutuhan, sehingga memastikan ketersediaan yang tinggi dan kinerja yang stabil.
5. Pengurangan Risiko: Dengan pemantauan kualitas yang ketat dan pengujian yang otomatis, DevOps membantu mengurangi risiko kesalahan manusia dan masalah keamanan. Ini berdampak positif pada keandalan dan keamanan data dalam aplikasi E-Monev.
6. Adaptasi Terhadap Perubahan: Dalam dunia yang terus berubah, DevOps memberikan fleksibilitas yang diperlukan dalam mengadaptasi aplikasi E-Monev terhadap perubahan

regulasi, kebijakan, atau kebutuhan pengguna. Perubahan dapat diterapkan dan diuji dengan cepat.

DAFTAR PUSTAKA

[1] Presiden RI, "UNDANG-UNDANG REPUBLIK INDONESIA NOMOR 20 TAHUN 2003 TENTANG SISTEM PENDIDIKAN NASIONAL," *Dep. Pendidik. Nas.*, vol. 19, no. 8, pp. 159–170, 2003.

[2] Kementrian Hukum dan HAM, "UU RI No. 12/2012 tentang Pendidikan Tinggi," *Undang Undang*, p. 18, 2012.

[3] P. Isaias and T. Issa, *High Level Models for Information Systems*. 2016.

[4] R. Rustiarni, E. Putri, V. Dawvi, and F. Wardani Kusuma, "Rancang Bangun Sistem Informasi Manajemen Kegiatan Yayasan Amfibi Reptile Indonesia," *Smart Comp Jurnalnya Orang Pint. Komput.*, vol. 11, no. 4, pp. 745–751, 2022, doi: 10.30591/smartcomp.v11i4.4266.

[5] I. Rusmana Akbar, R. Fauzi, and D. Pramesti, "Pengembangan Marketplace 'Nufish' Berbasis Web Untuk Meningkatkan Pemasaran Hasil Perikanan Menggunakan Metode Extreme Programming," *Smart Comp Jurnalnya Orang Pint. Komput.*, vol. 12, no. 1, 2023, doi: 10.30591/smartcomp.v12i1.4635.

[6] K. Akshaya, HL; Nisarga, Jagadish S; Vidya, J; Veena, "A Basic Introduction to DevOps Tools," *Int. J. Comput. Sci. Inf. Technol.*, vol. 6, no. March, pp. 1–4, 2015.

[7] J. Jaeni, N. A. S., and A. D. Laksito, "Implementasi Continuous Integration/Continuous Delivery (Ci/Cd) Pada Performance Testing Devops," *J. Inf. Syst. Manag.*, vol. 4, no. 1, pp. 62–66, 2022, doi: 10.24076/joism.2022v4i1.887.