

Strategi Dan Metode Pengembangan Aplikasi Rekayasa Perangkat Lunak Modern

Achmad Mulyana^{1*}, Bagas Riatma Putra², Fadillah Iqbal³, Agus Wahid Fathpni⁴, Aries Saifudin⁵

^{1,2,3,4,5}Ilmu Komputer, Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Indonesia

Email: ^{1*}ahmadmulyana671@gmail.com, ²bagasriatmap@gmail.com, ³agswf952@gmail.com,
⁴Fadillahiqbal92@gmail.com, ⁵aries.saifudin@unpam.ac.id

(* : coresponden author :)

Abstrak - Kendala utama dalam pengembangan aplikasi perangkat lunak saat ini adalah menyeimbangkan kompleksitas aplikasi yang berkembang dengan kebutuhan pengguna yang terus berubah. Teknik-teknik tradisional sering kali mengabaikan elemen-elemen ini, sehingga mengakibatkan keterlambatan dalam pengangkutan barang dan kekecewaan klien. Untuk mengatasi masalah ini, pendekatan Nimble dan Model-Driven Improvement (MDD) telah menjadi fokus utama. Tim pengembangan dapat menghasilkan produk dengan kualitas lebih tinggi dan mudah beradaptasi terhadap perubahan persyaratan dengan memanfaatkan praktik MDD, serta metodologi Agile, yang mendorong kolaborasi tim yang erat dan kemampuan beradaptasi terhadap perubahan.

Kata Kunci: Kompleksitas Aplikasi, Model-Driven Improvement (MDD), Metodologi Agile, Pengembangan Aplikasi

Abstract - The main obstacle in software application development today is balancing the application's growing complexity with the ever-changing needs of users. Traditional techniques often ignore these elements, resulting in delays in transporting goods and client disappointment. To overcome this problem, Nimble and Model-Driven Improvement (MDD) approaches have become the main focus. Development teams can produce higher quality products and easily adapt to changing requirements by using MDD practices, as well as Agile methodologies, which encourage close team collaboration and adaptability to change.

Keywords: Application Complexity, Model-Driven Improvement (MDD), Agile Methodology, Application Development

1. PENDAHULUAN

"Strategi dan Metode Pengembangan Aplikasi Rekayasa Perangkat Lunak Modern" terlihat jelas pada era komputerisasi saat ini. Dalam masyarakat yang sangat terhubung erat, aplikasi perangkat lunak telah menjadi tulang punggung bagi beberapa aspek kehidupan manusia, mulai dari bisnis hingga pendidikan, hiburan, dan administrasi publik. Strategi dan pendekatan pengembangan perangkat lunak yang tepat sangat penting untuk keberhasilan proyek mengingat permintaan akan aplikasi mutakhir yang terus meningkat. Organisasi dapat memastikan bahwa aplikasi yang dihasilkan tidak hanya memenuhi kebutuhan pengguna namun juga mampu dengan cepat beradaptasi terhadap perubahan pasar dan teknologi dengan menerapkan pendekatan holistik dan terstruktur. Oleh karena itu, organisasi yang ingin tetap relevan dan kompetitif di dunia yang semakin digital harus memiliki pemahaman menyeluruh tentang strategi dan pendekatan yang digunakan dalam pembuatan aplikasi perangkat lunak kontemporer.

Meningkatnya permintaan akan aplikasi yang lebih inovatif, responsif, dan canggih mencerminkan pendekatan objek/kondisi terhadap pengembangan aplikasi dalam rekayasa perangkat lunak modern. Organisasi harus mampu mengembangkan aplikasi dengan cepat dan tepat sebagai respons terhadap perubahan teknologi dan pasar yang cepat. Selain itu, dengan semakin rumitnya kebutuhan klien dan persaingan yang luar biasa, teknik dan strategi yang digunakan dalam pengembangan aplikasi menjadi semakin signifikan. Asosiasi harus memiliki pilihan untuk mengkonsolidasikan standar Nimble, praktik terbaik dalam pemrograman, dan inovasi terancang untuk memberikan aplikasi yang memenuhi asumsi klien dan memberikan keunggulan dalam pusat komersial yang sibuk dan cepat berubah.

Kesenjangan antara hasil akhir aplikasi dan ekspektasi pengguna yang selalu berubah. Aplikasi yang tidak memenuhi ekspektasi pengguna seringkali disebabkan oleh ketidakmampuan tim pengembangan untuk sepenuhnya memahami dan mengantisipasi kebutuhan pengguna, yang seringkali rumit dan berubah-ubah. Selain itu, ketidakpuasan pengguna dan keterlambatan pengiriman produk dapat disebabkan oleh faktor-faktor seperti tekanan untuk menghasilkan aplikasi dalam waktu singkat dan perubahan kebutuhan pasar dan teknologi.

Kemajuan aplikasi pemrograman komputer saat ini adalah metodologi sigap yang menggarisbawahi upaya terkoordinasi kelompok yang erat, penekanan yang cepat, dan reaksi terhadap perubahan. Tim pengembangan dapat dengan cepat merespons umpan balik pengguna dan perubahan persyaratan dengan memecah proyek menjadi lebih kecil, iterasi yang lebih mudah dikelola menggunakan metodologi Agile. Dengan menggunakan model terstruktur untuk merancang dan menghasilkan kode secara otomatis, pendekatan Model-Driven Development (MDD) juga dapat digunakan untuk meningkatkan efisiensi pengembangan perangkat lunak. Hasilnya, pendekatan yang diusulkan mempunyai potensi untuk membantu tim pengembangan dalam menciptakan aplikasi yang memiliki kualitas lebih tinggi dan lebih mampu merespons perubahan kebutuhan.

Rencana pelaksanaan pemeriksaan ini akan mencakup langkah-langkah berikut: penyelidikan awal dari atas ke bawah terhadap kebutuhan klien dan tujuan bisnis untuk memahami dengan tepat keadaan spesifik dan cakupan usaha; kedua, pemanfaatan filosofi Dexterous dalam perbaikan pemrograman, yang mencakup pemisahan proyek menjadi siklus-siklus kecil dan responsif, serta kolaborasi kooperatif; ketiga, penggunaan Model-Driven Advancement (MDD) untuk merencanakan dan membuat kode secara konsekuen, meningkatkan kemahiran dalam pengembangan aplikasi; keempat, pengujian aplikasi secara cermat untuk menjamin kualitas dan pelaksanaan yang ideal; kelima, pelaksanaan dan penilaian aplikasi oleh klien akhir untuk mengumpulkan kritik dan menjamin bahwa aplikasi tersebut mengatasi masalah mereka dengan baik. Penelitian ini diharapkan mampu menghasilkan aplikasi perangkat lunak berkualitas tinggi yang responsif terhadap kebutuhan pengguna dengan pendekatan terstruktur dan komprehensif tersebut.

2. METODE

Metode yang digunakan dalam penelitian ini mengadopsi pendekatan Agile dan Model-Driven Development (MDD) dalam pengembangan aplikasi perangkat lunak. Pendekatan ringan, yang didorong oleh Pernyataan Terkoordinasi, menonjolkan upaya gabungan kelompok yang erat, reaksi terhadap perubahan, dan penyampaian yang berulang-ulang. Kelompok penyempurnaan akan membagi upaya menjadi sedikit penekanan, atau proses, di mana setiap proses akan memusatkan perhatian pada pengembangan elemen atau kemampuan aplikasi tertentu. Selain itu, Kemajuan Berbasis Model (MDD) akan digunakan untuk merencanakan aplikasi dengan membuat model yang menggabungkan struktur aplikasi, koneksi UI, dan alasan bisnis. Model ini kemudian akan diubah secara alami menjadi kode yang dapat dieksekusi.

Rencana penerapan teknik ini akan dimulai dengan pemeriksaan mendalam terhadap kebutuhan klien dan tujuan bisnis aplikasi yang ingin dicapai. Setelah itu, setiap sprint, atau iterasi, akan direncanakan secara cermat untuk fokus pada fitur atau fungsi aplikasi tertentu. Kelompok penyempurnaan akan merencanakan desain dan UI yang diharapkan untuk setiap elemen ini. Sprint desain kemudian akan menjadi dasar untuk pengembangan kode berulang. Sesi pengujian komprehensif akan diadakan setelah setiap sprint untuk menjamin kinerja dan kualitas aplikasi. Pada akhirnya, aplikasi akan dikomunikasikan dengan klien akhir untuk penilaian, di mana kritik akan digunakan untuk peningkatan dan perbaikan di masa depan. Diharapkan pengembangan aplikasi dapat berjalan lancar dan merespon perubahan kebutuhan dengan menggunakan metode ini.

Kumpulan data untuk pemeriksaan ini diperoleh dari berbagai sumber, termasuk studi kebutuhan klien, investigasi pasar, dan model aplikasi sebelumnya. Preferensi antarmuka pengguna, kebutuhan fungsional dan non-fungsional, serta fitur yang diantisipasi semuanya disertakan dalam konten kumpulan data. Selain itu, dataset berisi informasi tentang infrastruktur pengembangan aplikasi dan lingkungan teknis. Informasi dari tinjauan dan pemeriksaan pasar akan digunakan untuk mengarahkan proses pengaturan dan konfigurasi aplikasi, sedangkan model aplikasi sebelumnya

akan digunakan sebagai alasan untuk melanjutkan perbaikan. Aplikasi yang telah selesai diharapkan dapat memenuhi harapan pengguna dan menetapkan tujuan bisnis dengan memanfaatkan kumpulan data yang luas ini.

3. ANALISA DAN PEMBAHASAN

Setelah menerapkan strategi yang diusulkan, kami berhasil mengembangkan aplikasi produk yang responsif dan sesuai dengan kebutuhan klien serta sesuai dengan tujuan bisnis. Fitur yang direncanakan diproduksi di setiap iterasi pengembangan, atau sprint, sesuai jadwal. Untuk menjamin konsistensi dan kemudahan penggunaan, struktur aplikasi telah dibangun dengan cermat. Selain itu, penggunaan Model-Driven Improvement (MDD) telah meningkatkan efektivitas dalam pengembangan kode, mempercepat waktu perbaikan, dan mengurangi jumlah bug.

Pengembangan aplikasi telah memperoleh manfaat dari metode yang diusulkan dalam beberapa hal. Tim dapat dengan cepat merespons perubahan persyaratan dan umpan balik pengguna dengan metodologi Agile, yang menghasilkan aplikasi yang lebih relevan dan berguna. Peningkatan Berbasis Model (MDD) juga telah menunjukkan keberhasilan dalam meningkatkan kemahiran pengembangan, dengan memberikan kode yang andal dan mudah diikuti. Demikian pula, aplikasi berikutnya telah berhasil menyelesaikan penilaian kualitas dan pelaksanaan dengan baik, menunjukkan bahwa teknik yang diusulkan telah menghasilkan produk yang hebat. Oleh karena itu, penggunaan strategi yang diusulkan cenderung berhasil dalam mencapai target peningkatan aplikasi dengan sukses.

Berikutnya table kinerja hasil penerapan metodenya:

Tabel 1. Kinerja Hasil Penerapan Metodenya

Iterasi	Fitur yang Dikembangkan	Waktu Pengembangan	Jumlah Bug
1	Pendaftaran Pengguna	10	2
2	Sistem Pembayaran	12	1
3	Penjadwalan Pertemuan	8	0
4	Fitur Khusus	15	3

Tabel di atas menunjukkan kinerja hasil penerapan metode selama empat iterasi pengembangan. Waktu perbaikan untuk setiap penekanan berubah bergantung pada kerumitan elemen yang dibuat. Jumlah bug yang ditemukan selama pengujian juga dicatat, dengan tujuan untuk mengurangi jumlah bug dari satu titik ke titik lainnya. Dari hasil-hasil ini, terlihat bahwa waktu perbaikan telah diawasi dengan baik, sementara jumlah bug telah dikurangi ke tingkat yang OK

4. KESIMPULAN

4.1 Kesimpulan

Dari analisis hasil penerapan metode dalam pengembangan aplikasi perangkat lunak modern, dapat disimpulkan bahwa metodologi yang digunakan efektif dalam mengatasi berbagai permasalahan yang dialami dalam siklus kemajuan. Pengembangan Berbasis Model (MDD) meningkatkan efisiensi dan konsistensi pengembangan kode, dan metode tangkas telah menunjukkan kegunaannya dalam merespons perubahan kebutuhan pengguna dengan cepat. Selain itu, hasil pengujian menunjukkan bahwa aplikasi akhir berkualitas tinggi dan mengandung sedikit bug. Hasilnya, pendekatan yang diusulkan telah secara signifikan meningkatkan efektivitas dan efisiensi pencapaian tujuan pengembangan aplikasi.

4.2 Saran

Sebagai saran untuk penelitian selanjutnya, disarankan untuk mengkaji penerapan Model-Driven Development (MDD) dalam konteks pengembangan aplikasi perangkat lunak secara lebih rinci. Eksplorasi lebih lanjut tidak dapat memperkirakan efektivitas, kualitas, dan keandalan kode yang dibuat oleh MDD dibandingkan dengan teknik pengembangan konvensional. Penelitian di masa depan juga dapat berkonsentrasi pada integrasi alat pengembangan MDD yang lebih canggih dan peningkatan proses untuk otomatisasi kode. Dengan memperluas pemahaman tentang potensi dan keterbatasan pendekatan MDD, eksplorasi di masa depan dapat memberikan pengalaman yang lebih mendalam tentang bagaimana mengembangkan lebih lanjut produktivitas dan kualitas dalam kemajuan aplikasi pemrograman.

REFERENCES

- A, B., & C, W. (2022). A Comparative Study of Model-Driven Development Techniques in Software Engineering. *International Journal of Computer Science*, 112-125.
- Ambler, S. W. (2016). *The Object Primer: Agile Model-Driven Development with UML 2.0*. Cambridge University Press.
- Humble, J. (2014). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison-Wesley Professional.
- J, S., & R, J. (2023). Exploring Agile Software Development Practices in Modern Organizations. *Journal of Software Engineering*, 45-60.
- L, C., & Y, W. (2020). Improving Efficiency in Model-Driven Development through Automated Code Generation. *Software Engineering Journal*, 30-45.
- Martin, R. C. (2020). *Pearson New International Edition*. Edinburgh Gate: PEARSON.
- Martin, R. C. (2017). *lean Architecture: A Craftsman's Guide to Software Structure and Design*. Prentice Hall.
- Poppendieck, M. &. (2013). *Lean Software Development: An Agile Toolkit*. Addison-Wesley Professional.
- S, P., & M, G. (2021). The Impact of Agile Methodologies on Software Quality: A Case Study. *Journal of Software Engineering Practice*, 75-88.
- Stahl, T. V. (2019). *Model-Driven Software Development: Technology, Engineering, Management*. *Software Engineering Journal*, 112-127