

FORWARD ENGINEERING PADA APLIKASI E-SURAT (STUDI KASUS PT. XYZ)

¹Arni Muarifah Amri*, ²Ilham Mudin, ³Lorenzo Laridho Sambiring, ⁴M Syarif Hidayatullah, ⁵Dyah Putri Rahmawati

^{1,2,3,4} Program Studi Rekayasa Perangkat Lunak, ¹ Program Studi Informatika, Fakultas Informatika,
Universitas Telkom
Jl Terusan Buah Batu Bandung, Indonesia

arnyrivah@telkomuniversity.ac.id, ilhammudin@student.telkomuniversity.ac.id,
lorenzosembiring@student.telkomuniversity.ac.id, syarifhidayatullah@student.telkomuniversity.ac.id
dyahputri@telkomuniversity.ac.id

Abstrak

E-surat adalah sistem persuratan yang diperuntukan untuk sebuah institusi. Aplikasi ini digunakan untuk memudahkan proses surat menyurat di lingkup sebuah institusi sehingga pengelolaan nomor surat, pembuatan surat, dan pengarsipan surat keluar dan masuk terkelola dengan baik. Namun, aplikasi yang ada saat ini masih memiliki kekurangan berupa terbatasnya kategori surat yang dapat diarsipkan, yakni hanya terbatas pada delapan jenis surat yang telah diatur dengan hardcode. Hal ini menjadi sebuah masalah karena pada prakteknya, pengguna membutuhkan kategori surat yang dapat disesuaikan terhadap kebutuhan pengguna. Melalui pendekatan forward engineering, penelitian ini akan mengidentifikasi dan mengatasi permasalahan tersebut dengan merancang ulang aplikasi sehingga mampu menangani dan menambahkan kategori surat baru secara mudah.

Kata kunci: E-Surat, Forward Engineering, Software Reengineering, Use Case, MySQL

Abstract

E-Surat is a correspondence system designed for an institution. This application is utilized to facilitate the process of correspondence within the scope of an institution, ensuring proper management of letter numbers, letter creation, and the organization of outgoing and incoming letters. However, the current application has limitations in terms of the categories of letters that can be archived, as it is restricted to only eight predefined types of letters hardcoded into the system. This poses a problem because, in practice, users need customizable letter categories based on their specific requirements. Through a forward engineering approach, this research aims to identify and address these issues by redesigning the application to handle and easily add new letter categories.

Keywords: Maximum E-Surat, Forward Engineering, Software Reengineering, Use Case, MySQL

1 Pendahuluan (or Introduction)

Aplikasi E-Surat merupakan sebuah aplikasi berbasis web untuk menyimpan dan mengelola dokumen arsip, dokumen surat masuk dan surat keluar agar lebih terstruktur dan efisien. Aplikasi ini sangat diperlukan untuk pengelolaan surat di suatu institusi. Dalam penggunaannya, seringkali karna kebutuhan yang meningkat seperti pada aplikasi E-surat di institusi XYZ, beberapa fitur masih perlu ditambahkan. Dari kebutuhan tersebut, diperlukan Software Reengineering yang tahapan dan tujuannya adalah melakukan analisis dan restrukturisasi software dengan merekonstruksi kembali dalam bentuk yang baru dan diimplementasikan selanjutnya dalam bentuk yang baru. Hal ini sering dilakukan untuk mengatasi permasalahan suatu aplikasi jika dibutuhkan penambahan fitur ataupun memodernisasi suatu software [1]. Selain itu, reengineering dilakukan untuk memberikan peningkatan



This work is licensed under a [Creative Commons Attribution 4.0 International License](http://creativecommons.org/licenses/by/4.0/).
<http://journal.stmikjayakarta.ac.id/index.php/JMIJayakarta>

DOI: <https://doi.org/10.52362/jmijayakarta.v4i4.1595>

dari segi performance dan reliability pada suatu software. Pada Aplikasi E-Surat yang sudah berjalan terdiri dari 8 jenis surat yang telah ditentukan sebelumnya. Dan pada saat implementasi, aplikasi terdapat keterbatasan yang dialami oleh user yaitu pada saat ingin menyesuaikan dan menambahkan jenis surat selain dari jenis surat yang sudah ada.

Pendekatan forward engineering diadopsi untuk mengatasi permasalahan tersebut dengan merancang ulang aplikasi E-surat. Forward engineering memungkinkan penyesuaian dan penambahan jenis surat baru secara sistematis, memastikan aplikasi tetap relevan dan adaptif terhadap kebutuhan yang terus berkembang [2], [3].

Melalui penelitian ini, diharapkan bahwa rekonstruksi E-surat akan memberikan solusi yang efektif terhadap kendala tersebut, meningkatkan fleksibilitas sistem, dan memastikan pengguna dapat dengan mudah mengelola berbagai jenis surat sesuai dengan kebutuhan spesifik mereka. Dengan demikian, peningkatan ini akan memberikan kontribusi positif terhadap efisiensi dan efektivitas proses persuratan. Dalam penelitian ini, reengineering dilakukan untuk menambahkan fitur penambahan jenis surat yang dibangun dengan bahasa pemrograman php dengan framework laravel dan terdapat perubahan basis data dari MySQL yang bertipe sql menjadi MongoDB yang bertipe nosql karena pendekatan penambahan tabel baru pada MySQL setiap kali penambahan jenis surat baru akan mengubah skema basis data [4], [5].

2 Tinjauan Literatur (or Literature Review)

Forward Reengineering adalah salah satu metode yang ada pada *reengineering software* membangun dari model atau konsep tingkat tinggi untuk membangun kompleksitas dan detail tingkat rendah. Aktivitas yang ada pada reengineering dimulai dari analysis, design, coding dan testing. Pada forward reengineering biasanya dilakukan perubahan dari bahasa pemrograman, perubahan desain, penambahan fitur dan perubahan database. Beberapa penelitian telah dilakukan untuk membahas terkait forward reengineering pada software, Berikut adalah penelitian yang dilakukan pada 5 tahun terakhir. Penelitian dilakukan oleh Mauludin dan Sidik pada tahun 2020 yang melakukan penelitian forward engineering pada aplikasi rekapitulasi nilai siswa. Latar belakang dari penelitian ini adalah beberapa fitur pada aplikasi yang lama, tidak dapat berfungsi dengan baik sehingga diusulkan untuk melakukan forward reengineering. Pada penelitian ini menghasilkan *re-design* fungsi dan bahasa pemrograman yang digunakan namun untuk databasenya masih menggunakan database yang lama dengan MySQL. Hasil dari reengineering pada penelitian tersebut menghasilkan beberapa fungsi dapat digunakan secara baik di beberapa *operating system*

3 Metode Penelitian (or Research Method)

3.1 Metode Pendekatan

Dalam Software reengineering, terdapat beberapa metode pendekatan yang umum digunakan, salah satunya adalah partial approach [6]. Dalam metode ini, hanya sebagian dari sistem yang mengalami reengineering, kemudian diintegrasikan dengan komponen atau bagian sistem yang tidak mengalami reengineering [6]. Tahapan partial approach terdiri dari beberapa phase diantaranya : sistem dibagi menjadi 2 bagian, satu bagian diidentifikasi untuk direengineering dan bagian lainnya yang tidak direengineering kemudian dilanjut ke phase berikutnya dilakukan integrasi keduanya. Pendekatan ini memiliki keuntungan mengurangi ruang lingkup reengineering dan juga akan menghemat waktu dan biaya pengembangan [6], [7]. Pada penelitian ini partial approach digunakan untuk penambahan fitur tambah jenis surat pada aplikasi E-Surat, pendekatan ini digunakan karena hanya sebagian yang akan dilakukan reengineering, seperti pada gambar 1.



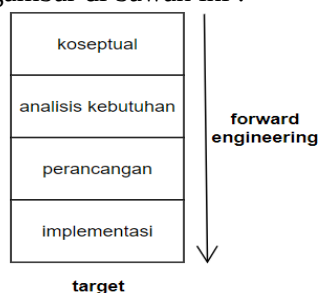
This work is licensed under a [Creative Commons Attribution 4.0 International License](http://creativecommons.org/licenses/by/4.0/).
<http://journal.stmikjayakarta.ac.id/index.php/JMIJayakarta>



Gambar 1. Partial Approach

3.2 Forward Engineering

Pada penelitian ini juga menggunakan pendekatan metode Forward Engineering yang melibatkan perpindahan dari abstraksi tingkat tinggi dan desain logis yang tidak tergantung pada implementasi untuk menuju implementasi fisik suatu system [8]. Dalam kata lain, forward engineering merupakan pembuatan rencana dan desain secara konsep sebelum membuat sesuatu dengan detail yang lebih spesifik. Forward engineering memiliki beberapa tahapan antara lain : merumuskan konsep system, requirement analysis, perancangan system, hingga mengimplementasikan desain. Sesuai pada gambar di bawah ini :



Gambar 2. Proses Forward engineering

4 Hasil dan Pembahasan (or Results and Analysis)

Pada metode forward engineering dimulai dari requirement analysis untuk mengetahui kebutuhan dari user [9]. Dalam hal ini peneliti melakukan requirement analysis atau menganalisis kebutuhan pada user aplikasi E-Surat, analisis kebutuhan dilakukan dengan metode wawancara ke user dan mendapatkan list requirement sebagai berikut :

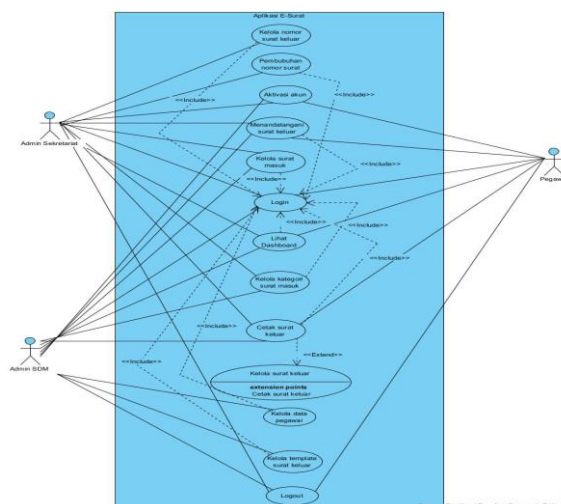
1	Fitur untuk menambahkan kategori surat
2	Fitur untuk kategori surat keluar
3	Modifikasi UI untuk tambah jenis surat baru

Kemudian dilanjut tahapan perancangan, penulis melakukan perancangan use case dan activity diagram dengan acuan perancangan yang sudah dilakukan sebelumnya. Use case adalah gambaran dari rangkaian aksi yang memiliki nilai terukur kepada seorang actor [10]. Use case juga merupakan gambaran dari fungsionalitas dan interaksi dari setiap aktor yang ada pada suatu sistem, di dalam use case terdiri dari aktor dan fungsional sistem [11]. Pada penelitian ini dilakukan analisis use case diagram yang lama, terdapat 13 use case diantaranya login, kelola nomor surat, pembubuhan nomor surat, aktivasi akun, menandatangani surat keluar, kelola surat masuk, lihat dashboard, kelola kategori surat masuk, cetak surat keluar, kelola surat keluar, kelola data pegawai, kelola template surat keluar, logout. Berikut gambaran use case sebelumnya pada gambar 3.



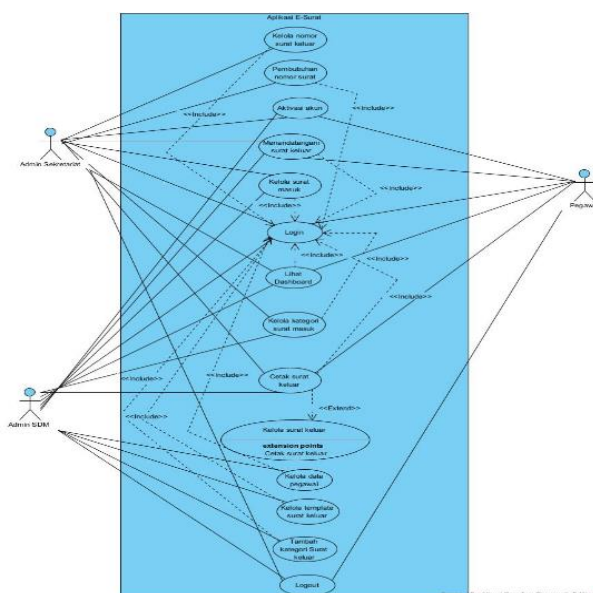
DOI

5



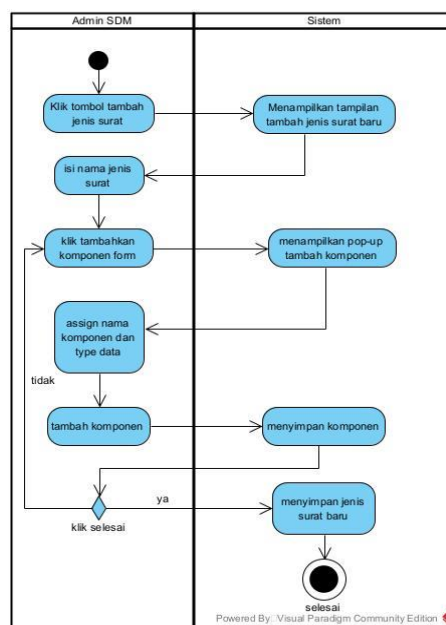
Gambar 3. Use case diagram sistem lama

Pada use case diagram di system yang baru memiliki 14 use case dengan penambahan 1 use case yaitu use case tambah jenis surat baru yang dapat dilihat pada gambar berikut



Gambar 4. Use case diagram sistem baru

Pada tahapan perancangan, peneliti juga melakukan perancangan activity diagram yang berfungsi untuk menggambarkan ataupun memodelkan aliran kerja yang terdiri dari aktifitas ataupun tindakan [12]. Activity diagram merupakan pengembangan dari use case dengan menggambarkan alur aktifitas setiap use case [13]. Pada tahapan ini digambarkan diagram activity untuk penambahan fitur yang baru yaitu activity diagram tambah jenis surat baru, sesuai pada gambar berikut :



Gambar 5. Activity Diagram Tambah Jenis Surat Baru

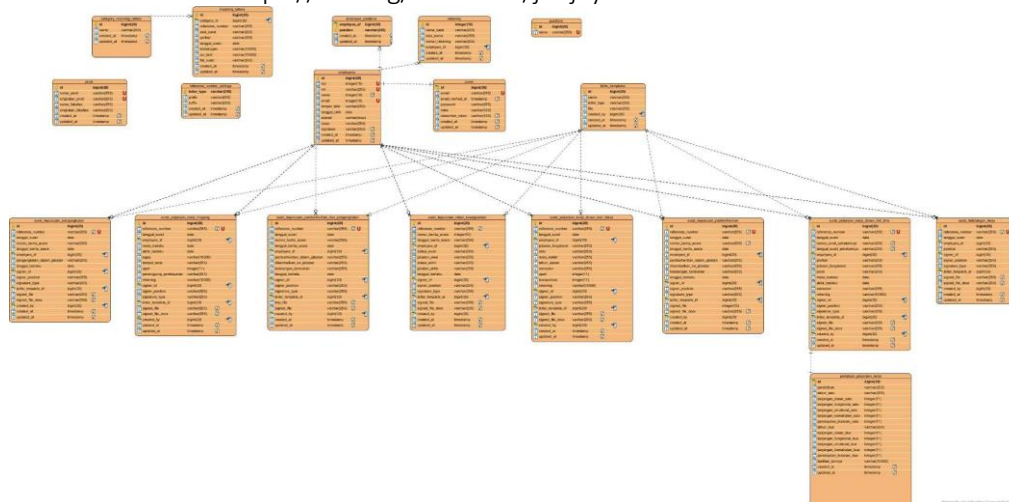
Tahapan analisis dan perancangan akan menghasilkan software documentation yang baru, yang merupakan bagian terpenting dari proses pengembangan software. proses redocumentation ini sangat dibutuhkan jika melakukan reengineering software. Reengineering adalah transformasi sistematis dari suatu system yang ada menjadi new system untuk mewujudkan kualitas yang lebih baik, kemampuan system dari segi fungsionalitas ataupun dari kinerja suatu system [14]. Pada case ini, reengineering dilakukan untuk menambahkan fungsionalitas yang baru dengan menambahkan beberapa tambahan fungsi di source code dan juga melakukan perubahan pada database. Reengineering dilakukan tidak mengubah Bahasa pemograman, sehingga perubahan pada source code tidak terlalu banyak.

Pada tahapan ini dilakukan implementasi reengineering pada fitur yang sudah ada, dengan mengubah beberapa fungsi yang ada di source code. Reengineering yang digunakan adalah forward reengineering, dimana membangun suatu sistem dari model atau konsep ke tingkat yang lebih tinggi untuk membangun kompleksitas. Tahapan dari reengineering ini terdiri dari tahapan desain kemudian dilanjutkan ke tahap implementasi [15]. Pada penelitian ini forward reengineering terfokus pada perubahan sebuah fitur yang ada di suatu sistem.

Secara prinsip, basis data merupakan kumpulan informasi yang terakumulasi selama periode waktu tertentu. Dalam konteks penggunaan sehari-hari, istilah basis data merujuk pada kumpulan data yang diorganisir dan dikelola oleh Sistem Manajemen Basis Data (DBMS) [10]. Pada sistem E-Surat ini, MySQL digunakan sebagai Sistem Manajemen Basis Data yang bersifat relasional [16].

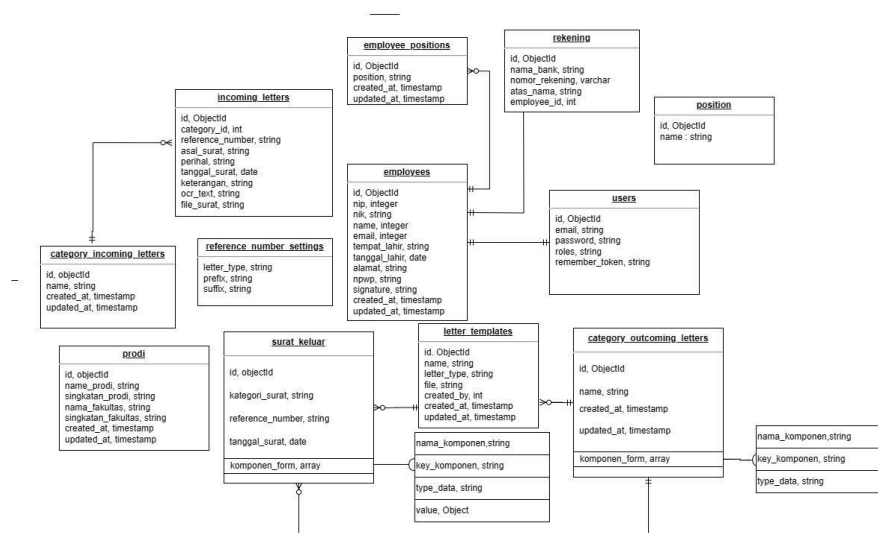
Basis data pada sistem E-Surat yang lama terdiri dari 19 tabel, dimana delapan diantaranya secara khusus berfungsi untuk menyimpan informasi terkait jenis surat. Setiap table memiliki peran dan struktur data yang unik, menciptakan suatu kerangka kerja yang kompleks untuk menyimpan dan mengelola informasi surat. Hubungan antar table dalam basis data tersebut diilustrasikan melalui Physical Data Model, yang menyajikan gambar visual tentang keterkaitan dan ketergantungan antar entitas dan basis data.

DOI: <https://doi.org/10.52362/jmijayakarta.v4i4.1595>



Gambar 6. *Physical data model* basis data lama

Database yang digunakan sistem lama tidak dapat memenuhi kebutuhan untuk menambah jenis surat baru karena setiap jenis surat memiliki atribut yang berbeda-beda. Oleh karena itu penulis memilih menggunakan MongoDB yang merupakan sistem manajemen basis data nosql sebagai basis data yang memungkinkan untuk penyimpanan data berbasis dokumen sehingga menggunakan memiliki fleksibilitas lebih tinggi dibandingkan basis data yang dibangun menggunakan sistem manajemen basis data relasional digunakan karena basis data ini dianggap sangat fleksibel dan dapat memenuhi kebutuhan untuk menambah jenis surat baru [5]. Selain itu, MongoDB juga memberikan kecepatan yang lebih sekitar 3,6 kali dari MySQL untuk menyimpan file dalam ukuran kecil. Berikut ini merupakan rancangan basis data yang baru dapat dilihat pada gambar 7.



Gambar 7. *Physical data model* basis data baru

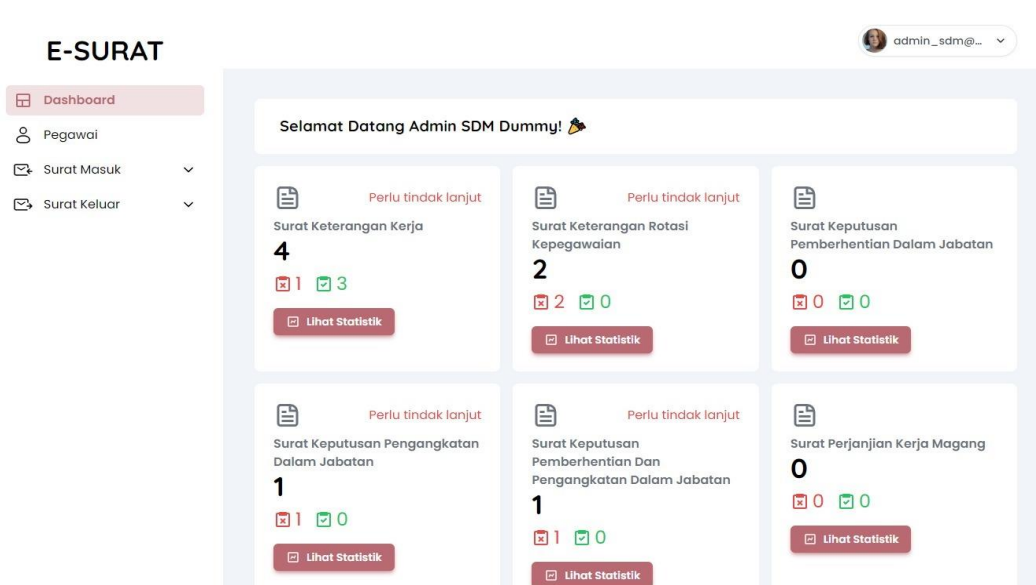
Antarmuka pengguna merujuk pada segmen komputer dan perangkat lunak yang dapat dipahami secara langsung oleh manusia melalui indra penglihatan, pendengaran, sentuhan, dan bahkan interaksi lisan [7]. Dalam konteks penerapan pada sistem e-surat yang lama, antarmuka yang lama akan dipertahankan dengan mempertimbangkan fitur-fitur yang sebagian besar tidak berubah. Selain itu, terdapat penambahan fitur untuk meningkatkan fungsionalitas dan kinerja sistem e-surat tersebut. Dengan melakukan penambahan pada fitur yang ada dalam sistem, maka diperlukan penambahan



This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).
<http://journal.stmikjayakarta.ac.id/index.php/JMIJayakarta>

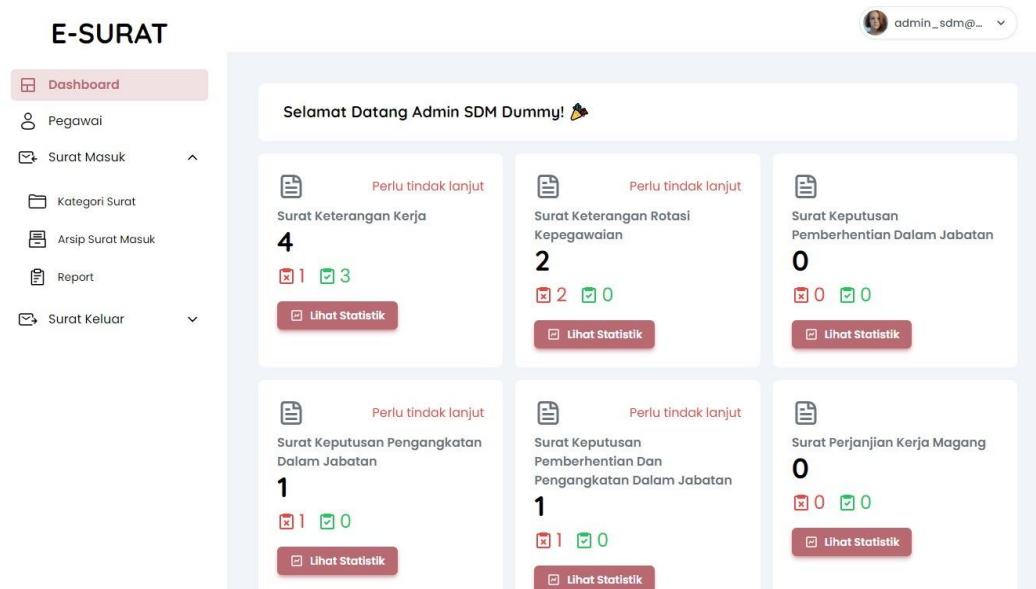
DOI: <https://doi.org/10.52362/jmijayakarta.v4i4.1595>

pada antarmuka sistem sehingga fitur baru dapat diakses oleh pengguna. Penambahan antarmuka tersebut berupa penambahan halaman, tombol, teks input dan sebagainya.



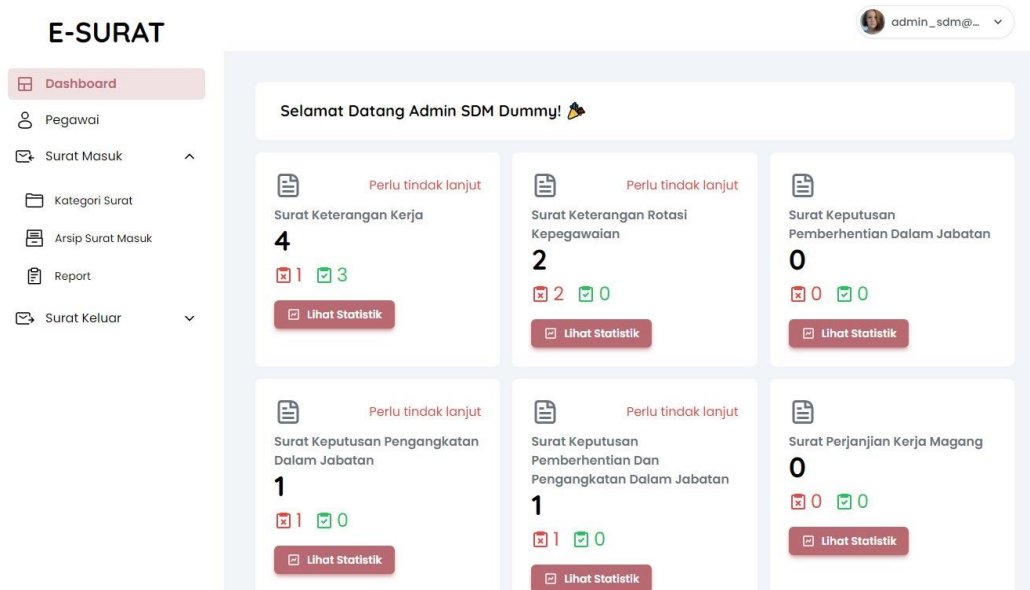
Gambar 8. Halaman *dashboard*

Pada gambar 8, terlihat halaman dashboard yang merupakan halaman utama setelah pengguna melakukan login ke dalam sistem. Halaman ini berisikan berbagai informasi statistik yang diperlukan oleh pengguna.



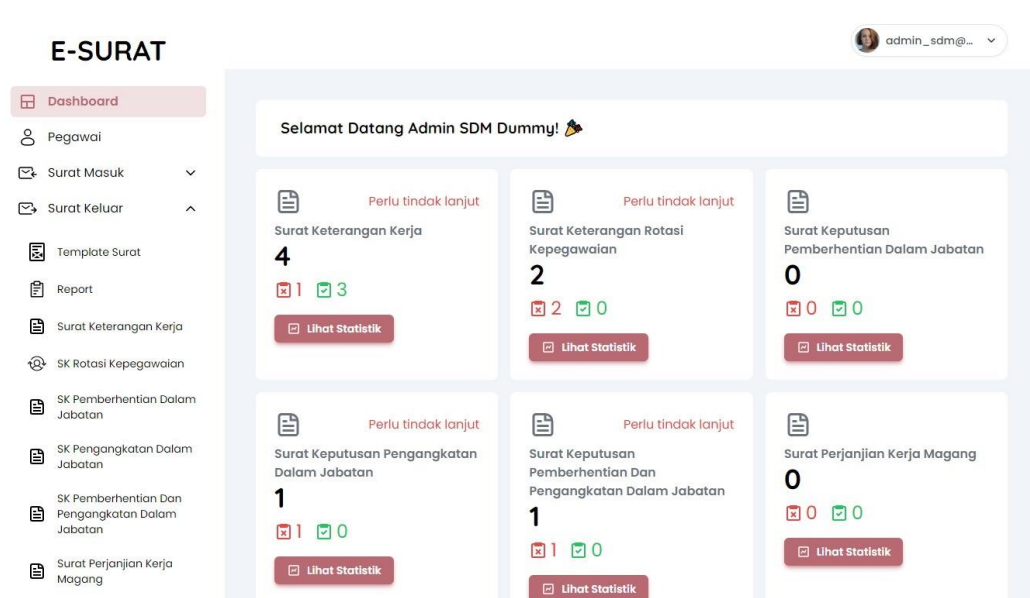
DOI: <https://doi.org/10.52362/jmijayakarta.v4i4.1595>

Pada gambar 8, terlihat halaman dashboard yang merupakan halaman utama setelah pengguna melakukan login ke dalam sistem. Halaman ini berisikan berbagai informasi statistik yang diperlukan oleh pengguna.



Gambar 9. Halaman *dashboard* dengan *sidebar* menu surat masuk ter-expand

Pada gambar 9, terlihat menu surat masuk pada sidebar yang ter-expand dan menampilkan beberapa pilihan menu di dalamnya, seperti kategori surat, arsip surat masuk dan report.

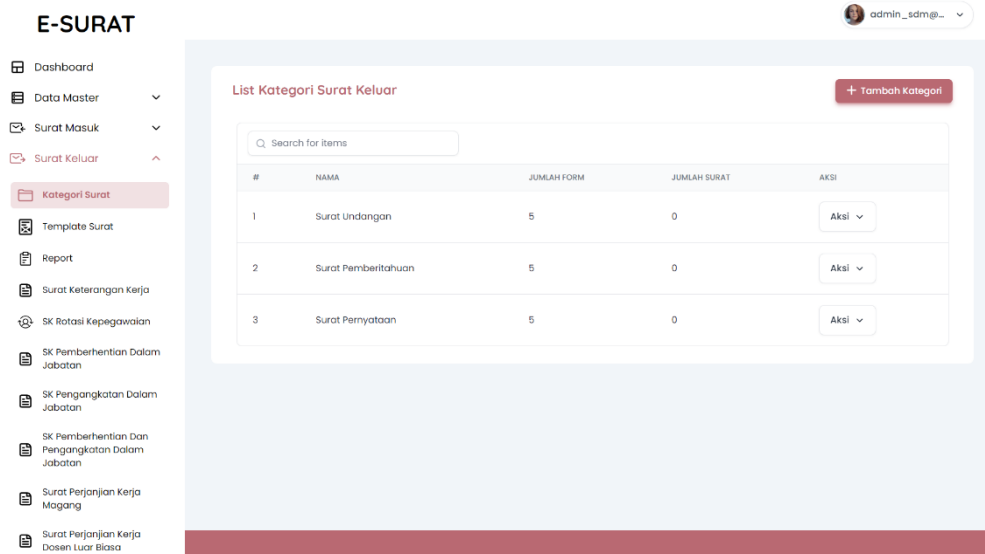


Gambar 10. Halaman *dashboard* dengan *sidebar* menu surat keluar ter-expand



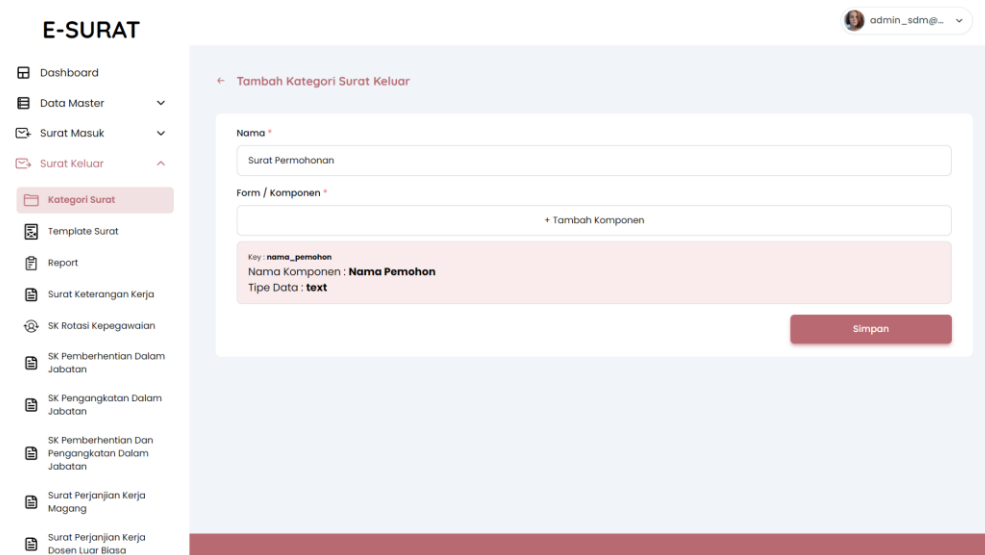
DOI: <https://doi.org/10.52362/jmijayakarta.v4i4.1595>

Pada gambar 10, terlihat menu surat keluar pada sidebar yang terexpand dan menampilkan beberapa pilihan menu di dalamnya, seperti template surat, report, dan daftar kategori surat masuk.



Gambar 11. Halaman baru list kategori surat keluar dengan tombol tambah kategori

Pada gambar 11, terlihat list kategori surat keluar yang menampilkan semua daftar kategori surat yang dapat dilakukan aksi, seperti tambah kategori, edit kategori dan hapus kategori.



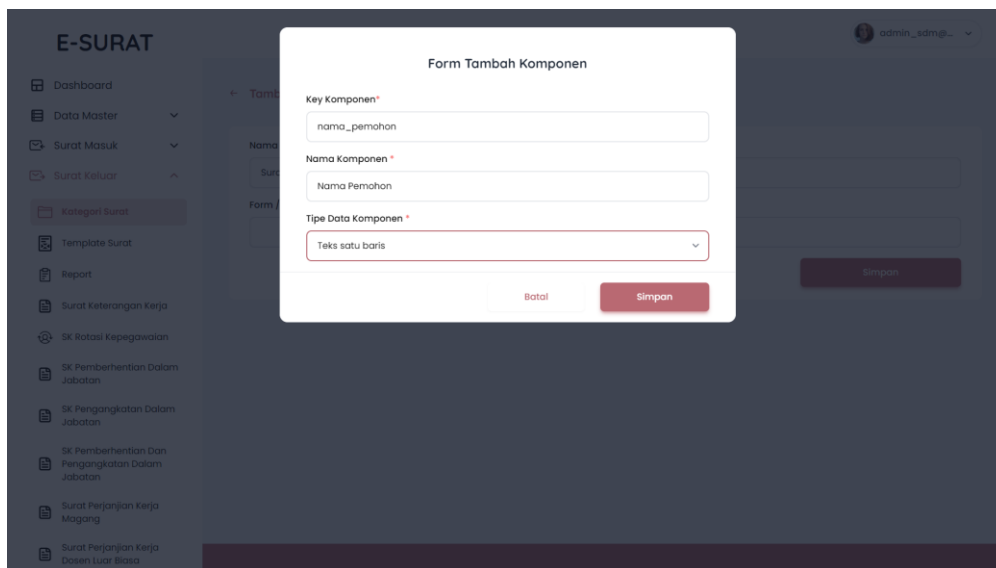
Gambar 12. Halaman baru tambah kategori surat keluar

Pada gambar 12, terlihat halaman tambah kategori surat keluar yang merupakan halaman yang muncul setelah melakukan klik pada tombol tambah kategori pada halaman list kategori surat keluar. Pada halaman ini pengguna perlu memasukkan nama kategori dan juga



DOI: <https://doi.org/10.52362/jmijayakarta.v4i4.1595>

menambahkan komponen surat yang dapat ditambahkan setelah menekan tombol tambah komponen.



Gambar 13. Tampilan *pop-up input* komponen surat

Pada gambar 13, tampak pop up form tambah komponen, dimana pengguna perlu menambahkan komponen apa yang akan masuk pada kategori surat yang telah dibuat dan juga mengatur tipe data dari komponen tersebut

5 Kesimpulan

Melalui pendekatan forward engineering, penelitian ini berhasil mengidentifikasi dan merancang ulang aplikasi e-surat sehingga dapat menangani serta menambahkan kategori surat baru dengan lebih mudah. Ini dapat meningkatkan fleksibilitas aplikasi, sesuai dengan kebutuhan pengguna yang mungkin beragam.

Saran untuk penelitian ini adalah untuk terus memonitor dan mengumpulkan umpan balik dari pengguna setelah implementasi perubahan pada aplikasi. Hal ini dapat membantu mengevaluasi sejauh mana perubahan tersebut efektif dan apakah pengguna merasa lebih terbantu dengan penambahan kategori surat yang lebih fleksibel.

Referensi

- [1] M. Majthoub, M. H. Qutqut, and Y. Odeh, "Software re-engineering: An overview," in *2018 8th International Conference on Computer Science and Information Technology (CSIT)*, 2018, pp. 266–270.
- [2] P. Polack, "Beyond algorithmic reformism: Forward engineering the designs of algorithmic systems," *Big Data Soc.*, vol. 7, no. 1, p. 2053951720913064, 2020.
- [3] R. Renaldi, B. C. Santoso, and Y. Natasya, "Tinjauan Pustaka Sistematis terhadap Basis Data MongoDB," *J. Inov. Inform.*, vol. 5, no. 2, pp. 132–142, 2020.
- [4] S. Smirnova and A. Tezuysal, *MySQL Cookbook*. "O'Reilly Media, Inc.," 2022.
- [5] K. Banker, D. Garrett, P. Bakkum, and S. Verch, *MongoDB in action: covers MongoDB version 3.0*. Simon and Schuster, 2016.
- [6] P. Tripathy and K. Naik, *Software evolution and maintenance: a practitioner's approach*. John Wiley & Sons, 2014.
- [7] H. Garcia-Molina, *Database systems: the complete book*. Pearson Education India, 2008.



This work is licensed under a [Creative Commons Attribution 4.0 International License](http://creativecommons.org/licenses/by/4.0/).
<http://journal.stmikjayakarta.ac.id/index.php/JMIJayakarta>

DOI: <https://doi.org/10.52362/jmijayakarta.v4i4.1595>

- [8] T. Mens, S. Demeyer, K. Mens, and T. Tourwé, “Evolution issues in aspect-oriented programming,” *Softw. Evol.*, pp. 203–232, 2008.
- [9] A. Taryana, A. Fadli, E. Murdyantoro, and S. R. Nurshiami, “DevOps approach embraces forward and reverse engineering,” *Int. J. Appl. Inf. Technol.*, vol. 4, no. 02, 2020.
- [10] S. Ambler, “The Elements of UML Style Cambridge University Press,” *Cambrige, Inglaterra*, 2003.
- [11] A. W. Sudio and Y. Adharani, “Pengembangan Aplikasi E-Surat Berbasis Web,” *Just IT J. Sist. Informasi, Teknol. Inf. dan Komput.*, vol. 11, no. 3, 2022.
- [12] E. K. Putra, W. Witanti, I. V. Saputri, and S. Y. Pinasty, “Perancangan sistem informasi pengarsipan surat berbasis web di kecamatan xyz,” *IKRA-ITH Inform. J. Komput. dan Inform.*, vol. 4, no. 2, pp. 55–64, 2020.
- [13] W. O. Galitz, *The essential guide to user interface design: an introduction to GUI design principles and techniques*. John Wiley & Sons, 2007.
- [14] J. K. Bergey, L. O’Brien, and D. Smith, *Options analysis for reengineering (OAR): A method for mining legacy assets*. Carnegie Mellon University, Software Engineering Institute, 2001.
- [15] S. Mauluddin and R. Sidik, “Forward Engineering in Student Mark Recapitulation Application,” in *IOP Conference Series: Materials Science and Engineering*, 2020, vol. 879, no. 1, p. 12055.
- [16] M. P. Putri *et al.*, “Sistem Manajemen Basis Data Menggunakan MySQL,” 2023.

