

BAB III

PELAKSANAAN KERJA MAGANG

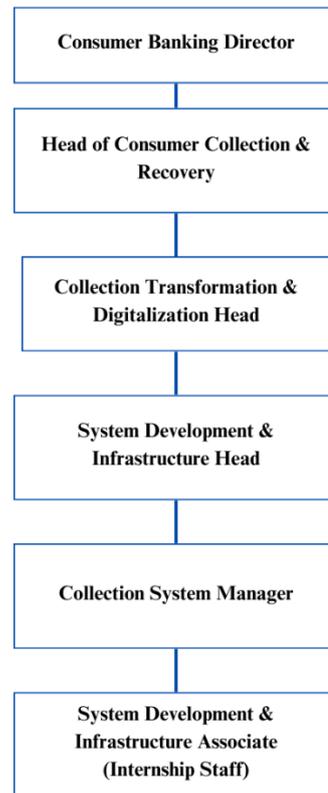
3.1 Kedudukan dan Koordinasi



Gambar 3.1 Tim System Development & Infrastructure Associate

Tim System Development & Infrastructure Associate terdiri dari 3 anggota mahasiswa yang berasal dari Universitas yang berbeda-beda dan dipersatukan melalui program magang yang diadakan oleh PT Bank XYZ melalui program Kampus Merdeka. Posisi peserta magang secara langsung bergerak dibawah naungan Pak Wantro selaku Collection System Manager dan Pak Riski Witjaksono selaku System Development & Infrastructure Head, namun posisi tersebut masih merupakan bagian dari divisi Collection Transformation & Digitalization yang

dinaungi oleh Pak Adrianus Hermawan. Berikut merupakan diagram hierarki yang lebih terperinci mengenai posisi peserta magang selama bekerja di PT Bank XYZ.

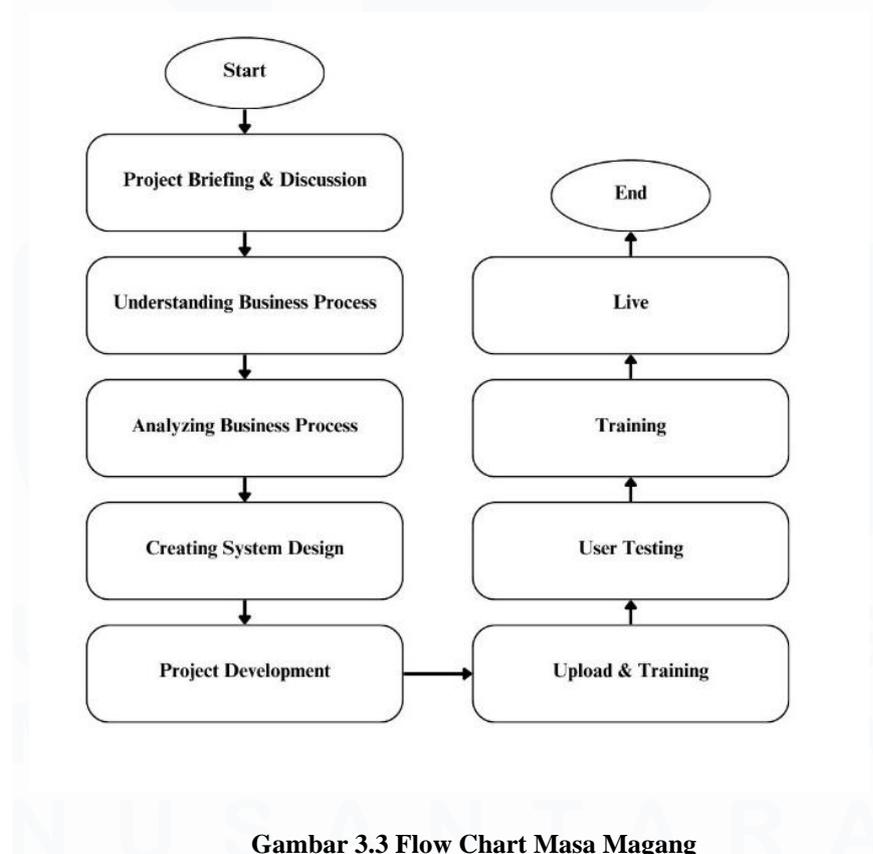


Gambar 3.2 Diagram Hierarki Posisi System Development & Infrastructure Associate

Berdasarkan diagram hierarki yang tertera pada Gambar 3.2, maka peserta magang diposisikan sesuai dengan penjelasan mengenai posisi dan naungan yang dijelaskan sebelumnya. Maka, peserta magang memiliki kewajiban untuk selalu menyampaikan laporan mengenai progress proyek harian bersama buddy dan mentor, serta menyampaikan laporan mengenai progress proyek mingguan bersama dengan kepala sub-divisi Collection Transformation & Digitalization Head.

Laporan mengenai progress proyek yang dilakukan oleh peserta magang setiap harinya harus dilakukan oleh peserta magang dengan menyampaikannya melalui media komunikasi utama yang digunakan oleh internal PT Bank XYZ yaitu Blackberry Access maupun verbal secara langsung. Dengan ini, setiap harinya

peserta magang harus menyampaikan to-do list, progres, maupun kesulitan yang sedang dialami di hari itu. Hal tersebut dapat membantu peserta magang untuk selalu tepat waktu dan mengetahui akan apa saja yang telah dilakukan pada proyek, serta membantu peserta magang untuk menerima kritik maupun saran yang dapat membantu jalannya proyek dari buddy maupun mentor. Selain itu, weekly meeting juga dilakukan untuk menyampaikan progres proyek yang telah diselesaikan bersama dengan buddy, mentor, Collection Performance Management & Strategy Head, Collection Transformation & Digitalization Head, maupun anggota divisi lainnya yang sekiranya nanti akan terpengaruhi oleh luaran proyek yang dikerjakan oleh peserta magang. Hal tersebut menjadi faktor yang sangat mendukung dalam keberlangsungannya progres proyek yang dilakukan peserta magang karena kritik dan saran yang disampaikan bersifat lebih rinci untuk meningkatkan fitur-fitur maupun tampilan sistem website yang dirancang oleh peserta magang guna memenuhi standar dan mempermudah kinerja sub-divisi nantinya.



Gambar 3.3 Flow Chart Masa Magang

Alur kerja yang telah dilalui oleh peserta magang mengimplementasikan metodologi agile, yang merupakan sebuah metodologi dimana pendekatannya melibatkan proses pemecahan proses menjadi beberapa fase dan sangat menekankan proses kolaborasi serta tidak menutup ruang untuk peningkatan terhadap proyek untuk menjadi lebih baik dan yang bersifat berkelanjutan, namun alur kerja tersebut tetap menjadi pedoman utama yang digunakan saat proyek berlangsung. Pada tahap pertama, peserta magang menerima briefing mengenai proyek oleh buddy dan melewati tahap diskusi mengenai alur dan ekspektasi terhadap proyek bersama dengan user. Setelah itu, peserta magang menggambarkan pemahaman mengenai proyek dan melakukan analisis terhadap hal-hal yang telah diterima dengan adanya template-template form beserta dengan merancang use case diagram dan activity diagram untuk lebih memvisualisasikan alur gambaran proyek yang akan direalisasikan nantinya.

Dari use case diagram dan activity diagram yang telah dibuat, peserta magang merancang system design untuk proyek agar alur yang nantinya akan terealisasi dapat lebih mudah untuk dipahami. Peserta magang melanjutkan pengerjaannya setelah gambaran alur mengenai proyek dapat mudah dipahami dengan jelas oleh buddy dan mentor, maka proses development atau perancangan mulai dilakukan. Bagian ini terlihat sangat jelas dimana metodologi agile berperan agar proyek yang dirancang dapat bekerja dengan optimal dan dapat menyediakan fitur-fitur yang memberi kontribusi inovatif pada proses alur kerja sebelumnya. Saat proyek dapat dibilang berhasil terealisasikan secara optimal, maka tahap upload pada sistem internal dan training dilakukan. Tahapan ini menjadi tahap krusial dalam majunya proyek ke tahap selanjutnya, dimana proyek akan digunakan oleh user untuk pertama kalinya. Lalu, ketika user sepakat dan nyaman atas proyek yang telah dihasilkan oleh peserta magang, maka tahap training akan dilakukan kepada anggota divisi yang sekiranya akan menggunakan luaran sistem web yang dihasilkan oleh proyek tersebut. Dengan ini, proyek yang telah dirancang berhasil dan akan digunakan secara langsung oleh divisi yang memerlukan untuk meningkatkan efektivitas dan efisiensi kerja divisi kedepannya.

3.2 Tugas dan Uraian Kerja Magang

System Development & Infrastructure Associate merupakan posisi peserta magang yang telah dipercaya untuk memiliki tanggung jawab utama dalam merealisasikan beberapa sistem web sebagai kontribusi inovatif yang dapat meningkatkan kinerja sub-direktorat Consumer Collection & Recovery secara keseluruhan pada PT Bank XYZ. Dengan ini, sistem-sistem yang dirancang oleh peserta magang juga harus memiliki fitur-fitur yang dapat memudahkan proses kerja anggota sub-direktorat divisi tersebut. Namun, dalam prosesnya seluruh tahap yang dilewati dalam proses perancangan sistem dengan bimbingan buddy dan mentor, peserta magang dibebaskan untuk menentukan tahapan metode analisis maupun penggunaan bahasa pemrograman dan tools yang digunakan hingga tahap development dengan syarat dapat memenuhi dan sesuai dengan standar perusahaan yang ada. Maka, keberlangsungan proyek dapat dikatakan sepenuhnya menjadi tanggung jawab peserta magang System Development & Infrastructure Associate selama masa program magang berlangsung.

Program magang yang diikuti oleh peserta magang dimulai secara resmi sejak tanggal 16 Februari 2024 dan akan berlangsung selama kurang lebih 4 bulan kedepan, sehingga seiring dengan laporan ini disusun, program magang masih dalam proses keberlangsungan hingga tanggal yang telah ditentukan yaitu tepat pada tanggal 30 Juni 2024. Dalam keberlangsungannya, peserta magang System Development & Infrastructure telah dipercaya untuk merancang beberapa sistem yang dapat meningkatkan kinerja divisi. Sistem yang sedang dalam proses perancangan untuk saat ini adalah sistem web berjudul eCOLL-Letter, eCOLL-Request Form, dan proyek EUC RnR CC namun laporan ini pembahasannya akan berfokus pada eCOLL-Request Form dan sekilas mengenai proyek EUC RnR CC karena merupakan sepenuhnya tanggung jawab penulis sebagai peserta magang di PT Bank XYZ. Berikut merupakan penjabaran mengenai agenda timeline yang telah dan akan dilalui oleh peserta magang selama program magang masih berlangsung di PT Bank XYZ, agenda tersebut juga telah didiskusikan dan disetujui

oleh beberapa pihak yang terlibat seperti kepala sub-direktorat, mentor, maupun buddy pada proyek ini.

Tabel 3.1 Timeline Agenda System Development & Infrastructure

No	Aktivitas	Minggu ke-	Tanggal Mulai Aktivitas	Tanggal Akhir Aktivitas
1	Orientasi Lingkungan Perusahaan dan Briefing Proyek	1	16 Februari 2024	19 Februari 2024
2	Analisis Proyek eCOLL-Letters & eCOLL-Request Form	1 - 2	19 Februari 2024	26 Februari 2024
3	Mengerjakan Proyek eCOLL-Letters & eCOLL-Request Form	2 - 14	20 Februari 2024	30 Mei 2024
4	Analisis Proyek EUC RnR CC	10	29 April 2024	30 April 2024
5	Mengerjakan Proyek EUC RnR CC	11 - 19	1 Mei 2024	24 Juni 2024
6	Weekly Meeting	2 - 18	23 Februari 2024	25 Juni 2024

7	Mempersiapkan Final Presentation	17 - 18	11 Juni 2024	18 Juni 2024
8	Final Presentation	18	25 Juni 2024	28 Juni 2024

Tabel 3.1 merepresentasikan timeline garis besar yang dilewati oleh peserta magang, dalam proses pengerjaan setiap proyeknya. Aktivitas yang tertera merupakan rangkaian yang telah dan akan dilakukan oleh peserta magang, sesuai dengan masa magang yang dijalankan. Adapun timeline detail yang fokus terhadap pengerjaan masing-masing proyek yang ada. Hal ini dilakukan agar peserta magang anggota System Development & Infrastructure Associate dapat berpacu kepada tenggat waktu serta terbiasa dengan sistem kerja yang bersifat sistematis sehari-harinya. Dengan ini, berikut merupakan penjabaran terperinci detail timeline pengerjaan eCOLL-Request Form yang akan menjadi topik utama pada laporan magang ini.

Tabel 3.2 Timeline Agenda Proyek eCOLL-Request Form

No	Aktivitas	Minggu ke-	Tanggal Mulai Aktivitas	Tanggal Akhir Aktivitas
1	Analisis As Is Process	1	16 Februari 2024	19 Februari 2024
2	Menyusun System Design & Prototype	1 - 2	16 Februari 2024	26 Februari 2024

3	Development eCOLL-Request Form	2 - 14	20 Februari 2024	30 Mei 2024
4	Integration on Dashboard	5 - 8	18 Maret 2024	12 April 2024
5	Upload Training	12 - 13	6 Mei 2024	17 Mei 2024
6	User Testing	12 - 13	8 Mei 2024	17 Mei 2024
7	Training	12 - 13	8 Mei 2024	17 Mei 2024
8	Go Live	14	20 Mei 2024	21 Mei 2024

Hasil akhir dari projek eCOLL-Request Form merupakan sebuah situs web yang tentunya dapat meningkatkan efektivitas dan efisiensi kinerja sub-direktorat Collection Performance Management & Strategy. Namun, sebelum tahap pengerjaannya, peserta magang melalui beberapa tahapan sebelum secara resmi menjadi anggota peserta magang. Maka, berikut merupakan penjabaran berdasarkan Tabel 3.1 dan Tabel 3.2 mengenai aktivitas yang telah dilewati dan akan dilewati oleh peserta magang System Development & Infrastructure Associate selama program magang berlangsung.

3.2.1 Minggu 1: Orientasi Lingkungan Perusahaan dan Briefing Proyek

Orientasi lingkungan perusahaan PT Bank XYZ dilakukan untuk secara resmi menyambut peserta magang baru dari program Kampus Merdeka yang diikuti oleh PT Bank XYZ. Kegiatan ini diselenggarakan pada tanggal 16 Februari 2024 yang dikenal dengan sebutan Onboarding di salah satu cabang PT Bank XYZ. Seluruh

peserta magang, dengan total 99 peserta, diwajibkan untuk mengikuti seluruh rangkaian acara dari kegiatan ini yang mencakup materi mengenai company profile PT Bank XYZ dan sharing session mengenai pengalaman menjadi peserta magang di PT Bank XYZ. Maka peserta magang yang saat itu masih berada di luar daerah, harus bergabung secara online agar dapat berpartisipasi juga pada kegiatan ini.

PT Bank XYZ sendiri juga menyediakan pelatihan yang disediakan dan wajib untuk dilakukan oleh peserta magang selama program magang berlangsung. Pelatihan ini dilakukan melalui sebuah aplikasi yang dapat diakses melalui perangkat elektronik pribadi masing-masing peserta magang, baik Android maupun IOS, dengan tujuan untuk membuka ataupun meningkatkan wawasan peserta magang mengenai dunia perbankan maupun keterampilan yang diperlukan di dalam dunia kerja.

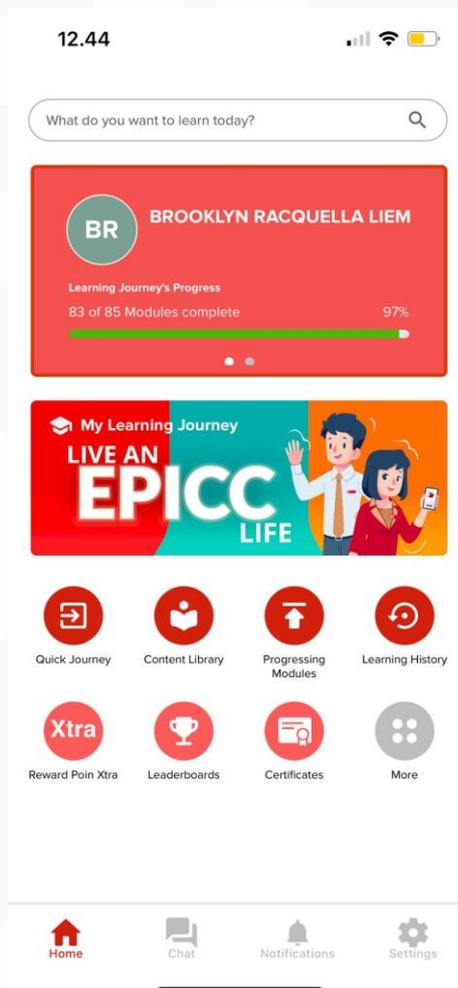


Gambar 3.4 Logo Aplikasi LOG PT Bank XYZ

Aplikasi tersebut dikenal dengan LOG, yang merupakan sebuah aplikasi sebagai sarana pelatihan dan digunakan secara aktif oleh staff PT Bank XYZ. Dengan ini, aplikasi tersebut juga digunakan oleh para peserta magang PT Bank XYZ dalam menerima wawasan baru mengenai dunia kerja dan perbankan. LOG telah dirancang untuk menyediakan 85 modul sebagai sarana pelatihan bagi peserta

magang dan wajib untuk dikerjakan selama program magang berlangsung. Modul-modul tersebut dibagi menjadi 3 bagian utama, yaitu:

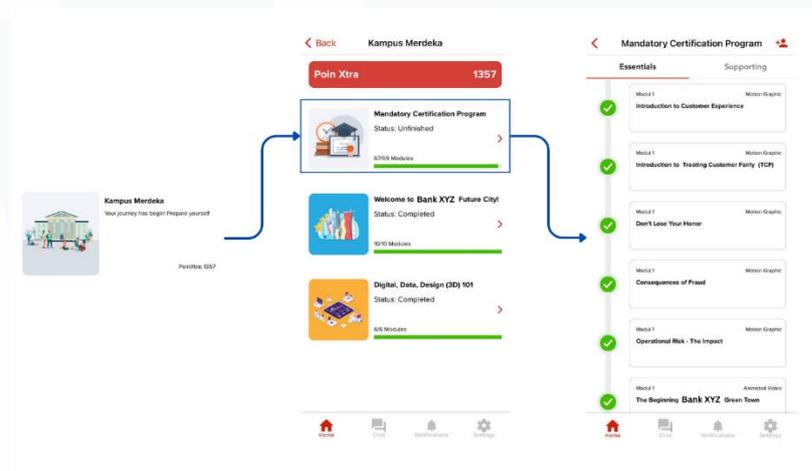
- 69 Modul Mandatory Certification Program
- 10 Modul Welcome to Bank XYZ Future City
- 6 Modul Digital, Data, Design (3D) 101



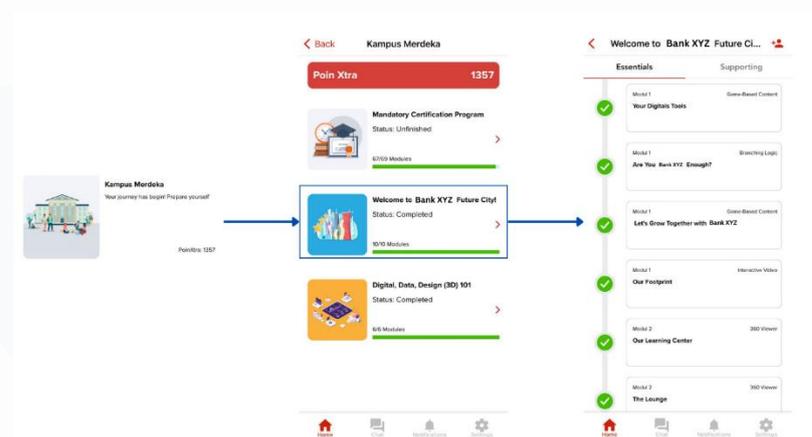
Gambar 3.5 Tampilan Aplikasi LOG

Modul-modul tersebut bersifat wajib untuk para peserta magang kerjakan selama program magang berlangsung dan menjadi salah satu syarat kelulusan program magang. Namun, berbagai fitur maupun modul yang ada di LOG ini sudah

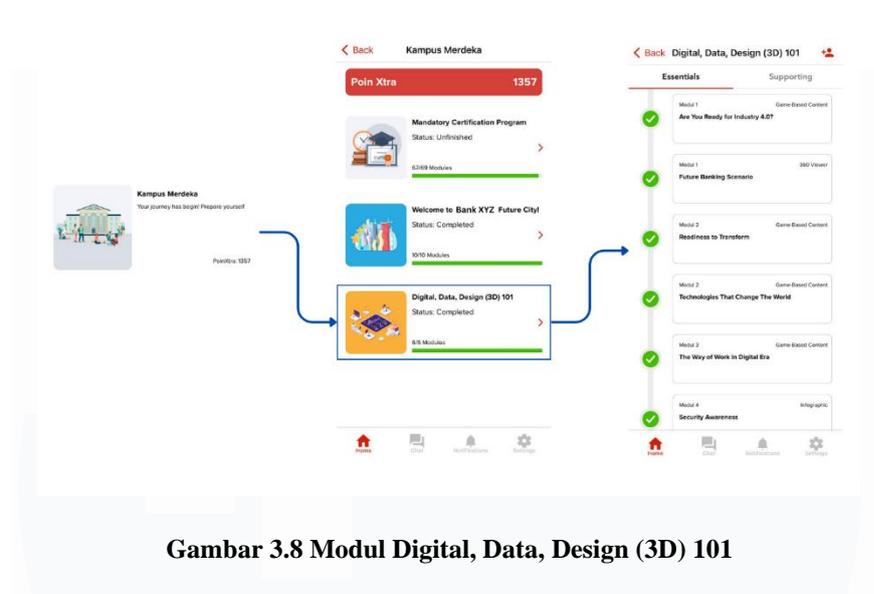
dirancang agar menarik dan mudah untuk dipahami oleh para penggunanya agar wawasan dapat diterima oleh peserta magang dengan baik melalui modul-modul yang bersifat interaktif. Berikut merupakan rincian secara garis besar modul-modul yang disegmentasikan menjadi 3 bagian utama oleh PT Bank XYZ.



Gambar 3.6 Modul Mandatory Certification Program



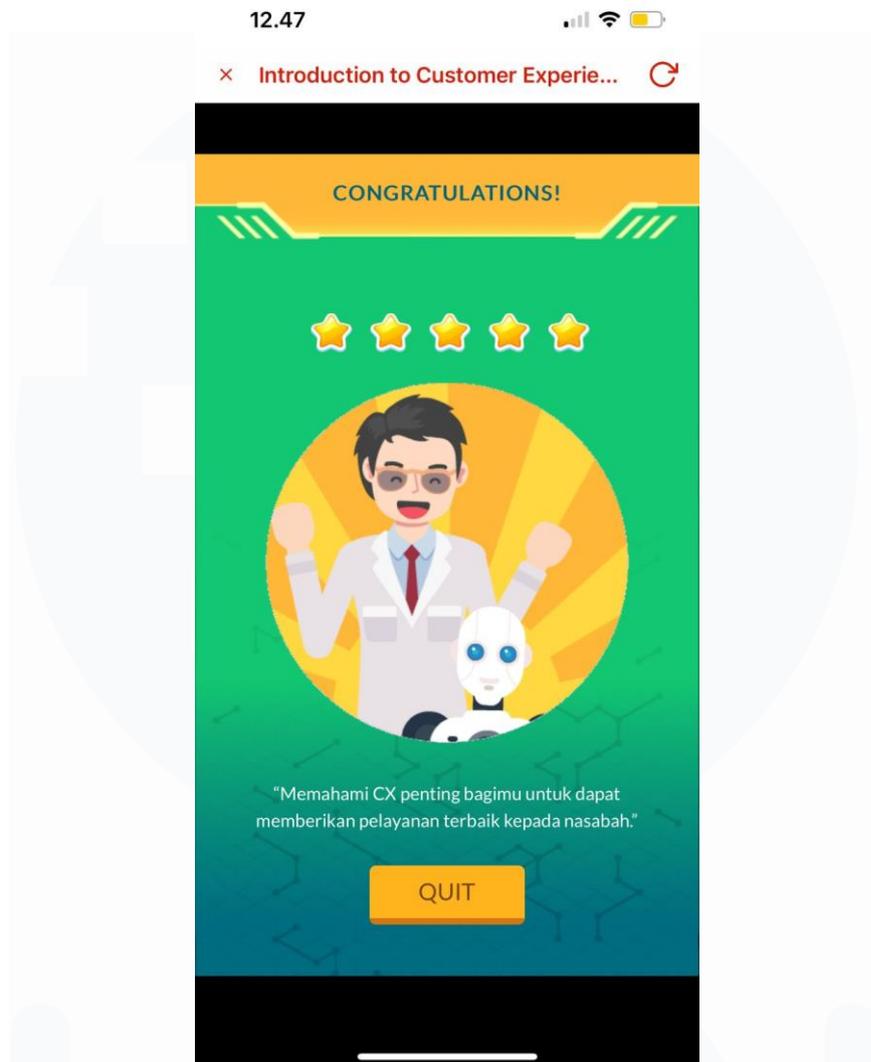
Gambar 3.7 Modul Welcome to Bank XYZ Niaga Future City



Gambar 3.8 Modul Digital, Data, Design (3D) 101

Sistem pada LOG juga mengimplementasikan fitur poin, dimana saat modul selesai maka pengguna akan menerima sejumlah poin yang sebanding dengan nilai pengerjaan modul yang telah dilakukan. Poin-poin tersebut dalam bentuk numerikal yang nantinya dapat dikalkulasikan nilai total akhir dari setiap modul yang berhasil diselesaikan. Dengan ini, sistem penilaian akhir peserta magang juga dapat ditarik melalui performa pengerjaan modul-modul yang ada pada aplikasi LOG. Berikut merupakan gambaran poin yang dapat diterima oleh peserta magang jika berhasil menyelesaikan suatu modul dengan baik.





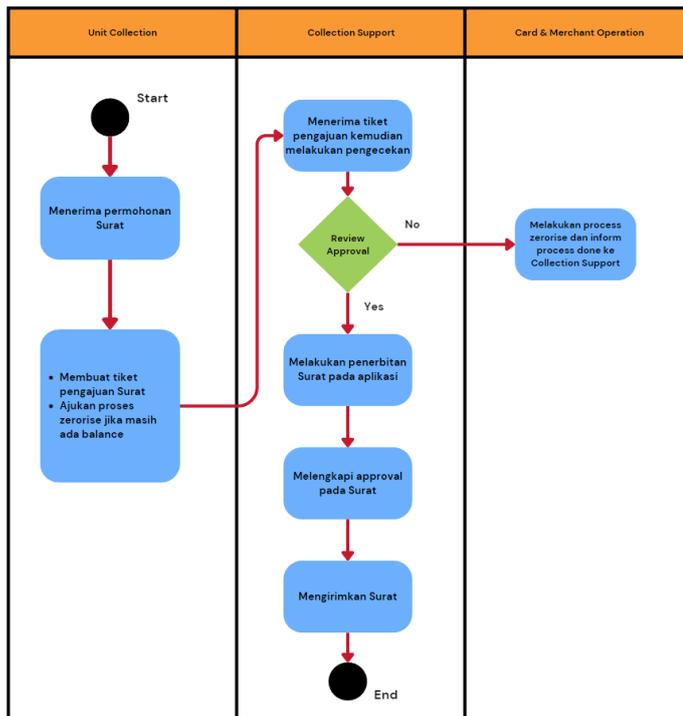
Gambar 3.9 Tampilan Hasil Menyelesaikan Modul

Selain mengenai kegiatan Onboarding dan pelatihan LOG yang didapatkan oleh peserta magang, minggu pertama ini juga merupakan masa dimana tim System Development & Infrastructure Associate bertemu dan berkenalan dengan rekan-rekan kerja pada divisi Consumer Collection & Recovery. Briefing bersama dengan mentor, buddy, dan user mengenai proyek yang akan dilakukan selama program magang berlangsung juga dilalui oleh tim System Development & Infrastructure Associate.

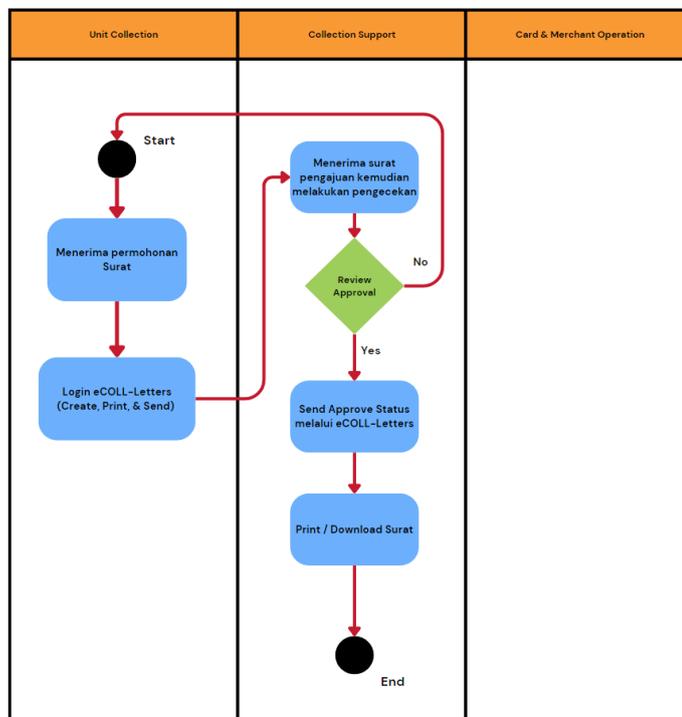
Briefing tersebut merupakan gambaran besar mengenai proyek yang akan dilaksanakan selama program magang berlangsung mengenai sebuah sistem yang dibutuhkan. Sistem tersebut harus mengimplementasikan proses digitalisasi pembuatan surat ataupun form pengajuan yang menjadi kebutuhan utama dari alur kerja divisi Consumer Collection & Recovery sehingga dapat menggantikan sistem manual yang selama ini digunakan. Dengan ini, proyek eCOLL-Letters & eCOLL-Request Form secara langsung mulai direalisasikan oleh tim System Development & Infrastructure Associate dengan bimbingan mentor dan buddy seiring dengan proses perancangannya.

3.2.2 Minggu 1-2: Analisis Proyek eCOLL-Letters & eCOLL-Request Form

Setelah melalui rangkaian acara Onboarding dan menerima briefing mengenai proyek yang akan dilakukan, peserta magang segera melakukan analisis terhadap alur proses yang sekiranya sesuai dengan alur proses sebelumnya agar dapat diimplementasikan ke dalam sistem eCOLL-Letters dan eCOLL-Request Form yang akan diajukan dan dirancang oleh tim System Development & Infrastructure Associate sehingga mampu meningkatkan efektivitas dan efisiensi sistem kinerja divisi Consumer Collection & Recovery. Berikut merupakan gambaran secara garis besar bagaimana As Is Process atau alur proses sebelumnya yang selama ini digunakan oleh divisi Consumer Collection & Recovery dan proses yang diajukan sebelum proyek ini dilakukan.



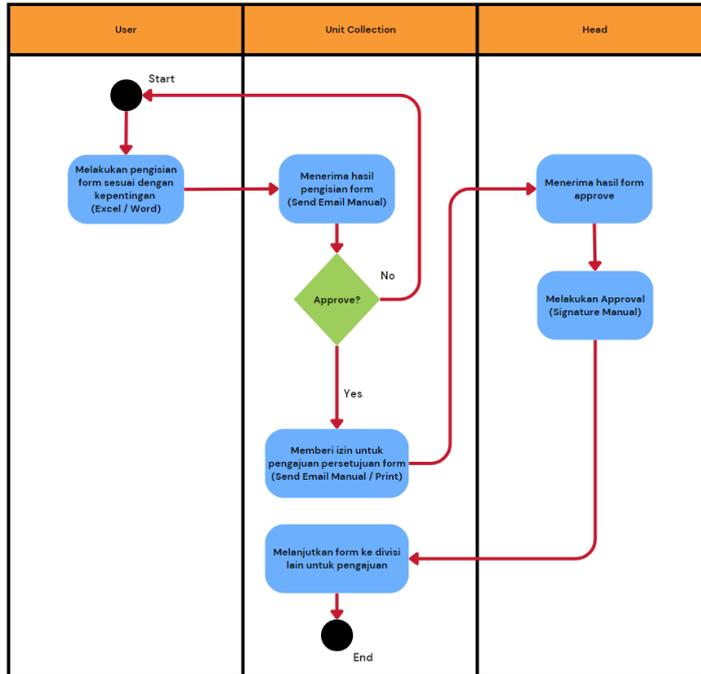
Gambar 3.10 As Is Process eCOLL-Letters



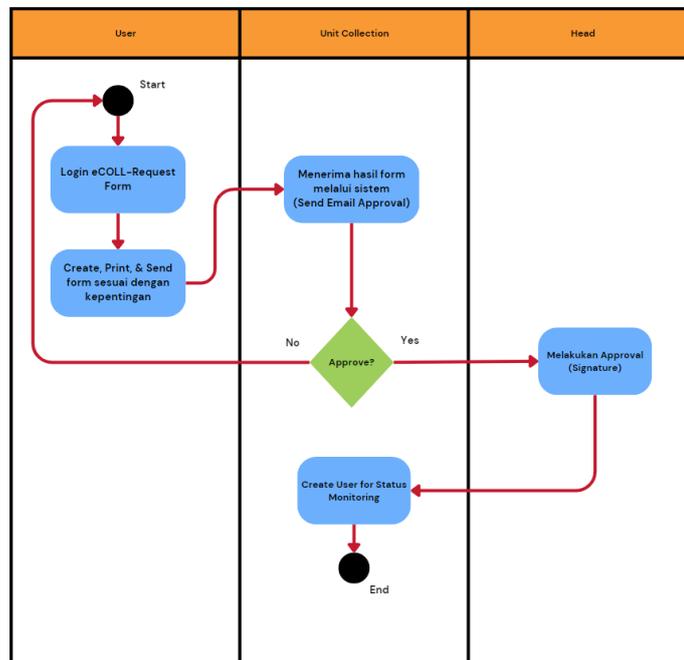
Gambar 3.11 Proses eCOLL-Letters

Pada As Is Process eCOLL-Letters, seorang user, yang dimana merupakan salah satu anggota dari Unit Collection, akan menerima sebuah permohonan mengenai surat yang ingin diajukan. Lalu, user harus membuat sebuah tiket mengenai pengajuan surat secara manual menggunakan tools yang sekiranya dapat membantu proses pembuatan surat dan mengajukan sebuah proses yang dinamakan zerorise jika masih terindikasi memiliki balance. Tiket tersebut akan diteruskan ke Collection Support yang akan mengecek dan memberi persetujuan terhadap tiket tersebut. Jika tiket tersebut mendapat persetujuan, maka akan dilakukan sebuah penerbitan terhadap surat dan akan dilanjutkan hingga proses pengiriman surat, namun jika tiket tersebut tidak menerima persetujuan maka akan kembali dilakukan proses zerorise dan secara langsung Unit Collection akan diberi pernyataan mengenai tindakan tersebut.

Sedangkan dengan eCOLL-Letters, proses tersebut akan tentunya menjadi lebih efisien. Hal ini dibuktikan dengan tidak lagi terlibatnya bagian Card & Merchant Operation di dalam alur prosesnya, serta seluruh proses maker & checker dapat dilakukan di dalam satu sistem tanpa harus menggunakan tools lainnya yang sekiranya dapat membantu proses persetujuan. Selain itu, user dapat dengan mudah mengakses sistem eCOLL-Letters yang sudah memiliki segenap fitur yang dapat membantu user dalam membuat surat dari tiket permohonan yang ada. Lalu proses pengecekan atau maker & checker juga sudah terintegrasi di dalam sistem sehingga user tidak perlu memberi pemberitahuan menggunakan tools lainnya, melainkan hal tersebut dapat dilakukan di dalam sistem eCOLL-Letters hingga penerimaan status setuju ataupun tidak setuju terhadap surat yang diajukan. Collection Support juga dapat dengan mudah mengabari status persetujuan yang telah ditentukan kepada user hingga mengunduh surat tersebut hanya dengan memanfaatkan fitur-fitur yang disediakan oleh eCOLL-Letters.



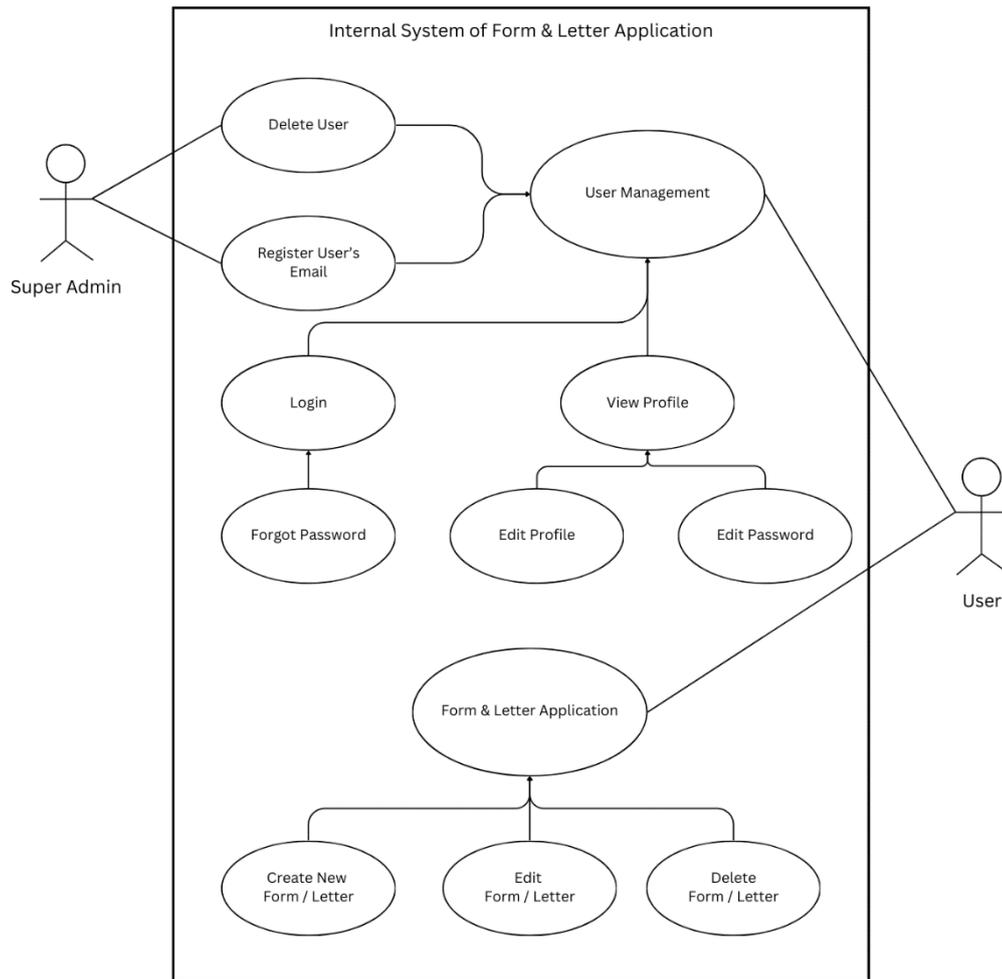
Gambar 3.12 As Is Process eCOLL-Request Form



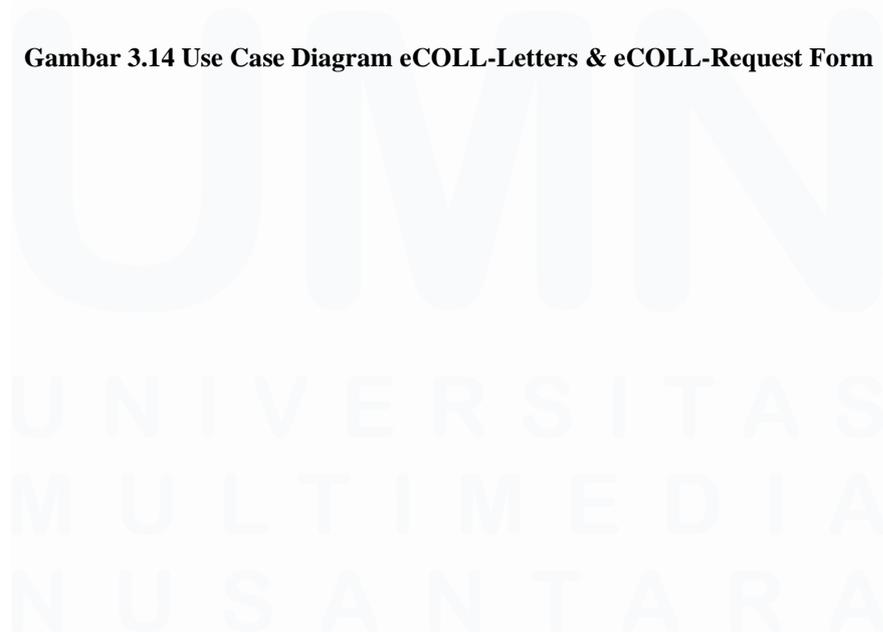
Gambar 3.13 Proses eCOLL-Request Form

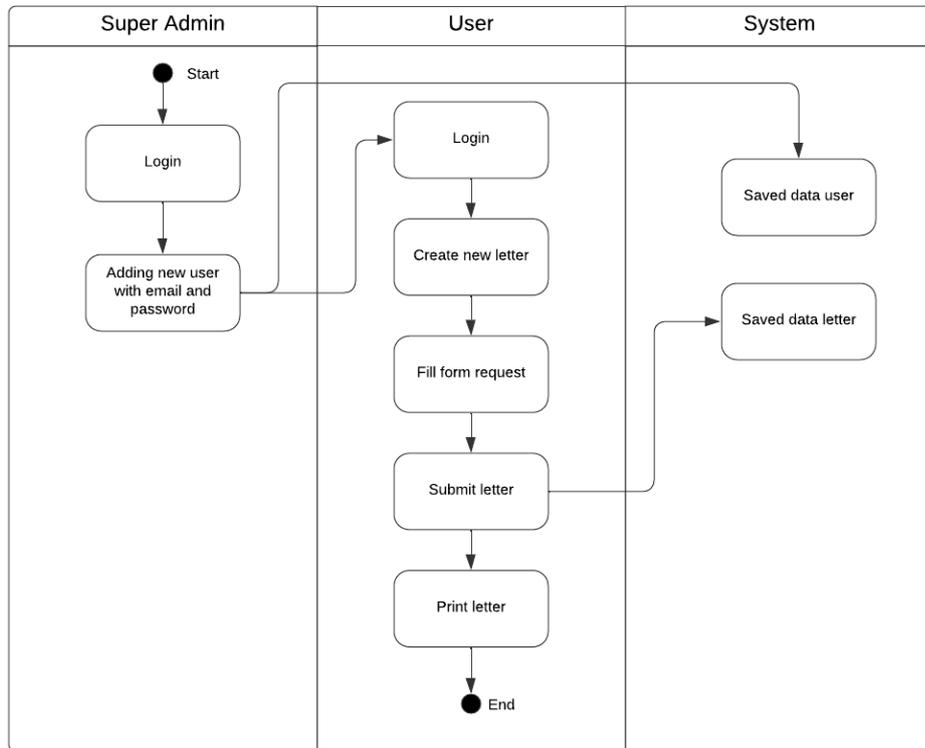
Pada As Is Process eCOLL-Request Form, proses ini juga bersifat manual walaupun bagian-bagiannya tidak memiliki kompleksitas yang sebanding dengan As Is Process eCOLL-Letters. Namun dengan mengimplementasikan eCOLL-Request Form, dapat meningkatkan efisiensi dan efektivitas pengerjaan. Pada proses sebelumnya, memanfaatkan tools seperti Excel ataupun Word dalam membuat form User ID Management, Change Rules Classification-Allocation Management, dan Request Action Code merupakan hal yang wajib. Lalu ketika form tersebut telah dibuat, user harus mengirim hasil pengisian form tersebut kepada bagian dari Unit Collection untuk persetujuan dengan mengirimkan email secara manual dan tentunya menunggu hasil persetujuan dari form yang telah diajukan. Jika form tersebut berhasil untuk menerima persetujuan, maka proses akan dilanjutkan untuk di tanda tangan sebagai tanda persetujuan atas form yang diajukan, namun jika tidak, akan dikembalikan kembali ke user dengan revisi ataupun tidak. Lalu ketika form tersebut sudah memiliki tanda persejutuannya, maka dapat dilanjutkan ke divisi yang bersangkutan untuk mengeksekusi permintaan dari form tersebut.

Sedangkan dengan eCOLL-Request Form tentunya seperti yang sudah disinggung sebelumnya, dapat mendukung waktu alur proses yang ada menjadi lebih singkat karena hanya dengan memanfaatkan fitur-fitur yang terdapat pada sistem eCOLL-Request Form, sistem ini dapat membantu proses pengajuan form yang diinginkan hanya dengan memanfaatkan satu sistem yang sudah terintegrasi hingga menerima persetujuan terhadap form yang telah diajukan. Berikut juga merupakan Use Case Diagram dan Activity Diagram yang telah digunakan oleh tim untuk mengajukan alur proses eCOLL-Letters dan eCOLL-Request Form dan tentunya mendukung pernyataan alur proses yang lebih efektif dan efisien saat mengimplementasikan dan memanfaatkan fitur-fitur yang sudah terintegrasi dan dirancang agar dapat memenuhi kebutuhan dan menggantikan As Is Process yang ada dari kedua sistem tersebut.



Gambar 3.14 Use Case Diagram eCOLL-Letters & eCOLL-Request Form

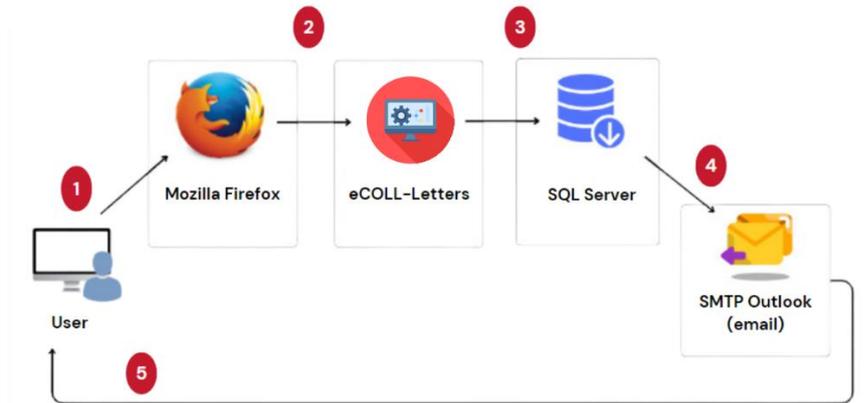




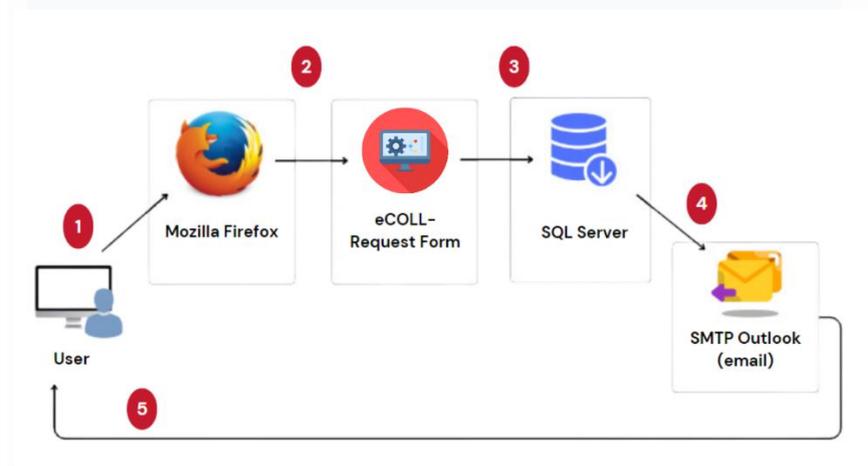
Gambar 3.15 Activity Diagram eCOLL-Letters & eCOLL-Request Form

3.2.3 Minggu 2-14: Mengerjakan Proyek eCOLL-Request Form

Setelah melalui tahap analisis yang diperlukan untuk memulai proyek tim System Development & Infrastructure Associate, selanjutnya tim memasuki minggu kedua dimana proses development proyek mulai dilakukan. Tahap ini dimulai dengan menyusun system design & prototype yang diperlukan agar menjadi pedoman selama tahap development proyek berlangsung. Berikut merupakan system design yang telah disusun oleh tim untuk memudahkan pemahaman saat mempresentasikan materi dengan berpacu pada Use Case Diagram maupun Activity Diagram yang telah dibuat.



Gambar 3.16 System Design eCOLL-Letters

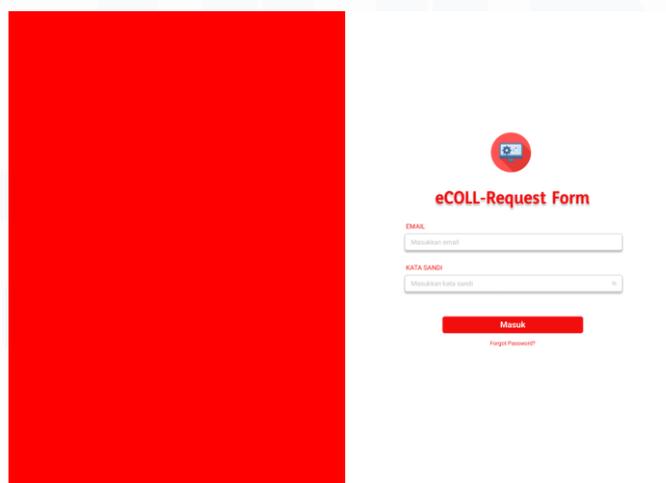


Gambar 3.17 System Design eCOLL-Request Form

System Design eCOLL-Letters dan eCOLL-Request Form sesuai dengan Gambar 3.16 dan Gambar 3.17 memiliki alur proses yang sama, hanya saja memiliki sistem web yang berbeda. Proses sistem diawali ketika user membuka web browser yang menjadi standar perusahaan yaitu Mozilla Firefox. Lalu saat situs web eCOLL-Letters atau eCOLL-Request Form yang akan digunakan sudah berhasil tertampil di layar user, maka user harus melakukan login sesuai dengan akun yang dimiliki dan dapat memulai proses pembuatan form ataupun surat hasil permohonan yang telah diterima sebelumnya. Saat user memasukkan data-data untuk

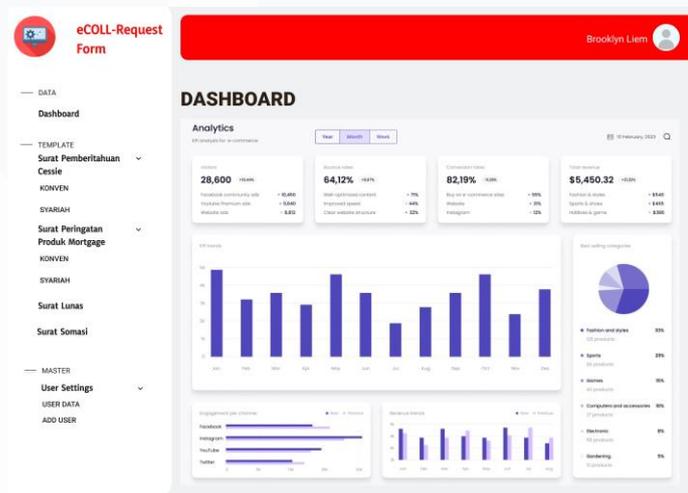
kepentingan pemenuhan form atau surat tersebut, secara langsung surat ataupun form yang disimpan beserta dengan data-datanya untuk saat ini tersimpan di database lokal melalui SQL Server. Setelah form atau surat berhasil tersimpan dan user dapat melihat hasil akhirnya, maka proses pengajuan persetujuan akan secara langsung berlangsung pada sistem. Maka, implementasi dari proses Send to Email untuk pengajuan persetujuan akan secara otomatis terjadi dan alur proses yang sama akan selalu terjadi saat user memanfaatkan sistem eCOLL-Letters atau eCOLL-Request Form dalam memenuhi kepentingan pembuatan form atau surat yang diperlukan.

Selain System Design yang dapat digunakan sebagai media pedoman alur sistem dalam perancangan eCOLL-Letters dan eCOLL-Request Form. Tampilan sistem juga menjadi aspek penting dalam proses perancangan sistem ini, maka prototype eCOLL-Letters dan eCOLL-Request Form juga dirancang sebagai gambaran awal bagi tim agar memiliki bayangan mengenai apa yang diinginkan dalam segi tampilan di dalam sistem. Maka berikut merupakan rancangan desain awal tampilan sistem eCOLL-Request Form, sistem yang menjadi tanggung jawab utama penulis, yang telah dirancang oleh tim secara bersama-sama menggunakan Figma sebagai tools pendukung.



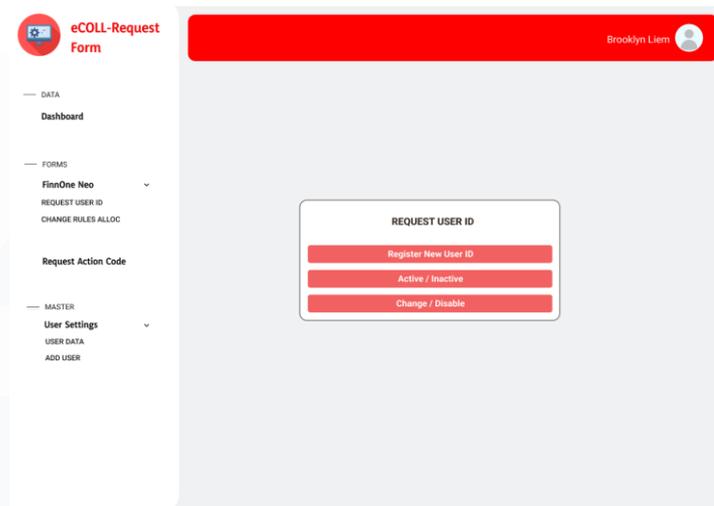
Gambar 3.18 Page Login eCOLL-Request Form

Gambar 3.18 merupakan tampilan desain page login pada eCOLL-Request Form. Tampilan pada kedua sistem tersebut tentunya akan memiliki kemiripan di hampir seluruh komponen yang ada, seperti pilihan warna atau font. Hal ini dilakukan sebagai indikasi bahwa proyek tersebut merupakan hasil yang dilakukan oleh proyek program magang dari angkatan tim System Development & Infrastructure Associate Batch ke-6. Sebagai contoh pada tampilan page login ini, hanya judul dari sistem tersebut yang menunjukkan perbedaan pada sistem eCOLL-Letters.



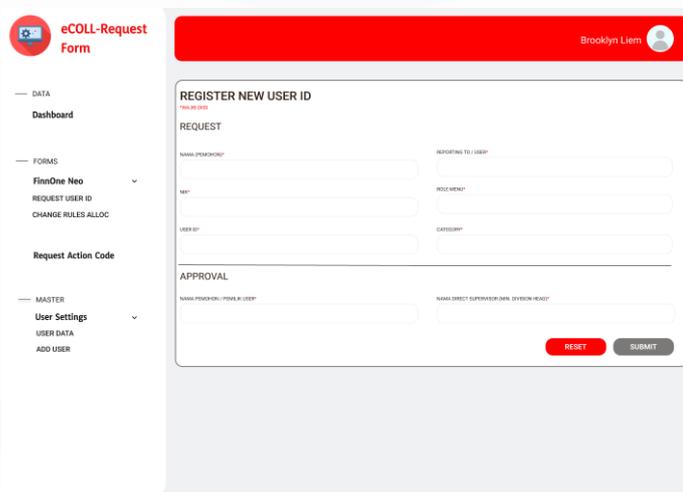
Gambar 3.19 Page Dashboard eCOLL-Request Form

Gambar 3.19 merupakan tampilan desain halaman utama atau dashboard pada eCOLL-Request Form. Dashboard akan menjadi tampilan halaman pertama yang nantinya akan dilihat oleh user yang melakukan login pada eCOLL-Request Form. Maka, visualisasi grafik yang dapat menggambarkan data-data yang sudah ada di dalam sistem disisipkan untuk memudahkan user dalam mengetahui progress sistem eCOLL-Request Form setiap harinya.



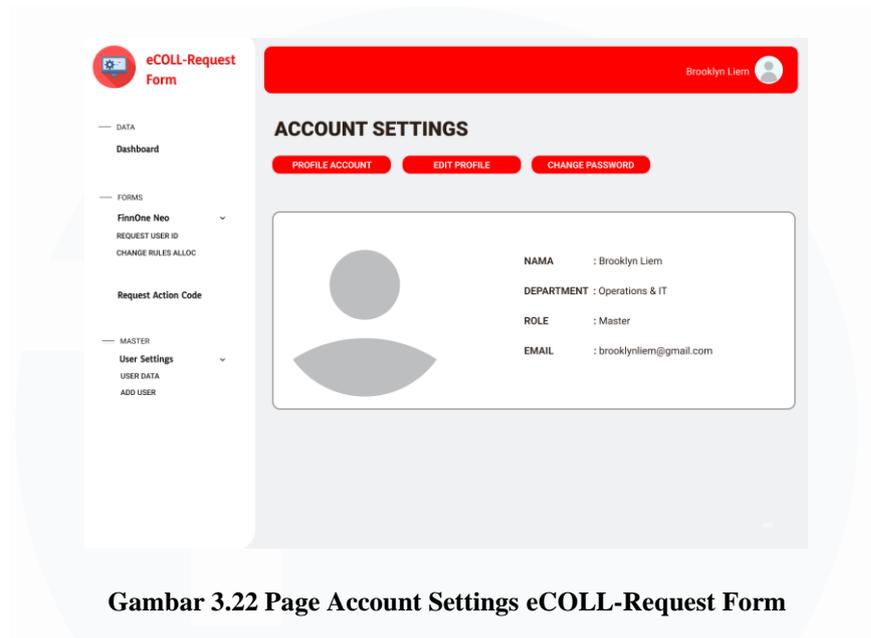
Gambar 3.20 Page Menu Pop Up eCOLL-Request Form

Gambar 3.20 merupakan tampilan desain dari menu pop up yang nantinya akan digunakan untuk memudahkan user dalam memilih form apa yang ingin dibuat di dalam sistem eCOLL-Request Form.



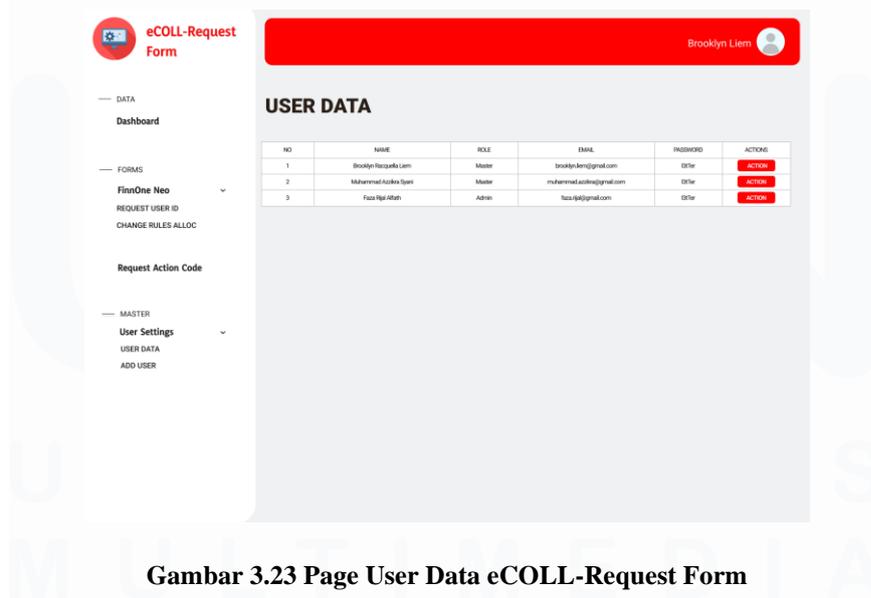
Gambar 3.21 Page Input Form eCOLL-Request Form

Gambar 3.21 merupakan salah satu contoh dari tampilan desain input form yang ada pada eCOLL-Request Form. Disini tim nantinya akan memanfaatkan shortcuts yang sekiranya dapat mempersingkat proses dan memudahkan pengisian form dengan eCOLL-Request Form.



Gambar 3.22 Page Account Settings eCOLL-Request Form

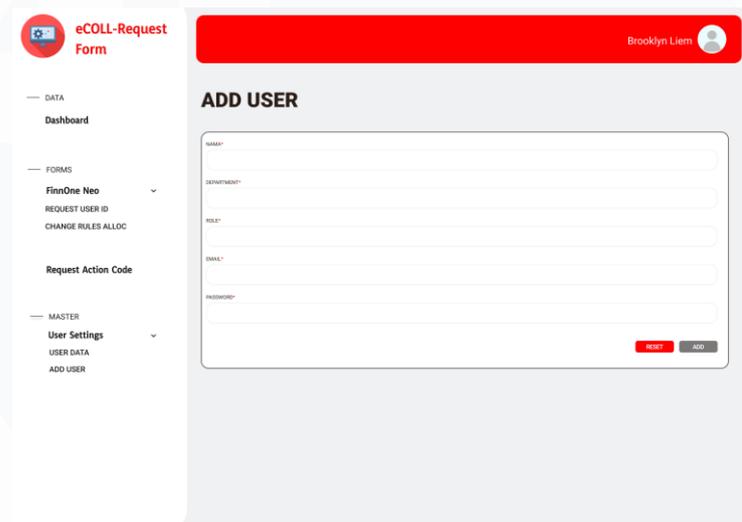
Gambar 3.22 merupakan tampilan desain halaman pengaturan akun. Halaman ini akan berisikan informasi-informasi pribadi yang sekiranya diperlukan dalam proses pengisian input form pada eCOLL-Request Form.



Gambar 3.23 Page User Data eCOLL-Request Form

Gambar 3.23 merupakan tampilan desain halaman data user, dimana nantinya akan dimiliki akses oleh Admin saja. Hal ini dilakukan agar admin dapat selalu

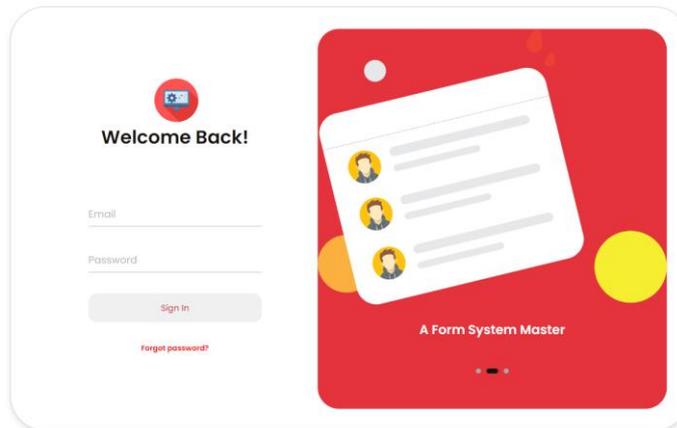
mengatur dan mengawasi data-data user yang ada di dalam sistem. Halaman ini nantinya akan dibatasi agar tidak semua pengguna eCOLL-Request Form memiliki akses karena terdapat data-data pribadi yang bersifat rahasia.

The image shows a web application interface for 'eCOLL-Request Form'. At the top, there is a red navigation bar with the application logo on the left and the user's name 'Brooklyn Liem' on the right. A sidebar on the left side of the page lists various menu items: 'Dashboard' under 'DATA', 'FinnOne Neo' and 'REQUEST USER ID' under 'FORMS', and 'Request Action Code' under 'Request Action Code'. Under the 'MASTER' section, there are 'User Settings', 'USER DATA', and 'ADD USER'. The main content area is titled 'ADD USER' and contains a form with several input fields: 'NAME', 'DEPARTMENT', 'ROLE', 'EMAIL', and 'PASSWORD'. There are also 'RESET' and 'ADD' buttons at the bottom right of the form.

Gambar 3.24 Page Add User eCOLL-Request Form

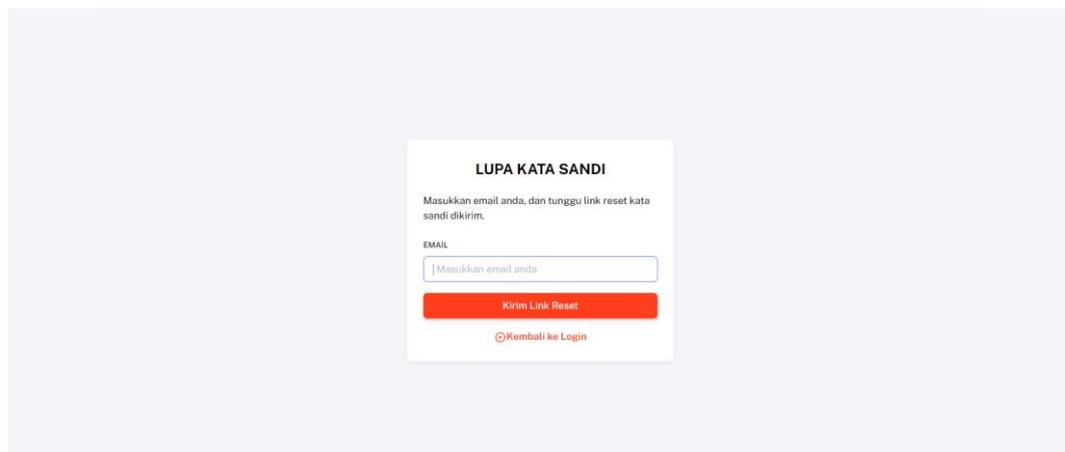
Gambar 3.24 merupakan tampilan desain halaman penambahan user, dimana tentunya akan hanya dimiliki akses oleh Admin untuk memudahkan admin dalam melakukan input data-data user nantinya yang sekiranya harus memiliki akun pada eCOLL-Request Form.

Setelah menyusun bagian-bagian yang sekiranya diperlukan dan dapat membantu proses perancangan sistem, tim mulai memasuki tahap development dimana proses perancangan sistem dimulai. Salah satu tools yang digunakan dalam proses perancangan sistem adalah Visual Studio Code sebagai source-code editor dan untuk saat ini PostgreSQL sebagai alat untuk pengolahan database ke local host pada sistem eCOLL-Request Form. Bahasa pemrograman yang digunakan untuk merancang sistem adalah PHP beserta dengan framework Laravel untuk merealisasikan fitur-fitur yang dapat memenuhi standar proses kerja divisi. Maka, berikut merupakan penjelasan mengenai codingan serta tampilan sistem eCOLL-Request Form berdasarkan fitur-fitur yang disediakan oleh eCOLL-Request Form.



Gambar 3.25 Page Login eCOLL-Request Form

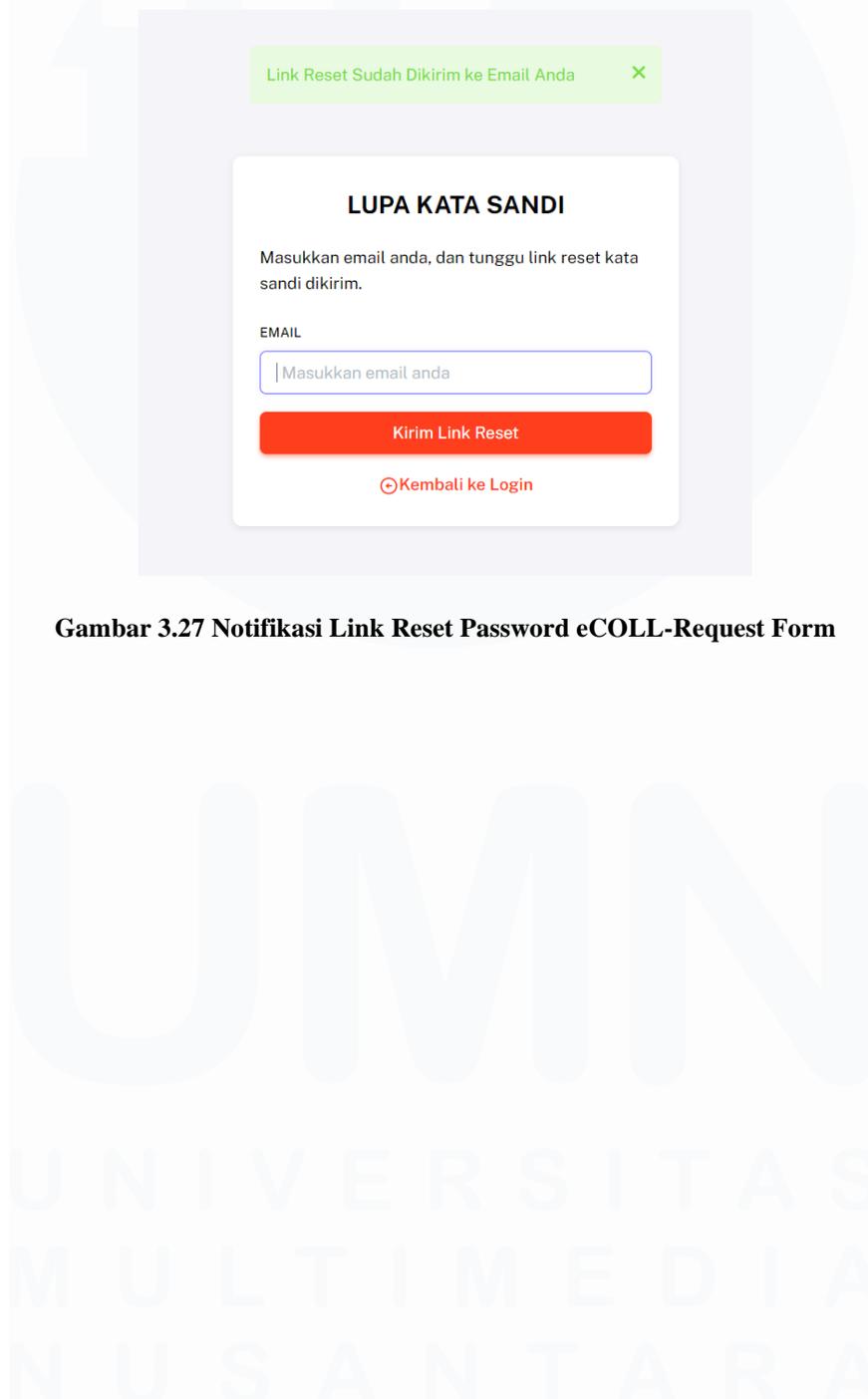
Gambar 3.25 merupakan page Login dari sistem eCOLL-Request Form. Ketika user membuka sistem, page tersebut merupakan tampilan pertama yang akan dilihat oleh user saat mengakses sistem eCOLL-Request Form. Tampilan Login bersikan bagian input dimana user harus mengisi email dan password sesuai dengan akun yang telah terdaftar di dalam sistem. Selain itu, terdapat tombol forgot password? yang dapat digunakan jika user lupa akan password yang dimiliki saat melakukan Login pada sistem dan ingin mengubah password yang dimiliki. Berikut merupakan tampilan page forgot password? yang dapat diakses oleh user.



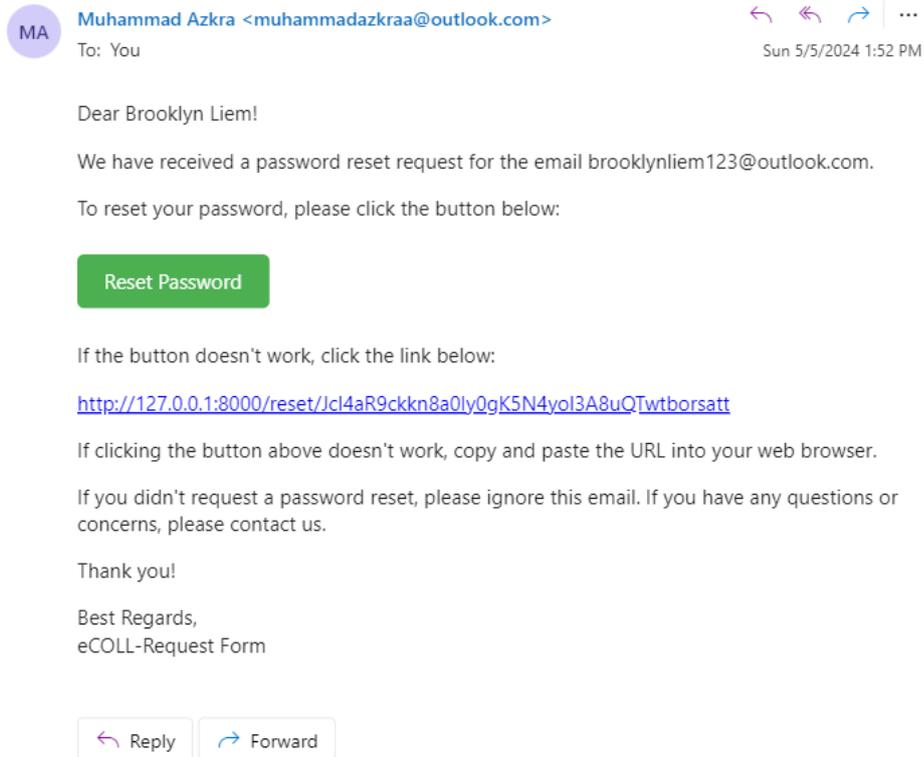
Gambar 3.26 Page Forgot Password eCOLL-Request Form

Pada page tersebut terdapat sebuah tampilan card yang dapat diakses oleh user untuk melakukan input mengenai keterangan email yang ingin digunakan

untuk tindakan perubahan password akun yang dimiliki. Ketika user telah menginput email yang dimiliki untuk tindakan tersebut, berikut merupakan tampilan notifikasi dan email yang dapat diterima user secara otomatis ketika user memilih untuk memilih tombol link reset password.

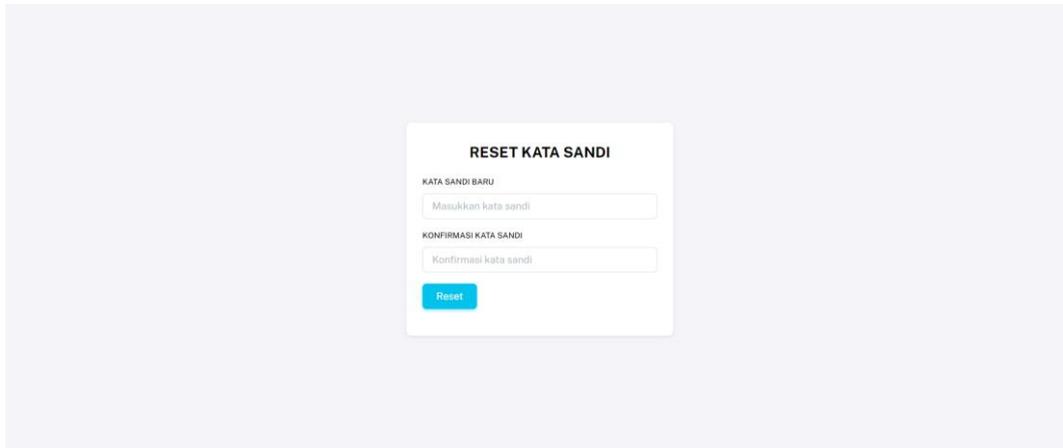


Gambar 3.27 Notifikasi Link Reset Password eCOLL-Request Form



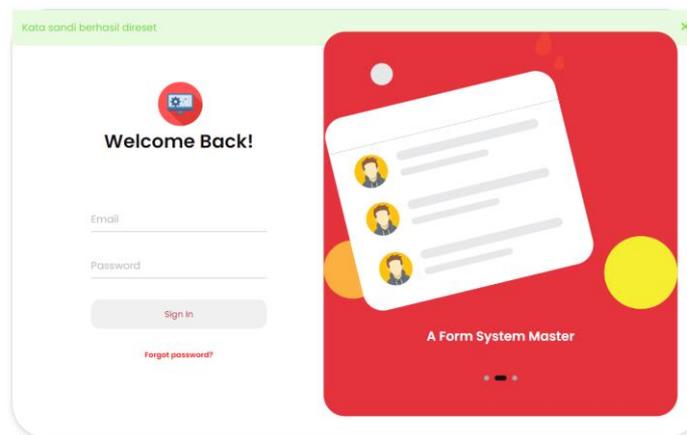
Gambar 3.28 Email Link Reset Password eCOLL-Request Form

Untuk saat ini, user akan masih menerima email dari akun Admin yang terdaftar ke dalam sistem namun hal ini akan digantikan nantinya ketika proyek ini dapat diakses dan di upload pada sistem internal PT Bank XYZ. Lalu user dapat memilih untuk melanjutkan tindakan pergantian password dengan memilih tombol Reset Password ataupun link yang telah disisipkan pada body email tersebut. Berikut merupakan tampilan yang akan dilihat oleh user ketika memilih aksi tersebut.



Gambar 3.29 Page Reset Password eCOLL-Request Form

Sesuai dengan Gambar 3.29, user dapat secara langsung melakukan reset password dengan memasukkan password baru yang diinginkan. Lalu user dapat memilih tombol Reset dan secara otomatis user akan dikembalikan ke tampilan page Login pada sistem eCOLL-Request Form dan terdapat notifikasi bahwa password telah berhasil direset.



Gambar 3.30 Notifikasi Reset Password Berhasil eCOLL-Request Form

Maka, user dapat secara langsung melakukan Login pada sistem menggunakan email yang sudah terdaftar di dalam sistem dengan password baru yang telah berhasil di reset. Selain bagian tersebut, pada tampilan Login juga terdapat tampilan teks card yang dapat berganti sesuai dengan detik yang telah ditentukan pada codingan JavaScript yang terhubung. Teks tersebut terdiri dari 3

bagian yang mencakup teks “eCOLL-Request Form”, “A Form System Master”, dan “Generate forms for business interests”. Berikut merupakan source code JavaScript dan html yang terhubung dengan tampilan teks pada card tersebut.

```

<div class="carousel">
  <div class="images-wrapper">
    
  </div>

  <div class="text-slider">
    <div class="text-wrap">
      <div class="text-group">
        <h2 class="text" data-value="1">eCOLL-Request Form</h2>
        <h2 class="text" data-value="2">A Form System Master</h2>
        <h2 class="text" data-value="3">Invite students to your class</h2>
      </div>
    </div>

    <div class="bullets">
      <span class="active" data-value="1"></span>
      <span data-value="2"></span>
      <span data-value="3"></span>
    </div>
  </div>
</div>

```

Gambar 3.31 Source Code html Teks Card

```

function showToast(message, type = 'info') {
  const toastContainer = document.getElementById('toast-container');
  const toast = document.createElement('div');
  toast.classList.add('toast', `toast-${type}`);
  toast.innerText = message;
  toastContainer.appendChild(toast);
  setTimeout(() => {
    toast.classList.add('show');
    setTimeout(() => {
      toast.classList.remove('show');
      setTimeout(() => {
        toast.remove();
      }, 300); // Fade out duration
    }, 2000); // Toast display duration
  }, 50); // Delay before showing toast
}

```

Gambar 3.32 Source Code JavaScript Teks Card

Dengan ini, tampilan page Login dapat dengan mudah dipahami dan menarik perhatian user saat mengakses sistem eCOLL-Request Form. Sistem Login pada eCOLL-Request Form juga telah terkonfigurasi dengan database tabel users yang berisikan keterangan data-data pengguna yang sudah terdaftar dan memiliki

akun pada sistem. Untuk memiliki akun pada sistem, user harus meminta akses kepada user yang memiliki role Admin atau dalam sistem ini disebut dengan Master untuk dibuatkan akun ke dalam sistem eCOLL-Request Form. Lalu Master harus menginput data pengguna user tersebut ke dalam sistem untuk saat ini dengan menambahkan keterangan data-data yang dimiliki oleh user melalui Page Add User.

Gambar 3.33 Tampilan Page Add User eCOLL-Request Form

Pada Gambar 3.33 terdapat beberapa cell pada card yang memiliki akses untuk dilakukan input text secara manual ataupun dropdown. Walaupun begitu, semua data-data yang diinput di card tersebut akan secara otomatis tersimpan pada database tabel users yang menggunakan local host pada Postgres SQL.

	id [PK] bigint	name character varying (255)	email character varying (255)	role character varying (255)	nik character varying (255)
1	1	Brooklyn Liem	brooklynliem123@outlook.com	Master	00000011880
2	2	Olin	olin@outlook.com	User	00000011882

Gambar 3.34 Tabel Users eCOLL-Request Form

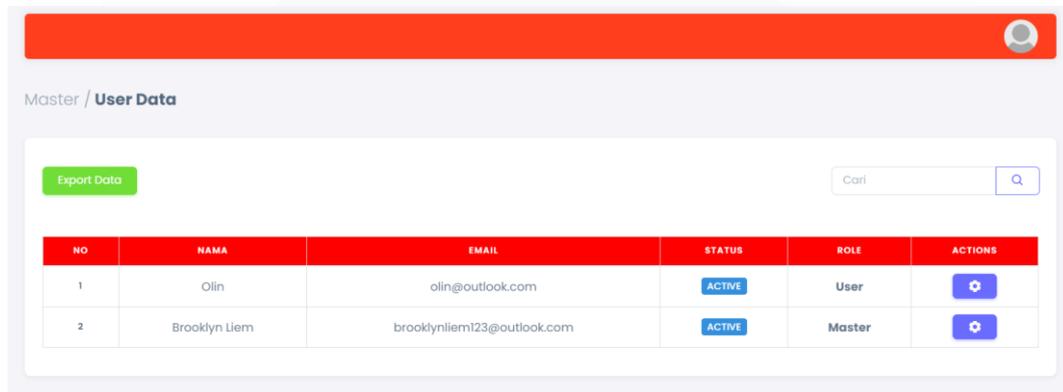
Pada Gambar 3.34 merupakan tampilan cuplikan dari tabel users yang dimiliki oleh sistem eCOLL-Request Form. Untuk saat ini data tersebut merupakan data dummy karena sistem belum diupload pada sistem internal PT Bank XYZ. Selain kolom yang terlihat pada Gambar 3.34, tabel users juga terdiri dari beberapa kolom lainnya. Berikut merupakan tampilan source code dari tabel users yang mencakup kolom-kolom yang dimiliki oleh tabel users.

```
public function up()
{
    Schema::create('users', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->string('email')->unique();
        $table->string('role');
        $table->string('nik');
        $table->string('userid');
        $table->string('department');
        $table->string('status');
        $table->string('password');
        $table->string('image')->default('user.png');
        $table->timestamp('email_verified_at')->nullable();
        $table->rememberToken();
        $table->timestamps();
    });
}
```

Gambar 3.35 Source Code Tabel Users eCOLL-Request Form

Berdasarkan Gambar 3.35, dapat dilihat bahwa tabel users terdiri dari kolom untuk name, email, role, nik, userid, department, status, password, image, dan timestamps. Kolom-kolom tersebut mewakili data-data yang sekiranya diperlukan untuk users pada sistem eCOLL-Request Form. Dengan ini tabel tersebut juga dibuat melalui back-end dan tidak secara manual pada PostgreSQL dengan menggunakan php artisan migrate:fresh sehingga tabel dapat secara langsung tersusun dan terkoneksi dengan PostgreSQL. Details dari tabel users selain dapat dilihat melalui PostgreSQL, tim juga telah menyiapkan tampilan page yang hanya dapat diakses oleh Master dari sistem untuk melihat details dari user yang sudah memiliki akun pada sistem eCOLL-Request Form. Berikut merupakan

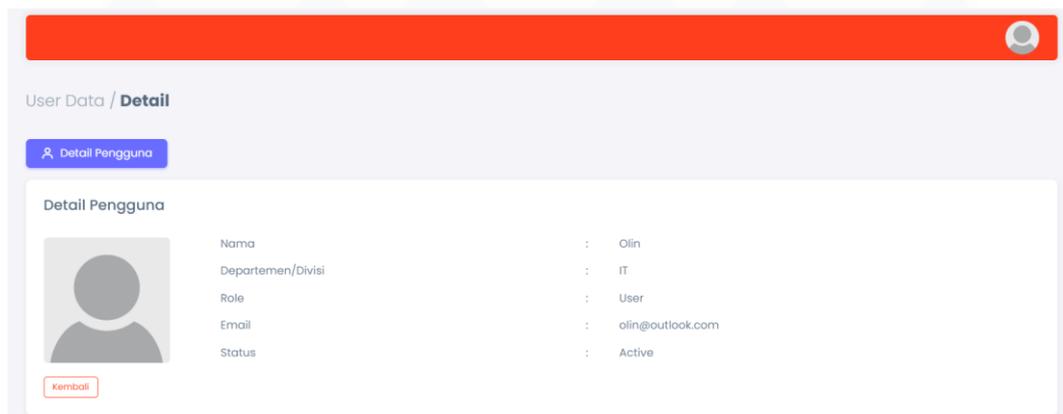
tampilan page User Data yang berisikan detail akun-akun yang telah terdaftar pada sistem.



NO	NAMA	EMAIL	STATUS	ROLE	ACTIONS
1	Olin	olin@outlook.com	ACTIVE	User	
2	Brooklyn Liem	brooklynliem123@outlook.com	ACTIVE	Master	

Gambar 3.36 Tampilan Page User Data eCOLL-Request Form

Pada Gambar 3.36 merupakan tampilan page yang berisikan details dari tabel users. Namun tampilan tersebut hanya menampilkan kolom-kolom yang sekiranya diperlukan untuk mengawasi akun-akun sehari-harinya oleh Master. Jika Master ingin melakukan pendataan yang memerlukan data yang lebih mendetail pada setiap akunnya, Master dapat mengakses tombol yang terdapat pada kolom Actions pada sub-menu detail. Berikut merupakan tampilan page pada sub-menu detail yang dimiliki pada sistem eCOLL-Request Form.

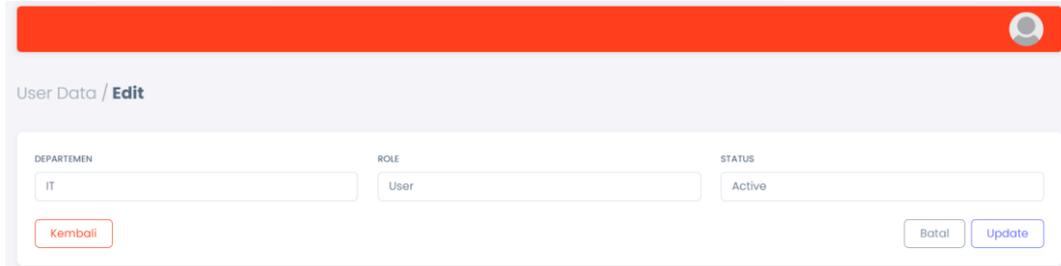


Nama	:	Olin
Departemen/Divisi	:	IT
Role	:	User
Email	:	olin@outlook.com
Status	:	Active

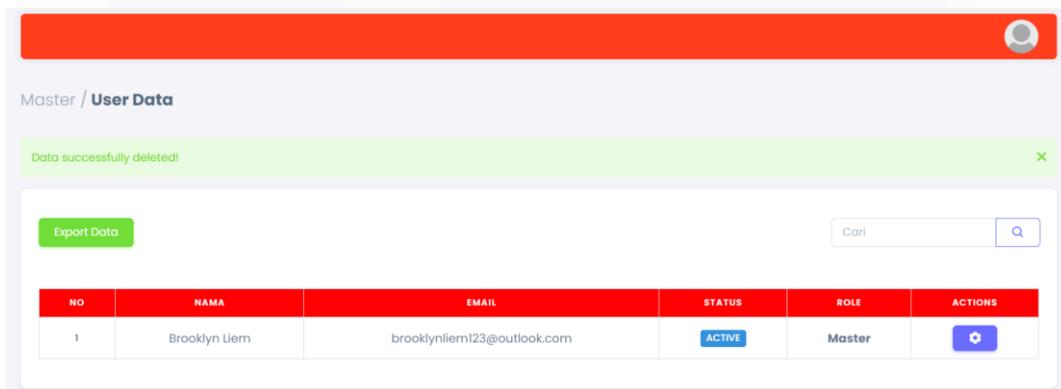
Gambar 3.37 Tampilan Page Detail User eCOLL-Request Form

Selain tampilan Detail User, Master juga memiliki akses ke beberapa submenu lainnya yang dimiliki oleh tombol Actions antara lain pilihan untuk Edit

dan Delete akun yang dimiliki oleh User. Berikut merupakan tampilan dari masing-masing page Edit dan notifikasi Delete pada eCOLL-Request Form.



Gambar 3.38 Tampilan Page Edit User eCOLL-Request Form



NO	NAMA	EMAIL	STATUS	ROLE	ACTIONS
1	Brooklyn Liem	brooklynliem123@outlook.com	ACTIVE	Master	

Gambar 3.39 Tampilan Page Delete User eCOLL-Request Form

Gambar 3.38 merupakan tampilan page Edit ketika Master memilih sub-menu Edit dan ingin melakukan edit pada akun yang sudah terdaftar pada sistem eCOLL-Request Form. Untuk saat ini Master hanya dapat mengedit department, role, dan status yang dimiliki oleh masing-masing akun dan ketika Master memilih tombol Update pada card di Page Edit maka data akun tersebut akan secara otomatis teredit pada tampilan serta database tabel users.

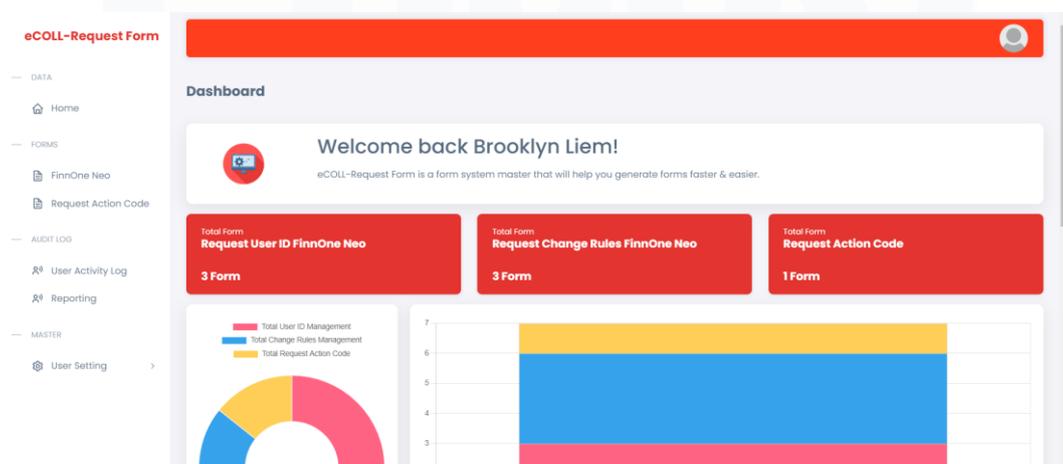
Selain itu, pada sub-menu Actions ketika user memilih Delete maka akun yang terdaftar pada sistem secara langsung akan terhapus pada tampilan dan database tabel users. Jika akun tersebut berhasil terhapus, page akan secara otomatis menampilkan notifikasi sesuai pada Gambar 3.39 yang menampilkan teks Data successfully deleted! atau berarti bahwa data telah berhasil dihapus.

Pada Page User Data juga terdapat tombol Export Data sebagai fitur dimana Master dapat melakukan ekspor pada data-data akun yang telah terdaftar pada sistem. Ketika User memilih untuk melakukan ekspor pada data-data akun, Master dapat menerima data-data tersebut dalam bentuk file Excel. Berikut merupakan tampilan file Excel yang dapat diterima oleh Master ketika melakukan ekspor.

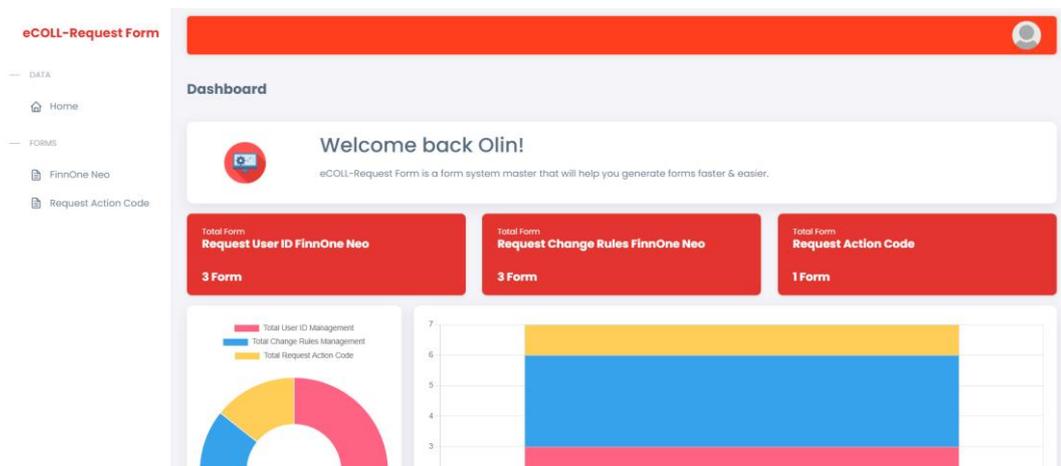
	A	B	C	D	E	F	G	H	I	J
1	Nama	Departme	Role	NIK	User ID	Status	Email			
2	Brooklyn I	Consumer	Master	00000011	00881	Active	brooklynliem123@outlook.com			
3										
4										
5										
6										
7										

Gambar 3.40 Hasil Ekspor Tabel Users eCOLL-Request Form

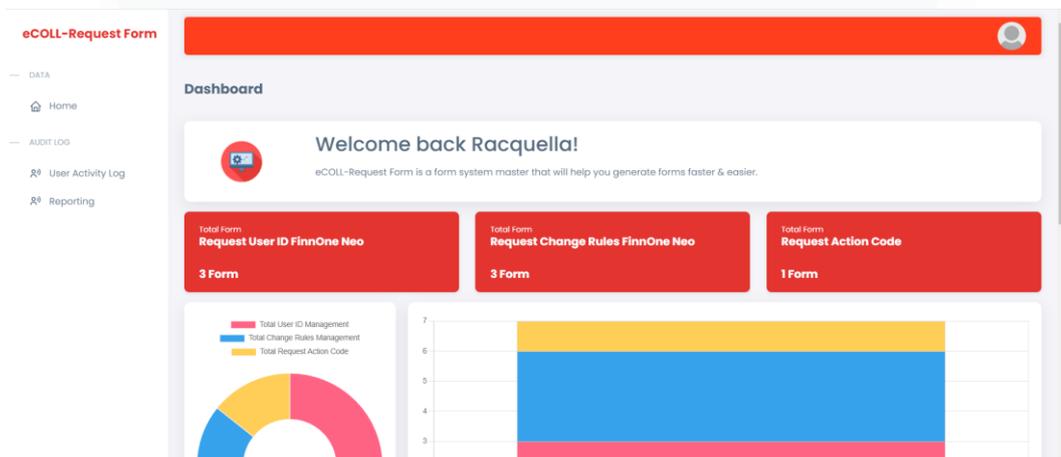
Seperti yang telah disinggung sebelumnya, seluruh akses untuk data-data akun yang telah terdaftar pada sistem eCOLL-Request Form hanya dimiliki oleh user yang memegang role Master. Maka fitur-fitur yang telah dibahas sebelumnya akan dibatasi dengan adanya middleware yang mengatur. Dengan ini, seluruh sistem yang hanya dimiliki akses oleh masing-masing akun tidak akan terganggu oleh user yang sekiranya tidak memiliki kepentingan dalam mengakses fitur tersebut. Namun terdapat satu fitur atau tampilan yang dapat diakses oleh seluruh role yang ada pada sistem eCOLL-Request Form yaitu pada page Dashboard dan Vertical Menu dengan batasan sesuai dengan role yang login ke dalam sistem.



Gambar 3.41 Page Dashboard dan Vertical Menu Master eCOLL-Request Form

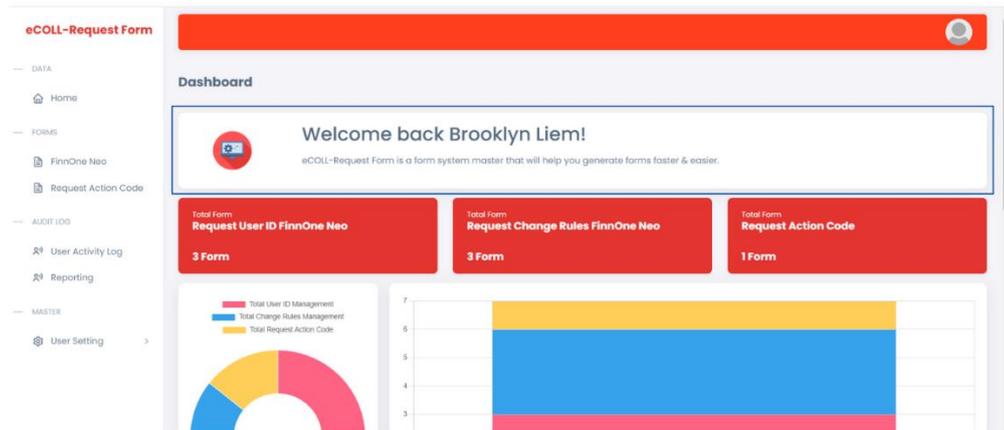


Gambar 3.42 Page Dashboard dan Vertical Menu User eCOLL-Request Form



Gambar 3.43 Page Dashboard dan Vertical Menu Head eCOLL-Request Form

Pada Gambar 3.41, 3.42, dan 3.43 merupakan tampilan Dashboard dan Vertical Menu yang dimiliki oleh user dengan role Master, role User, dan role Head. Berikut merupakan tampilan dan penjelasan detail mengenai komponen-komponen yang menyusun page Dashboard pada eCOLL-Request Form.



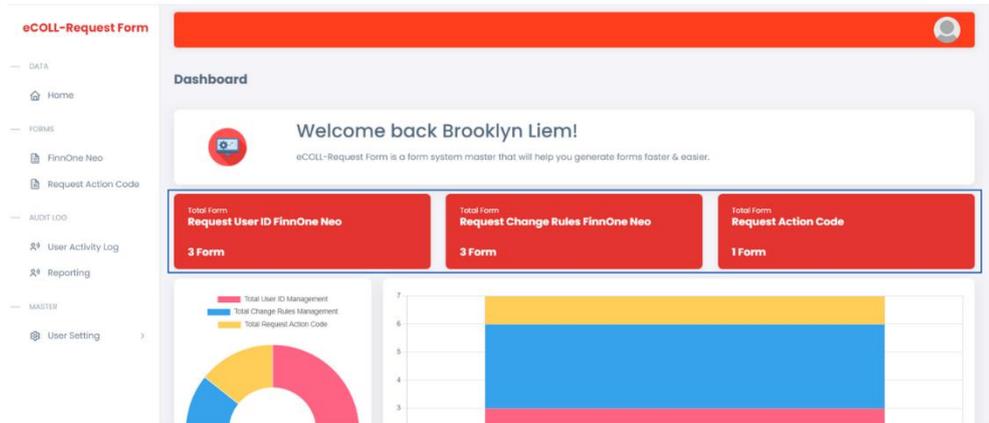
Gambar 3.44 Card Welcome Back

```

<div class="headBox">
  <div class="box">
    <div class="box-content">
      <div class="welcome-text">
        <div class="box-logo">
          
        </div>
        <div class="text-content">
          @if (Auth::check())
            <h2>Welcome back {{ Auth::user()->name }}!</h2>
          @else
            <h2>Welcome!</h2>
          @endif
          <p>eCOLL-Request Form is a form system master that will help you generate forms faster & easier.</p>
        </div>
      </div>
    </div>
  </div>
</div>
  
```

Gambar 3.45 Source Code Card Welcome Back

Pada Gambar 3.44 komponen yang ditandai dengan kotak outline biru merupakan tampilan Card Welcome Back pada page Dashboard. Card tersebut berisikan teks “Welcome back” dan dilanjutkan dengan nama user yang sedang Login dan mengakses sistem eCOLL-Request Form saat itu, beserta dengan teks penjelasan singkat mengenai sistem eCOLL-Request Form. Gambar 3.45 merupakan source code yang mengatur tampilan dan logic menampilkan nama user yang sedang mengakses sistem.



Gambar 3.46 Card Total Form

```

<div class="col-md-4 mb-3 mt-3">
  <div class="totalForm-userid">
    <div class="card" style="background: #e3342f; height: 130px">
      <div class="card-header" style="margin-top: -5px">
        <small class="text-muted d-block"><span style="color: #fff">Total Form</span></small>
        <h5 class="mt-0" style="color: #fff">
          <strong class="d-flex justify-content-between">
            <p>Request User ID FinnOne Neo</p>
          </strong>
        </h5>
        <h5 class="mt-0" style="color: #fff">
          <strong class="d-flex justify-content-between">
            <p>{{ $totalCount }} Form</p>
          </strong>
        </h5>
      </div>
    </div>
  </div>
</div>

```

Gambar 3.47 Source Code Card Total Form Request User ID

```

<div class="col-md-4 mb-3 mt-3">
  <div class="totalForm-changerules">
    <div class="card" style="background: #e3342f; height: 130px">
      <div class="card-header" style="margin-top: -5px">
        <small class="text-muted d-block"><span style="color: #fff">Total Form</span></small>
        <h5 class="mt-0" style="color: #fff">
          <strong class="d-flex justify-content-between">
            <p>Request Change Rules FinnOne Neo</p>
          </strong>
        </h5>
        <h5 class="mt-0" style="color: #fff">
          <strong class="d-flex justify-content-between">
            <p>{{ $totalChangeRules }} Form</p>
          </strong>
        </h5>
      </div>
    </div>
  </div>
</div>

```

Gambar 3.48 Source Code Card Total Form Change Rules

```

<div class="col-md-4 mb-3 mt-3">
  <div class="totalsurat">
    <div class="card" style="background: #e3342f; height: 130px">
      <div class="card-header" style="margin-top: -5px">
        <small class="text-muted d-block"><span style="color: #fff">Total Form</span></small>
        <h5 class="mt-0" style="color: #fff">
          <strong class="d-flex justify-content-between">
            <p>Request Action Code</p>
          </strong>
        </h5>
        <h5 class="mt-0" style="color: #fff">
          <strong class="d-flex justify-content-between">
            <p>{{ $totalRequestAction }} Form</p>
          </strong>
        </h5>
      </div>
    </div>
  </div>
</div>

```

Gambar 3.49 Source Code Card Total Form Request Action Code

Pada Gambar 3.46 komponen yang ditandai dengan kotak outline biru merupakan tampilan Card Total Form pada page Dashboard. Card tersebut berisikan teks yang menampilkan masing-masing jumlah form yang sudah tersimpan di dalam database pada masing-masing model sesuai dengan apa yang ditampilkan pada source code masing-masing card di Gambar 3.47, 3.48, dan 3.49. Jumlah setiap form yang dibuat telah dikalkulasi masing-masing total jumlah formnya dengan fungsi count() sesuai dengan model pada masing-masing form. Berikut merupakan logic count() yang digunakan untuk menampilkan jumlah total pada masing-masing form pada card yang ada.

```

$changeRulesData = ChangeRules::select(
  DB::raw(
    'DATE_PART(\'year\', created_at) as year, DATE_PART(\'month\', created_at) as month, COUNT(*) as count',
  ),
)
->groupBy('year', 'month')
->get();

$requestActionData = RequestAction::select(
  DB::raw(
    'DATE_PART(\'year\', created_at) as year, DATE_PART(\'month\', created_at) as month, COUNT(*) as count',
  ),
)
->groupBy('year', 'month')
->get();

$registerNewCountCard = RegisterNew::count();
$activeInactiveCountCard = ActiveInactive::count();
$changeDisableCountCard = ChangeDisable::count();

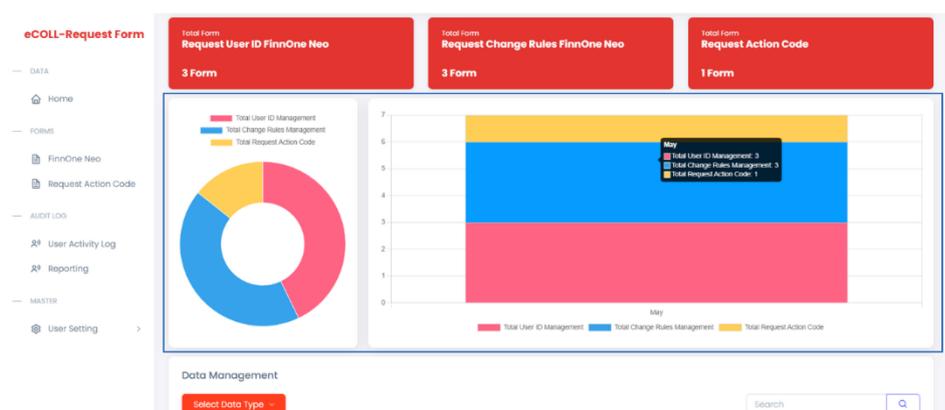
$totalCountData = $registerNewsData->merge($activeInactivesData)->merge($changeDisablesData);
$totalChangeRulesData = $changeRulesData;
$totalRequestActionData = $requestActionData;

$totalCount = $registerNewCountCard + $activeInactiveCountCard + $changeDisableCountCard;
$totalChangeRules = $totalChangeRulesData->sum('count');
$totalRequestAction = $totalRequestActionData->sum('count');

```

Gambar 3.50 Source Code Logic count() untuk Card Total Form

Source Code diatas menggambarkan logic count() yang digunakan untuk implementasi perhitungan jumlah total masing-masing form yang ada pada database. Untuk \$totalCountData merupakan logic untuk menghitung total yang tersimpan pada model RegisterNew, ActiveInactive, dan ChangeDisable atau yang termasuk pada form User ID Management. Untuk \$totalChangeRulesData merupakan logic untuk menghitung total yang tersimpan pada model ChangeRules atau yang termasuk pada form Change Rules Classification-Allocation Management. Sedangkan untuk \$totalRequestActionData merupakan logic untuk menghitung total yang tersimpan pada model RequestAction atau yang termasuk pada form Request Action Code. Dengan ini, setiap Card Total Form yang ada pada Dashboard sudah tersegmentasi dengan logic perhitungan yang berbeda-beda sehingga dapat terlihat jelas dan mudah dipahami oleh user yang ingin melihat jumlah form yang ada pada setiap kategorinya.



Gambar 3.51 Card Grafik Doughnut dan Stacked Bar Chart

Pada Gambar 3.51 komponen yang ditandai dengan kotak outline biru merupakan tampilan Card Grafik Doughnut dan Stacked Bar Chart. Pada bagian Doughnut Chart menampilkan total jumlah setiap form berdasarkan kategori form yang ada. Sedangkan pada Stacked Bar Chart akan menampilkan jumlah total form pada masing-masing bulannya sesuai dengan timestamp yang ada pada setiap form yang berhasil tersimpan pada database. Berikut merupakan source code untuk

masing-masing chart dengan JavaScript yang mengatur logic filter pada masing-masing chart.

```
<div class="graphBox mb-3">
  <div class="box">
    <canvas id="myChart"></canvas>
  </div>
  <div class="box">
    <canvas id="myChartBar"></canvas>
  </div>
</div>
```

Gambar 3.52 Source Code Card Grafik Doughnut dan Stacked Bar Chart

```
new Chart(ctx, {
  type: 'doughnut',
  data: {
    labels: ['Total User ID Management', 'Total Change Rules Management',
      'Total Request Action Code'],
    datasets: [{
      data: [totalCount, totalChangeRules, totalRequestAction],
      backgroundColor: [
        'rgba(255, 99, 132, 1)',
        'rgba(54, 162, 235, 1)',
        'rgba(255, 206, 86, 1)',
      ],
      borderWidth: 1
    }]
  },
  options: {
    maintainAspectRatio: false, // Disable aspect ratio
    responsive: true,
    // Adjust width and height as needed
    width: 100,
    height: 100
  }
});
```

Gambar 3.53 Source Code JavaScript Card Grafik Doughnut

```

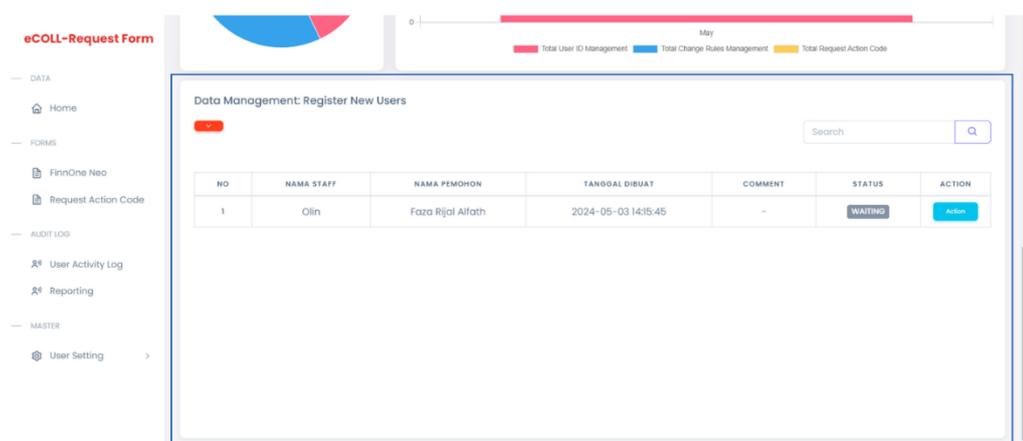
new Chart(bar, {
  type: 'bar',
  data: {
    labels: labels.map(month => {
      const date = new Date();
      date.setMonth(month - 1);
      return date.toLocaleString('default', {
        month: 'long'
      });
    }),
    datasets: [{
      label: 'Total User ID Management',
      data: [{ { $totalCount } }],
      backgroundColor: 'rgba(255, 99, 132, 1)',
      borderWidth: 1
    },
    {
      label: 'Total Change Rules Management',
      data: [{ { $totalChangeRules } }],
      backgroundColor: 'rgba(54, 162, 235, 1)',
      borderWidth: 1
    },
    {
      label: 'Total Request Action Code',
      data: [{ { $totalRequestAction } }],
      backgroundColor: 'rgba(255, 206, 86, 1)',
      borderWidth: 1
    }
  ]
},
options: {
  maintainAspectRatio: false, // Disable aspect ratio
  responsive: true,
  scales: {
    x: {
      stacked: true,
    },
    y: {
      stacked: true,
    }
  },
  plugins: {
    legend: {
      position: 'bottom',
    },
    tooltip: {
      mode: 'index',
    }
  }
}
});

```

Gambar 3.54 Source Code JavaScript Card Grafik Stacked Bar Chart

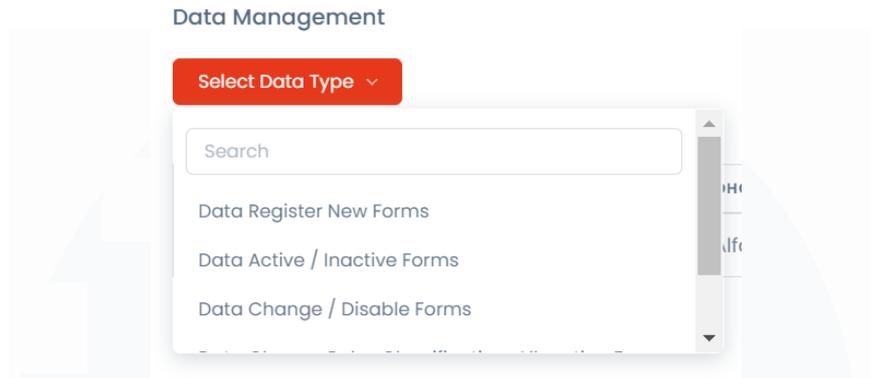
Pada Gambar 3.52 merupakan source code untuk memanggil JavaScript dari masing-masing chart, dalam konteks ini myChart mewakilkan JavaScript untuk

Doughnut Chart dan myChartBar mewakili JavaScript untuk Stacked Bar Chart. Selanjutnya Gambar 3.53 merupakan source code JavaScript dari Doughnut Chart yang memiliki fungsi untuk memvisualisasikan jumlah data pada masing-masing kategori form yang ada, serta mensegmentasikannya sesuai dengan warna yang berbeda-beda agar mudah untuk dibedakan dan menarik untuk dilihat. Gambar 3.54 merupakan source code JavaScript dari Stacked Bar Chart, yang memiliki fungsi untuk menampilkan total jumlah form pada masing-masing kategori berdasarkan dengan timestamp yang ada dan ditampilkan pada x axis dari Stacked Bar Chart tersebut. Dengan adanya, kedua chart ini tampilan Dashboard dapat terkesan lebih variatif dan interaktif untuk digunakan oleh user yang mengakses sistem eCOLL-Request Form.



Gambar 3.55 Card Data Management Table Master

Gambar 3.55 merupakan tampilan Card Data Management Table yang dimana card tersebut menyediakan beberapa fungsi atau fitur yang berbeda untuk setiap role user yang ada pada sistem eCOLL-Request Form. Data Management Table yang tertera pada Gambar 3.55 merupakan tampilan yang dimiliki oleh Master dimana master memiliki akses untuk memilih data form yang ingin ditampilkan dengan menggunakan tombol menu dropdown yang ada, menu tombol ini berlaku untuk semua role. Berikut merupakan tampilan menu dropdown pada Data Management Table.



Gambar 3.56 Menu Dropdown Data Management Table

Menu dropdown tersebut berisikan pilihan untuk masing-masing form yang ada pada sistem agar user dapat memilih data form yang ingin ditampilkan pada card agar dapat terlihat jelas pada Dashboard. Berikut merupakan source code yang terhubung pada menu dropdown tersebut.

```

<div class="card" style="height: 600px">
  <h5 class="card-header" id="dataManagementTitle">Data Management</h5>
  <div class="card-body">
    <div class="row justify-content-between gap-2">
      <div class="col-md-3">
        <div class="dropdown">
          <button class="btn btn-danger dropdown-toggle" type="button" id="dropdownMenuButton"
            data-bs-toggle="dropdown" aria-expanded="false">
            Select Data Type
          </button>
          <ul class="dropdown-menu" aria-labelledby="dropdownMenuButton"
            style="max-height: 200px; overflow-y: auto;">
            <li style="padding: 10px"><input class="form-control search-input" type="text"
              placeholder="Search"></li>
            <li><a class="dropdown-item" href="#" onclick="selectDataType('registerNew')">Data
              Register New Forms</a></li>
            <li><a class="dropdown-item" href="#" onclick="selectDataType('activeInactive')">Data
              Active / Inactive Forms</a></li>
            <li><a class="dropdown-item" href="#" onclick="selectDataType('changeDisable')">Data
              Change / Disable Forms</a></li>
            <li><a class="dropdown-item" href="#" onclick="selectDataType('changeRules')">Data
              Change
              Rules Classification-Allocation Forms</a></li>
            <li><a class="dropdown-item" href="#" onclick="selectDataType('requestAction')">Data
              Request Action Code Forms</a></li>
          </ul>
        </div>
      </div>
    </div>
  </div>
</div>

```

Gambar 3.57 Source Code Menu Dropdown Data Management Table

Sesuai dengan source code yang dapat dilihat pada Gambar 3.57, menu dropdown yang dimiliki oleh Card Data Management Table dapat menjadi interaktif dengan bantuan JavaScript untuk menampilkan tampilan dari masing-masing `selectDataType` yang ada. Sebagai contoh ketika user memilih untuk menampilkan Data Register New Forms pada menu dropdown, berikut merupakan

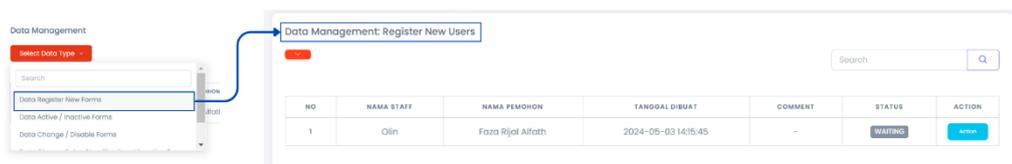
JavaScript yang akan terpanggil untuk menyesuaikan tampilan card dengan menu pilihan yang dipilih oleh user.

```
function selectDataType(dataType, buttonText) {
    document.getElementById("dropdownMenuButton").textContent = buttonText;

    localStorage.setItem('selectedDataType', dataType);
    showSearchForm(dataType);
    event.preventDefault();
    var title = "Data Management: ";
    if (dataType === 'registerNew') {
        title += "Register New Users";
    } else if (dataType === 'activeInactive') {
        title += "Active / Inactive Users";
    } else if (dataType === 'changeDisable') {
        title += "Change / Disable Users";
    } else if (dataType === 'changeRules') {
        title += "Change Rules Classification-Allocation";
    } else if (dataType === 'requestAction') {
        title += "Request Action Code";
    }
    document.getElementById('dataManagementTitle').innerText = title;
}
```

Gambar 3.58 Source Code JavaScript Data Management Table Title

Gambar 3.58 merupakan source code JavaScript yang mengatur tampilan judul pada Card Data Management Title, maka ketika user memilih Data Register New Forms pada menu dropdown judul daripada card akan berubah menjadi Data Management: Register New Users, seperti yang dapat dilihat pada Gambar 3.58.



Gambar 3.59 Tampilan Title Card Data Management Register New Forms

Selain judul daripada Card Data Management Table akan berubah, tentunya tabel yang ada pada card akan berubah sesuai dengan data-data form yang ada pada form Register New Users. Berikut merupakan source code dan juga alur proses yang terjadi saat menu dropdown tersebut terpilih.

```
function showSearchForm(dataType) {
  if (dataType === 'registerNew') {
    $('.searchRegisterNew').show();
    $('.searchActiveInactive').hide();
    $('.searchChangeDisable').hide();
    $('.searchChangeRules').hide();
    $('.searchRequestAction').hide();

    $('#registerNewTable').show();
    $('#activeInactiveTable').hide();
    $('#changeDisableTable').hide();
    $('#changeRulesTable').hide();
    $('#requestActionTable').hide();
  }
}
```

Gambar 3.60 Source Code JavaScript Data Management Table

Pada Gambar 3.60, JavaScript diatas akan membantu untuk menampilkan hanya tabel registerNewTable dan search bar searchRegisterNew ketika Data Register New Forms dipilih oleh user. Sehingga berikut merupakan alur proses yang terjadi untuk memanggil fungsi tersebut.

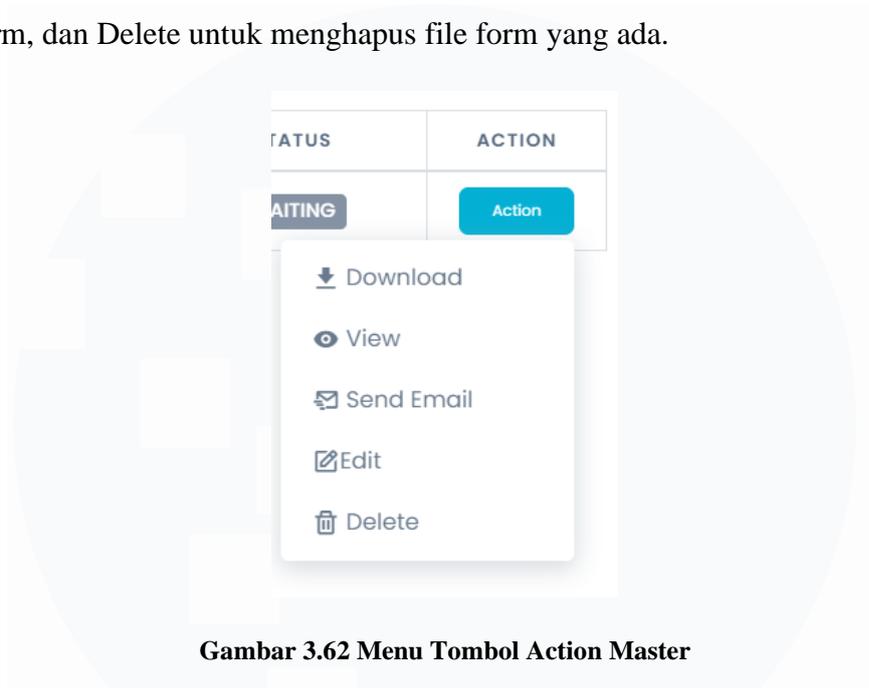


Gambar 3.61 Tampilan Tabel Card Data Management Register New Forms

Gambar 3.61 menampilkan tampilan tabel pada Card Data Management ketika user memilih Data Register New Forms pada menu dropdown. Tabel tersebut tentunya akan menampilkan data yang dimiliki oleh Register New Forms yang berhasil tersimpan pada database. Fungsi penjelasan yang telah dijelaskan untuk Data Register New Forms juga berlaku untuk pilihan pada menu dropdown lainnya pada Card Data Management Table.

Selain itu, user Master juga memiliki akses untuk memilih sub-menu yang ada pada tombol di kolom Action pada setiap tabel yang ada pada dropdown tersebut. Sub-menu yang ada pada tombol Action tersebut yang dimiliki akses oleh user Master terdiri atas pilihan Download untuk mengunduh file form, View untuk

melihat file form, Send Email untuk mengirimkan file form, Edit untuk menyunting file form, dan Delete untuk menghapus file form yang ada.



Gambar 3.62 Menu Tombol Action Master

Berikut merupakan source code beserta penjelasan untuk masing-masing sub-menu pilihan yang ada pada tombol Action di Data Register New Forms.

```
public function downloadRegisterNew($id)
{
    // Construct the file path
    $filePath = 'public/form_register_news/' . $id . '.pdf';

    //User Activity Report
    $users = auth()->user();
    $audit_trail = [
        'users_id' => auth()->id(), // Assuming $users is the authenticated user
        'name' => $users->name, // Assuming $users is the authenticated user
        'date' => Carbon::parse(now()->toDateString(), // Current date
        'activity' => "{$users->name} downloaded Form Request User ID FinnOne Neo (Register New User ID)",
    ];
    $audit_trail = AuditTrail::query()->create($audit_trail);

    // Check if the file exists
    if (Storage::exists($filePath)) {
        // If the file exists, return a response to download the file
        return response()->download(
            storage_path('app/' . $filePath),
            'Form Request User ID FinnOne Neo.pdf'
        );
    } else {
        // If the file does not exist, return a 404 Not Found response
        abort(404);
    }
}
```

Gambar 3.63 Source Code Sub-Menu Download

Gambar 3.63 merupakan source code back-end yang mengatur sub-menu Download pada Register New Form. Dimana ketika user memilih untuk melakukan download, file tersebut akan dicari sesuai dengan tempat penyimpanannya pada file path public/form_register_news/ yang dimana akan dicari berdasarkan id file form yang ingin diunduh. Setelah itu, aksi tersebut akan memicu fungsi Audit Trail, yang nanti akan dijelaskan lebih lanjut, berupa pernyataan bahwa telah terjadinya pengunduhan pada file form terkait. Lalu user dapat menerima hasil unduhan file dengan format pdf yang sudah tersimpan dari file path yang dipanggil di awal dengan nama file Form Request User ID. Namun source code tersebut hanya akan berfungsi ketika tombol Download sudah terhubung dengan route web yang berkorelasi dengan source code terkait, maka berikut merupakan route web yang dimiliki untuk fungsi download beserta dengan source code untuk memanggil fungsi tersebut pada bagian tombol Action di tabel Data Register New Forms.

```
Route::get('downloadsuratlunas/{file}', $controller_path . '\dashboard\Data@downloadSuratLunas')->name('download')->middleware(['auth']);
Route::get('downloadformregisternewusers/{file}', $controller_path . '\dashboard\Data@downloadRegisterNew')->name('download')->middleware(['auth']);
Route::get('downloadformactiveinactiveusers/{file}', $controller_path . '\dashboard\Data@downloadActiveInactive')->name('download')->middleware(['auth']);
Route::get('downloadformchangedisableusers/{file}', $controller_path . '\dashboard\Data@downloadChangeDisable')->name('download')->middleware(['auth']);
Route::get('downloadformchangerulesusers/{file}', $controller_path . '\dashboard\Data@downloadChangeRules')->name('download')->middleware(['auth']);
Route::get('downloadformrequestaction/{file}', $controller_path . '\dashboard\Data@downloadRequestAction')->name('download')->middleware(['auth']);
```

Gambar 3.64 Route Fungsi Download File

```
<a href="downloadformregisternewusers/{ { $register->id } }"
  class="dropdown-item" download>
  <i class="bx bxs-download me-1"></i>Download
</a>
```

Gambar 3.65 Sub-menu Download Register New Forms

Gambar 3.64 menampilkan route dari masing-masing form yang ada untuk fungsi Download, sedangkan Gambar 3.65 menampilkan source code untuk memanggil fungsi Download pada tombol Action yang dimiliki oleh Register New Form. Sehingga pilihan Download dapat berfungsi sesuai dengan yang diinginkan hingga user dapat menerima file terkait yang ingin diunduh. Berikut merupakan tampilan file form hasil dari Download yang dilakukan pada salah satu Data Register New Forms.



Gambar 3.66 Hasil Download File Register New Forms

Fungsi Download yang ada pada tombol Action di Register New Form memiliki alur fungsi yang sama dengan form lainnya. Maka, selanjutnya merupakan pilihan untuk View atau melihat tampilan form pada Data Register New Form terkait.

```

public function showPdfRegisterNew($id)
{
    $filePath = 'public/form_register_news/' . $id . '.pdf';

    //User Activity Report
    $users = auth()->user();
    $audit_trail = [
        'users_id' => auth()->id(), // Assuming $users is the authenticated user
        'name' => $users->name, // Assuming $users is the authenticated user
        'date' => Carbon::parse(now())->toDateString(), // Current date
        'activity' => "{$users->name} viewed Form Request User ID FinnOne Neo (Register New User ID)",
    ];
    $audit_trail = AuditTrail::query()->create($audit_trail);

    if (Storage::exists($filePath)) {
        return response()->file(storage_path('app/' . $filePath));
    } else {
        abort(404);
    }
}

```

Gambar 3.67 Source Code Sub-Menu View

Pilihan View memiliki fungsi hanya untuk menampilkan file form terkait dalam bentuk pdf sesuai dengan hasil file form yang dapat diunduh oleh user pada bagian fungsi Download. Maka, fungsi tersebut akan mencari file terkait sesuai file path yang dimiliki yaitu public/form_register_news/ dan sesuai dengan id pada file terkait. Selanjutnya aksi tersebut akan memicu fungsi Audit Trail, dimana akan

menyatakan bahwa user telah melihat form tersebut. Lalu, jika terdapat file yang sesuai dengan id yang dicari, maka file akan ditampilkan namun jika tidak maka file tidak akan ditampilkan. Namun source code tersebut hanya akan berfungsi ketika tombol View sudah terhubung dengan route web yang berkorelasi dengan source code terkait, maka berikut merupakan route web yang dimiliki untuk fungsi view beserta dengan source code untuk memanggil fungsi tersebut dan tampilan hasil dari fungsi View pada bagian tombol Action di tabel Data Register New Forms.

```
Route::get('show-form-register-new-users/{id}', $controller_path . '\dashboard\Data@showPdfRegisterNew')->middleware(['auth']);
Route::get('show-form-active-inactive-users/{id}', $controller_path . '\dashboard\Data@showPdfActiveInactive')->middleware(['auth']);
Route::get('show-form-change-disable-users/{id}', $controller_path . '\dashboard\Data@showPdfChangeDisable')->middleware(['auth']);
Route::get('show-form-change-rules-users/{id}', $controller_path . '\dashboard\Data@showPdfChangeRules')->middleware(['auth']);
Route::get('show-form-request-action/{id}', $controller_path . '\dashboard\Data@showPdfRequestAction')->middleware(['auth']);
```

Gambar 3.68 Route Fungsi View File

```
<a href="show-form-register-new-users/{ { $register->id } }"
  class="dropdown-item" target="_blank">
  <i class="bx bxs-show me-1"></i>View
</a>
```

Gambar 3.69 Sub-menu View Register New Forms



Gambar 3.70 Hasil View File Register New Forms

Setelah pilihan untuk View dan Download, terdapat pilihan untuk Send Email atau mengirimkan file yang dipilih dengan email yang terdaftar pada setiap input form yang dilakukan. Fitur Send Email ini memiliki fungsi utama untuk

mendukung proses dari fitur Maker & Checker yang akan dijelaskan nantinya. Berikut merupakan alur dari pengisian data email hingga fitur Send Email dapat digunakan dengan benar pada sistem eCOLL-Request Form.

The screenshot shows a web-based form titled "Register User ID Baru". At the top, there are navigation tabs: "Register User ID Baru" (selected), "Active/Inactive", and "Change/Disable". Below the tabs, the form is organized into several sections:

- JENIS PERMOHONAN:** A dropdown menu showing "Register User ID Baru".
- NIK STAFF:** A dropdown menu with the value "0000011882".
- REPORTING TO/USER:** A dropdown menu with the value "Brooklyn Liem".
- NAMA STAFF:** A text input field containing "Olin".
- ROLE MENU:** A dropdown menu with the value "COLL".
- USER ID STAFF:** A text input field containing "00882".
- CATEGORY:** A dropdown menu with the value "DC".
- APPROVAL:** A section with two fields:
 - NAMA DIRECT SUPERVISOR (MIN DIVISION HEAD):** A dropdown menu with the value "Racquella".
 - EMAIL PIC:** A text input field containing "brooklynliem123@outlook.com".

At the bottom right of the form, there are two buttons: "RESET" and "SUBMIT".

Gambar 3.71 Page Input Form Register New Users

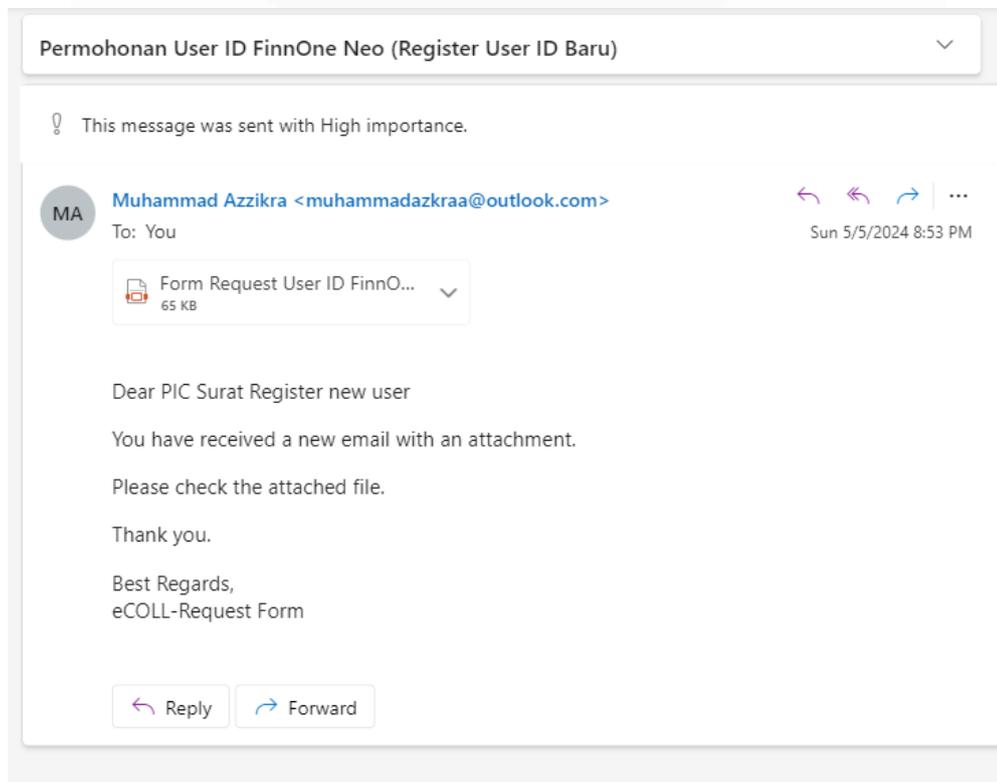
Pertama, user harus melakukan input data email melalui page input form. Disini user tidak perlu secara manual mengisi alamat email yang dituju karena sudah terdapat logic autofill yang secara otomatis dapat mengisi alamat email yang dituju ketika user memilih PIC pada bagian Reporting To/User. Ketika user sudah selesai menginput data-data untuk melalui proses input form, maka data tersebut dapat di submit dan secara otomatis dapat tersimpan pada database dan akan terlihat pada tampilan Card Data Management Table pada page Dashboard.

The screenshot shows the 'Data Management: Register New Users' section of the eCOLL-Request Form dashboard. It features a search bar and a table with the following data:

NO	NAMA STAFF	NAMA PEMOHON	TANGGAL DIBUAT	COMMENT	STATUS	ACTION
1	Olin	Faza Rijal Alfath	2024-05-03 14:15:45	-	WAITING	Action
2	Olin	Brooklyn Liem	2024-05-05 20:46:45	-	WAITING	Action

Gambar 3.72 Data Register New Forms Pada Card Data Management Table

Jika data yang sudah terinput sudah berhasil tersimpan dan terlihat pada page Dashboard seperti pada Gambar 3.72, maka pilihan untuk Send Email dapat digunakan. Berikut merupakan tampilan email yang diterima oleh alamat email yang terdaftar pada form terkait.



Gambar 3.73 Email Data Register New Forms

Email tersebut mengandung subject agar user dapat mengetahui kepentingan mengapa email tersebut dikirim, body email mengenai penjelasan singkat tentang kepentingan yang ingin disampaikan, beserta dengan sisipan file form yang sama dengan file form jika dilakukan fungsi Download ataupun View. Berikut merupakan source code yang terhubung pada fungsi Send Email.

```
public function sendRegisterNewUsersMail($id)
{
    $register_news = RegisterNews::find($id);

    //User Activity Report
    $users = auth()->user();
    $audit_trail = [
        'users_id' => auth()->id(), // Assuming $users is the authenticated user
        'name' => $users->name, // Assuming $users is the authenticated user
        'date' => Carbon::parse(now()->toDateString()), // Current date
        'activity' => "($users->name) sent Form Request User ID FinnOne Neo (Register New User ID) Mail",
    ];
    $audit_trail = AuditTrail::query()->create($audit_trail);

    if ($register_news) {
        // Construct the file path for the attachment
        $filePath = 'public/Form_register_news/' . $register_news->id . '.pdf';

        // Check if the file exists
        if (Storage::exists($filePath)) {
            // Retrieve the file path
            $filePath = storage_path('app/' . $filePath);
            $fileName = 'Form Request User ID FinnOne Neo.pdf';

            // Send email with attachment
            Mail::send('content.emails.surat-register-new-users', ['register_news' => $register_news], function ($message) use ($register_news, $filePath, $fileName) {
                $message->to($register_news->emailPIC);
                $message->subject('Permohonan User ID FinnOne Neo (Register User ID Baru)');

                $message->from('muhammadazkraa@outlook.com', 'Muhammad Azzikra');
                $message->sender('muhammadazkraa@outlook.com', 'Muhammad Azzikra');
                $message->replyTo('muhammadazkraa@outlook.com', 'Muhammad Azzikra');

                $message->priority(1);

                // Attach the file
                $message->attach($filePath, [
                    'as' => $fileName,
                    'mime' => 'application/pdf',
                ]);
            });

            // Check if the email sending was queued successfully
            if (count(Mail::failures()) > 0) {
                // Email sending failed
                return redirect('/')->with('error', 'Failed to send email.');
```

Gambar 3.74 Source Code Sub-Menu Send Email

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Surat Register New User</title>
7 </head>
8 <body>
9   <p>Dear PIC Surat Register new user</p>
10  <p>You have received a new email with an attachment.</p>
11  <p>Please check the attached file.</p>
12  <p>Thank you.</p>
13  <p>Best Regards, <br> eCOLL-Request Form</p>
14 </body>
15 </html>

```

Gambar 3.75 Source Code Body Email

Pada Gambar 3.74 dapat dilihat bahwa fungsi Send Email dimulai dengan mencari file berdasarkan id yang dipicu untuk melakukan fungsi tersebut. Lalu fungsi Audit Trail juga ikut terpanggil dan menyatakan bahwa user telah mengirim email terkait. Lalu jika file yang dipicu telah ditemukan, maka fungsi Send Email akan dilakukan besertakan dengan subject, body email, dan sisipan file data form yang telah ditentukan. Jika berhasil, maka alamat email yang terdaftar menerima notifikasi dan email dari sistem dengan file terkait, namun jika tidak maka email tidak akan berhasil terkirim. Namun source code tersebut hanya akan berfungsi ketika tombol Send Email sudah terhubung dengan route web yang berkorelasi dengan source code terkait, maka berikut merupakan route web yang dimiliki untuk fungsi mengirimkan email, source code untuk memanggil fungsi tersebut, dan JavaScript yang terhubung pada fungsi tersebut di bagian tombol Action tabel Data Register New Forms.

```

Route::match(['get', 'post'], 'send-email-register/{id}', $controller_path . '\dashboard\Data@sendRegisterNewUsersMail')->name('send-email-register')->middleware(['auth']);
Route::match(['get', 'post'], 'send-email-active/{id}', $controller_path . '\dashboard\Data@sendActiveInactiveMail')->name('send-email-active')->middleware(['auth']);
Route::match(['get', 'post'], 'send-email-changeds/{id}', $controller_path . '\dashboard\Data@sendChangeDisableMail')->name('send-email-changeds')->middleware(['auth']);
Route::match(['get', 'post'], 'send-email-changerules/{id}', $controller_path . '\dashboard\Data@sendChangeRulesMail')->name('send-email-changerules')->middleware(['auth']);
Route::match(['get', 'post'], 'send-email-requests/{id}', $controller_path . '\dashboard\Data@sendRequestActionMail')->name('send-email-requests')->middleware(['auth']);

```

Gambar 3.76 Route Fungsi Send Email

```

@if (auth()->check() && (auth()->user()->role === 'Master' || auth()->user()->role === 'User'))
  <a href="#" class="dropdown-item send-email-register"
    data-id="{ $register->id }">
    <i class="bx bx-mail-send me-1"></i>Send Email
  </a>
@endif

```

Gambar 3.77 Sub-menu Send Email Register New Forms

```

$(document).ready(function() {
    $('.send-email-register').click(function(e) {
        e.preventDefault();
        var id = $(this).data('id');

        // Get CSRF token from meta tag
        var csrfToken = $('meta[name="csrf-token"]').attr('content');

        // Include CSRF token in AJAX request headers
        $.ajaxSetup({
            headers: {
                'X-CSRF-TOKEN': csrfToken
            }
        });

        // Send AJAX POST request
        $.post("{{ route('send-email-register', ['id' => ':id']) }}".replace(':id', id), function(
            data) {
                // Handle success response
                console.log(data);
                alert("Email sent successfully.");
            }).fail(function(xhr, status, error) {
                // Check if the error is due to a non-success status code
                if (xhr.status !== 200) {
                    console.error("Error sending email:", error);
                }
            });
    });
});

```

Gambar 3.78 Source Code JavaScript Send Email Register New Forms

Pada Gambar 3.77 yang merupakan source code untuk sub-menu tampilan pilihan Send Email memiliki pernyataan if yang menyatakan bahwa pilihan ini hanya dapat diakses user dengan role Master dan User saja karena user dengan role Head tidak memiliki kepentingan untuk mengakses pilihan ini.

Selanjutnya merupakan pilihan Edit pada tombol Action, fungsi tersebut kurang lebih memiliki fungsi yang sama dengan bagian input form, yang nanti akan dijelaskan lebih detail, karena fungsi ini digunakan untuk mengedit data inputan yang telah dikumpulkan oleh user saat melakukan input form pada awalnya. Fitur ini dapat menggantikan data hasil input hingga pada file pdf yang sudah tergenerasi oleh sistem. Berikut merupakan tampilan page Edit pada data Register New Forms yang dapat dilihat oleh user saat ingin melakukan edit pada salah satu form yang telah tersimpan pada database.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Gambar 3.79 Page Edit Data Register New Forms

Selanjutnya merupakan pilihan Edit pada tombol Action, fungsi tersebut kurang lebih memiliki fungsi yang sama dengan bagian input form, yang nanti akan dijelaskan lebih detail, karena fungsi ini digunakan dengan cara menginput data-data untuk memenuhi form dan mengedit data inputan yang telah dikumpulkan oleh user di awal. Berikut merupakan source code dari fungsi Edit pada Data Register New Forms.

```

public function edit($id)
{
    $register_news = RegisterNew::findOrFail($id);
    $users = User::all();
    return view('content.edit-form-finnone.edit-register-new-users', compact('register_news', 'users'));
}

public function update(Request $request, $id)
{
    $request->validate([
        'status' => 'nullable|in:WAITING,APPROVED,REJECTED',
    ]);

    $register_news = RegisterNew::findOrFail($id);

    // Update each field only if it is provided in the request
    if ($request->has('nama')) {
        $register_news->nama = $request->nama;
    }
    if ($request->has('nik')) {
        $register_news->nik = $request->nik;
    }
    if ($request->has('userid')) {
        $register_news->userid = $request->userid;
    }
    if ($request->has('reportinguser')) {
        $register_news->reportinguser = $request->reportinguser;
    }
}

```

Gambar 3.80 Source Code Sub-Menu Edit

Pada Gambar 3.80 merupakan tampilan untuk source code sub-menu Edit pada data Register New Forms. Untuk fungsi edit, terdapat dua fungsi utama yang harus dimiliki pada bagian back-end yaitu edit untuk memanggil database yang diperlukan dalam memperbaharui data-data pada hasil input form yang telah tersimpan pada database dan update untuk menggantikan data-data yang telah tersimpan. Fungsi update tersebut juga memiliki fungsi Audit Trail yang akan menyatakan bahwa form tersebut telah teredit oleh user terkait yang melakukan aksi tersebut. Berikut merupakan source code dari fungsi Audit Trail tersebut.

```
//User Activity Report
$users = auth()->user();
$audit_trail = [
    'users_id' => auth()->id(), // Assuming $users is the authenticated user
    'name' => $users->name, // Assuming $users is the authenticated user
    'date' => Carbon::parse(now()->toDateString(), // Current date
    'activity' => "{$users->name} edited Form Request User ID FinnOne Neo (Register New User ID)",
];
$audit_trail = AuditTrail::query()->create($audit_trail);
```

Gambar 3.81 Source Code Fungsi Audit Trail

Selain fungsi tersebut juga terdapat code yang secara khusus mengatur ketika form berhasil ataupun tidak di update ke dalam sistem. Berikut merupakan tampilan source code tersebut.

```
// Save the updated model
if ($register_news->save()) {
    // Generate new PDF
    $pdf = $this->generatePdf($register_news);

    // Delete previous PDF from storage
    $previousPdfPath = storage_path('public/form_register_news/' . $id . '.pdf');
    if (file_exists($previousPdfPath)) {
        unlink($previousPdfPath);
    }

    // Save the new PDF to storage
    Storage::put('public/form_register_news/' . $id . '.pdf', $pdf->output());

    $user = $request->user();
    if ($user && ($user->role === 'Master')) {
        return redirect('/');
    } elseif ($user && ($user->role === 'Division Head' || $user->role === 'Group Head' || $user->role === 'Director')) {
        return redirect('/');
    }
} else {
    // Handle the case where saving fails
    return back()->with('error', 'Failed to update surat');
}
```

Gambar 3.82 Source Code Fungsi Update

Pada Gambar 3.82 merupakan source code untuk mengupdate data-data yang berhasil tersimpan pada database. Pertama, ketika user telah menyimpan inputan melalui submit, hal tersebut akan memicu fungsi dan membuat file pdf baru dengan data-data baru hasil inputan dari fungsi edit yang baru saja dilakukan lalu akan tersimpan secara otomatis pada file path sebelumnya menggantikan file lama dengan id yang sama. Ketika file tersebut berhasil disimpan, user dapat mengecek keberhasilannya melalui pilihan sub-menu View ataupun Download pada page Dashboard yang telah dijelaskan sebelumnya. Namun source code tersebut hanya akan berfungsi ketika tombol Edit sudah terhubung dengan route web yang berkorelasi dengan source code terkait, maka berikut merupakan route web yang dimiliki untuk fungsi edit dan update beserta dengan source code untuk memanggil fungsi tersebut di bagian tombol Action tabel Data Register New Forms.

```
Route::get('/edit_register_new_users/{id}', $controller_path . '\form_finnone\RegisterNewUsers@edit')->name('input_register_new_users.edit');
Route::put('/edit_register_new_users/{id}', $controller_path . '\form_finnone\RegisterNewUsers@update')->name('input_register_new_users.update');
```

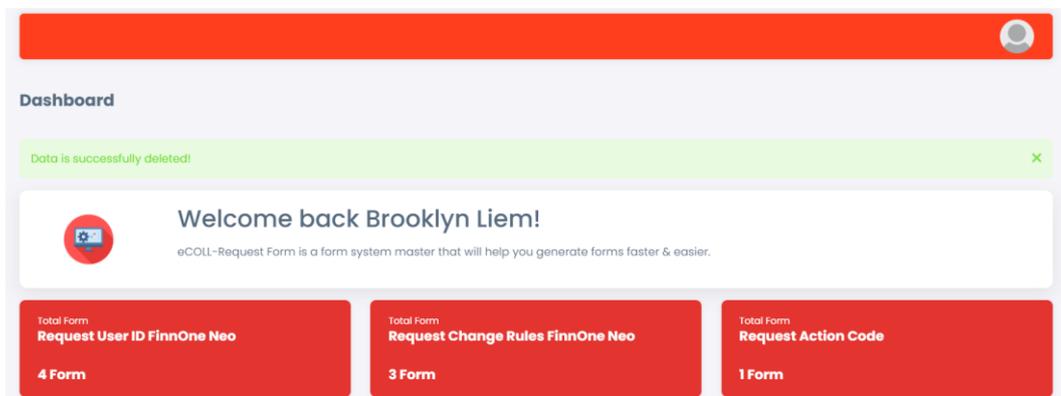
Gambar 3.83 Route Fungsi Edit

```
@if (auth()->check() && auth()->user()->role === 'Master' || auth()->user()->role === 'User' && $register->status !== 'APPROVED')
    <a href="/edit_register_new_users/{ $register->id }"
        class="dropdown-item">
        <i class="bx bx-edit"></i>Edit
    </a>
@endif
```

Gambar 3.84 Sub-menu Edit Register New Forms

Pada Gambar 3.84 yang merupakan source code untuk sub-menu tampilan pilihan Send Email memiliki pernyataan if yang menyatakan bahwa pilihan ini hanya dapat diakses user dengan role Master dan User saja karena user dengan role Head tidak memiliki kepentingan untuk mengakses pilihan ini. Namun, akses tersebut hanya akan muncul jika status dari form tersebut tidak APPROVED atau REJECT oleh Master yang memberikan persetujuan pada form.

Berikutnya merupakan pilihan sub-menu terakhir yang terdapat pada tombol Action yaitu pilihan Delete, fungsi untuk menghapus data form. Fungsi ini hanya dapat diakses oleh Master dari sistem agar tidak terjadi kelalaian penghapusan data yang tidak diinginkan oleh user lainnya. Berikut merupakan tampilan notifikasi yang ada ketika pilihan delete dilakukan.



Gambar 3.85 Notifikasi Fungsi Delete Berhasil

Gambar 3.85 menampilkan notifikasi yang muncul saat sub-menu Delete dilakukan dan jika berhasil terhapus pada database. Saat data berhasil terhapus pada database, maka jumlah total pada Card form terkait akan berkurang dan data tersebut tidak akan ada pada tabel Card Data Management. Berikut merupakan source code dari fungsi Delete di pada Data Register New Forms.

```

public function deleteregisterNew($id)
{
    $register_news = RegisterNew::find($id);

    //User Activity Report
    $users = auth()->user();
    $audit_trail = [
        'users_id' => auth()->id(), // Assuming $users is the authenticated user
        'name' => $users->name, // Assuming $users is the authenticated user
        'date' => Carbon::parse(now()->toDateString(), // Current date
        'activity' => "{$users->name} deleted Form Request User ID FinnOne Neo (Register New User ID)",
    ];
    $audit_trail = AuditTrail::query()->create($audit_trail);

    if ($register_news) {
        $register_news->delete();
        return redirect('/')->with('success', 'Data is successfully deleted!');
    } else {
        // If the model does not exist, return a 404 Not Found response
        abort(404);
    }
}

```

Gambar 3.86 Source Code Fungsi Delete

Fungsi Delete berdasarkan source code yang ada pada Gambar 3.86 dimulai dengan mencari file berdasarkan id pada model yang ingin dihapus. Lalu terdapat fungsi Audit Trail yang menyatakan bahwa data tersebut berhasil dihapus oleh user. Ketika data berhasil ditemukan maka fungsi Delete akan melakukan penghapusan

pada data tersebut dan memunculkan notifikasi sesuai dengan penjelasan sebelumnya pada Gambar 3.85. Namun source code tersebut hanya akan berfungsi ketika tombol Delete sudah terhubung dengan route web yang berkolerasi dengan source code terkait, maka berikut merupakan route web yang dimiliki untuk fungsi penghapusan data update beserta dengan source code untuk memanggil fungsi tersebut di bagian tombol Action tabel Data Register New Forms.

```
Route::get('data/delete/{id}', $controller_path . '\dashboard\Data@delete')->name('delete')->middleware(['auth']);
Route::get('data/deleteregisternew/{id}', $controller_path . '\dashboard\Data@deleteregisterNew')->name('deleteregisternew')->middleware(['auth']);
Route::get('data/deleteactiveinactive/{id}', $controller_path . '\dashboard\Data@deleteactiveInactive')->name('deleteactive')->middleware(['auth']);
Route::get('data/deletechangeisable/{id}', $controller_path . '\dashboard\Data@deletechangeDisable')->name('deletechanges')->middleware(['auth']);
Route::get('data/deletechangerules/{id}', $controller_path . '\dashboard\Data@deletechangeRules')->name('deletechangerules')->middleware(['auth']);
Route::get('data/deleterequstaction/{id}', $controller_path . '\dashboard\Data@deleterequstAction')->name('deleterequstaction')->middleware(['auth']);
```

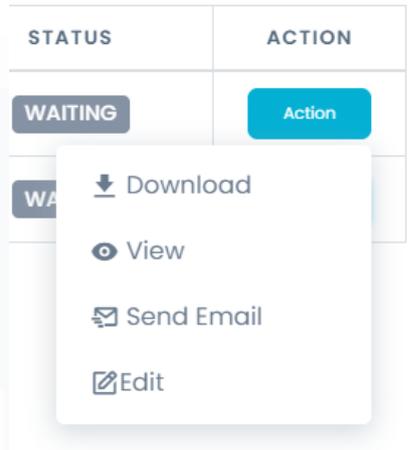
Gambar 3.87 Route Fungsi Delete

```
@if (auth()->check() && auth()->user()->role === 'Master')
    <a href="data/deleteregisternew/{ { $register->id } }"
        class="dropdown-item">
        <i class="bx bx-trash me-1"></i>Delete
    </a>
@endif
```

Gambar 3.88 Sub-menu Delete Register New Forms

Sesuai dengan yang telah disinggung sebelumnya bahwa akses terhadap fungsi Delete hanya dimiliki oleh user dengan role Master, maka Gambar 3.88 memiliki pernyataan if untuk mengimplementasikan batasan tersebut.

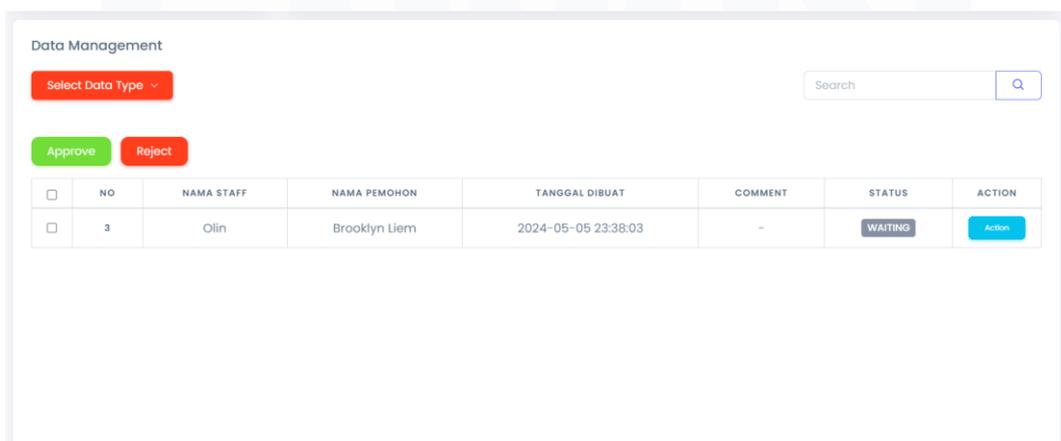
Fungsi-fungsi pada sub-menu tombol Action yang telah dijelaskan di atas berlaku untuk semua form yang dapat diakses oleh Master pada card Data Management Table. Dengan ini, tidak semua fungsi yang telah dijelaskan dapat diakses oleh User ataupun Head, maupun sebaliknya. Maka berikut merupakan penjelasan mengenai perbedaan untuk masing-masing card Data Management Table berdasarkan role User dan Head.



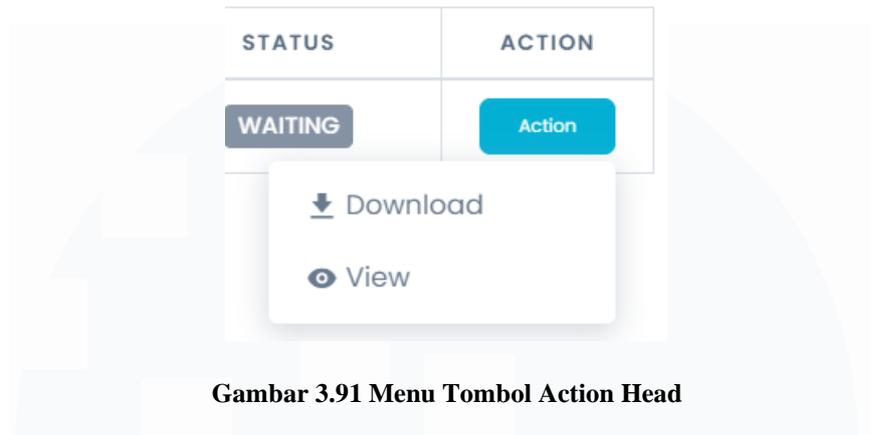
Gambar 3.89 Menu Tombol Action User

Gambar 3.89 merupakan tampilan menu pada tombol Action yang dimiliki oleh role User. Pilihan untuk Download, View, Send Email, maupun Edit memiliki fungsi dan alur yang sama dengan pilihan pada menu tombol Action yang dimiliki oleh role Master. Seluruh fungsi juga mengimplementasikan fungsi Audit Trail dengan pernyataan yang serupa dengan implementasi Audit Trail pada menu pilihan Master.

Sedangkan pada Card Data Management Table yang dimiliki oleh role Head memiliki beberapa fitur tambahan yang tidak dimiliki oleh role User dan Master. Berikut merupakan tampilan Card Data Management Table dan menu Action ketika di akses user dengan role Division Head, Group Head, ataupun Director.



Gambar 3.90 Tampilan Card Data Management Table Master



Gambar 3.91 Menu Tombol Action Head

Gambar 3.90 menampilkan Card Data Management Table yang dimiliki oleh role Master, dapat dilihat bahwa terdapat tombol Approve dan Reject untuk memberi persetujuan terhadap form yang diajukan oleh user. Sedangkan pada tombol menu Action, memiliki pilihan yang sekiranya diperlukan untuk mempertimbangkan persetujuan yang akan diberikan oleh Head pada form yang diajukan, yaitu terdiri dari pilihan untuk Download dan View saja. Maka, berikut merupakan penjelasan dan source code yang dimiliki untuk fitur Approve dan Reject pada Data Register New Forms.

```

public function approveregisterNew(Request $request)
{
    $ids = $request->input('ids');

    // Check if IDs were passed and if they are in array format
    if (!empty($ids) && is_array($ids)) {
        // Filter the IDs to ensure they are numeric
        $validIds = Arr::where($ids, function ($value, $key) {
            return is_numeric($value);
        });

        // Check if there are valid IDs to update
        if (!empty($validIds)) {
            // Use the update method with whereIn to update multiple records
            RegisterNew::whereIn('id', $validIds)->update(['status' => 'APPROVED', 'commentregister' => '-']);

            return response()->json(['success' => true, 'message' => 'Successfully approved']);
        } else {
            return response()->json(['success' => false, 'message' => 'There is no data to approve!'], 400);
        }
    } else {
        return response()->json(['success' => false, 'message' => 'There is no data you have chosen to approve'], 400);
    }
}

```

Gambar 3.92 Source Code fungsi Approve

```

public function rejectregisterNew(Request $request)
{
    // Retrieve IDs, status, and comment from the request
    $ids = $request->input('ids');
    $status = $request->input('status');
    $commentregister = $request->input('commentregister');

    // Check if IDs were passed and if they are in array format
    if (!empty($ids) && is_array($ids)) {
        // Filter the IDs to ensure they are numeric
        $validIds = Arr::where($ids, function ($value, $key) {
            return is_numeric($value);
        });

        // Check if there are valid IDs to update
        if (!empty($validIds)) {
            // Use the update method with whereIn to update multiple records
            RegisterNew::whereIn('id', $validIds)
                ->update(['status' => $status, 'commentregister' => $commentregister]);

            // Log the rejection comment if available
            if (!empty($commentregister)) {
                Log::info('Rejection Comment:', ['commentregister' => $commentregister]);
            }

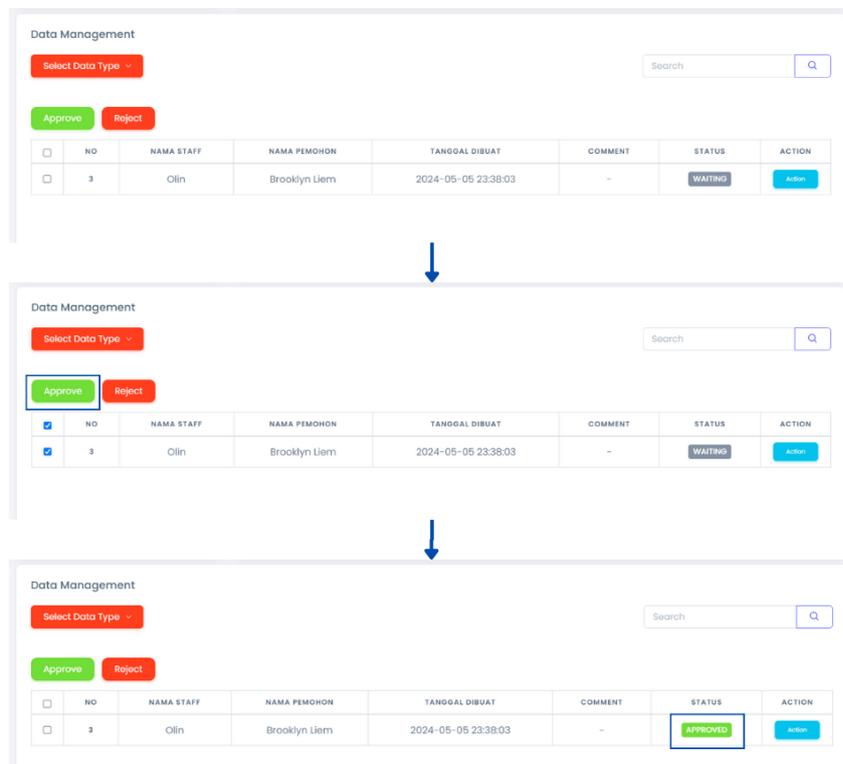
            return response()->json(['success' => true, 'message' => 'Successfully rejected']);
        } else {
            return response()->json(['success' => false, 'message' => 'There is no data to reject!'], 400);
        }
    } else {
        return response()->json(['success' => false, 'message' => 'There is no data you choose to reject'], 400);
    }
}

```

Gambar 3.93 Source Code fungsi Reject

Pada Gambar 3.92 merupakan source code dari fungsi Approve untuk data Register New Forms dan Gambar 3.93 merupakan source code dari fungsi Reject. Kedua fungsi yang ada diawali dengan pencarian id yang sesuai pada database, sama seperti fungsi-fungsi sebelumnya. Lalu untuk fungsi Approve akan melakukan deteksi jika id yang ditemukan dalam format numerik atau tidak. Ketika berhasil dideteksi dengan format yang diinginkan, maka fungsi Approve akan melakukan pengecekan terhadap status yang sudah dimiliki oleh form terkait. Lalu ketika semua sesuai, fungsi Approve akan menggantikan status dari form yang sebelumnya WAITING menjadi APPROVE tanpa keterangan pada kolom comment yang berarti form tersebut telah menerima persetujuan Head terkait. Berikut merupakan alur proses yang terjadi saat form yang berhasil mendapatkan persetujuan.

MULTIMEDIA
NUSANTARA



Gambar 3.94 Alur Proses Approve Form

Sesuai dengan alur proses yang ada pada Gambar 3.94, ketika form menerima tombol Approve maka tampilan pada kolom status akan berubah menjadi Approve. Berikut merupakan source code dan JavaScript yang memicu hal tersebut.

```

<td>@php
    $status = $register['status'];
    $badgeClass =
        $status === 'APPROVED'
        ? 'bg-success'
        : ($status === 'REJECTED'
        ? 'bg-danger'
        : 'bg-secondary');
@endphp
    <span class="badge {{ $badgeClass }}">{{ $status }}</span>
</td>

```

Gambar 3.95 Source Code Kolom Status

```

<script>
  // Event listener for approve and reject buttons
  document.getElementById('approveBtnRegister').addEventListener('click', function() {
    updateStatus('APPROVED', '-');
  });

  document.getElementById('rejectBtnRegister').addEventListener('click', function() {
    $('#commentModalRegister').modal('show');
  });

  document.getElementById('saveCommentBtnRegister').addEventListener('click', function() {
    const commentregister = document.getElementById('commentregister').value; // Update the comment variable
    updateStatus('REJECTED', commentregister);
  });

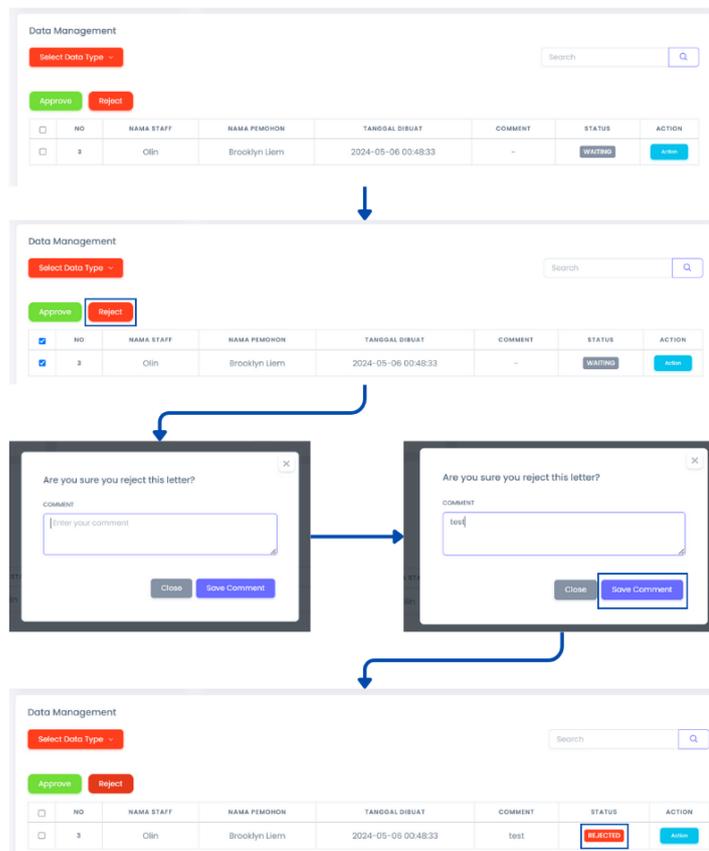
  // Function to send AJAX request to update status
  function updateStatus(status, commentregister) {
    var ids = []; // Array to store selected IDs
    var checkboxes = document.querySelectorAll('.checkbox_ids_register:checked'); // Get all checked checkboxes
    checkboxes.forEach(function(checkbox) {
      ids.push(checkbox.value); // Push the ID to the array
    });

    // Send AJAX request
    $.ajax({
      type: 'POST',
      url: status === 'APPROVED' ? '{{ route('approve_status_register') }}' :
        '{{ route('reject_status_register') }}', // Update with your route
      data: {
        _token: '{{ csrf_token() }}', // Add CSRF token if using Laravel
        ids: ids,
        status: status,
        commentregister: commentregister
      },
      success: function(response) {
        console.log(response);
        // Refresh the page or update the UI as needed
        location.reload(); // Example: reload the page
      },
      error: function(xhr, status, error) {
        console.error(xhr.responseText);
        // Handle errors
      }
    });
  }
}
</script>

```

Gambar 3.96 Source Code JavaScript Kolom Status

Fungsi Reject juga memiliki alur proses yang serupa dengan fungsi Approve, namun yang membedakan adalah fungsi ini disertakan dengan kewajiban untuk mengisi keterangan atau alasan mengapa form tersebut tidak disetujui dengan mengisi comment pada Modal Comment yang ada untuk setiap formnya. Berikut merupakan alur proses yang terjadi saat form yang tidak berhasil mendapatkan persetujuan.



Gambar 3.97 Alur Proses Reject Form

Sesuai dengan alur proses yang ada pada Gambar 3.97, ketika form menerima tombol Reject maka terdapat pop up modal yang berisikan area untuk input komentar atau alasan mengenai tidak persetujuan yang akan dilakukan terhadap form. Lalu saat keterangan sudah terisi dan tombol Save Comment sudah terpilih, maka comment dan status akan secara otomatis mengalami update untuk form terkait. Source code dan JavaScript dapat dilihat pada Gambar 3.95 dan 3.96. Selain itu, untuk memudahkan proses tersebut pada role Head sudah diimplementasikan filter melalui pernyataan if agar hanya data yang melibatkan user Head tersebut yang akan muncul pada Card Data Management. Berikut merupakan fungsi filter yang diimplementasikan.

```

@if (auth()->check() &&
    (auth()->user()->role === 'Division Head' ||
    auth()->user()->role === 'Group Head' ||
    auth()->user()->role === 'Director'))
    @if ($register['namadirectsupervisor'] === auth()->user()->name)
        <!-- Filter data by $name -->

```

Gambar 3.98 Filter Data Form

Dengan ini, dapat mempermudah proses Maker & Checker atau Approve & Reject pada form yang melibatkan user Head yang terkait. Lalu untuk menyambungkan setiap fungsi yang berkaitan untuk pilihan Approve & Reject, berikut merupakan route dari fitur Approve & Reject untuk Data Register New Forms.

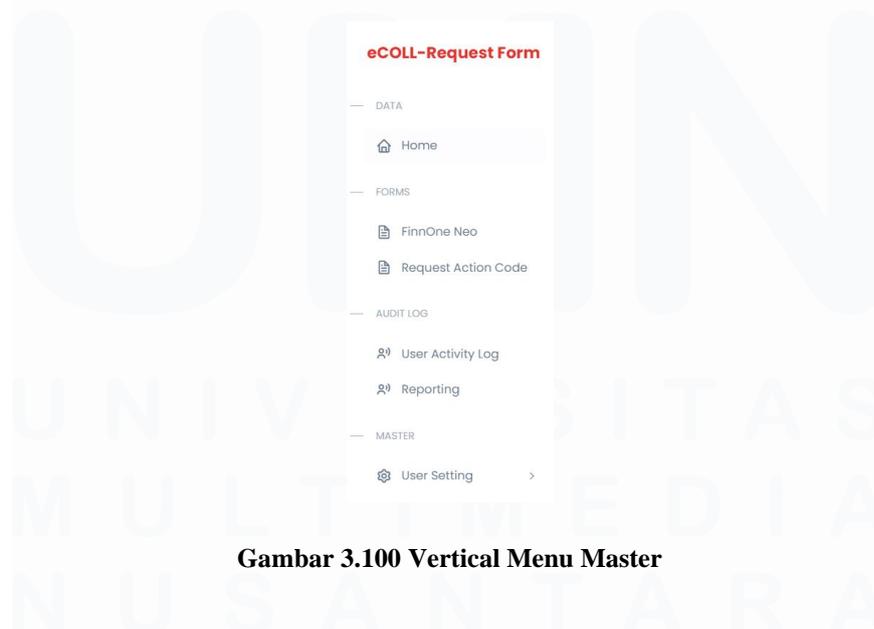
```

Route::get('/approve_status_register', $controller_path . '\dashboard\Data@approveregisterNew')->middleware(['auth']);
Route::post('/approve_status_register', $controller_path . '\dashboard\Data@approveregisterNew')->name('approve_status_register')->middleware(['auth']);
Route::get('/reject_status_register', $controller_path . '\dashboard\Data@rejectregisterNew')->middleware(['auth']);
Route::post('/reject_status_register', $controller_path . '\dashboard\Data@rejectregisterNew')->name('reject_status_register')->middleware(['auth']);

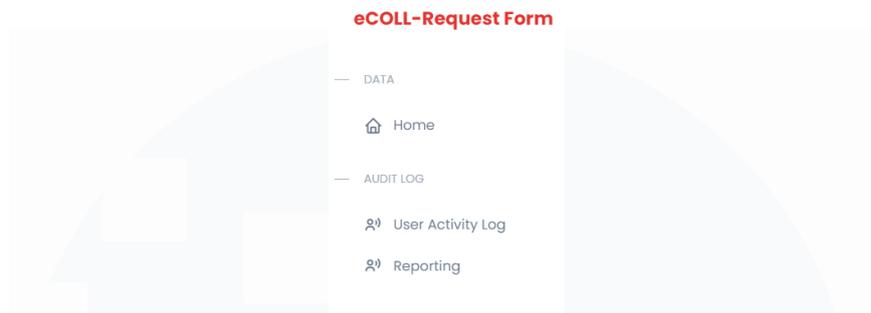
```

Gambar 3.99 Route Fungsi Approve & Reject

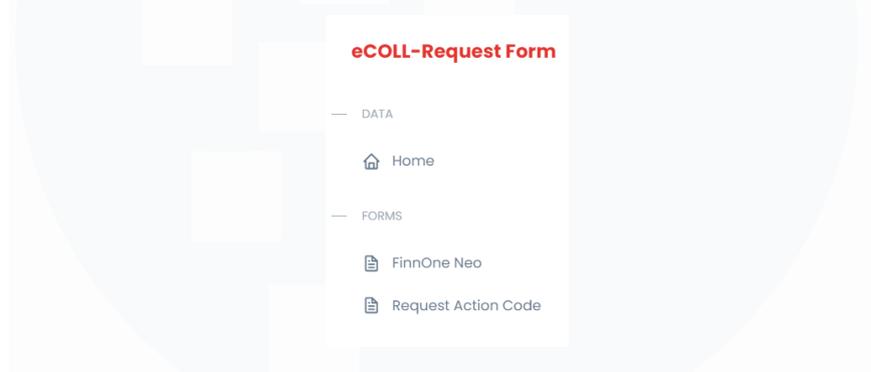
Selain Dashboard, terdapat Vertical Menu yang merupakan menu tampilan yang dapat diakses berdasarkan masing-masing role pada sistem eCOLL-Request Form. Berikut merupakan tampilan Vertical Menu berdasarkan role yang ada pada sistem.



Gambar 3.100 Vertical Menu Master



Gambar 3.101 Vertical Menu Head



Gambar 3.102 Vertical Menu User

Berdasarkan Gambar 3.100, 3.101, dan 3.102, role dengan Vertical Menu yang memiliki akses untuk fitur terbanyak adalah role Master. Dimana, role master memiliki akses untuk menu membuat form pada pilihan dan Request Action Code, akses untuk melihat User Activity Log dan Reporting, dan akses untuk melakukan pengaturan pada User Setting terhadap akun-akun yang ada pada sistem eCOLL-Request Form. Gambar 3.101 merupakan tampilan pada Vertical Menu untuk role Head, dimana user dengan role Head hanya memiliki akses untuk melihat User Activity Log dan Reporting untuk keperluan pengawasan audit jika diperlukan. Berikut merupakan penjelasan mengenai tampilan page dan source code untuk User Activity Log dan Reporting.

NO	NAME	DATE	ACTIVITY	UPDATE
1	Brooklyn Liem	2024-05-06	Brooklyn Liem Logged In To The Website	2024-05-06 01:48:55
2	Olin	2024-05-06	Olin Logged Out From The Website	2024-05-06 01:42:45
3	Olin	2024-05-06	Olin Logged In To The Website	2024-05-06 01:34:32
4	Brooklyn Liem	2024-05-06	Brooklyn Liem Logged Out From The Website	2024-05-06 01:34:19
5	Brooklyn Liem	2024-05-06	Brooklyn Liem Logged In To The Website	2024-05-06 01:33:01
6	Brooklyn Liem	2024-05-06	Brooklyn Liem Logged Out From The Website	2024-05-06 01:32:42
7	Brooklyn Liem	2024-05-06	Brooklyn Liem Logged In To The Website	2024-05-06 01:32:26
8	Brooklyn Liem	2024-05-06	Brooklyn Liem Logged Out From The Website	2024-05-06 01:32:11

Gambar 3.103 Page User Activity Log

Page User Activity Log yang terdapat pada Gambar 3.103 hanya dapat diakses oleh user yang memiliki role Master dan Head saja karena umumnya hanya digunakan untuk keperluan audit. Pada page tersebut berisikan history aktivitas yang dilakukan oleh masing-masing akun yang terdaftar pada sistem eCOLL-Request Form dalam bentuk tabel. Dengan mempertimbangkan keperluan Audit, kolom-kolom yang tersedia pada page User Activity Log meliputi kolom Name sebagai nama user yang mengakses, Date sebagai tanggal saat user melakukan aktivitas tersebut, Activity berisikan pernyataan history aktivitas yang dilakukan oleh seluruh user pada sistem, dan Update yang berisikan timestamp spesifik untuk setiap aktivitasnya. Berikut merupakan source code yang berkaitan dengan tampilan tabel pada page User Activity Log.

```
public function up()
{
    Schema::create('audit_trails', function (Blueprint $table) {
        $table->bigIncrements('id');
        $table->string('users_id');
        $table->string('name');
        $table->string('date');
        $table->text('activity');
        $table->timestamps();
        $table->softDeletes();
    });
}
```

Gambar 3.104 Source Code Tabel audit_trails

```

<div class="table-responsive text-nowrap mt-5">
  <table class="table table-bordered text-center">
    <thead class="custom-header">
      <tr>
        <th>No</th>
        <th>Name</th>
        <th>date</th>
        <th>Activity</th>
        <th>Update</th>
      </tr>
    </thead>
    <tbody>
      <@php
        $no = 1;
      @endphp>
      <@foreach ($audit_trails as $audit)>
        <tr>
          <th scope="audit">{{ $no++ }}</th>
          <td>{{ $audit['name'] }}</td>
          <td>{{ $audit['date'] }}</td>
          <td>{{ $audit['activity'] }}</td>
          <td>{{ $audit['updated_at']->timezone('Asia/Jakarta')->format('Y-m-d H:i:s') }}</td>
        </tr>
      <@endforeach>
    </tbody>
  </table>
  <div class="d-flex" style="margin-top: 15px; float:right">
    <{!! $audit_trails->links() !!>
  </div>
</div>

```

Gambar 3.105 Source Code Tampilan Page User Activity Log

Gambar 3.104 merupakan source code yang membantu untuk membuat tabel `audit_trails` yang terhubung pada tabel User Activity Log yang ada. Sedangkan Gambar 3.105 merupakan source code yang menampilkan tabel pada page User Activity Log. Selain itu, history aktivitas pada masing-masing akun untuk mengisi tabel `audit_trails` tersebut dapat terisi melalui source code yang ada pada setiap fungsi atau fitur yang ada seperti pada Gambar 3.86, yang mewakili aktivitas Delete yang dilakukan oleh user.

Salah satu penyebab page User Activity Log ini ada yaitu untuk kepentingan memudahkan proses Audit perusahaan nantinya, maka disediakan fitur untuk mengunduh melalui tombol Download yang dapat diakses oleh user Master dan Head yang bersangkutan. Berikut merupakan source code untuk fitur Download tersebut.

MULTIMEDIA
NUSANTARA

```
public function downloadAuditTrails()
{
    return Excel::download(new DownloadAuditTrail, 'Data Audit Trail.xlsx');
}
```

Gambar 3.106 Source Code Fitur Download User Activity Log

```
<div class="download mb-3 mb-md-0">
  <a href="{{ route('downloadaudittrail') }}" target="_blank" class="btn btn-success">Download</a>
</div>
```

Gambar 3.107 Source Code Tampilan Tombol Download

Dengan ini, berikut merupakan hasil dari fitur download yang dapat dilakukan pada tabel User Activity Log sistem eCOLL-Request Form sehingga memudahkan proses Audit yang mungkin terjadi pada divisi Consumer Collection & Recovery.

	A	B	C	D
1	Name	Activity	Updated At	
2	Brooklyn I	Brooklyn I	2024-05-03 14:15:47	
3	Brooklyn I	Brooklyn I	2024-05-03 14:16:06	
4	Brooklyn I	Brooklyn I	2024-05-03 14:16:30	
5	Brooklyn I	Brooklyn I	2024-05-03 14:17:01	
6	Brooklyn I	Brooklyn I	2024-05-03 14:17:32	
7	Brooklyn I	Brooklyn I	2024-05-03 14:18:03	
8	Brooklyn I	Brooklyn I	2024-05-03 14:19:26	
9	Brooklyn I	Brooklyn I	2024-05-03 14:31:53	
10	Brooklyn I	Brooklyn I	2024-05-03 14:55:31	
11	Brooklyn I	Brooklyn I	2024-05-03 14:55:44	

Gambar 3.108 Hasil Download Tabel User Activity Log

Pada page Reporting juga memiliki alur proses dan fungsi yang hampir serupa dengan User Activity Log, hanya saja page tersebut telah dibuat khusus untuk history aktivitas yang memiliki hubungan dengan data form yang ada pada sistem eCOLL-Request Form.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Master / User Activity Report

Download

Carl

NO	NAME	DATE	ACTIVITY	UPDATE
1	Brooklyn Liem	2024-05-06	Brooklyn Liem created Request User ID FinnOne Neo (Register New User ID)	2024-05-06 00:48:35
2	Brooklyn Liem	2024-05-05	Brooklyn Liem created Request User ID FinnOne Neo (Register New User ID)	2024-05-05 23:38:05
3	Brooklyn Liem	2024-05-05	Brooklyn Liem created Request User ID FinnOne Neo (Register New User ID)	2024-05-05 22:59:37
4	Brooklyn Liem	2024-05-05	Brooklyn Liem created Request User ID FinnOne Neo (Register New User ID)	2024-05-05 20:46:51
5	Brooklyn Liem	2024-05-03	Brooklyn Liem created Form Request Action Code	2024-05-03 14:19:26
6	Brooklyn Liem	2024-05-03	Brooklyn Liem created Request Change FinnOne Neo Rules Classification- Allocation	2024-05-03 14:18:03
7	Brooklyn Liem	2024-05-03	Brooklyn Liem created Request Change FinnOne Neo Rules Classification- Allocation	2024-05-03 14:17:32
8	Brooklyn Liem	2024-05-03	Brooklyn Liem created Request Change FinnOne Neo Rules Classification- Allocation	2024-05-03 14:17:01

Gambar 3.109 Page Reporting

Sesuai dengan tampilan page Reporting pada Gambar 3.109, memiliki kemiripan yang hampir serupa dengan page User Activity Log. Namun lebih terkhusus hanya untuk mencatat aktivitas-aktivitas pembuatan form yang telah terjadi oleh akun yang terdaftar pada sistem eCOLL-Request Form. Tabel tersebut juga dapat diunduh dengan tersedianya fitur Download pada page Reporting ini. Maka, berikut merupakan source code yang memiliki korelasi dengan adanya tabel dan fitur Download pada page Reporting.

```
public function up()
{
    Schema::create('audit_reportings', function (Blueprint $table) {
        $table->bigIncrements('id');
        $table->string('users_id');
        $table->string('name');
        $table->string('date');
        $table->text('activity');
        $table->timestamps();
        $table->softDeletes();
    });
}
```

Gambar 3.110 Source Code Tabel audit_reportings

```

<div class="table-responsive text-nowrap mt-5">
  <table class="table table-bordered text-center">
    <thead class="custom-header">
      <tr>
        <th>No</th>
        <th>Name</th>
        <th>date</th>
        <th>Activity</th>
        <th>Update</th>
      </tr>
    </thead>
    <tbody>
      @php
        $no = 1;
      @endphp
      @foreach ($audit_reportings as $reporting)
        <tr>
          <th scope="reporting">{{ $no++ }}</th>
          <td>{{ $reporting['name'] }}</td>
          <td>{{ $reporting['date'] }}</td>
          <td>{{ $reporting['activity'] }}</td>
          <td>{{ $reporting['updated_at']->timezone('Asia/Jakarta')->format('Y-m-d H:i:s') }}</td>
        </tr>
      @endforeach
    </tbody>
  </table>
  <div class="d-flex" style="margin-top: 15px; float:right">
    {!! $audit_reportings->links() !!}
  </div>
</div>

```

Gambar 3.111 Source Code Tampilan Page Reporting

```

public function downloadAuditReportings()
{
    return Excel::download(new DownloadAuditReporting, 'Data Audit Trail.xlsx');
}

```

Gambar 3.112 Source Code Fitur Download Reporting

```

<div class="download mb-3 mb-md-0">
  <a href="{{ route('downloadauditreporting') }}" target="_blank" class="btn btn-success">Download</a>
</div>

```

Gambar 3.113 Source Code Tampilan Tombol Download

Dengan ini, berikut merupakan hasil dari fitur download yang dapat dilakukan pada tabel Reporting sistem eCOLL-Request Form sehingga memudahkan proses Audit yang mungkin terjadi pada divisi Consumer Collection & Recovery.

	A	B	C	D
1	Name	Activity	Updated At	
2	Brooklyn I	Brooklyn I	2024-05-03 14:15:47	
3	Brooklyn I	Brooklyn I	2024-05-03 14:16:06	
4	Brooklyn I	Brooklyn I	2024-05-03 14:16:30	
5	Brooklyn I	Brooklyn I	2024-05-03 14:17:01	
6	Brooklyn I	Brooklyn I	2024-05-03 14:17:32	
7	Brooklyn I	Brooklyn I	2024-05-03 14:18:03	
8	Brooklyn I	Brooklyn I	2024-05-03 14:19:26	
9	Brooklyn I	Brooklyn I	2024-05-05 20:46:51	
10	Brooklyn I	Brooklyn I	2024-05-05 22:59:37	
11	Brooklyn I	Brooklyn I	2024-05-05 23:38:05	

Gambar 3.114 Hasil Download Tabel Reporting

Penjelasan dari kedua page tersebut merupakan akhir dari fitur-fitur yang dapat diakses oleh role Head. Maka, berikut merupakan penjelasan mengenai menu pembuatan form yang dapat diakses oleh role Master dan User yang merupakan tujuan utama dirancangnya sistem eCOLL-Request Form. Berikut merupakan tampilan dari masing-masing page input form yang ada untuk setiap formnya.

Gambar 3.115 Page Input Form User ID Management

Gambar 3.115 merupakan tampilan dari page Input Form untuk User ID Management yang terdiri dari Register User ID Baru, Active Inactive, dan Change/Disable. Pada setiap input form yang ada, diharapkan dapat meningkatkan proses pembuatan form sehingga bersifat lebih efektif dan efisien. Maka pada proses input terdapat fitur seperti autofill dan dropdown menu. Berikut merupakan

salah satu source code dari form Register User ID yang terhubung dengan fitur terkait.

```
<div class="inputmb-3 col-md-6">
  <label for="nik" class="form-label">NIK STAFF</label>
  <select class="form-select" id="nik" name="nik">
    <option value="" disabled selected hidden>Pilih NIK</option>
    @foreach ($users as $userdata)
      @if ($userdata->role == 'User')
        <option value="{{ $userdata->nik }}" data-nama="{{ $userdata->name }}"
          data-userid="{{ $userdata->userid }}">
          {{ $userdata->nik }}</option>
      @endif
    @endforeach
  </select>
  @error('nik')
    <span class="text-danger">{{ $message }}</span>
  @enderror
</div>
```

Gambar 3.116 Source Code Memanggil Fitur AutoFill

```
document.getElementById('nik').addEventListener('change', function() {
  var selectedOption = this.options[this.selectedIndex];
  var nama = selectedOption.getAttribute('data-nama');
  var userid = selectedOption.getAttribute('data-userid');

  document.getElementById('nama').value = nama;
  document.getElementById('userid').value = userid;
});
```

Gambar 3.117 Source Code JavaScript Fitur AutoFill

Pada Gambar 3.116 merupakan kolom NIK yang merupakan menu dropdown, yang ketika salah satu nilai dropdown tersebut dipilih akan memicu fitur AutoFill pada kolom nama dan userid. Hal tersebut dibantu dengan adanya JavaScript yang mengatur pada Gambar 3.117. Berikut merupakan tampilan ketika AutoFill berhasil untuk kolom nama dan userid pada input form Register User ID Baru.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

NIK STAFF

00000011882

NAMA STAFF

Olin

USER ID STAFF

00882

Gambar 3.118 Tampilan Fitur AutoFill

Fitur AutoFill yang dimiliki pada input form Register User ID Baru juga diimplementasikan pada seluruh page input form yang ada untuk menghindari kelalaian dalam menginput data-data penting, sehingga proses pembuatan form dapat berjalan lebih cepat dengan hasil yang maksimal.

eCOLL-Request Form

Change Rule Classification Change Rule Allocation Others

NOMOR SURAT
NO : CPMS/FinnoneNeo/FRC/NRCA/2024/004

JENIS PERMOHONAN
Change Rule Classification

DESCRIPTION

REASON REQUEST

REQUEST DATE

FROM: da/mm/yyyy UNTIL: da/mm/yyyy

APPROVAL

NAMA REQUESTOR: Pilih Nama Requestor

NAMA DIVISION HEAD: Pilih Nama Division Head

Gambar 3.119 Page Input Form Change Rules Classification - Allocation

Gambar 3.119 merupakan tampilan dari page Input Form untuk Change Rules Classification – Allocation yang terdiri dari Change Rule Classification, Change Rule Allocation, dan Others. Selain fitur AutoFill pada page input form ini, mengimplementasikan fitur Auto Generated pada nomor form. Berikut merupakan source code terkait yang membantu untuk mengimplementasikan fitur tersebut.

```

public function generateCode()
{
    // Get the last used number and year from the database
    $lastTemplate = ChangeRules::latest()->first();
    $currentYear = date('Y'); // Set to the actual current year

    if ($lastTemplate) {
        $lastYear = date('Y', strtotime($lastTemplate->created_at));
        $lastCode = $lastTemplate->nomorsurat;
        preg_match('/(\d+)/', $lastCode, $matches); // Extract the number part from the code
        $currentNumber = ($currentYear == $lastYear) ? ($matches[0] + 1) : 1;

        while (ChangeRules::where('nomorsurat', "NO : CPMS/FinnoneNeo/FRC/NRCA/({$currentYear})/({$currentNumber})->exists()") {
            $currentNumber++;
        }
    } else {
        $currentNumber = 1;
    }

    function arabicToRoman($number)
    {
        $map = [
            'M' => 1000,
            'CM' => 900,
            'D' => 500,
            'CD' => 400,
            'C' => 100,
            'XC' => 90,
            'L' => 50,
            'XL' => 40,
            'X' => 10,
            'IX' => 9,
            'V' => 5,
            'IV' => 4,
            'I' => 1,
        ];

        $result = '';

        foreach ($map as $roman => $arabic) {
            while ($number >= $arabic) {
                $result .= $roman;
                $number -= $arabic;
            }
        }

        return 'NO : CPMS/FinnoneNeo/FRC/NRCA/' . $currentYear . '/' . str_pad($currentNumber, 3, '0', STR_PAD_LEFT) . arabicToRoman($currentNumber);
    }
}

```

Gambar 3.120 Source Code Auto Generated Code

```

<div class="inputmb-3 col-md-6" style="margin-bottom:15px;">
    <label for="nomorsurat" class="form-label">Nomor Surat</label>
    <input class="form-control" name="nomorsurat" id="nomorsurat" value="{{ $newCode }}"
        readonly required>
</div>

```

Gambar 3.121 Source Code Auto Generated Code

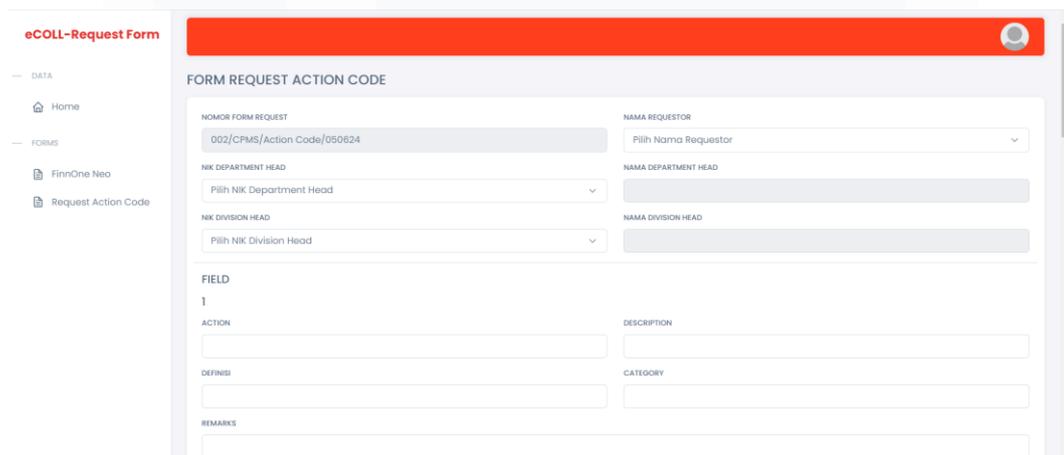
Dengan mengimplementasikan fitur Auto Generated Code melalui codingan pada Gambar 3.120, maka user tidak perlu khawatir untuk memikirkan nomor form pada form yang akan dibuat. Berikut merupakan tampilan implementasi dari fitur Auto Generated Code pada salah satu form yang menggunakan fitur tersebut yaitu Change Rule Classification.

NOMOR SURAT

NO : CPMS/FinnoneNeo/FRC/NRCA/2024/004

Gambar 3.122 Tampilan Fitur Auto Generated Code

Fitur Auto Generated Code ini juga diimplementasikan pada form Request Action Code, hanya saja format dari nomor form berbeda dengan apa yang digunakan pada form Change Rules Classification – Allocation. Berikut merupakan tampilan dari page Input Form untuk Request Action Code yang merupakan template form terakhir dari sistem eCOLL-Request Form untuk saat ini beserta dengan implementasi fitur Auto Fill dan Auto Generated Code yang dapat terlihat pada Gambar 3.123.



The screenshot displays the 'eCOLL-Request Form' interface. On the left is a sidebar menu with 'Request Action Code' selected. The main content area is titled 'FORM REQUEST ACTION CODE' and contains several input fields: 'NOMOR FORM REQUEST' (002/CPMS/Action Code/050624), 'NAMA REQUESTOR' (dropdown), 'NIK DEPARTMENT HEAD' (dropdown), 'NAMA DEPARTMENT HEAD' (text), 'NIK DIVISION HEAD' (dropdown), and 'NAMA DIVISION HEAD' (text). Below these is a 'FIELD' section with a table for '1' containing 'ACTION', 'DESCRIPTION', 'DEFINIS', 'CATEGORY', and 'REMARKS' fields.

Gambar 3.123 Page Input Form Request Action Code

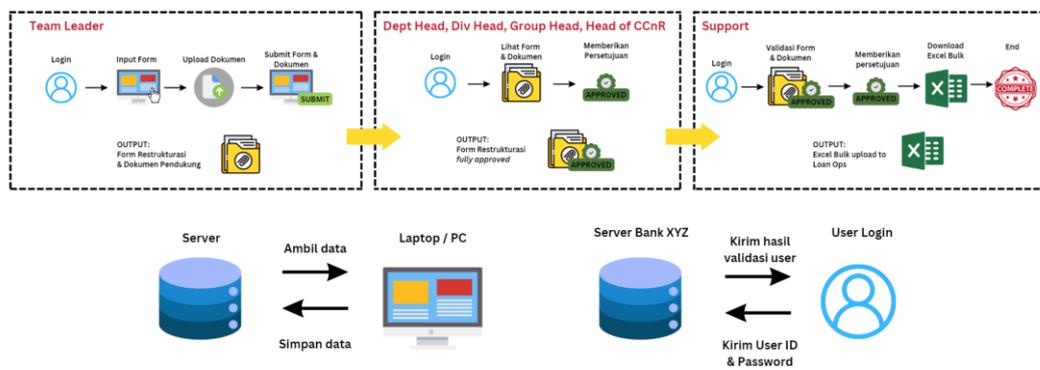
Dengan ini, tahap development pada sistem eCOLL-Request Form untuk saat ini berakhir pada fitur tersebut. Maka selanjutnya merupakan tahap upload ke dalam sistem internal divisi Consumer Collection & Recovery PT Bank XYZ.

Pada tahap upload, tim awalnya mengalami kendala dalam membantu proses instalasai salah satu komponen yang diperlukan untuk menginstalasi package Laravel yaitu composer karena komponen tersebut tidak masuk ke dalam standar aplikasi atau sistem yang digunakan oleh PT Bank XYZ maka harus dilakukannya pengajuan atau mengganti bahasa pemrograman dengan yang lain.

Tanpa adanya composer, package-package Laravel yang membantu adanya fitur-fitur seperti Send Email dapat hilang dan tidak dapat digunakan pada sistem eCOLL-Request Form. Maka, untuk saat ini tim masih dalam proses mengajukan tiket untuk Laravel agar mendapatkan persetujuan dan dapat digunakan pada sistem eCOLL-Request Form dan sekaligus mempelajari bahasa pemrograman PHP Native, jika pengajuan tersebut tidak berhasil dalam mendapatkan persetujuan. Namun ketika nantinya kendala tersebut ditemukan solusinya, tim akan membantu proses tersebut hingga sistem eCOLL-Request Form dan eCOLL-Letters dapat berjalan dengan baik pada sistem internal Bank XYZ. Dengan melakukan testing, menyediakan user manual, hingga training untuk divisi Consumer Collection & Recovery hingga sistem dapat Go Live pada akhir Mei.

3.2.4 Minggu 10: Analisis Proyek EUC RnR CC

Pada Minggu ke-10, tim mulai memasuki masa untuk menjalankan sebuah proyek baru yaitu proyek EUC RnR CC. Proyek ini memiliki alur proses yang kurang lebih sama dengan alur proses eCOLL-Letters maupun eCOLL-Request Form. Berikut merupakan alur proses dari sistem EUC RnR CC.



Gambar 3.124 Page Add User eCOLL-Request Form

Pada proyek EUC RnR CC terdapat 3 proses utama berdasarkan user yang akan memiliki akses pada sistem ini berdasarkan pada Gambar 3.124. Pada Team Leader, akan memiliki akses untuk proses pembuatan form yang nantinya akan tersimpan secara otomatis ke dalam server local hingga tahap pengumpulan dengan luaran

Form Restrukturasi beserta dengan dokumen pendukungnya. Lalu luaran tersebut akan dikumpulkan dan dilanjutkan ke proses selanjutnya yang akan melibatkan Department Head, Division Head, Group Head, maupun Head of Consumer Collection and Recovery. Tahap ini akan mencakup proses persetujuan yang nantinya berupa pembaharuan status untuk form yang telah diterima agar bisa lanjut ke proses selanjutnya, jika sudah menerima persetujuan. Lalu proses tersebut nantinya akan melibatkan Collection Support dimana proses validasi terhadap dokumen akan dilakukan dan data-data tersebut akan ditarik dan menghasilkan luaran berupa File Excel yang dapat diunduh oleh bagian Collection Support sebagai akhir dari alur proses proyek EuC RnR CC.

3.2.5 Minggu 11-19: Mengerjakan Proyek EUC RnR CC

Proyek EUC Rnr CC merupakan proyek baru yang diberikan kepada tim System Development & Infrastructure Associate yang sedang dalam progress pengerjaan sekarang, maka tim akan melanjutkan pengerjaan proyek tersebut ketika data progress sudah diteruskan oleh pihak yang bertanggung jawab kepada tim.

Pada minggu ke-11, tim telah mulai menerima briefing dan melewati tahap analisis untuk proyek EUC RnR CC selagi menunggu kendala yang dialami eCOLL-Letters dan eCOLL-Request Form terpecahkan. Hal yang dialami oleh proyek eCOLL-Letters dan eCOLL-Request Form, tidak menjadi kendala pada proyek ini karena bahasa pemograman yang digunakan oleh proyek EUC RnR CC sudah menjadi standar perusahaan dan sering digunakan oleh PT Bank XYZ dalam sistem internal perusahaan yaitu PHP Native. Berhubungan laporan magang ini akan diteruskan kepada pihak kampus pada awal Mei, maka progress mengenai proyek ini belum dapat disampaikan oleh penulis sebagai peserta magang pada PT Bank XYZ karena masih belum terdapat progress dari proyek secara signifikan yang dapat disampaikan karena masa pengerjaan proyek nantinya akan berlangsung selama bulan Mei hingga Juni saat program magang tim System Development & Infrastructure berakhir.

3.2.6 Minggu 2-18: Weekly Meeting

Sejak minggu ke-2 berlangsungnya program magang tim System Development & Infrastructure Associate hingga minggu ke-18 nanti, telah dan akan selalu dilaksanakannya weekly meeting mengenai progress update terhadap proyek yang dilaksanakan oleh tim selama program magang berlangsung setiap minggunya. Pertemuan ini merupakan pertemuan yang dihadiri oleh pihak-pihak yang terlibat dan sekiranya akan mengalami dampak dari adanya sistem eCOLL-Letters ataupun eCOLL-Request Form nantinya.

Topik utama yang menjadi pembahasan pada setiap pertemuan yang ada tentunya seputar proyek yang dikerjakan oleh tim System Development & Infrastructure Associate, seputar kritik maupun saran terhadap proyek. Namun selain itu, tidak menutup kemungkinan juga untuk tim menyampaikan kendala-kendala yang dialami selama menjadi peserta magang pada divisi Consumer Collection & Recovery yang melibatkan pihak perusahaan ataupun kampus.

3.2.7 Minggu 17-18: Mempersiapkan Final Presentation

Pada minggu ke-17 hingga ke-18, tim telah ditugaskan untuk mempersiapkan final presentation sebagai laporan terakhir yang dilakukan oleh tim mengenai proyek yang selama ini dilaksanakan semasa program magang berlangsung pada PT Bank XYZ. Dengan ini, tim harus kerap melakukan pertemuan untuk menyusun presentasi final agar nantinya dapat tersampaikan dengan jelas dan baik. Sehingga dapat memberi dampak positif guna keberlangsungan sistem kinerja divisi Consumer Collection & Recovery pada PT Bank XYZ.

3.2.8 Minggu 18: Final Presentation

Pada minggu ke-18, kegiatan terakhir yang akan dilakukan oleh tim System Development & Infrastructure Associate adalah final presentation mengenai proyek yang selama ini telah dilakukan oleh tim saat berlangsungnya masa magang pada divisi Consumer Collection & Recovery. Dengan ini, materi yang dipresentasikan akan dalam bentuk PowerPoint dan dilanjutkan dengan demo sistem akhir yang

telah dirancang oleh tim untuk proyek eCOLL-Letters, eCOLL-Request Form, dan Proyek EUC RnR CC untuk sistem kinerja internal PT Bank XYZ.

3.3 Kendala yang Ditemukan

Selama berlangsungnya program magang yang diikuti oleh penulis sebagai anggota tim System Development & Infrastructure Associate pada divisi Consumer Collection & Recovery, tentunya terdapat beberapa kendala yang ditemukan selama prosesnya. Berikut merupakan beberapa kendala yang telah dilewati oleh penulis bersama dengan tim System Development & Infrastructure Associate di PT Bank XYZ.

1. Adanya kendala dalam mengakses tools yang digunakan oleh perusahaan sebagai standar proses kerja sehari-harinya seperti sarana Microsoft Teams, yang menjadi alat komunikasi utama selain menggunakan sistem BlackBerry Access.
2. Kendala terhadap list standar aplikasi yang sudah terdaftar pada sistem operasional PT Bank XYZ yang tidak menjadi pertimbangan utama sebelum melaksanakan proyek yang diberikan.
3. Pengetahuan dan pemahaman yang kurang mengenai bahasa pemrograman yang digunakan.
4. Pengetahuan mengenai beberapa istilah yang digunakan oleh divisi Consumer Collection & Recovery terkadang masih terdengar asing di telinga penulis sebagai peserta magang.

3.4 Solusi atas Kendala yang Ditemukan

Walaupun terdapat beberapa kendala dalam berlangsungnya program magang penulis sebagai anggota tim System Development & Infrastructure pada PT Bank XYZ, berikut merupakan beberapa solusi adaptif yang diimplementasikan oleh penulis dalam menyelesaikan kendala-kendala tersebut.

1. Untuk kendala akses Microsoft Teams, tim menggunakan Microsoft Teams pribadi yang tidak terhubung dengan alamat domain perusahaan, sehingga walaupun dapat bekerja dengan baik namun masih memiliki akses yang terbatas dalam penggunaannya.
2. Melakukan pengajuan tiket untuk tools yang diperlukan dalam keberlangsungan proyek pada sistem internal PT Bank XYZ dan mempelajari bahasa pemrograman alternatif sebagai rencana cadangan.
3. Untuk meningkatkan pengetahuan dan pemahaman mengenai bahasa pemrograman yang digunakan, penulis sebagai peserta magang harus melakukan riset dan belajar lebih untuk dapat merancang proyek hingga mencapai hasil akhir dan dapat beroperasi dengan baik.
4. Memastikan atau menanyakan ulang mengenai apa yang telah disampaikan sehingga tidak terjadinya miskomunikasi.