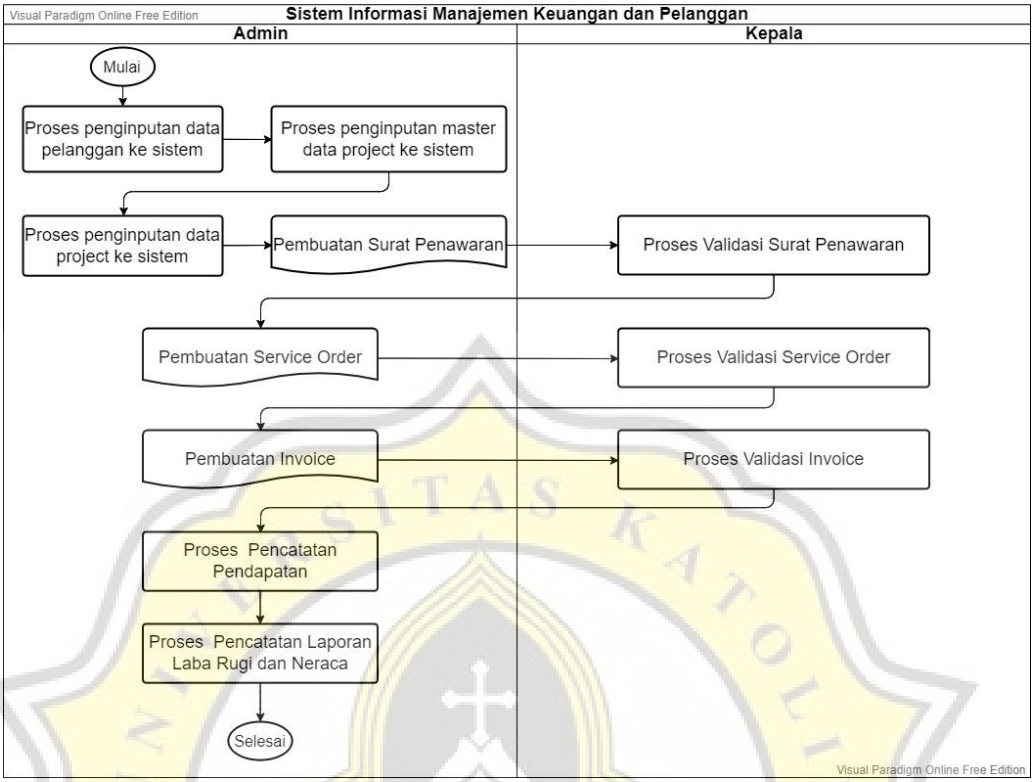
BAB IV
PEMBAHASAN

4.1 Proses Bisnis

Sistem informasi manajemen keuangan dan pelanggan ini dibuat dengan maksud untuk menunjang kinerja CCPA dan P3A di beberapa proses bisnisnya, yakni dalam proses pencatatan dan pengolahan data pelanggan, data project serta data dokumen keuangan dan juga proses pencatatan dan pengolahan data keuangan. Proses pencatatan dan pengolahan data pelanggan dimulai dari penginputan data pelanggan ke sistem, penghubungan data konsumen dengan data project, dan akan diakhiri dengan penghubungan data konsumen dengan data keuangan yang terkait (seperti piutang). Alur proses pencatatan dan pengolahan data project dan data dokumen akan dimulai ketika pengguna menginputkan master data ke sistem, penginputan detail data project ke sistem, pembuatan dokumen surat penawaran, validasi dokumen surat penawaran, pembuatan dokumen service order, validasi dokumen service order, dan akan diakhir dengan pembuatan dokumen invoice serta validasi dokumen invoice. Sedangkan alur proses pencatatan dan pengolahan data keuangan akan dimulai dengan penginputan pendapatan, beban, maupun transaksi lainnya, proses pembuatan laporan laba rugi project, proses pembuatan laporan laba rugi, dan akan diakhiri dengan proses pembuatan laporan keuangan atau neraca. Gambaran alur besar sistem informasi manajemen keuangan dan pelanggan dapat dilihat melalui gambar di bawah ini.

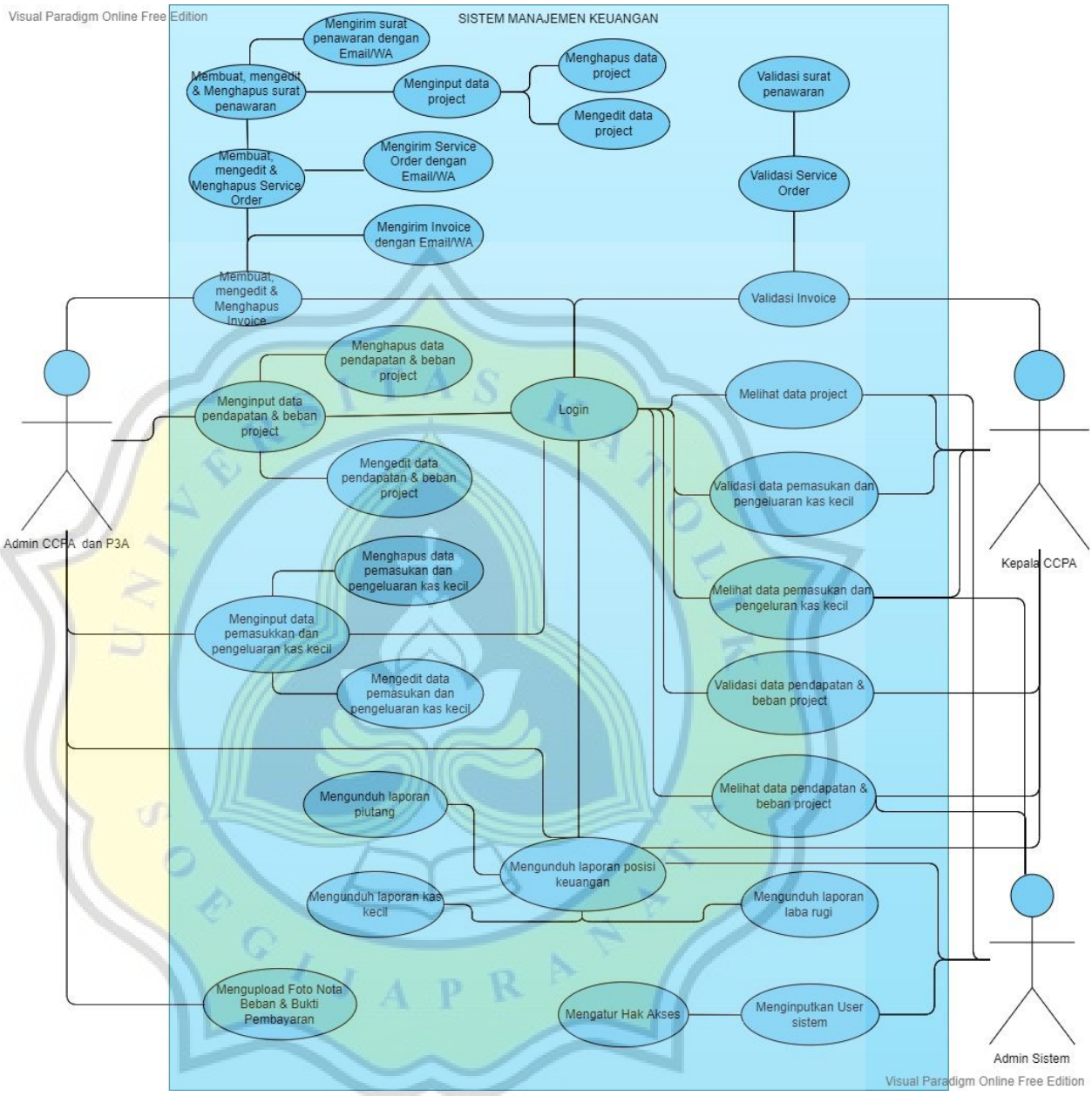


Gambar 4.1 Alur Sistem Informasi Keuangan dan Manajemen Pelanggan

4.2 Perancangan Sistem

4.2.1 Perancangan *Use Case Diagram*

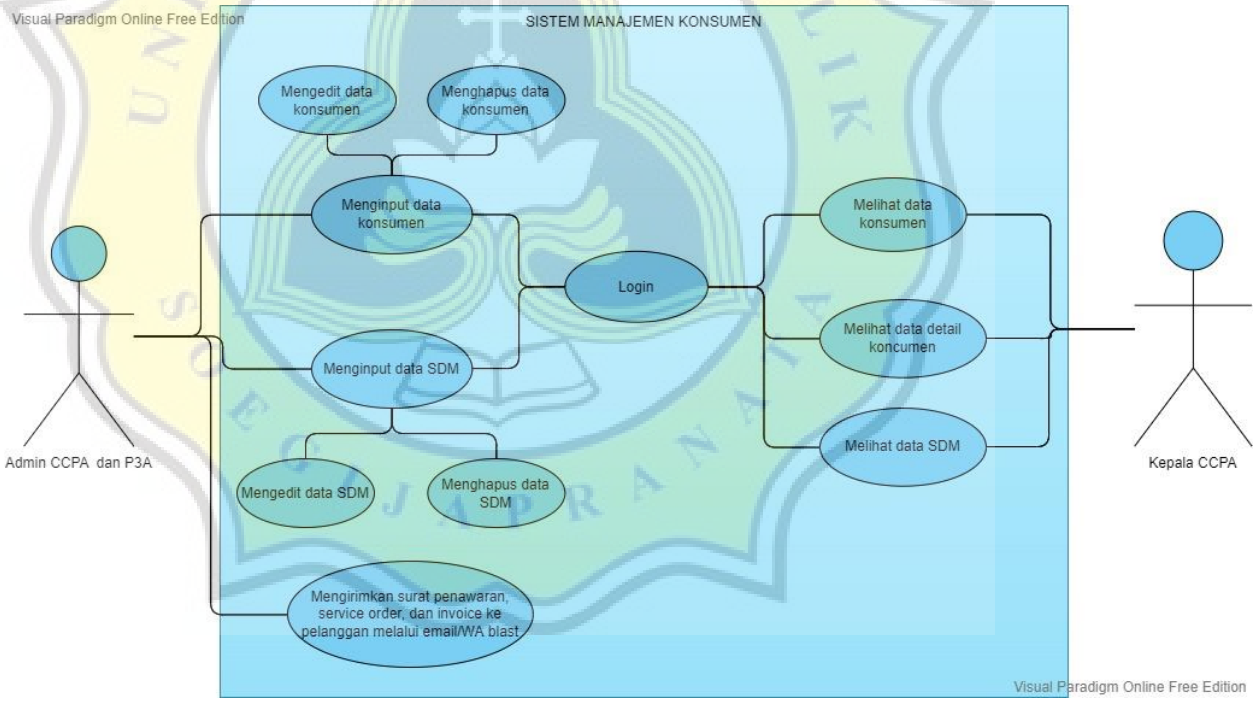
Sistem informasi manajemen keuangan dan pelanggan ini akan digunakan oleh tiga jenis pengguna/*user* yang berbeda yakni kepala CCPA & P3A, admin CCPA & P3A dan admin sistem. Masing-masing pengguna ini memiliki batasan fitur dan tanggung jawab yang berbeda-beda sesuai dengan hak akses yang sudah ditetapkan sebelumnya. Ketiga pengguna ini harus melakukan proses login sebelum dapat menggunakan sistem ini. Website yang dibuat ini merupakan contoh dari website yang bersifat statis dimana isi dari website ini tidak sering berubah.



Gambar 4.2 Use Case Sistem Informasi Keuangan

Melalui gambar diatas ini dapat kita lihat bahwa dalam sistem informasi manajemen keuangan, ketiga *user* ini memiliki batasan dan tanggungjawab yang berbeda-beda. Admin CCPA dalam sistem ini dapat menginput, mengubah serta menghapus data project, surat penawaran, *service order*, *invoice*, menginput dan mengubah data

pendapatan dan beban, menginput kas keluar dan masuk serta dapat mengunduh laporan laba rugi project, laporan laba rugi serta laporan keuangan/neraca. Kepala CCPA dalam sistem ini dapat melihat data project, memvalidasi surat penawaran, *service order*, *invoice*, melihat rincian serta memvalidasi data pendapatan, beban, kas serta piutang, dan mengunduh laporan laba rugi project, laporan laba rugi serta laporan keuangan/neraca. Sedangkan admin sistem dalam sistem ini dapat melihat data project, data dokumen (seperti surat penawaran, *service order, dan invoice*), mengunduh laporan laba rugi project, laporan laba rugi serta laporan keuangan/neraca dan admin sistem dapat mengatur hak akses dari masing-masing user lainnya.



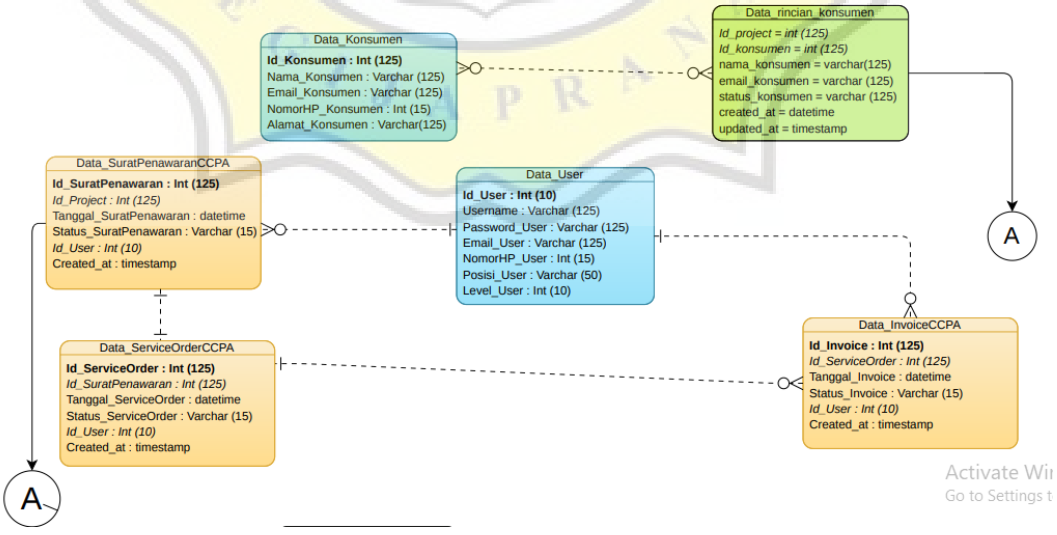
Gambar 4.3 Use Case Sistem Manajemen Pelanggan

Sama halnya dengan sistem informasi manajemen keuangan, sistem informasi manajemen pelanggan juga memiliki batasan fitur untuk masing-masing *user*nya. Dalam sistem ini Admin CCPA dapat menginputkan, mengubah serta menghapus data pelanggan

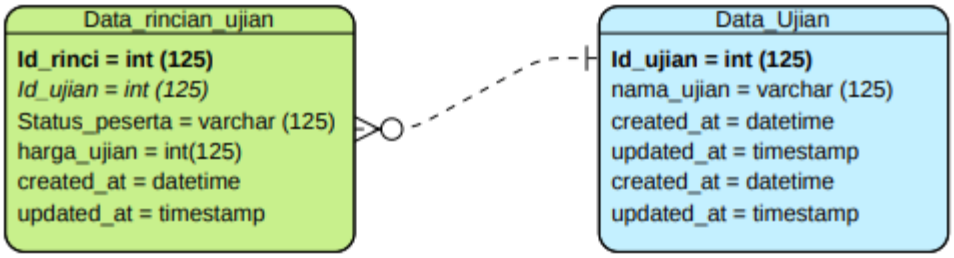
dan menginputkan, mengubah dan menghapus data SDM (dosen yang terlibat dalam project). Sementara kepala CCPA hanya dapat melihat data konsumen dan data SDM saja.

4.2.2 Perancangan *Entity Relationship Diagram*

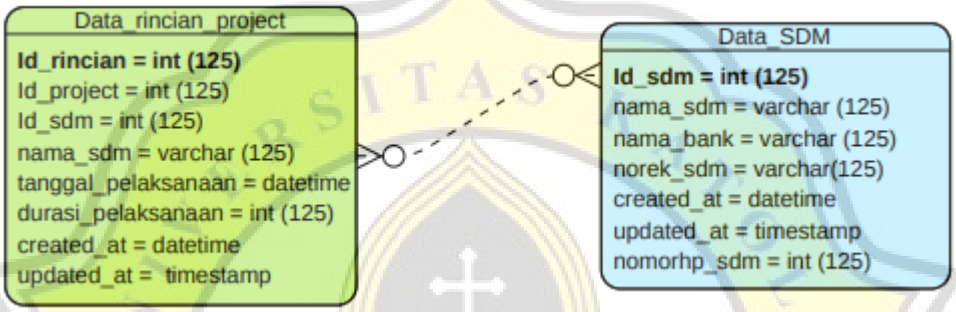
Sistem informasi manajemen keuangan dan pelanggan ini dirancang dengan menggunakan delapan belas tabel entitas. Delapan belas tabel entitas tersebut dibagi menjadi enam tabel entitas untuk master data, tiga tabel entitas untuk rincian data, tiga tabel entitas untuk dokumen dan enam tabel entitas untuk proses pencatatan dan pengolahan data keuangan. Tabel master data yang ada di sistem ini meliputi tabel entitas data konsumen, data project, data akun, data SDM, data user serta data ujian. Tabel rincian data yang digunakan di sistem ini meliputi tabel data rincian konsumen, tabel data rincian ujian dan tabel data rincian project. Sedangkan tabel-tabel yang digunakan untuk menyimpan data dokumen-dokumen, meliputi tabel data surat penawaran, tabel data *service order*, dan tabel data *invoice.* Dan tabel-tabel yang digunakan untuk mencatat dan menyimpan transaksi keuangan adalah tabel data bank, tabel beban, tabel kas, tabel pendapatan, tabel piutang dan tabel utang.



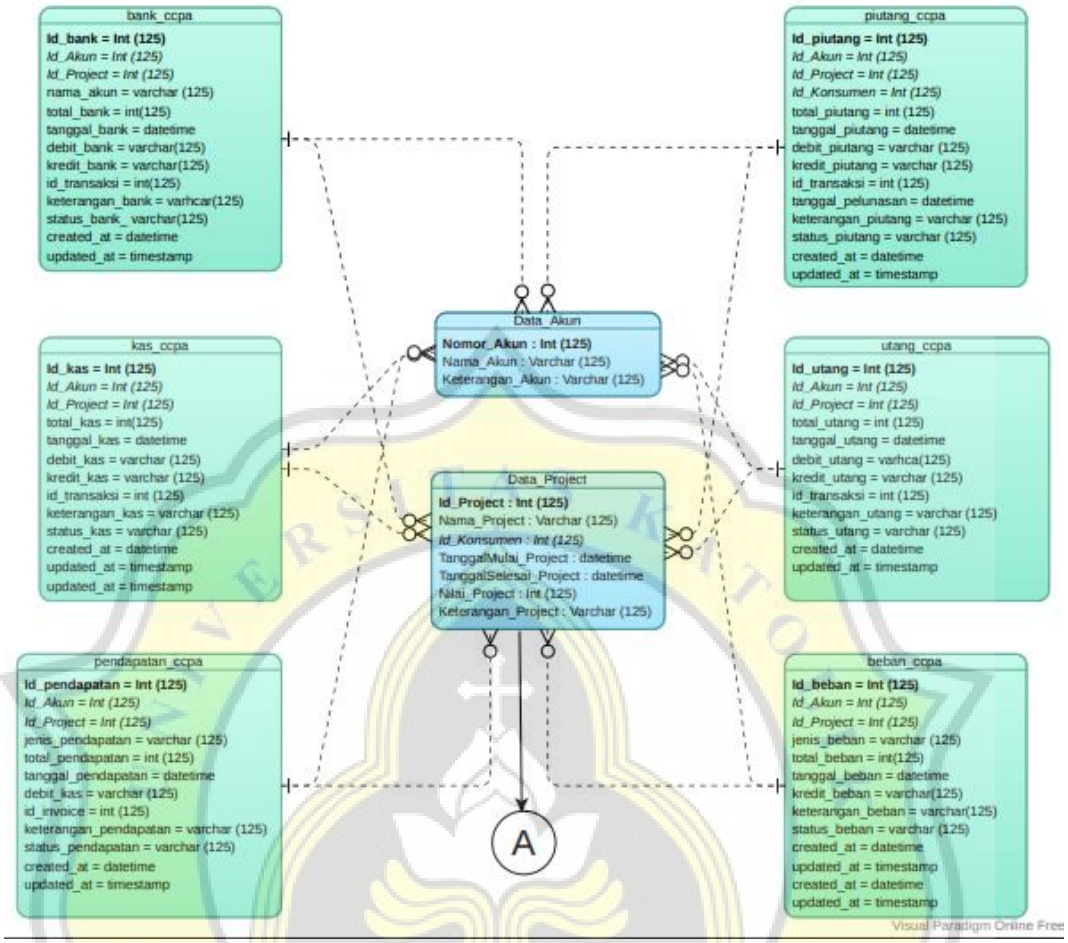
Gambar 4.4.a ERD Sistem Informasi Keuangan dan Manajemen Pelanggan



Gambar 4.4.b ERD Sistem Informasi Keuangan dan Manajemen Pelanggan



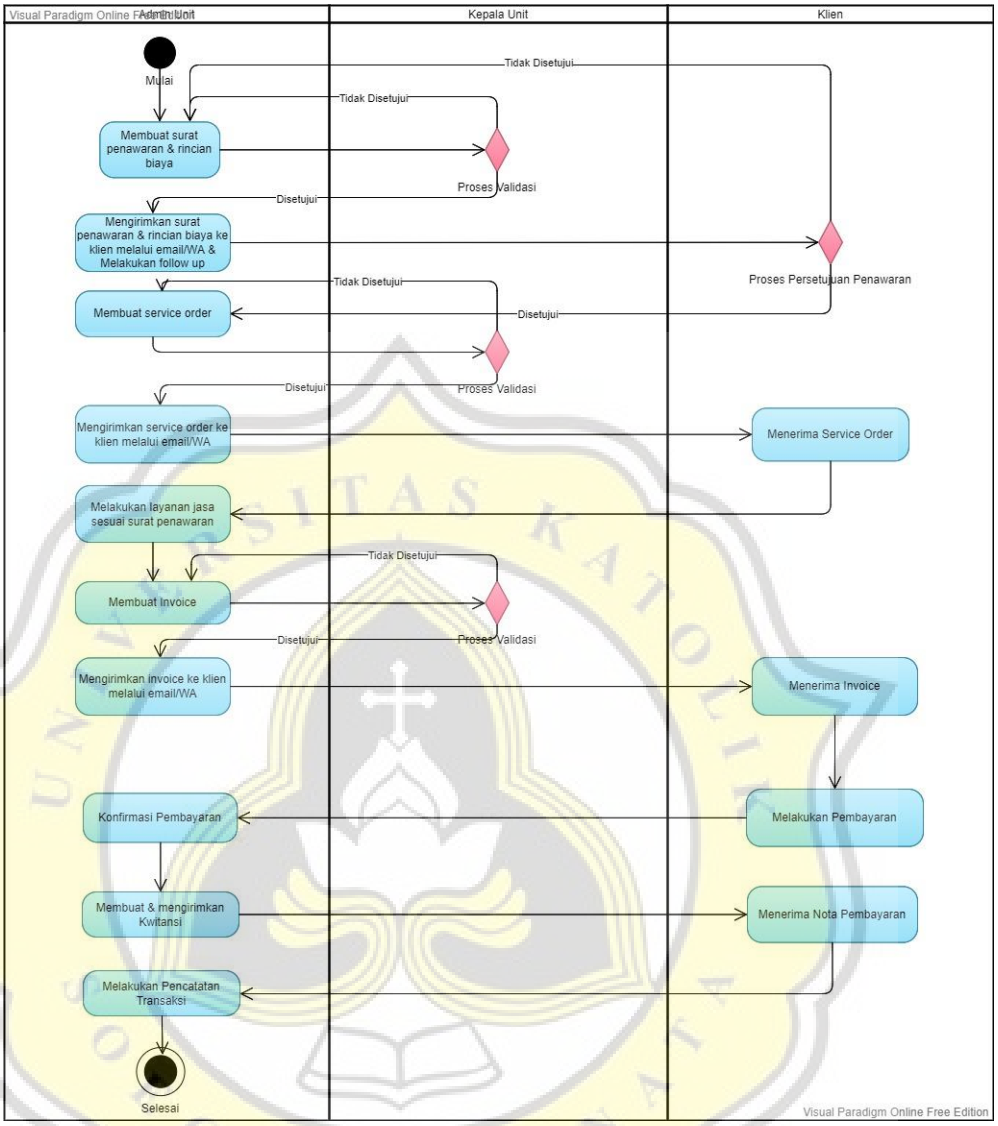
Gambar 4.4.c ERD Sistem Informasi Keuangan dan Manajemen Pelanggan



Gambar 4.4.d ERD Sistem Informasi Keuangan dan Manajemen Pelanggan

4.2.3 Perancangan *Activity Diagram*

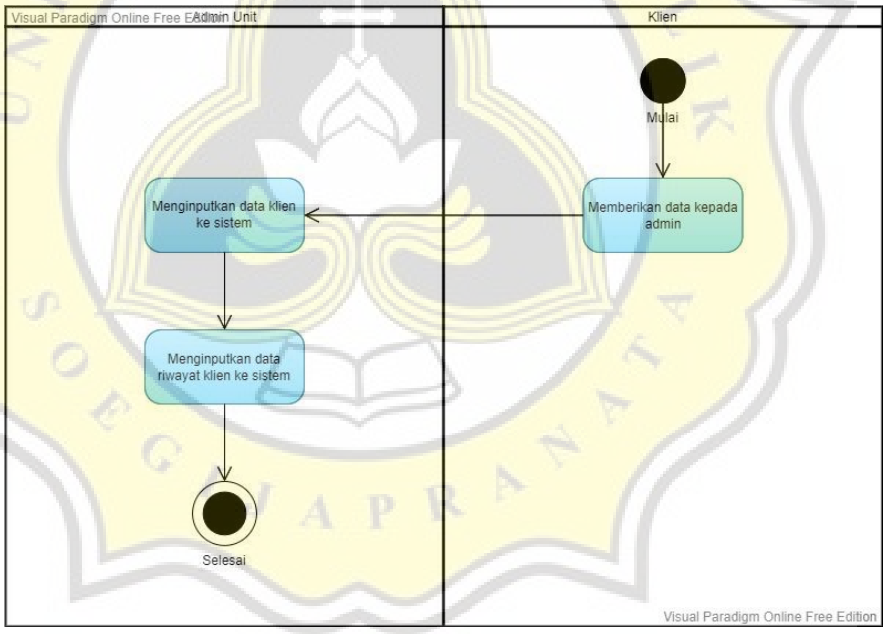
Sistem informasi manajemen keuangan dan pelanggan ini berjalan dengan didasari dua buah alur utama yakni alur untuk sistem informasi manajemen keuangan dan alur untuk sistem informasi manajemen pelanggan.



Gambar 4.5 Activity Diagram Sistem Informasi Keuangan

Gambar diatas ini merupakan gambar dari alur sistem informasi manajemen keuangan. Alur sistem ini dimulai ketika admin CCPA membuat surat penawaran dan rincian biaya untuk suatu project, selanjutnya surat penawaran tersebut akan divalidasi oleh kepala CCPA, apabila disetujui maka admin akan mengirimkan surat penawaran tersebut kepada pelanggan namun apabila tidak disetujui maka admin akan mengubah atau membuat surat penawaran yang baru. Kemudian setelah dikirim dan disetujui oleh pelanggan, maka admin CCPA akan membuat *service order* untuk project tersebut

dan setelah selesai membuat *service order*, admin CCPA akan menunggu validasi *service order* dari kepala CCPA. Bila telah disetujui, maka *service order* tersebut akan dikirimkan ke pihak pelanggan dan CCPA akan mulai melakukan layanan jasa sesuai dengan kontrak dan kesepakatan yang ada di *service order* dan surat penawaran. Setelah selesai melakukan layanan jasa, maka admin CCPA akan membuat *invoice* yang kemudian harus divalidasi oleh kepala CCPA sebelum dikirimkan ke pihak pelanggan. Dan apabila pelanggan telah melakukan pembayaran jasa, maka admin CCPA akan melakukan pencatatan pendapatan serta beban-beban yang dikeluarkan selama pelaksanaan project tersebut serta mengirimkan kwitansi pembayaran ke pihak pelanggan.



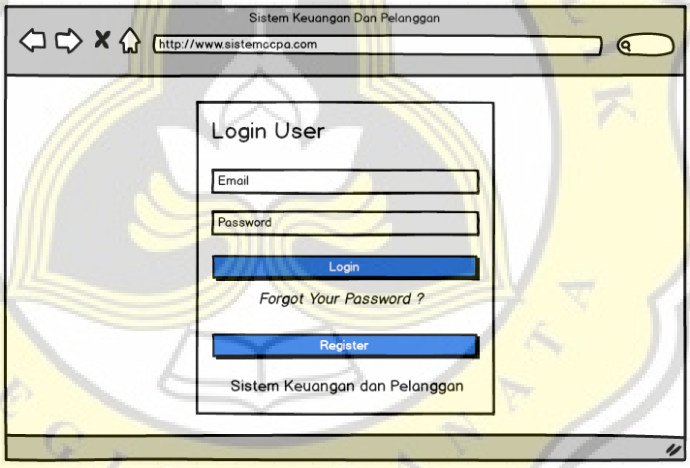
Gambar 4.6 Activity Diagram Sistem Manajemen Pelanggan

Gambar diatas merupakan alur dari sistem informasi manajemen pelanggan. Sistem ini akan dimulai ketika pelanggan memberikan datanya, yang kemudian akan diinputkan oleh admin CCPA ke dalam sistem serta menyimpan data pelanggan tersebut di dalam sistem.

4.2.4 Perancangan Desain Tampilan Depan (*User Interface)*

a. Desain Tampilan Depan Halaman *Login* dan *Register*

Gambar di bawah ini merupakan rancangan desain dari halaman *login* yang terdapat dalam sistem ini. Halaman ini merupakan halaman awal yang harus diisi oleh user sebelum dapat menggunakan sistem. Di halaman *login* ini, user diharuskan untuk mengisikan *username* dan *password*, dan apabila *username* dan *passwordnya* sesuai dengan data yang ada, maka user akan diarahkan ke halaman *dashboard* sesuai dengan hak aksesnya. Namun apabila *username* dan *password* tidak sesuai dengan data yang ada, maka akan muncul notifikasi dan *user* diharuskan untuk mengisi ulang *username* dan *password*nya.



Gambar 4.7 Desain Tampilan Login

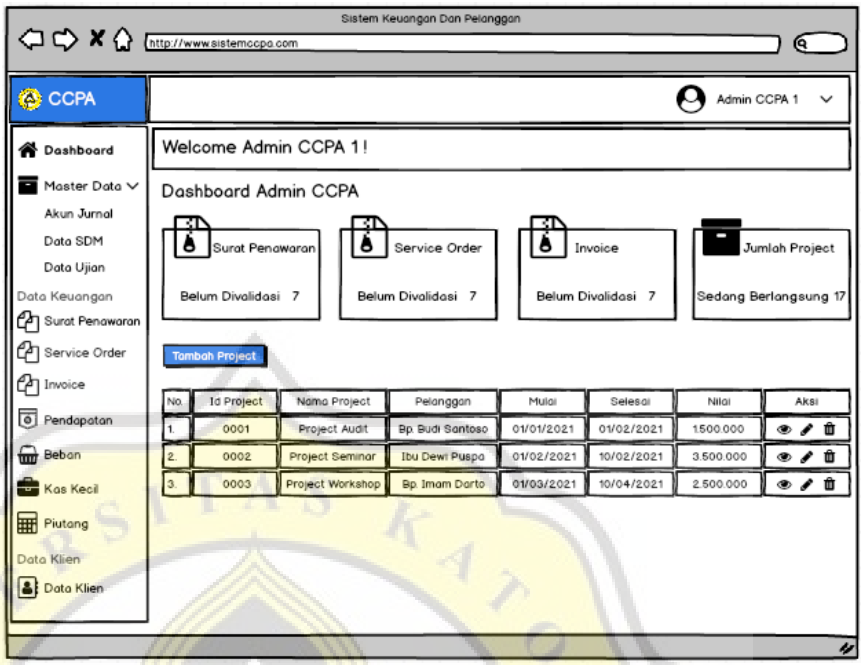
Gambar di bawah ini merupakan rancangan desain dari halaman *register* yang ada di sistem ini. Halaman *register* ini berguna untuk membuat akun baru bagi *user* yang belum memiliki akun. Di halaman ini, *user* diharuskan untuk menginputkan *username, password,* dan *email.* Lalu setelah itu klik tombol *register* dan akun baru akan otomatis tercatat dan tersimpan dalam *database*.

Gambar 4.8 Desain Tampilan Register

b. Desain Tampilan Depan Untuk Admin

1. Desain Halaman *Dashboard* Admin

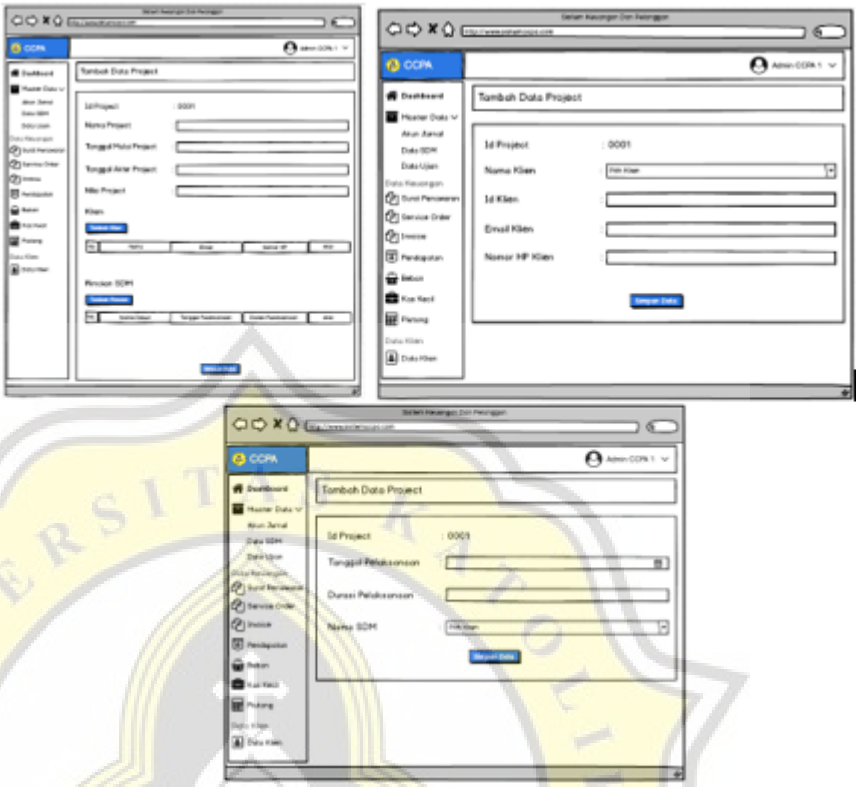
Gambar di bawah ini merupakan rancangan desain dari halaman *dashboard* untuk user dengan hak akses sebagai admin CCPA. Di halaman ini terdapat beberapa informasi, di sebelah kiri terdapat *navbar* yang ditujukan sebagai penunjuk untuk halaman lainnya, di bagian tengah pada halaman ini terdapat informasi tentang jumlah surat penawaran, *service order*, dan *invoice* yang belum divalidasi. Selain itu di bagian tengah pada halaman ini juga terdapat informasi tentan jumlah project yang masih berlangsung dan tabel yang berisi daftar project yang sedang berlangsung maupun yang sudah selesai. Pada tabel daftar project di baris paling kanan dari masing-masing project terdapat tiga aksi yakni lihat yang disimbolkan dengan ikon mata, edit yang disimbolkan dengan ikon pensil dan hapus yang disimbolkan dengan ikon sampah. Di halaman ini pula user dapat menginputkan project baru dengan cara menekan tombol tambah project.



Gambar 4.9 Desain Dashboard Admin

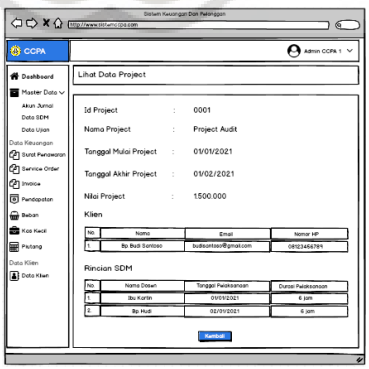
2. Desain Halaman Project

Gambar di bawah ini merupakan rancangan desain dari halaman tambah project yang akan muncul setelah user menekan tombol tambah project yang berada di halaman *dashboard*. Halaman tambah project ini terbagi menjadi tiga bagian. Di bagian pertama, user diharuskan untuk menginputkan nama project, tanggal mulai dan tanggal selesai project, serta nilai project. Di bagian yang kedua, user diharuskan untuk menginputkan data pelanggan yang terlibat dalam project tersebut. Dan di bagian ketiga, user diharuskan untuk menginputkan data SDM yang terlibat dalam pelaksanaan project tersebut.



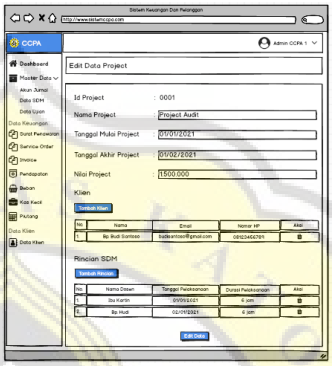
Gambar 4.10 Desain Halaman Tambah Project

Gambar di bawah ini adalah rancangan desain halaman lihat project. Melalui halaman ini user dapat melihat data detail dari suatu project. Mulai dari Id project, nama project, tanggal mulai dan tanggal selesai project, nilai project, data pelanggan yang terlibat dan data SDM yang terlibat dalam project tersebut. Halaman ini akan muncul setelah user menekan ikon mata yang terdapat di tabel daftar project pada halaman *dashboard*.



Gambar 4.11 Desain Halaman Lihat Project

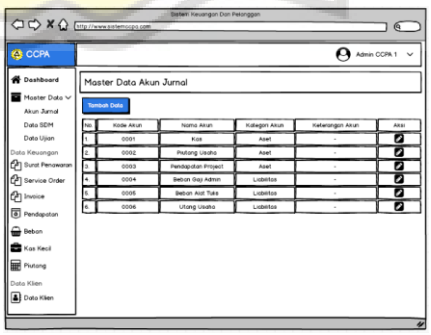
Gambar di bawah ini adalah rancangan desain dari halaman edit project. Melalui halaman inilah user dapat mengubah data project apabila terjadi kesalahan input. Halaman ini akan muncul ketika user menekan ikon pensil yang ada di tabel daftar project pada halaman *dashboard*.



Gambar 4.12 Desain Halaman Edit Project

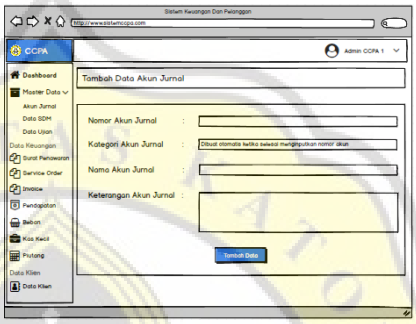
3. Desain Halaman Master Data Akun

Gambar di bawah ini menggambarkan rancangan desain halaman awal untuk master data akun. Di halaman ini user dapat melihat tabel yang berisi daftar akun yang sudah ada sebelumnya lengkap dengan kode akunnya. Di halaman ini pula user dapat menginputkan akun baru dengan cara menekan tombol tambah akun. Dan user dapat mengubah data dari suatu akun yang sudah ada dengan cara menekan ikon pensil yang ada di masing-masing baris akun yang akan diedit.



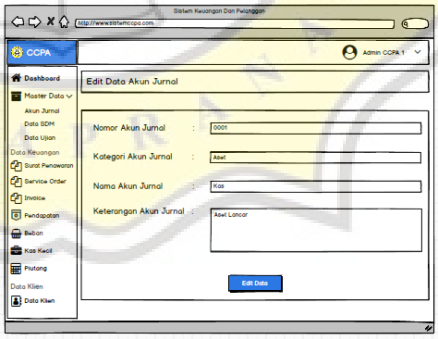
Gambar 4.13 Desain Halaman Master Data Akun

Gambar di bawah ini adalah rancangan desain dari halaman tambah akun. Di halaman ini user dapat menginputkan dan lalu menyimpan data akun baru. Di halaman ini, user diharuskan untuk menginputkan kode, nama, dan keterangan akun. Halaman ini akan muncul setelah user menekan tombol tambah akun yang terdapat pada halaman awal master data akun.



Gambar 4.14 Desain Halaman Tambah Master Data Akun

Gambar di bawah ini adalah rancangan desain untuk halaman edit akun,. Di halaman ini user dapat mengubah data dari suatu akun yang sudah ada sebelumnya. Halaman ini akan muncul setelah user menekan ikon pensil yang ada di tabel daftar akun pada halaman awal master data akun.

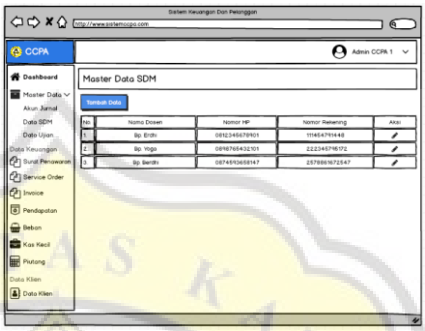


Gambar 4.15 Desain Halaman Edit Master Data Akun

4. Desain Halaman Master Data SDM

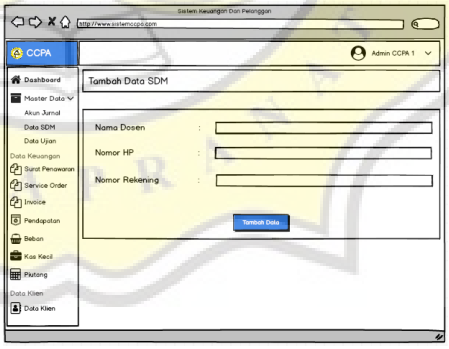
Gambar di bawah ini merupakan rancangan desain dari halaman awal untuk master data SDM. Di halaman ini user dapat melihat tabel yang berisikan data SDM yang sudah ada sebelumnya.

Dan di halaman ini pula user dapat menekan tombol tambah data SDM yang bertujuan untuk menambah data SDM baru maupun menekan ikon pensil yang ada di dalam tabel untuk mengedit data SDM yang sudah ada.



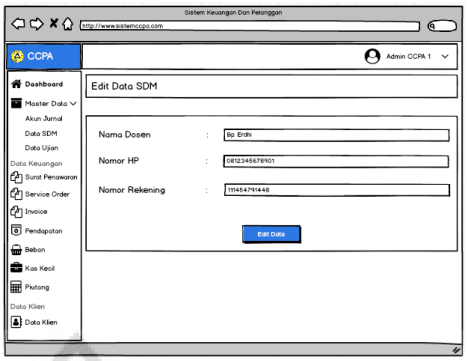
Gambar 4.16 Desain Halaman Master Data SDM

Gambar di bawah ini adalah rancangan desain dari halaman tambah sdm. Di halaman ini user dapat menginputkan dan lalu menyimpan data SDM baru. Di halaman ini, user diharuskan untuk menginputkan nama dosen, nomor hp dan nomor rekening. Halaman ini akan muncul setelah user menekan tombol tambah data sdm yang terdapat pada halaman awal master data sdm.



Gambar 4.17 Desain Halaman Tambah Master Data SDM

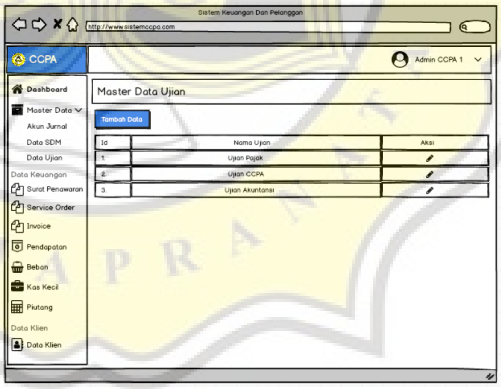
Gambar di bawah ini adalah rancangan desain untuk halaman edit SDM. Di halaman ini user dapat mengubah data dari suatu SDM yang sudah ada sebelumnya. Halaman ini akan muncul setelah user menekan ikon pensil yang ada di tabel daftar akun pada halaman awal master data SDM.



Gambar 4.18 Desain Halaman Edit Master Data SDM

5. Desain Halaman Master Data Ujian

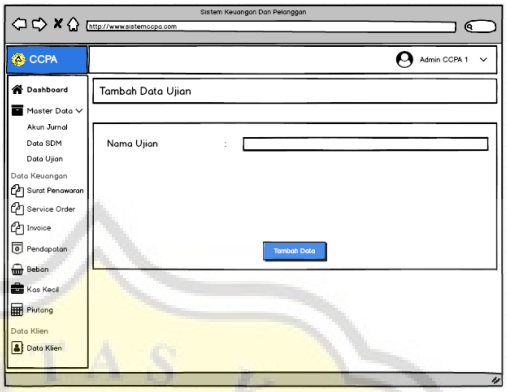
Gambar di bawah ini adalah rancangan desain untuk halaman awal master data ujian. Di halaman ini terdapat informasi berupa tabel yang berisi nama project ujian yang sudah diinputkan sebelumnya. Di halaman ini user dapat mulai menginputkan data ujian baru dengan menekan tombol tambah data ujian serta dapat mengedit data yang sudah ada dengan cara menekan ikon pensil yang ada di tabel daftar data ujian.



Gambar 4.19 Desain Halaman Master Data Ujian

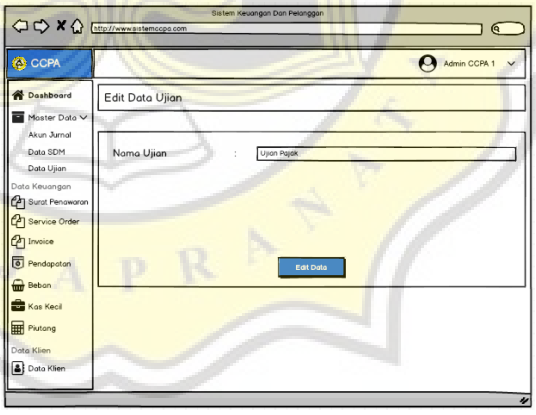
Gambar di bawah ini adalah rancangan desain dari halaman tambah ujian. Di halaman ini user dapat menginputkan dan lalu menyimpan data ujian baru. Di halaman ini, user diharuskan untuk menginputkan nama ujian dan id dari data ujian akan otomatis dibuatkan oleh sistem. Halaman ini akan muncul setelah user

menekan tombol tambah data ujian yang terdapat pada halaman awal master data ujian.



Gambar 4.20 Desain Halaman Tambah Master Data Ujian

Gambar di bawah ini adalah rancangan desain untuk halaman edit ujian,. Di halaman ini user dapat mengubah data dari suatu ujian yang sudah ada sebelumnya. Halaman ini akan muncul setelah user menekan ikon pensil yang ada di tabel daftar akun pada halaman awal master data ujian.

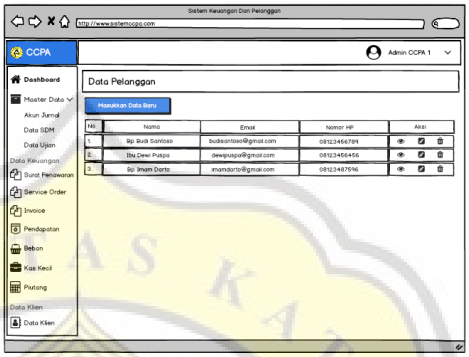


Gambar 4.21 Desain Halaman Edit Master Data Ujian

6. Desain Halaman Master Data Pelanggan

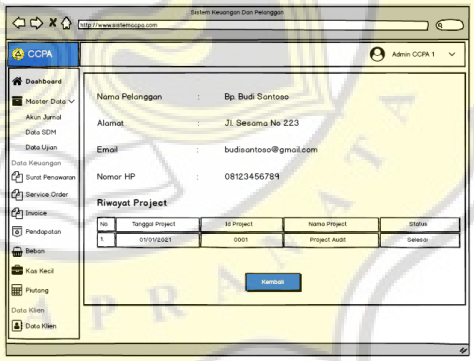
Gambar di bawah ini merupakan rancangan desain untuk halaman awal dari master data pelanggan. Di halaman ini user dapat melihat sebuah tabel yang berisikan data pelanggan yang sudah diinputkan sebelumnya. Melalui halaman ini pula, user dapat mulai

menginputkan data pelanggan baru dengan cara menekan tombol tambah data baru dan melihat rincian data pelanggan yang sudah ada dengan cara menekan ikon mata, serta mengedit data yang sudah ada sebelumnya dengan menekan ikon pensil yang ada di dalam tabel.



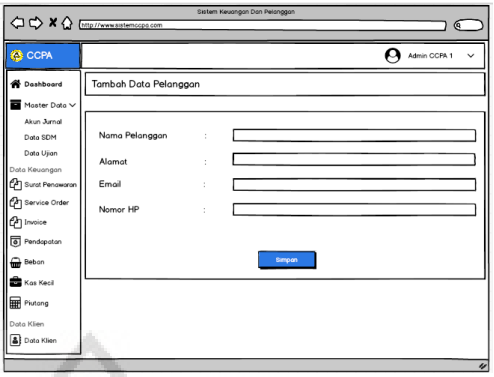
Gambar 4.22 Desain Halaman Master Data Pelanggan

Gambar di bawah ini merupakan rancangan desain untuk halaman lihat rincian data pelanggan. Melalui halaman ini user dapat melihat data rinci dari seorang pelanggan.



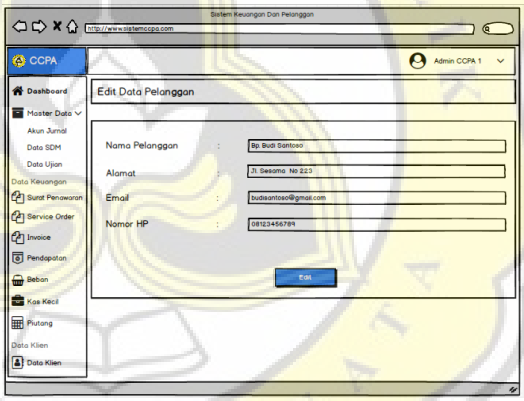
Gambar 4.23 Desain Halaman Lihat Master Data Pelanggan

Gambar di bawah ini merupakan rancangan desain untuk halaman tambah data pelanggan. Melalui halaman inilah user menginputkan menyimpan data pelanggan baru. Di halaman ini user diharukan untuk menginputkan nama pelanggan, alamat pelanggan, email pelanggan dan nomor hp pelanggan.



Gambar 4.24 Desain Halaman Tambah Master Data Pelanggan

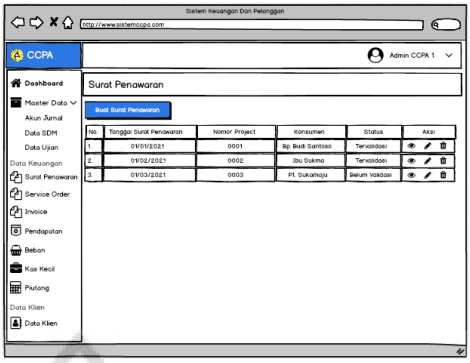
Gambar di bawah ini merupakan rancangan desain untuk halaman edit data pelanggan. Di halaman ini, user dapat mengubah data pelanggan yang sudah diinputkan sebelumnya apabila terjadi salah input.



Gambar 4.25 Desain Halaman Edit Master Data Pelanggan

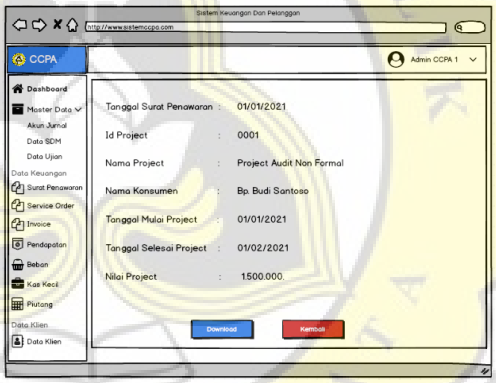
7. Desain Halaman Surat Penawaran

Gambar di bawah ini merupakan rancangan desain untuk halaman awal surat penawaran. Di halaman ini user dapat melihat tabel yang berisikan data surat penawaran yang sudah dibuat sebelumnya. Di halaman ini pula user dapat mulai membuat surat penawaran baru, melihat, mengedit, dan menghapus data surat penawaran yang sudah dibuat sebelumnya.



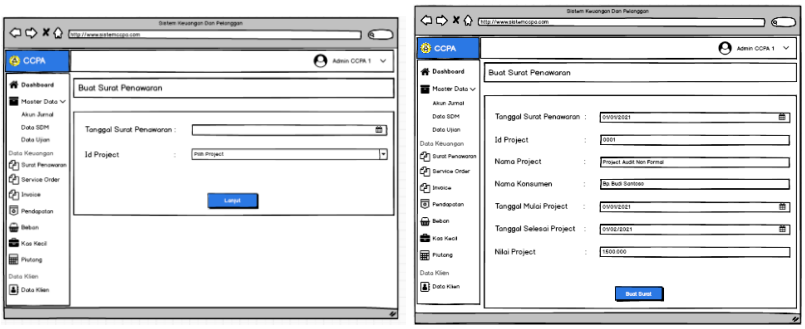
Gambar 4.26 Desain Halaman Surat Penawaran

Gambar di bawah ini merupakan rancangan desain untuk halaman lihat data surat penawaran. Di halaman ini user dapat melihat data rinci dari suatu surat penawaran yang sudah dibuat sebelumnya.



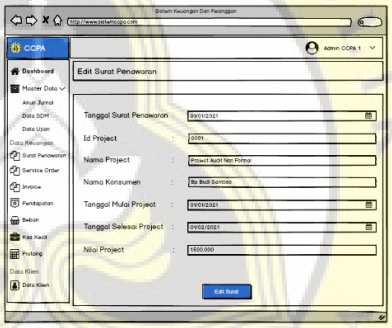
Gambar 4.27 Desain Halaman Lihat Surat Penawaran

Gambar di bawah ini merupakan rancangan desain untuk halaman tambah surat penawaran baru. Proses pembuatan surat penawaran baru terbagi menjadi dua bagian, di bagian pertama user diharuskan untuk menginputkan tanggal surat penawaran. Sedangkan di bagian kedua, user diharuskan untuk memilih id dari project yang akan dibuatkan surat penawarannya. Halaman buat surat penawaran baru akan muncul setelah user menekan tombol buat surat penawaran yang ada pada halaman awal surat penawaran.



Gambar 4.28 Desain Halaman Tambah Surat Penawaran

Gambar di bawah ini merupakan rancangan desain untuk halaman edit data surat penawaran. Di halaman ini, user dapat mengubah data surat penawaran yang sudah diinputkan sebelumnya apabila terjadi salah input.



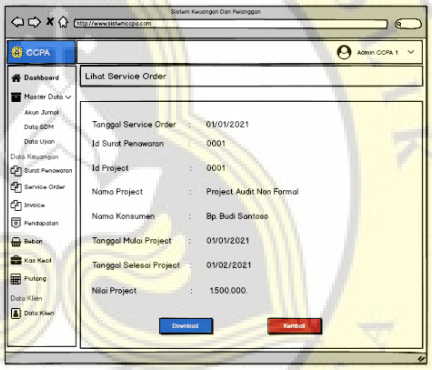
Gambar 4.29 Desain Halaman Edit Surat Penawaran

8. Desain Halaman *Service Order*

Gambar di bawah ini merupakan rancangan desain untuk halaman awal *service order*. Di halaman ini user dapat melihat tabel yang berisikan data *service order* yang sudah dibuat sebelumnya. Di halaman ini pula user dapat mulai membuat *service order* baru, melihat, mengedit, dan menghapus data *service order* yang sudah dibuat sebelumnya.

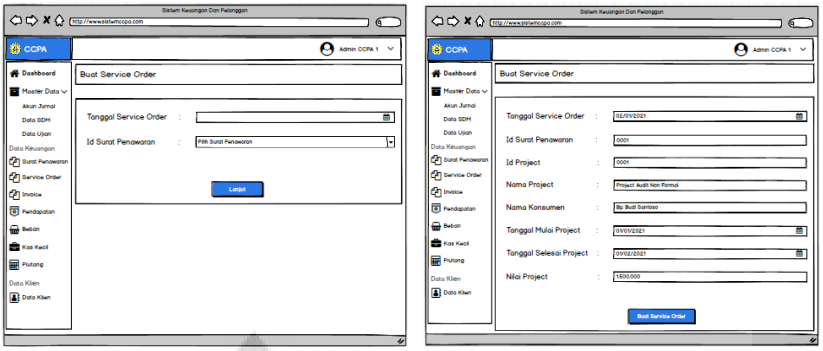
Gambar 4.30 Desain Halaman Service Order

Gambar di bawah ini merupakan rancangan desain untuk halaman lihat data *service order*. Di halaman ini user dapat melihat data rinci dari suatu *service order* yang sudah dibuat sebelumnya.



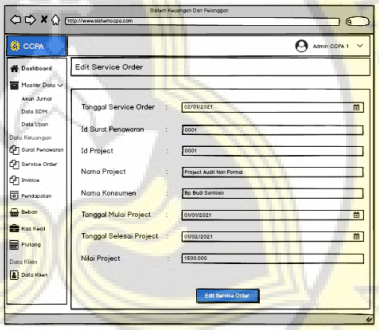
Gambar 4.31 Desain Halaman Lihat Service Order

Gambar di bawah ini merupakan rancangan desain untuk halaman tambah *service order* baru. Proses pembuatan surat penawaran baru terbagi menjadi dua bagian, di bagian pertama user diharuskan untuk menginputkan tanggal *service order*. Sedangkan di bagian kedua, user diharuskan untuk memilih id dari surat penawaran yang akan dibuatkan *service order*. Dan nantinya Id dari *service order* yang baru akan otomatis dibuatkan oleh sistem. Halaman buat *service order* baru akan muncul setelah user menekan tombol buat *service order* yang ada pada halaman awal *service order*.



Gambar 4.32 Desain Halaman Tambah Service Order

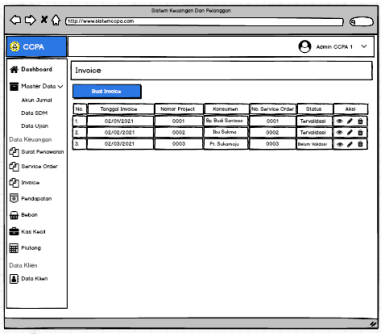
Gambar di bawah ini merupakan rancangan desain untuk halaman edit data *service order*. Di halaman ini, user dapat mengubah data *service order* yang sudah diinputkan sebelumnya apabila terjadi salah input.



Gambar 4.33 Desain Halaman Edit Service Order

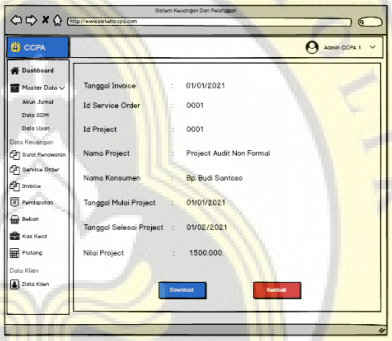
9. Desain Halaman *Invoice*

Gambar di bawah ini merupakan rancangan desain untuk halaman awal *invoice*. Di halaman ini user dapat melihat tabel yang berisikan data *invoice* yang sudah dibuat sebelumnya. Di halaman ini pula user dapat mulai membuat *invoice* baru, melihat, mengedit, dan menghapus data *invoice* yang sudah dibuat sebelumnya.



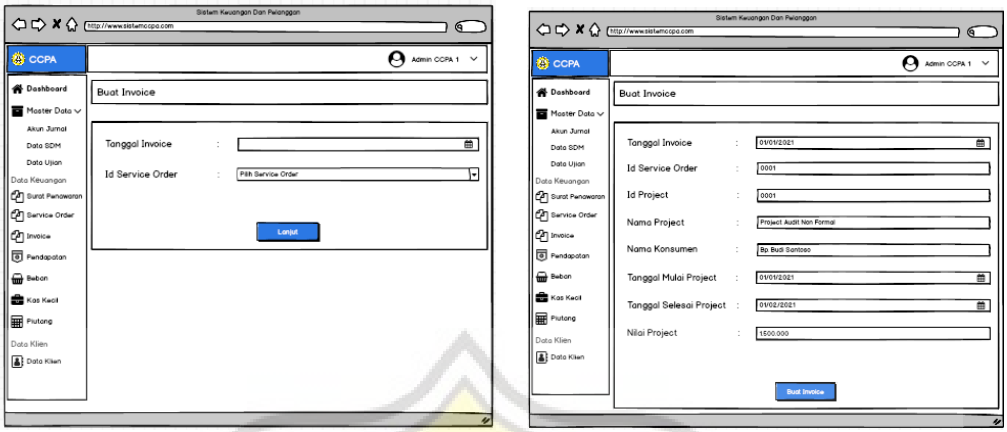
Gambar 4.34 Desain Halaman Invoice

Gambar di bawah ini merupakan rancangan desain untuk halaman lihat data *invoice*. Di halaman ini user dapat melihat data rinci dari suatu *invoice* yang sudah dibuat sebelumnya.



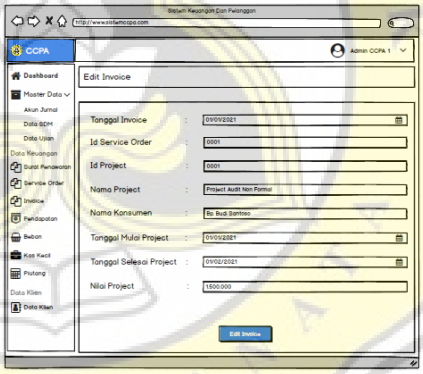
Gambar 4.35 Desain Halaman Lihat Invoice

Gambar di bawah ini merupakan rancangan desain untuk halaman tambah *invoice* baru. Proses pembuatan surat penawaran baru terbagi menjadi dua bagian, di bagian pertama user diharuskan untuk menginputkan tanggal *invoice*. Sedangkan di bagian kedua, user diharuskan untuk memilih id dari *service order* yang akan dibuatkan *invoice*. Dan nantinya Id dari *invoice* yang baru akan otomatis dibuatkan oleh sistem. Halaman buat *invoice* baru akan muncul setelah user menekan tombol buat *invoice* yang ada pada halaman awal *invoice.*



Gambar 4.36 Desain Halaman Tambah Invoice

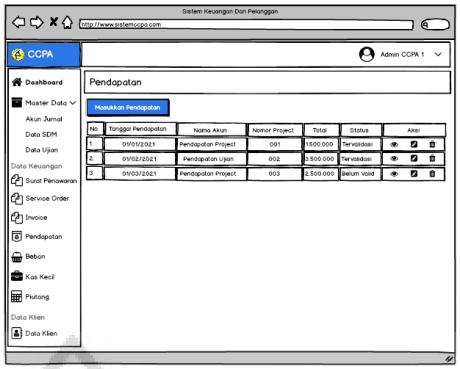
Gambar di bawah ini merupakan rancangan desain untuk halaman edit data *invoice*. Di halaman ini, user dapat mengubah data *invoice* yang sudah diinputkan sebelumnya apabila terjadi salah input.



Gambar 4.37 Desain Halaman Edit Invoice

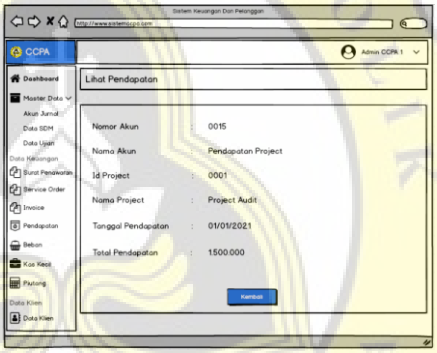
10. Desain Halaman Pendapatan

Gambar di bawah ini merupakan rancangan desain untuk halaman awal pendapatan. Di halaman ini user dapat melihat tabel yang berisikan data pendapatan yang sudah dibuat sebelumnya. Di halaman ini pula user dapat mulai membuat data pendapatan baru, melihat, mengedit, dan menghapus data pendapatan yang sudah dibuat sebelumnya.



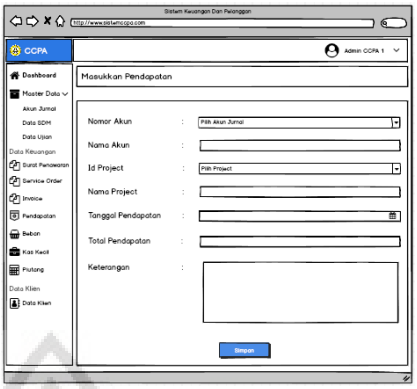
Gambar 4.38 Desain Halaman Pendapatan

Gambar di bawah ini merupakan rancangan desain untuk halaman lihat data pendapatan. Di halaman ini user dapat melihat data rinci dari suatu pendapatan yang sudah dibuat sebelumnya.



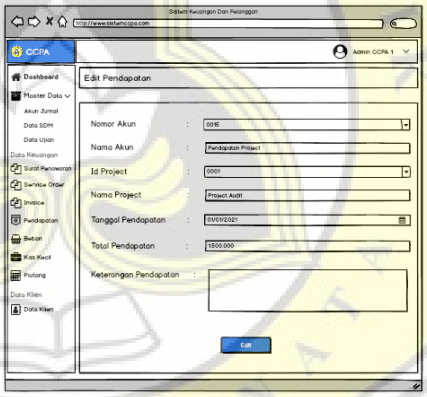
Gambar 4.39 Desain Halaman Lihat Pendapatan

Gambar di bawah ini merupakan rancangan desain untuk halaman tambah data pendapatan baru. Melalui halaman inilah user dapat menginputkan dan menyimpan data pendapatan baru. Di halaman ini user diharuskan untuk mengisi kode akun tujuan, memilih id project serta mengisi keterangan pendapatan bila ada.



Gambar 4.40 Desain Halaman Tambah Pendapatan

Gambar di bawah ini merupakan rancangan desain untuk halaman edit data pendapatan. Di halaman ini, user dapat mengubah data pendapatan yang sudah diinputkan sebelumnya apabila terjadi salah input.



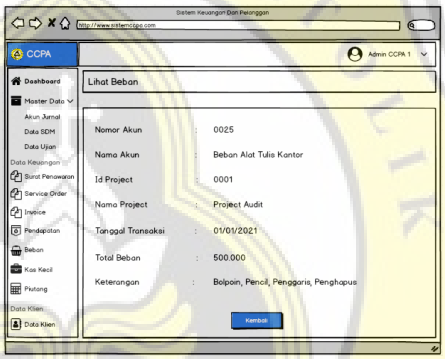
Gambar 4.41 Desain Halaman Edit Pendapatan

11. Desain Halaman Beban

Gambar di bawah ini merupakan rancangan desain untuk halaman awal beban. Di halaman ini user dapat melihat tabel yang berisikan data beban yang sudah dibuat sebelumnya. Di halaman ini pula user dapat mulai membuat data beban baru, melihat, mengedit, dan menghapus data beban yang sudah dibuat sebelumnya.

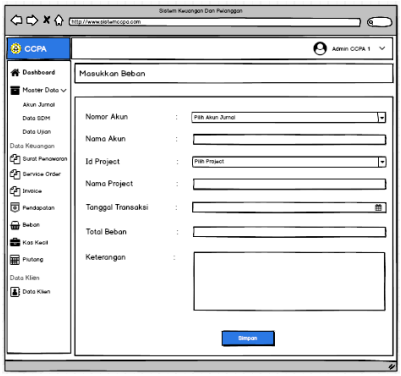
Gambar 4.42 Desain Halaman Beban

Gambar di bawah ini merupakan rancangan desain untuk halaman lihat data beban. Di halaman ini user dapat melihat data rinci dari suatu beban yang sudah dibuat sebelumnya.



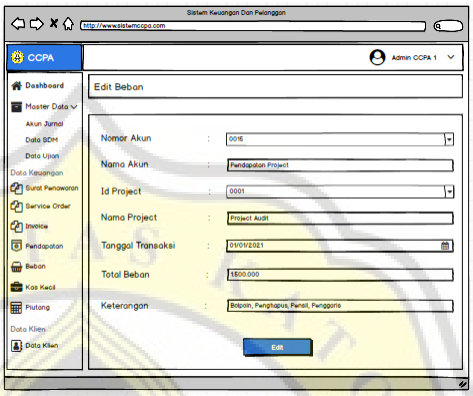
Gambar 4.43 Desain Halaman Lihat Beban

Gambar di bawah ini merupakan rancangan desain untuk halaman tambah data beban baru. Melalui halaman inilah user dapat menginputkan dan menyimpan data beban baru. Di halaman ini user diharuskan untuk mengisi kode akun tujuan, memilih id project serta mengisi keterangan beban bila ada.



Gambar 4.44 Desain Halaman Tambah Beban

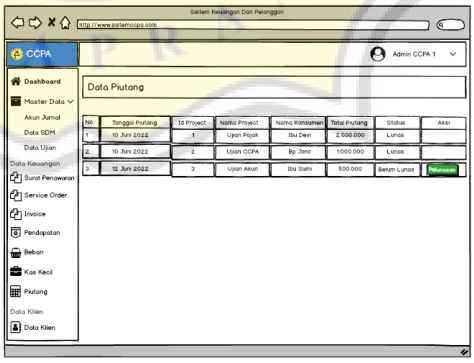
Gambar di bawah ini merupakan rancangan desain untuk halaman edit data beban. Di halaman ini, user dapat mengubah data beban yang sudah diinputkan sebelumnya apabila terjadi salah input.



Gambar 4.45 Desain Halaman Edit Beban

12. Desain Halaman Piutang

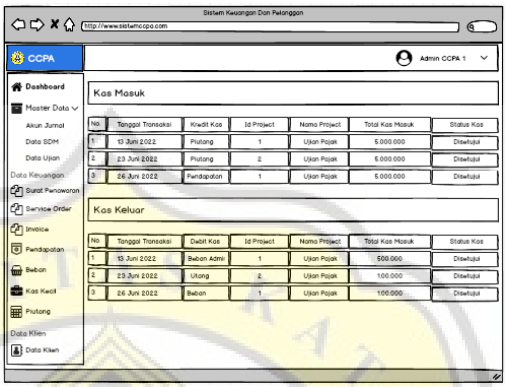
Gambar di bawah ini merupakan rancangan desain dari halaman piutang. Di halaman ini tersedia informasi daftar piutang yang sudah lunas maupun yang belum lunas. Di halaman ini pula user dapat mengubah status piutang menjadi lunas ketika sudah adanya pembayaran dari pelanggan dengan cara menekan tombol pelunasan piutang.



Gambar 4.46 Desain Halaman Piutang

13. Desain Halaman Kas Kecil

Gambar di bawah ini merupakan rancangan desain untuk halaman kas kecil dimana di halaman ini tersedia informasi terkait dengan arus keluar masuk yang dapat dilihat oleh user.

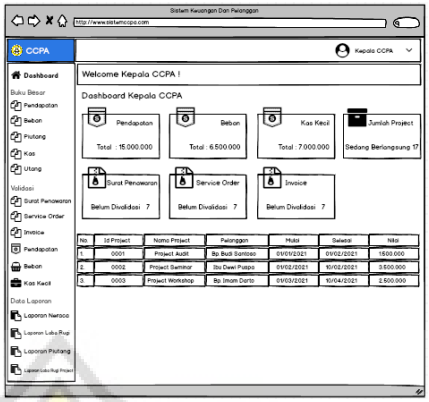


Gambar 4.47 Desain Halaman Kas Kecil

c. Desain Tampilan Depan Untuk Kepala

1. Desain Halaman *Dashboard* Kepala

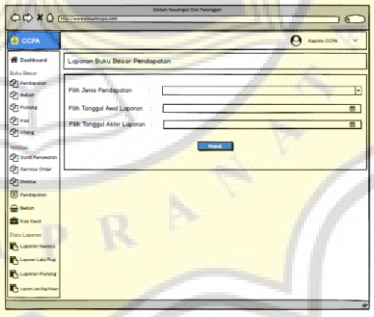
Gambar di bawah ini adalah rancangan desain dari halaman *dashboard* untuk user yang login sebagai kepala. Di halaman ini terdapat informasi yang lebih banyak dan detail bila dibandingkan dengan *dashboard* milik admin. Di halaman ini user dapat melihat jumlah total pendapatan, beban dan kas dalam suatu periode. Di halaman ini pula terdapat informasi tentang jumlah surat penawaran, *service order*, dan *invoice* yang belum divalidasi. Dan di halaman ini juga, user dapat melihat jumlah project yang sedang berlangsung dan tabel yang berisi daftar project yang sudah selesai dan sedang berlangsung.



Gambar 4.48 Desain Dashboard Kepala

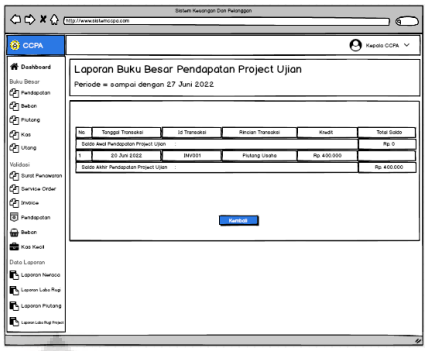
2. Desain Halaman Buku Besar Pendapatan

Gambar di bawah ini menggambarkan rancangan desain untuk halaman buku besar dari akun pendapatan. Sebelum masuk ke data buku besar akun pendapatan, user diharuskan untuk memilih jenis pendapatanya dan mengisi tanggal awal dan tanggal selesai periode. Setelah itu user diharuskan menekan tombol masuk. Maka setelah itu buku besar untuk jenis pendapatan yang dipilih sebelumnya akan terbuka di halaman selanjutnya.



Gambar 4.49 Desain Halaman Awal BB Pendapatan

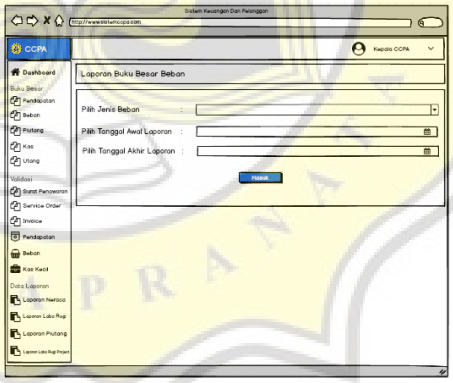
Gambar di bawah ini merupakan rancangan desain untuk halaman lanjutan setelah user selesai mengisi dan menekan tombol masuk. Di halaman ini user dapat melihat buku besar beserta dengan detail transaksinya untuk akun pendapatan yang telah dipilih sebelumnya.



Gambar 4.50 Desain Halaman BB Pendapatan

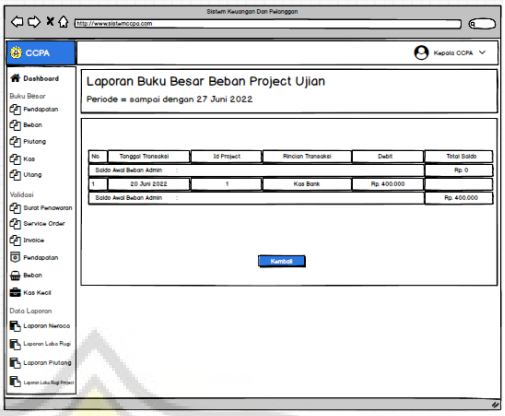
3. Desain Halaman Buku Besar Beban

Gambar di bawah ini menggambarkan rancangan desain untuk halaman buku besar dari akun beban. Sebelum masuk ke data buku besar akun beban, user diharuskan untuk memilih jenis bebannya dan mengisi tanggal awal dan tanggal selesai periode. Setelah itu user diharuskan menekan tombol masuk. Maka setelah itu buku besar untuk jenis beban yang dipilih sebelumnya akan terbuka di halaman selanjutnya.



Gambar 4.51 Desain Halaman Awal BB Beban

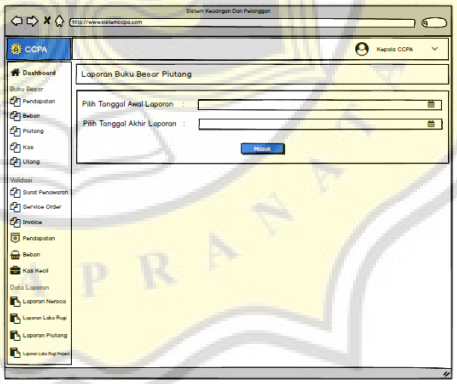
Gambar di bawah ini merupakan rancangan desain untuk halaman lanjutan setelah user selesai mengisi dan menekan tombol masuk. Di halaman ini user dapat melihat buku besar beserta dengan detail transaksinya untuk akun beban yang telah dipilih sebelumnya.



Gambar 4.52 Desain Halaman BB Beban

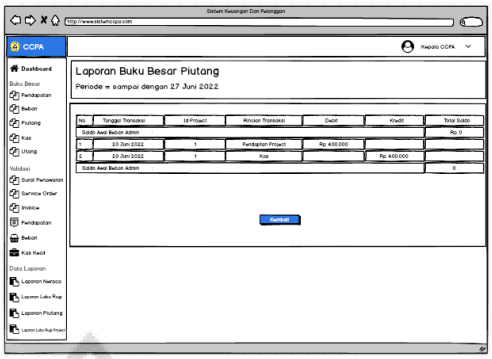
4. Desain Halaman Buku Besar Piutang

Gambar di bawah ini adalah rancangan desain untuk halaman buku besar akun piutang. Sebelum masuk ke detail buku besar akun piutang, user diharuskan untuk mengisi tanggal awal dan tanggal akhir dari periode buku besar yang ingin dilihat. Lalu setelah itu user menekan tombol masuk, dan sistem akan otomatis membawa user ke halaman selanjutnya.



Gambar 4.53 Desain Halaman Awal BB Piutang

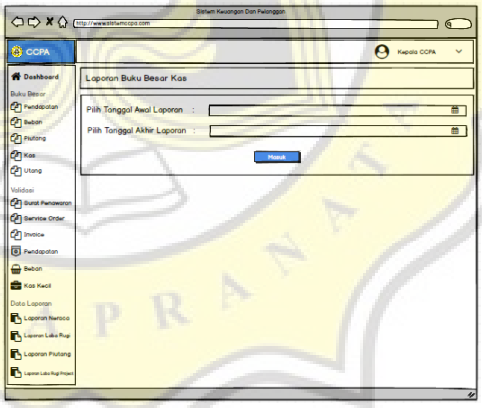
Gambar di bawah ini merupakan rancangan desain untuk halaman lanjutan setelah user selesai mengisi dan menekan tombol masuk. Di halaman ini user dapat melihat buku besar beserta dengan detail transaksinya untuk akun piutang.



Gambar 4.54 Desain Halaman BB Piutang

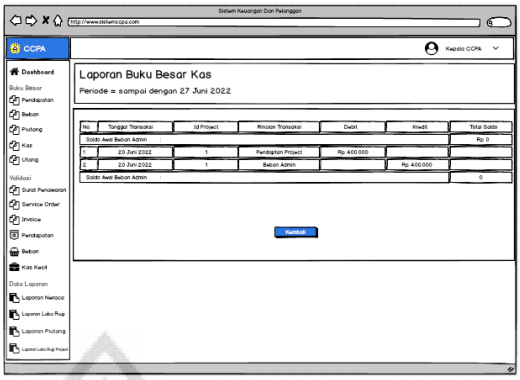
5. Desain Halaman Buku Besar Kas

Gambar di bawah ini adalah rancangan desain untuk halaman buku besar akun kas. Sebelum masuk ke detail buku besar akun kas, user diharuskan untuk mengisi tanggal awal dan tanggal akhir dari periode buku besar yang ingin dilihat. Lalu setelah itu user menekan tombol masuk, dan sistem akan otomatis membawa user ke halaman selanjutnya.



Gambar 4.55 Desain Halaman Awal BB Kas

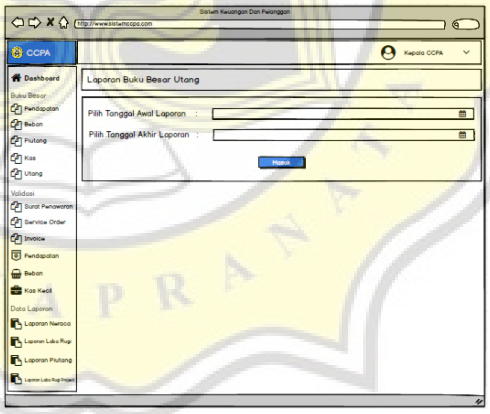
Gambar di bawah ini merupakan rancangan desain untuk halaman lanjutan setelah user selesai mengisi dan menekan tombol masuk. Di halaman ini user dapat melihat buku besar beserta dengan detail transaksinya untuk akun kas.



Gambar 4.56 Desain Halaman BB Kas

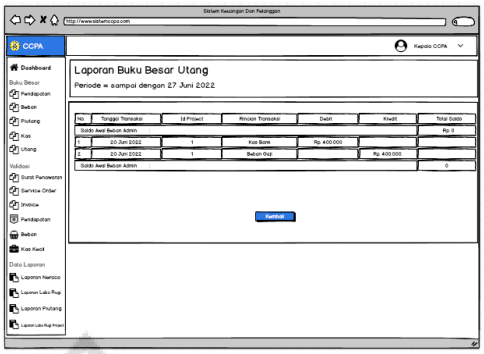
6. Desain Halaman Buku Besar Utang

Gambar di bawah ini adalah rancangan desain untuk halaman buku besar akun utang. Sebelum masuk ke detail buku besar akun utang, user diharuskan untuk mengisi tanggal awal dan tanggal akhir dari periode buku besar yang ingin dilihat. Lalu setelah itu user menekan tombol masuk, dan sistem akan otomatis membawa user ke halaman selanjutnya.



Gambar 4.57 Desain Halaman Awal BB Utang

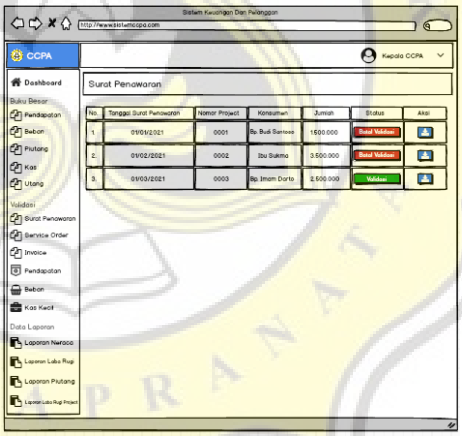
Gambar di bawah ini merupakan rancangan desain untuk halaman lanjutan setelah user selesai mengisi dan menekan tombol masuk. Di halaman ini user dapat melihat buku besar beserta dengan detail transaksinya untuk akun utang.



Gambar 4.58 Desain Halaman BB Utang

7. Desain Halaman Validasi Surat Penawaran

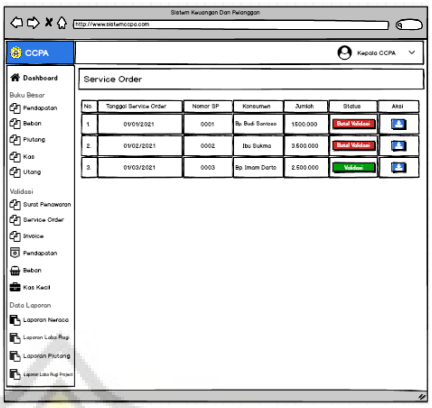
Gambar di bawah ini menggambarkan rancangan desain untuk halaman validasi surat penawaran. Di halaman ini user dapat melakukan validasi surat penawaran maupun membatalkan validasi dari suatu surat penawaran yang sudah divalidasi. Di halaman ini pula user dapat mengunduh surat penawaran.



Gambar 4.59 Desain Halaman Validasi Surat Penawaran

8. Desain Halaman Validasi *Service Order*

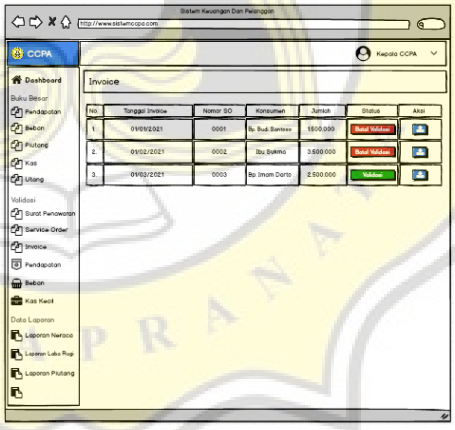
Gambar di bawah ini menggambarkan rancangan desain untuk halaman validasi *service order*. Di halaman ini user dapat melakukan validasi *service order* maupun membatalkan validasi dari suatu *service order* yang sudah divalidasi. Di halaman ini pula user dapat mengunduh *service order*.



Gambar 4.60 Desain Halaman Validasi Service Order

9. Desain Halaman Validasi *Invoice*

Gambar di bawah ini menggambarkan rancangan desain untuk halaman validasi *invoice*. Di halaman ini user dapat melakukan validasi *invoice* maupun membatalkan validasi dari suatu *invoice* yang sudah divalidasi. Di halaman ini pula user dapat mengunduh *invoice*.

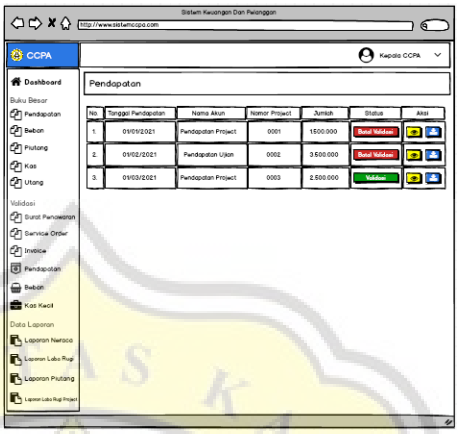


Gambar 4.61 Desain Halaman Validasi Invoice

10. Desain Halaman Validasi Pendapatan

Gambar di bawah ini menggambarkan rancangan desain untuk halaman validasi pendapatan. Di halaman ini user dapat melakukan validasi pendapatan maupun membatalkan validasi dari suatu pendapatan yang sudah divalidasi. Di halaman ini pula user

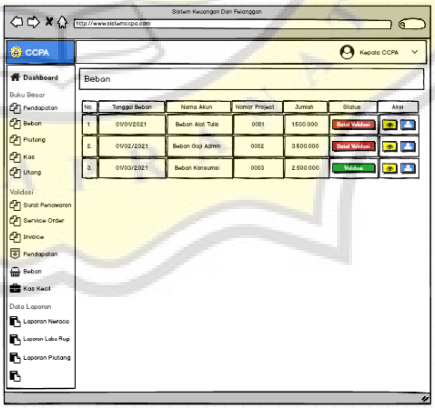
dapat mengunduh pendapatan serta melihat rincian data dari suatu transaksi pendapatan.



Gambar 4.62 Desain Halaman Validasi Pendapatan

11. Desain Halaman Validasi Beban

Gambar di bawah ini menggambarkan rancangan desain untuk halaman validasi beban. Di halaman ini user dapat melakukan validasi beban maupun membatalkan validasi dari suatu beban yang sudah divalidasi. Di halaman ini pula user dapat mengunduh beban serta melihat rincian data dari suatu transaksi beban.

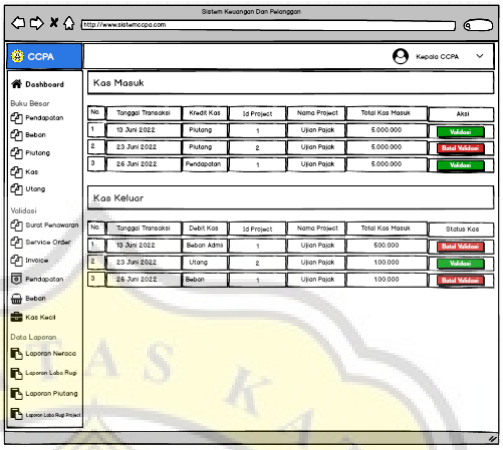


Gambar 4.63 Desain Halaman Validasi Beban

12. Desain Halaman Validasi Kas Kecil

Gambar di bawah ini menggambarkan rancangan desain untuk halaman validasi kas masuk dan keluar. Di halaman ini user

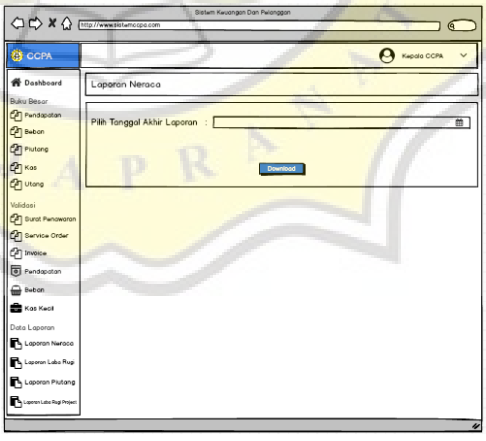
dapat melakukan validasi beban maupun membatalkan validasi dari suatu kas masuk dan keluar yang sudah divalidasi.



Gambar 4.64 Desain Halaman Validasi Kas Kecil

13. Desain Halaman Laporan Neraca

Gambar di bawah ini adalah rancangan desain untuk halaman yang harus diisi user sebelum user dapat mengunduh laporan neraca dalam suatu periode. Di halaman ini, user diharuskan untuk mengisi tanggal akhir laporan lalu tekan tombol download dan laporan neraca akan terunduh secara otomatis setelah itu.

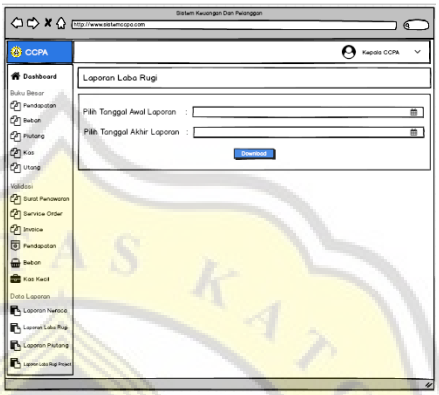


Gambar 4.65 Desain Halaman Laproan Neraca

14. Desain Halaman Laporan Laba Rugi

Gambar di bawah ini adalah rancangan desain untuk halaman yang harus diisi user sebelum user dapat mengunduh

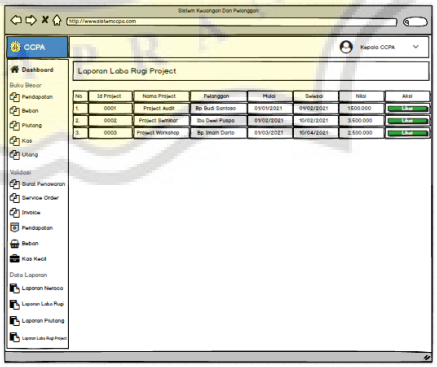
laporan laba rugi dalam suatu periode. Di halaman ini, user diharuskan untuk mengisi tanggal awal dan tanggal akhir laporan lalu tekan tombol download dan laporan laba rugi dalam periode yang dipilih akan terunduh secara otomatis setelah itu.



Gambar 4.66 Desain Halaman Laporan Laba Rugi

15. Desain Halaman Laporan Laba Rugi Project

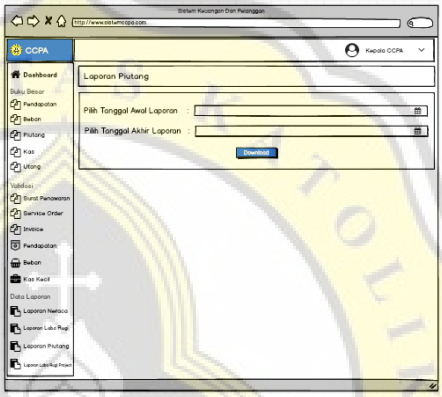
Gambar di bawah ini merupakan rancangan desain untuk halaman laporan laba rugi project. Di halaman ini tersedia sebuah tabel yang berisikan daftar project yang ada dan di baris paling kanan dari masing-masing projectnya terdapat tombol lihat yang apabila ditekan akan mengarahkan user ke halaman yang menampilkan detail laporan laba rugi dari project yang dipilih.



Gambar 4.67 Desain Halaman Laporan Laba Rugi Project

16. Desain Halaman Laporan Piutang

Gambar di bawah ini adalah rancangan desain untuk halaman yang harus diisi user sebelum user dapat mengunduh laporan piutang dalam suatu periode. Di halaman ini, user diharuskan untuk mengisi tanggal awal dan tanggal akhir laporan lalu tekan tombol download dan laporan piutang dalam periode yang dipilih akan terunduh secara otomatis setelah itu.



Gambar 4.68 Desain Halaman Laporan Piutang

4.3 Pembuatan Sistem

4.3.1 Pembuatan Database

a. Tabel Data User

Tabel data user adalah sebuah tabel yang digunakan untuk menyimpan data user beserta dengan *password*nya dan level hak aksesnya masing-masing. Tabel ini akan digunakan ketika proses login yang dilakukan oleh user.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	name	varchar(255)	utf8mb4_unicode_ci		No	None		
3	email	varchar(255)	utf8mb4_unicode_ci		No	None		
4	email_verified_at	timestamp			Yes	NULL		
5	password	varchar(255)	utf8mb4_unicode_ci		No	None		
6	level	int(10)			Yes	3		
7	remember_token	varchar(100)	utf8mb4_unicode_ci		Yes	NULL		
8	created_at	timestamp			Yes	NULL		
9	updated_at	timestamp			Yes	NULL		

Gambar 4.69 Tabel Data User

b. Tabel Data Konsumen

Tabel data konsumen adalah sebuah tabel yang digunakan untuk menyimpan data pelanggan. Tabel ini akan digunakan ketika user sistem membuat project, membuat surat penawaran, membuat *service order*, serta membuat *invoice*.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id_konsumen	int(125)			No	None		AUTO_INCREMENT
2	nama_konsumen	varchar(125)	utf8mb4_general_ci		No	None		
3	jabatan_konsumen	varchar(125)	utf8mb4_general_ci		No	None		
4	instansi_konsumen	varchar(125)	utf8mb4_general_ci		No	None		
5	alamat_konsumen	varchar(125)	utf8mb4_general_ci		No	None		
6	kota_konsumen	varchar(125)	utf8mb4_general_ci		No	None		
7	email_konsumen	varchar(125)	utf8mb4_general_ci		No	None		
8	nomorhp_konsumen	varchar(25)	utf8mb4_general_ci		No	None		
9	created_at	datetime			No	current_timestamp()		
10	updated_at	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()

Gambar 4.70 Tabel Data Konsumen

c. Tabel Data Project

Tabel data project adalah suatu tabel yang memiliki fungsi untuk menyimpan data project yang telah diinputkan oleh admin sistem sebelumnya.Tabel ini akan terlibat dalam proses pembuatan dan pengolahan data keuangan hingga menjadi laporan keuangan.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id_project	int(125)			No	None		AUTO_INCREMENT
2	nama_project	varchar(125)	utf8mb4_general_ci		No	None		
3	tanggalmulai_project	datetime			No	None		
4	tanggalselesai_project	datetime			No	None		
5	nilai_project	int(125)			No	None		
6	status_project	varchar(15)	utf8mb4_general_ci		Yes	Berlangsung		
7	status_hapus	varchar(15)	utf8mb4_general_ci		Yes	No		
8	created_at	datetime			No	current_timestamp()		
9	updated_at	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()

Gambar 4.71 Tabel Data Project

d. Tabel Data SDM

Tabel data SDM merupakan suatu tabel yang ada di dalam sistem, yang berguna untuk menyimpan data dosen/tenaga pendidik yang terlibat aktif dalam pelaksanaan suatu project.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id_sdm	int(125)			No	None		AUTO_INCREMENT
2	nama_sdm	varchar(125)	utf8mb4_general_ci		No	None		
3	nama_bank	varchar(125)	utf8mb4_general_ci		No	None		
4	norek_sdm	varchar(125)	utf8mb4_general_ci		No	None		
5	nomorhp_sdm	varchar(25)	utf8mb4_general_ci		No	None		
6	created_at	datetime			No	current_timestamp()		
7	updated_at	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()

Gambar 4.72 Tabel Data SDM

e. Tabel Data Ujian

Tabel data ujian adalah suatu tabel entitas yang digunakan untuk menyimpan master data untuk project-project.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id_ujian	int(125)			No	None		AUTO_INCREMENT
2	nama_ujian	varchar(125)	utf8mb4_general_ci		No	None		
3	created_at	datetime			No	current_timestamp()		
4	updated_at	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()

Gambar 4.73 Tabel Data Ujian

f. Tabel Data Akun

Tabel data akun adalah suatu tabel entitas yang digunakan untuk menyimpan data akun-akun akuntansi. Tabel ini akan terlibat aktif

dalam proses pencatan dan pengolahan transaksi keuangan yang terjadi.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id_akun	int(125)			No	None		
2	nama_akun	varchar(125)	utf8mb4_general_ci		No	None		
3	kategori_akun	varchar(125)	utf8mb4_general_ci		No	None		
4	keterangan_akun	varchar(125)	utf8mb4_general_ci		Yes	NULL		
5	created_at	datetime			No	current_timestamp()		
6	updated_at	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()

Gambar 4.74 Tabel Data Akun

g. Tabel Rincian Konsumen

Tabel rincian konsumen adalah suatu tabel yang menjadi penghubung antara tabel data project dan tabel data konsumen. Tabel ini berperan aktif dalam proses pengolahan data pelanggan karna dengan tabel inilah user dapat melihat seorang pelanggan telah ikut di berapa project dan dapat melihat pelanggan-pelanggan yang ada di suatu project.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id_project	int(125)			No	None		
2	id_konsumen	int(125)			No	None		
3	nama_konsumen	varchar(125)	utf8mb4_general_ci		No	None		
4	email_konsumen	varchar(125)	utf8mb4_general_ci		No	None		
5	nomorhp_konsumen	varchar(15)	utf8mb4_general_ci		No	None		
6	status_konsumen	varchar(125)	utf8mb4_general_ci		No	Umum		

Gambar 4.75 Tabel Data Konsumen

h. Tabel Rincian Project

Tabel rincian project adalah suatu tabel yang menjadi penghubung antara tabel data project dengan tabel data SDM. Tabel ini berfungsi untuk menyimpan dan mencatat SDM (dosen) yang terlibat dalam suatu project.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id_rincian	int(125)			No	None		AUTO_INCREMENT
2	id_project	int(125)			No	None		
3	id_sdm	int(125)			No	None		
4	nama_sdm	varchar(125)	utf8mb4_general_ci		No	None		
5	tanggal_pelaksanaan	datetime			No	None		
6	durasi_pelaksanaan	int(125)			No	None		
7	created_at	datetime			No	current_timestamp()		
8	updated_at	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()

Gambar 4.76 Tabel Data Rincian Project

i. Tabel Rincian Ujian

Tabel rincian ujian ini merupakan suatu tabel yang mencatat dan menyimpan data harga project per pelanggan berdasarkan statusnya. Tabel ini menjadi kunci utama dalam proses penghitungan nilai total suatu project.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id_rinci	int(125)			No	None		AUTO_INCREMENT
2	id_ujian	int(125)			No	None		
3	status_peserta	varchar(125)	utf8mb4_general_ci		No	None		
4	harga_ujian	varchar(125)	utf8mb4_general_ci		No	None		
5	created_at	datetime			No	current_timestamp()		
6	updated_at	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()

Gambar 4.77 Tabel Data Rincian Ujian

j. Tabel Surat Penawaran

Tabel surat penawaran adalah suatu tabel yang menyimpan dan mencatat data tentang surat penawaran. Melalui tabel ini jugalah user dapat melihat detail data dari suatu surat penawaran.Tabel ini menjadi dasar ketika proses pembuatan surat penawaran.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id_suratpenawaran	varchar(125)	utf8mb4_general_ci		No	None		
2	id_project	int(125)			No	None		
3	tanggal_suratpenawaran	datetime			No	None		
4	keterangan_suratpenawaran	varchar(500)	utf8mb4_general_ci		Yes	NULL		
5	status_suratpenawaran	varchar(125)	utf8mb4_general_ci		No	Belum Disetujui		
6	status_hapus	varchar(125)	utf8mb4_general_ci		No	No		
7	created_at	datetime			No	current_timestamp()		
8	updated_at	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()

Gambar 4.78 Tabel Data Surat Penawaran

k. Tabel *Service Order*

Tabel *service order* adalah suatu tabel yang berguna untuk menyimpan dan mencatat data tentang *service order*. Tabel ini menjadi dasar ketika proses pembuatan *service order*.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id_serviceorder	varchar(125)	utf8mb4_general_ci		No	None		
2	id_suratpenawaran	varchar(125)	utf8mb4_general_ci		No	None		
3	id_project	int(125)			Yes	NULL		
4	tanggal_serviceorder	datetime			No	None		
5	status_serviceorder	varchar(125)	utf8mb4_general_ci		No	Belum Disetujui		
6	status_hapus	varchar(125)	utf8mb4_general_ci		No	No		
7	created_at	datetime			No	current_timestamp()		
8	updated_at	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()

Gambar 4.79 Tabel Data Service Order

l. Tabel *Invoice*

Tabel *Invoice* merupakan suatu tabel yang berguna untuk menyimpan dan mencatat data tentang *Invoice* dari suatu project. Tabel ini menjadi dasar ketika proses pembuatan *invoice*. Tabel ini nantinya akan berperan dalam proses pencatatan dan pengolahan pendapatan dan piutang.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id_invoice	varchar(125)	utf8mb4_general_ci		No	None		
2	id_suratpenawaran	varchar(125)	utf8mb4_general_ci		No	None		
3	id_serviceorder	varchar(125)	utf8mb4_general_ci		No	None		
4	id_project	int(125)			No	None		
5	id_konsumen	int(125)			Yes	NULL		
6	tanggal_invoice	datetime			No	None		
7	status_invoice	varchar(125)	utf8mb4_general_ci		No	Belum Disetujui		
8	status_hapus	varchar(125)	utf8mb4_general_ci		No	No		
9	created_at	datetime			No	current_timestamp()		
10	updated_at	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()

Gambar 4.80 Tabel Data Invoice

m. Tabel Kas

Tabel kas merupakan suatu tabel yang menyimpan dan mencatat transaksi-transaksi keuangan yang melibatkan kas. Tabel ini akan berguna dalam proses pembuatan dan pengolahan laporan keuangan

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id_kas	int(125)			No	None		AUTO_INCREMENT
2	id_akun	int(125)			No	101		
3	id_project	int(125)			No	None		
4	total_kas	int(125)			No	None		
5	tanggal_kas	datetime			No	None		
6	debit_kas	varchar(125)	utf8mb4_general_ci		Yes	NULL		
7	kredit_kas	varchar(125)	utf8mb4_general_ci		Yes	NULL		
8	id_transaksi	varchar(125)	utf8mb4_general_ci		No	None		
9	keterangan_kas	varchar(125)	utf8mb4_general_ci		Yes	NULL		
10	status_kas	varchar(125)	utf8mb4_general_ci		No	Belum Disetujui		
11	created_at	datetime			No	current_timestamp()		
12	updated_at	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()

Gambar 4.81 Tabel Data Kas

n. Tabel Bank

Tabel bank merupakan tabel yang berguna untuk menyimpan dan mencatat transaksi-transaksi keuangan yang berkaitan dengan dengan kas bank. Tabel ini juga berguna untuk proses pengolahan data keuangan menjadi laporan keuangan.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id_bank	int(125)			No	None		AUTO_INCREMENT
2	id_akun	int(125)			No	None		
3	id_project	int(125)			No	None		
4	nama_akun	varchar(125)	utf8mb4_general_ci		No	None		
5	total_bank	int(125)			No	None		
6	tanggal_bank	datetime			No	None		
7	debit_bank	varchar(125)	utf8mb4_general_ci		Yes	NULL		
8	kredit_bank	varchar(125)	utf8mb4_general_ci		Yes	NULL		
9	id_transaksi	varchar(125)	utf8mb4_general_ci		No	None		
10	keterangan_bank	varchar(125)	utf8mb4_general_ci		Yes	NULL		
11	status_bank	varchar(125)	utf8mb4_general_ci		No	Belum Disetujui		
12	created_at	datetime			No	current_timestamp()		

Gambar 4.82 Tabel Data Bank

o. Tabel Piutang

Tabel piutang merupakan tabel yang berguna untuk menyimpan dan mencatat transaksi keuangan khusunya transaksi piutang. Tabel ini juga berguna dalam proses pembuatan laporan piutang.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id_piutang	int(125)			No	None		AUTO_INCREMENT
2	id_akun	int(125)			No	110		
3	id_project	int(125)			No	None		
4	id_konsumen	int(125)			Yes	NULL		
5	total_piutang	int(125)			No	None		
6	tanggal_piutang	datetime			No	None		
7	debit_piutang	varchar(125)	utf8mb4_general_ci		Yes	NULL		
8	kredit_piutang	varchar(125)	utf8mb4_general_ci		Yes	NULL		
9	id_transaksi	varchar(125)	utf8mb4_general_ci		No	None		
10	tanggal_pelunasan	datetime			Yes	NULL		
11	keterangan_piutang	varchar(125)	utf8mb4_general_ci		Yes	NULL		
12	status_piutang	varchar(125)	utf8mb4_general_ci		No	Belum Disetujui		
13	created_at	datetime			No	current_timestamp()		
14	updated_at	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()

Gambar 4.83 Tabel Data Piutang

p. Tabel Utang

Tabel utang merupakan tabel yang menyimpan dan mencatat transaksi keuangan khususnya transaksi utang.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id_utang	int(125)			No	None		AUTO_INCREMENT
2	id_akun	int(125)			No	210		
3	id_project	int(125)			No	None		
4	total_utang	int(125)			No	None		
5	tanggal_utang	datetime			No	None		
6	debit_utang	varchar(125)	utf8mb4_general_ci		Yes	NULL		
7	kredit_utang	varchar(125)	utf8mb4_general_ci		Yes	NULL		
8	id_transaksi	varchar(125)	utf8mb4_general_ci		No	None		
9	keterangan_utang	varchar(125)	utf8mb4_general_ci		Yes	NULL		
10	status_utang	varchar(125)	utf8mb4_general_ci		No	Belum Disetujui		
11	created_at	datetime			No	current_timestamp()		
12	updated_at	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()

Gambar 4.84 Tabel Data Utang

q. Tabel Pendapatan

Tabel pendapatan merupakan tabel yang menyimpan dan mencatat transaksi-transaksi pendapatan. Tabel ini akan berguna dalam proses pengolahan data keuangan menjadi laporan laba rugi baik itu laba rugi project maupun laporan laba rugi periodik.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id_pendapatan	varchar(125)	utf8mb4_general_ci		No	None		
2	id_akun	int(125)			No	None		
3	id_project	int(125)			No	None		
4	jenis_pendapatan	varchar(125)	utf8mb4_general_ci		No	None		
5	total_pendapatan	int(125)			No	None		
6	tanggal_pendapatan	datetime			No	None		
7	debit_pendapatan	int(125)			Yes	NULL		
8	id_invoice	varchar(125)	utf8mb4_general_ci		Yes	NULL		
9	keterangan_pendapatan	varchar(125)	utf8mb4_general_ci		Yes	NULL		
10	status_pendapatan	varchar(125)	utf8mb4_general_ci		No	Belum Disetujui		
11	created_at	datetime			No	current_timestamp()		
12	updated_at	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()

Gambar 4.85 Tabel Data Pendapatan

r. Tabel Beban

Tabel beban merupakan tabel yang menyimpan dan mencatat transaksi-transaksi beban. Tabel ini akan berguna dalam proses pengolahan data keuangan menjadi laporan laba rugi baik itu laba rugi project maupun laporan laba rugi periodik.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id_beban	varchar(125)	utf8mb4_general_ci		No	None		
2	id_akun	int(125)			No	None		
3	id_project	int(125)			No	None		
4	jenis_beban	varchar(125)	utf8mb4_general_ci		No	None		
5	total_beban	int(125)			No	None		
6	tanggal_beban	datetime			No	None		
7	kredit_beban	int(125)			Yes	NULL		
8	keterangan_beban	varchar(125)	utf8mb4_general_ci		Yes	NULL		
9	status_beban	varchar(125)	utf8mb4_general_ci		No	Belum Disetujui		
10	created_at	datetime			No	current_timestamp()		
11	updated_at	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()

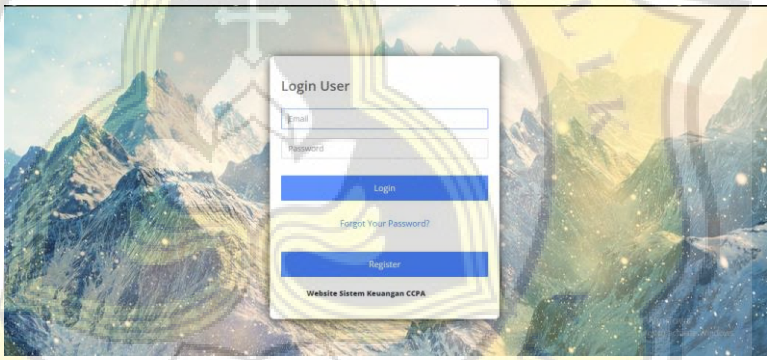
Gambar 4.86 Tabel Data Beban

4.3.2 Pembuatan Sistem

1. Halaman *Login, Logout,* dan *Register*

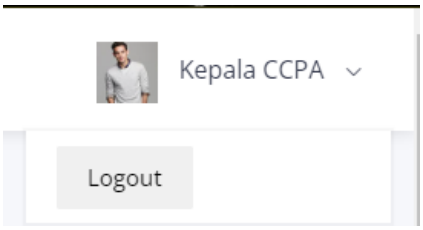
Gambar dibawah ini merupakan hasil implementasi dari halaman *login*. Halaman ini merupakan halaman pertama yang harus diisi oleh user sebelum user masuk dan menggunakan sistem ini. Di halaman ini user diharuskan untuk menginputkan *email* dan *password*. Apabila *email* dan *passwordnya* sesuai dengan data yang ada di tabel user maka user akan diarahkan ke halaman *dashboard*

sesuai dengan hak askesnya. Apabila user melakukan proses *login* dengan menggunakan user dengan hak akses sebagai admin, maka setelah berhasil *login*, maka user akan diarahkan ke halaman *dashboard* admin, namun apabila user melakukan proses *login* dengan menggunakan user dengan hak akses sebagai kepala, maka setelah berhasil *login*, maka user akan diarahkan ke halaman *dashboard* kepala. Namun apabila *email* dan *password* yang diinputkan tidak sesuai maka akan muncul notifikasi dan user diharuskan menginputkan ulang *email* dan *passwordnya*. Selain itu, terdapat fitur *forgot password* pada halaman *login*, dimana fitur tersebut akan membantu admin ketika admin lupa terhadap *password* akun yang digunakan.



Gambar 4.87 Halaman Login

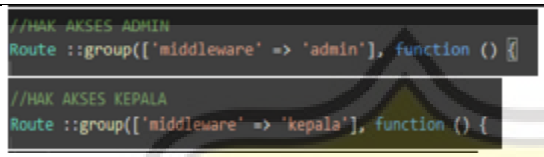
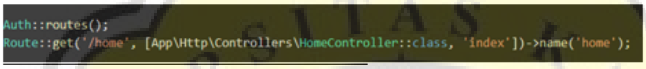
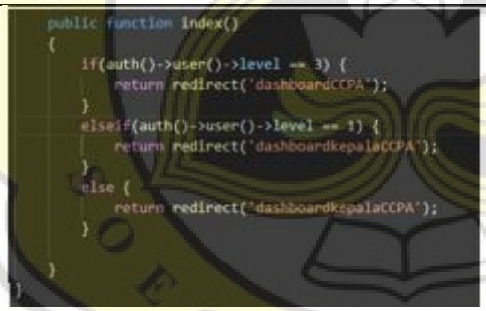
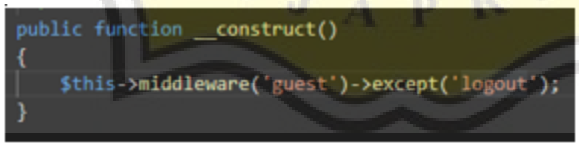
Gambar di bawah ini merupakan fitur *logout* yang ada di sistem ini. Ketika tombol ini ditekan maka user akan keluar dari sistem dan apabila user ingin menggunakan sistem ini kembali maka user diharuskan untuk melalui proses *login* ulang.



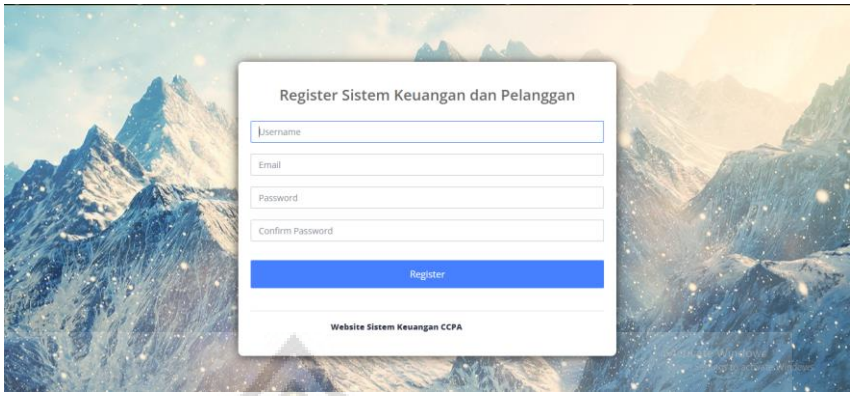
Gambar 4.88 Halaman Logout

Gambar di bawah ini merupakan gambar dari *route* dan juga *controller* yang berperan aktif dalam implementasi proses *login* dan *logout* dalam sistem ini.

Tabel 4.1 Route dan Controller Login & Logout

No	Gambar	Penjelasan
1.		Route yang membagi hak akses pengguna
2.		Route awal yang akan diakses oleh user dan route ini akan mengarahkan user ke controller home index
3.		Controller yang akan mengarahkan user ke tampilan
4.		Controller untuk melakukan proses logut

Gambar di bawah ini merupakan halaman *register* dari sistem ini. Halaman ini digunakan ketika user akan membuat akun baru. Di halaman ini user diharuskan untuk menginputkan *username, email,* dan *password* barulah bisa menekan tombol register dan akun user baru sudah tersimpan dalam *database.*



Gambar 4.89 Halaman Register

Gambar di bawah ini merupakan gambar dari *route* dan juga *controller* yang berperan aktif dalam implementasi proses *register* dalam sistem ini.

```php
protected function validator(array $data)
{
    return Validator::make($data, [
        'name' => ['required', 'string', 'max:255'],
        'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
        'password' => ['required', 'string', 'min:8', 'confirmed'],
    ]);
}

/**
 * Create a new user instance after a valid registration.
 *
 * @param  array  $data
 * @return \App\Models\User
 */
protected function create(array $data)
{
    return User::create([
        'name' => $data['name'],
        'email' => $data['email'],
        'password' => Hash::make($data['password']),
    ]);
}
```

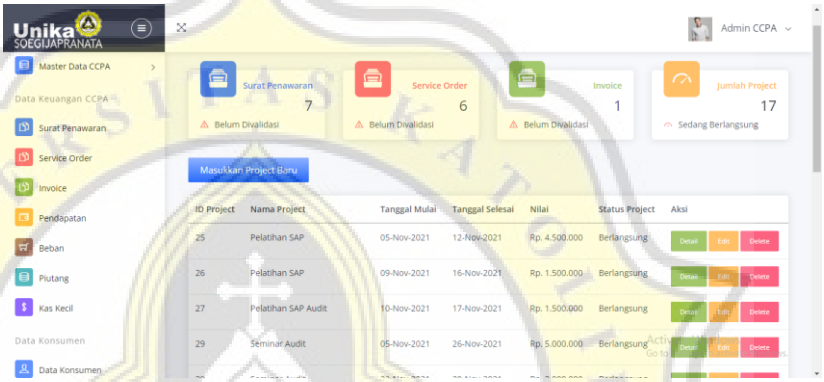
Gambar 4.90 Route dan Controller Register

2. Halaman Admin

a. Halaman *Dashboard* Admin

Gambar di bawah ini merupakan hasil dari implemnetasi untuk halaman *dashboard* admin. Di halaman ini admin dapat melihat beberapa informasi penting diantaranya jumlah surat penawaran, *service order*¸dan *invoice* yang belum divalidasi serta jumlah project yang berlangsung.Di halaman ini pula user dapat melihat tabel yang berisikan daftar project yang ada. Dimana di tiap projectnya user dapat melihat data detailnya, mengedit datanya apabila terjadi kesalahan, serta menghapus data project yang sudah

ada. Di halaman ini pula terdapat sebuah tombol yang akan mengarahkan user ke halaman khusus untuk menambahkan project baru, yakni tombol tambah project baru. Dibagian kiri dari halaman ini terdapat *navbar* yang nantinya memudahkan user untuk mengakses halaman lainnya, dan di bagian atas dari halaman ini terdapat fitur *logout* yang akan muncul ketika user menekan nama akunnya.



Gambar 4.91 Halaman Dashboard Admin

Gambar di bawah ini merupakan gambar dari *route* dan juga *controller* yang menunjang implementasi halaman *dashboard* admin dalam sistem ini.

Tabel 4.2 Route dan Controller Dashboard Admin

No.	Gambar	Penjelasan
1.	`/Route Project CCPA` `oute ::get('/dashboardCCPA', [ProjectController::class, 'index']);`	Route yang akan mengarahkan user ke tampilan dashboard
2.	`public function indexkepala()` `{` `$data = [` `'project'=> $this->ProjectModel->allData(),` `];` `$jumlah = [` `'jumlahproject'=> $this->ProjectModel->jumlahproject(),` `'jumlahsp'=> $this->ProjectModel->jumlahsp(),` `'jumlahso'=> $this->ProjectModel->jumlahso(),` `'jumlahinvoice'=> $this->ProjectModel->jumlahinvoice(),` `'jumlahpendapatan'=> $this->ProjectModel->jumlahpendapatan(),` `'jumlahbeban'=> $this->ProjectModel->jumlahbeban(),` `];` `return view ('CCPA_dashboardKepala', $data, $jumlah);` `}`	Controller yang digunakan untuk menampilkan halaman dashboard

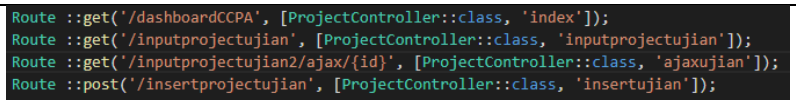
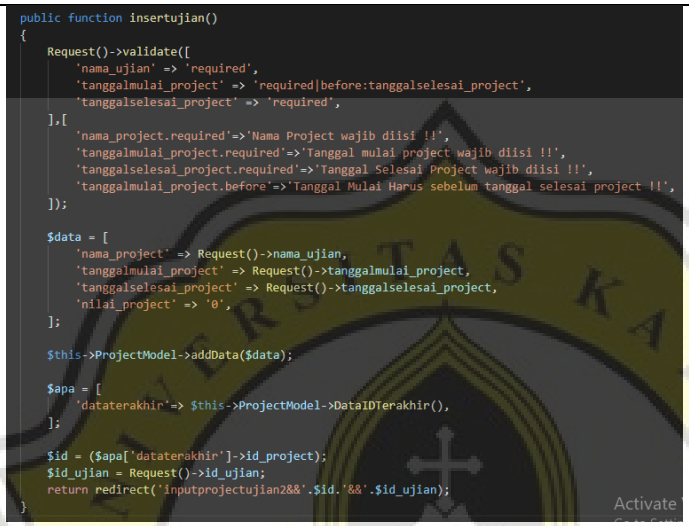
b. Halaman Project

Gambar di bawah ini merupakan hasil akhir untuk halaman tambah project. Halaman ini akan muncul setelah user menekan tombol tambah project baru yang ada di halaman *dashboard* admin. Melalui halaman ini user dapat menambahkan project baru, Proses pembuatan data project baru terbagi menjadi tiga bagian. Di bagian pertama, user diminta untuk mengisi nama project, tanggal mulai dan tanggal selesai project, serta nilai project. Di bagian yang kedua, user diharuskan untuk menginputkan data pelanggan yang terlibat dalam project tersebut. Dan di bagian ketiga, user diharuskan untuk menginputkan data SDM yang terlibat dalam pelaksanaan project tersebut.

Gambar 4.92 Halaman Tambah Project

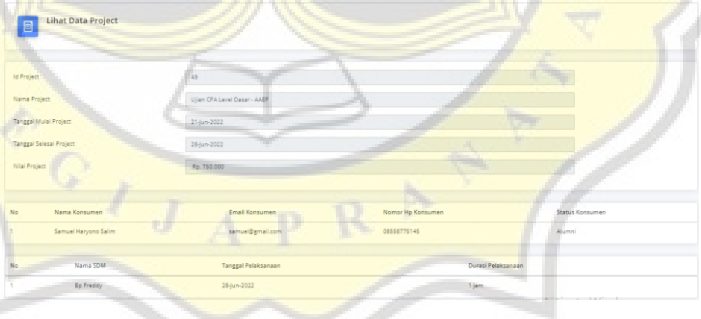
Gambar di bawah ini merupakan gambar dari *route* dan juga *controller* yang menunjang implementasi halaman tambah project dalam sistem ini. Proses menambah suatu project dibagi menjadi tiga tahap, di tahap pertama user diminta untuk menginputkan data detail project (seperti : nama project, tanggal pelaksanaan, dl), di tahap kedua user diminta untuk menginputkan data konsumen yang terlibat dalam project tersebut dan di tahap terakhir user diminta untuk menginputkan SDM (tenaga pendidik) yang terlibat dalam project tersebut.

Tabel 4.3 Route dan Controller Tambah Project

No.	Gambar	Penjelasan
1.		Route untuk proses menambah project
2.		Controller untuk menambahkan data detail project
3.		Controller untuk menginputkan data konsumen yang terlibat dalam project

4.		Controller untuk menginputkan data SDM yang terlibat

Gambar di bawah ini adalah hasil akhir untuk halaman lihat rincian project. Di halaman ini user dapat melihat data rinci dari suatu project, mulai dari Id project, nama project, tanggalm mulai dan tanggal selesai project, nilai project, daftar pelanggan yang terlibat dalam project tersebut, dan nama SDM yang terlibat dalam project tersebut. Halaman ini akan muncul ketika user menekan tombol detail yang ada di tabel daftar project di halaman *dashboard*.



Gambar 4.93 Halaman Lihat Project

Gambar di bawah ini merupakan gambar dari *route* dan juga *controller* yang menunjang implementasi halaman lihat detail project dalam sistem ini.

```
Route ::get('/lihatproject&&{id_project}', [ProjectController::class, 'lihat']);
```

```php
public function lihat($id_project)
{
    $lihat = [
        'lihatproject'=> $this->ProjectModel->LihatProject($id_project),
    ];
    $konsumen = [
        'konsumen' => $this->ProjectModel->LihatKonsumen($id_project),
        'sdm' => $this->ProjectModel->LihatSDM($id_project),
    ];
    return view('Project.lihatproject', $lihat, $konsumen);
}
```

Gambar 4.94 Route dan Controller Lihat Project

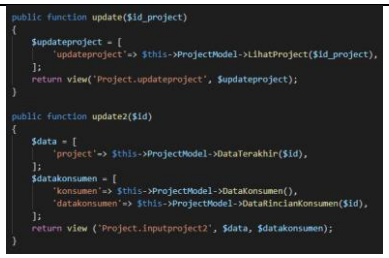
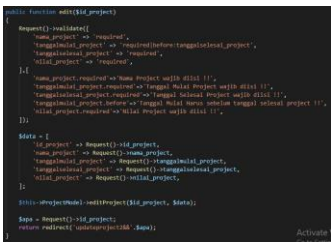
Gambar di bawah ini adalah hasil implementasi untuk halaman edit data project. Melalui halaman inilah user dapat mengubah data dari suatu project yang telah diinputkan sebelumnya.

Gambar 4.95 Halaman Edit Project

Gambar di bawah ini merupakan gambar dari *route* dan juga *controller* yang menunjang implementasi halaman edit project dalam sistem ini.

Tabel 4.4 Route dan Controller Edit Project

No.	Gambar	Penjelasan
1.	`Route ::get('/updateproject&&{id_project}', [ProjectController::class, 'update']);` `Route ::post('/editproject&&{id_project}', [ProjectController::class, 'edit']);` `Route ::get('/updateproject2&&{id_project}', [ProjectController::class, 'update2']);`	Route edit project

2.	 		Controller edit project

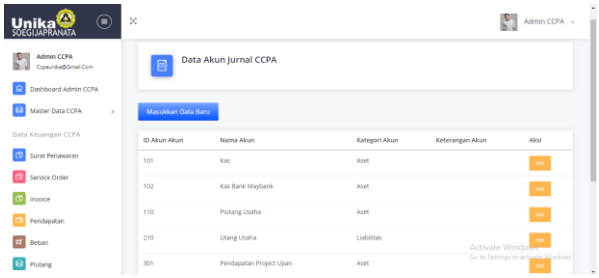
Gambar di bawah ini adalah gambar dari *route* dan juga *controller* yang mendukung fitur hapus data project dalam sistem ini.

Tabel 4.5 Route dan Controller Hapus Project

No.	Gambar	Penjelasan
1.	`Route ::get('/deleteproject&&{id_project}', [ProjectController::class, 'deleteproject']);`	Route delete project
2.	`public function deleteproject($id_project)` `{` `$this->ProjectModel->deleteDataProject($id_project);` `return redirect('dashboardCCPA')->with('pesan', 'Data Berhasil Dihapus !!');` `}`	Controller delete project

c. Halaman Master Data Akun

Gambar di bawah ini adalah hasil akhir dari halaman awal untuk master data akun. Di halaman ini user dapat melihat tabel yang berisikan data akun-akun yang sudah ada sebelumnya. Di halaman ini pula user dapat menekan tombol tambah data akun untuk menambahkan data akun baru ataupun menekan tombol edit untuk mengedit data akun yang sudah ada.



Gambar 4.96 Halaman Master Data Akun

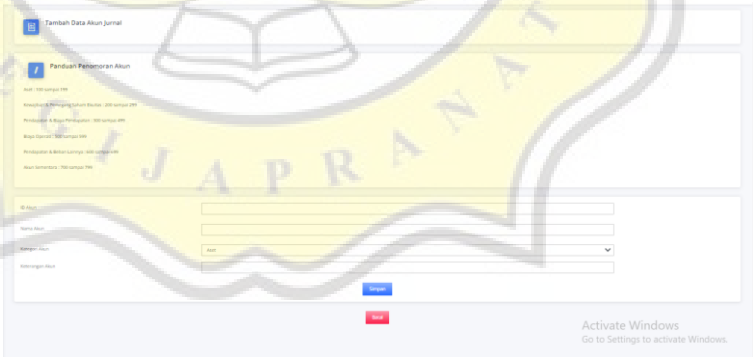
Gambar di bawah ini adalah gambar dari *route* dan juga *controller* yang menunjang berdirinya halaman awal untuk master data akun dalam sistem ini.

```
Route ::get('/dataakunccpa', [MasterCCPAController::class, 'indexakun']);
```

```
public function indexakun()
{
    $data = [
        'dataakun'=> $this->MasterCCPAModel->allDataAkun(),
    ];
    return view ('MasterCCPA.dataakunccpa', $data);
}
```

Gambar 4.97 Router dan Controller Master Data Akun

Gambar di bawah ini adalah gambar dari halaman yang ditujukan untuk menambahkan data akun baru. Di halaman ini user diminta untuk mengisi nama dan kategori akun beserta keterangannya. Di halaman ini pula terdapat catatan tentang pemberian kode untuk akun yang akan memudahkan user ketika akan memberi kode akun.



Gambar 4.98 Halaman Tambah Data Akun

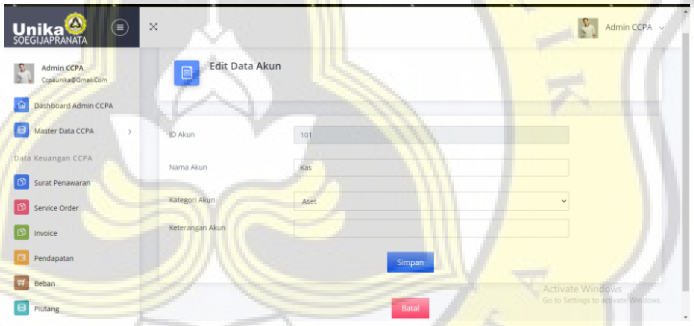
Gambar di bawah ini adalah gambar dari *route* dan juga *controller* yang mendukung proses pembuatan data akun baru dalam sistem ini.

```
Route ::post('/insertakun', [MasterCCPAController::class, 'insertakun']);
```

```php
public function insertakun()
{
    Request()->validate([
        'nama_akun' => 'required|max:255',
        'id_akun'=>'required',
    ],[
        'nama_akun.required'=>'Nama Akun wajib diisi !!',
        'id_akun.required'=>'ID Akun wajib diisi !!',
    ]);

    $dataakun = [
        'id_akun' => Request()->id_akun,
        'nama_akun' => Request()->nama_akun,
        'kategori_akun' => Request()->kategori_akun,
        'keterangan_akun' => Request()->keterangan_akun,
    ];

    $this->MasterCCPAModel->addDataAkun($dataakun);
    return redirect('dataakunccpa');
}
```

Gambar 4.99 Route dan Controller Tambah Data Akun

Gambar di bawah ini merupakan gambar dari halaman untuk mengubah data akun yang sudah ada. Di halaman ini user hanya dapat mengubah nama, kategori serta keterangan dari suatu akun.

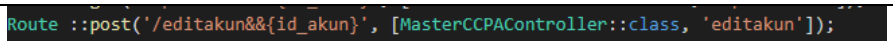
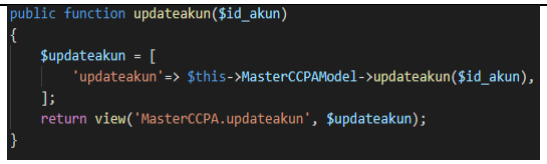


Gambar 4.100 Halaman Edit Master Data Akun

Gambar di bawah ini adalah gambar dari *route* dan juga *controller* yang mendukung proses pengeditan data akun yang sudah ada dalam sistem ini.

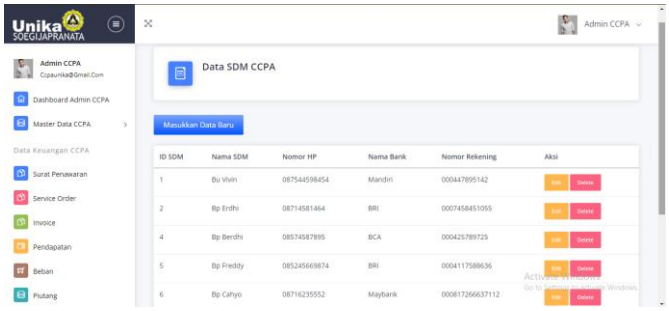
Tabel 4.6 Route dan Controller Edit Data Akun

No.	Gambar	Penjelasan
1.	`Route ::get('/updateakun&&{id_akun}', [MasterCCPAController::class, 'updateakun']);`	Route untuk membuka halaman edit akun

2.		Route untuk mengedit data akun
3.		Controller untuk membuka halaman edit akun
4.		Controller untuk mengedit data akun

d. Halaman Master Data SDM

Gambar di bawah ini merupakan hasil implementasi untuk halaman awal master data SDM. Di halaman ini user dapat melihat tabel yang berisikan data pelanggan yang sudah diinputkan sebelumnya. Di halaman ini pula user dapat mengubah atau pun menghapus data SDM yang sudah ada. Dan di halaman yang sama, user dapat menekan tombol tambah data baru yang nantinya akan mengarahkan user ke halaman khusus untuk menambahka data SDM yang baru.



Gambar 4.101 Halaman Master Data SDM

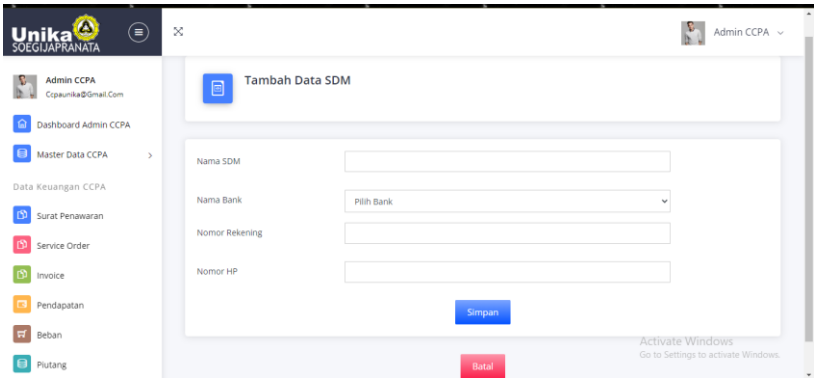
Gambar di bawah ini adalah gambar dari *route* dan juga *controller* yang menunjang berdirinya halaman awal untuk master data SDM dalam sistem ini.

```
//Route Master Data
Route ::get('/datasdmccpa', [MasterCCPAController::class, 'indexsdm']);
```

```
public function indexsdm()
{
    $data = [
        'datasdm'=> $this->MasterCCPAModel->allDataSDM(),
    ];
    return view ('MasterCCPA.datasdmccpa', $data);
}
```

Gambar 4.102 Route dan Controller Master Data Akun

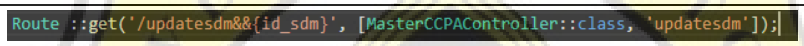
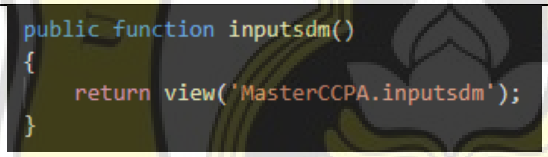
Gambar di bawah ini merupakan hasil akhir untuk halaman yang akan digunakan user untuk menambahkan data SDM yang baru. Di halaman ini user diminta untuk menginputkan nama, nama bank, nomor rekening dan nomor hp untuk data SDM yang baru.



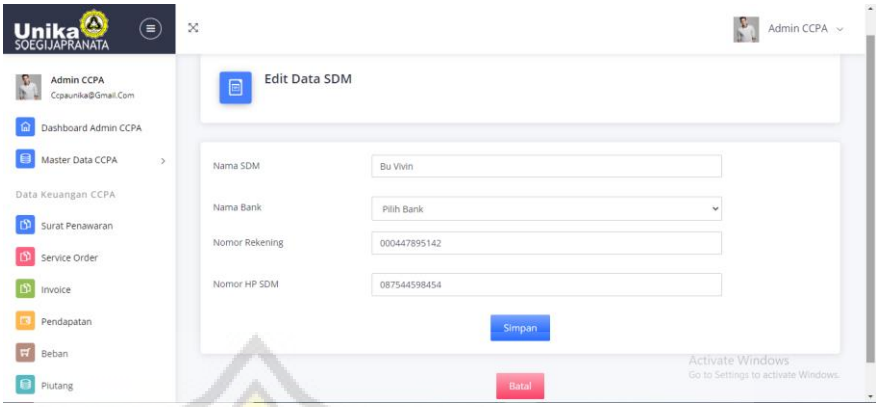
Gambar 4.103 Halaman Tambah Data SDM

Gambar di bawah ini adalah gambar dari *route* dan juga *controller* yang mendukung proses pembuatan data SDM baru dalam sistem ini.

Tabel 4.7 Route dan Controller Tambah Data SDM

No	Gambar	Penjelasan
1.		Route untuk menampilkan halaman input SDM
2.		Route untuk menyimpan data SDM
3.		Controller untuk menampilkan halaman input SDM
4.		Controller untuk menyimpan data SDM

Gambar di bawah ini merupakan hasil implementasi untuk halaman yang bertujuan untuk mengubah data SDM yang sudah ada sebelumnya. Di halaman ini user dapat mengubah nama, nama bank, nomor rekening serta nomor HP dari SDM yang sudah ada.

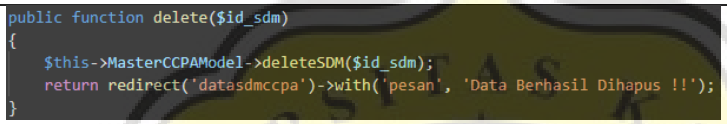


Gambar 4.104 Halaman Edit Data SDM

Gambar di bawah ini adalah gambar dari *route* dan juga *controller* yang mendukung proses pengeditan dan penghapusan data SDM yang sudah ada dalam sistem ini.

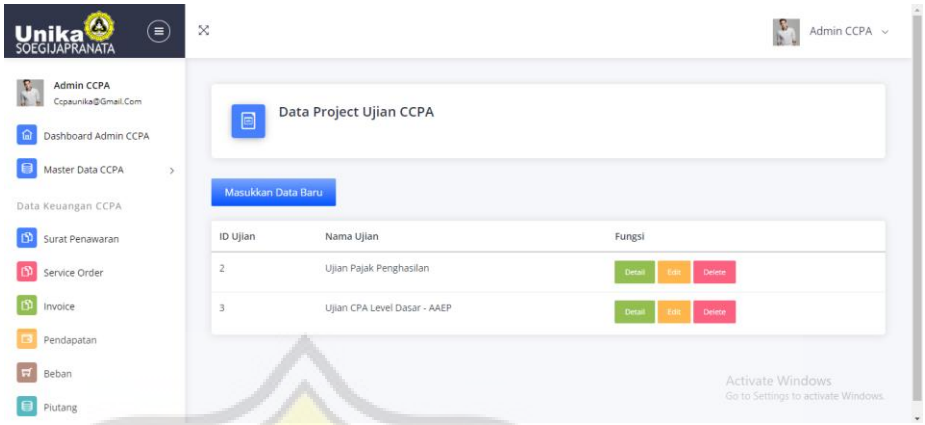
Tabel 4.8 Route dan Controller Edit dan Hapus Data SDM

No	Gambar	Penjelasan
1.	`Route ::get('/updatesdm&&{id_sdm}', [MasterCCPAController::class, 'updatesdm']);`	Route untuk menampilkan halaman edit data SDM
2.	`Route ::post('/updatesdm2&&{id_sdm}', [MasterCCPAController::class, 'updatesdm2']);`	Route untuk menyimpan perubahan data SDM
3.	`Route ::get('/deletesdm&&{id_sdm}', [MasterCCPAController::class, 'delete']);`	Route untuk menghapus data SDM
4.	`public function updatesdm($id_sdm)` `{` `$updatesdm = [` `'updatesdm'=> $this->MasterCCPAModel->UpdateSDM($id_sdm),` `];` `return view('MasterCCPA.updatesdm', $updatesdm);` `}`	Controller untuk menampilkan halaman edit data SDM

5.		Controller untuk menyimpan perubahan data SDM
6.		Controller untuk menghapus data SDM

e. Halaman Master Data Ujian

Gambar di bawah ini merupakan hasil akhir dari implementasi halaman awal untuk master data ujian. Di halaman awal ini terdapat daftar data ujian yang sudah diinputkan sebelumnya. Di halaman ini, user dapat menginputkan data ujian baru, mengedit maupun menghapus data ujian yang sudah ada. User dapat membuat data ujian baru dengan cara menekan tombol tambah data baru yang nantinya akan mengarahkan user ke halaman khusus untuk membuat data ujian baru. Dan apabila user ingin mengubah data dari data ujian yang sudah ada maka user cukup menekan tombol edit sesuai dengan kolom data ujian yang akan diubah. Dan user cukup menekan tombol hapus untuk menghapus suatu data ujian yang sudah ada.



Gambar 4.105 Halaman Master Data Ujian

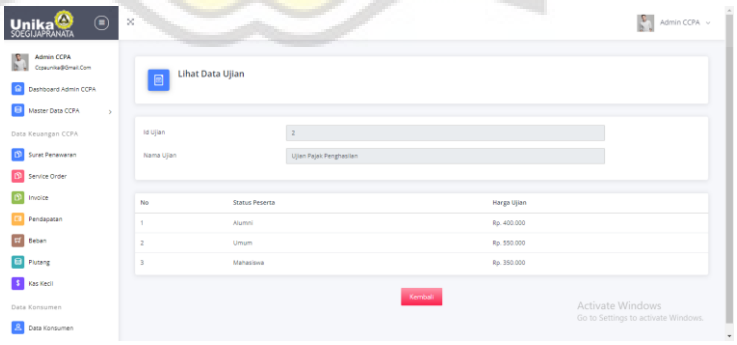
Gambar di bawah ini adalah gambar dari *route* dan juga *controller* yang menunjang berdirinya halaman awal untuk master data ujian dalam sistem ini.

```
Route ::get('/dataujianccpa', [MasterCCPAController::class, 'indexujian']);
```

```
public function indexujian()
{
    $data = [
        'dataujian'=> $this->MasterCCPAModel->allDataUjian(),
    ];
    return view ('MasterCCPA.dataujianccpa', $data);
}
```

Gambar 4.106 Route dan Controller Halaman Master Data Ujian

Gambar di bawah ini merupakan implementasi untuk halaman yang ditujukan untuk menampilkan informasi rinci terkait suatu project. Di halaman ini user dapat melihat daftar status pelanggan dengan tarifnya sesuai dengan status.



Gambar 4.107 Halaman Lihat Data Ujian

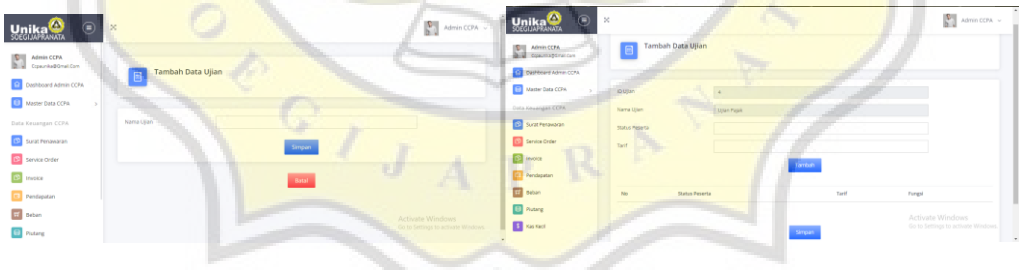
Gambar di bawah ini adalah gambar dari *route* dan juga *controller* yang menunjang berdirinya halaman lihat untuk master data ujian dalam sistem ini.

```
Route ::get('/lihatrinciujian&&{id_ujian}', [MasterCCPAController::class, 'lihatrinciujian']);
```

```
public function lihatrinciujian($id_ujian)
{
    $data = [
        'ujian'=> $this->MasterCCPAModel->DataUjian($id_ujian),
    ];
    $dataujian = [
        'rinci_ujian'=> $this->MasterCCPAModel->DataRinciUjian2($id_ujian),
    ];
    return view ('MasterCCPA.lihatrinciujian', $data, $dataujian);
}
```

Gambar 4.108 Route dan Controller Lihat Data Ujian

Gambar di bawah ini merupakan hasil akhir untuk halaman yang akan digunakan user untuk menambahkan data ujian yang baru. Di halaman ini user diminta untuk menginputkan nama ujian untuk data ujian yang baru. Lalu setelah selesai menginputkan nama ujian maka user diminta untuk menginputkan status pelanggan beserta dengan tarifnya untuk ujian tersebut.

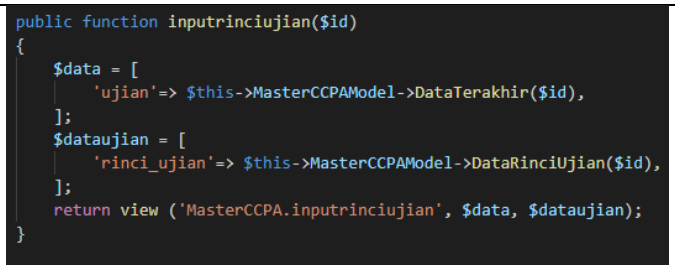


Gambar 4.109 Halaman Tambah Data Ujian

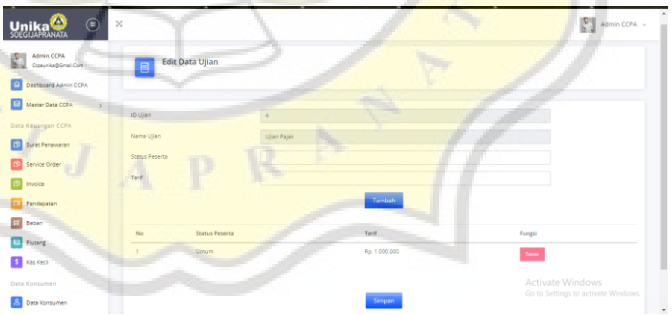
Gambar di bawah ini adalah gambar dari *route* dan juga *controller* yang mendukung proses pembuatan data ujian baru dalam sistem ini.

Tabel 4. 9 Route dan Controller Data Ujian

No	Gambar	Penjelasan
1.	`Route ::get('/inputujian', [MasterCCPAController::class, 'inputujian']);`	Route untuk menampilkan halaman input ujian
2.	`Route ::post('/insertujian', [MasterCCPAController::class, 'insertujian']);`	Route untuk menyimpan data ujian
3.	`Route ::get('/inputrinciujian&&{id_ujian}', [MasterCCPAController::class, 'inputrinciujian']);`	Route untuk menampilkan halaman input detail ujian
4.	`Route ::post('/insertrinciujian', [MasterCCPAController::class, 'insertrinciujian']);`	Route untuk menyimpan data detail ujian
5.	`public function inputujian()` `{` `  return view('MasterCCPA.inputujian');` `}`	Controller untuk menampilkan halaman input ujian
6.	`public function insertujian()` `{` `  Request()->validate([` `  'nama_ujian' => 'required|max:255',` `  ],[` `  'nama_ujian.required'=>'Nama Akun wajib diisi !!',` `  ]);` `  $dataujian = [` `  'nama_ujian' => Request()->nama_ujian,` `  ];` `  $this->MasterCCPAModel->addDataUjian($dataujian);` `  $apa = [` `  'datateraakhir'=> $this->MasterCCPAModel->DataIDTerakhir(),` `  ];` `  $id = ($apa['datateraakhir']->id_ujian);` `  return redirect('inputrinciujian&&'.$id);` `}`	Controller untuk menyimpan data ujian

7.		Controller untuk menampilkan halaman input detail ujian
8.		Controller untuk menyimpan data detail ujian

Gambar di bawah ini merupakan hasil implementasi untuk halaman yang bertujuan untuk mengubah data ujian yang sudah ada sebelumnya. Di halaman ini user dapat mengubah nama ujian serta mengubah status pelanggan dan tarif berdasarkan status pelanggan.

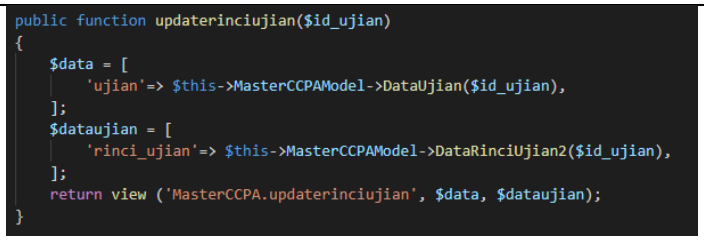
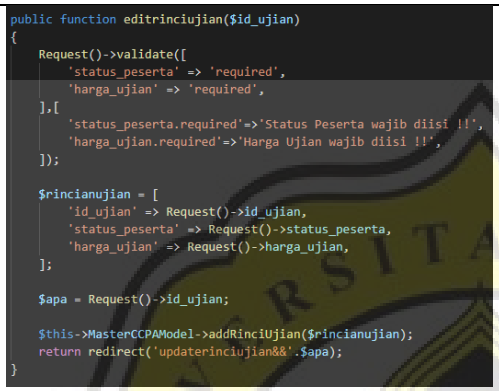
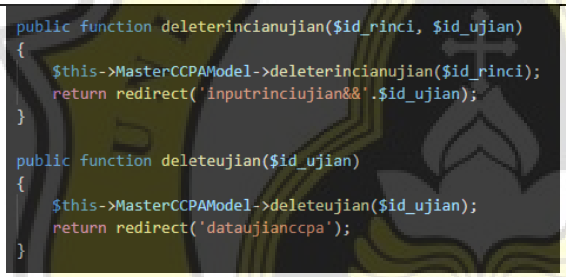


Gambar 4.110 Halaman Edit Data Ujian

Gambar di bawah ini adalah gambar dari *route* dan juga *controller* yang mendukung proses pengeditan dan penghapusan data ujian yang sudah ada dalam sistem ini.

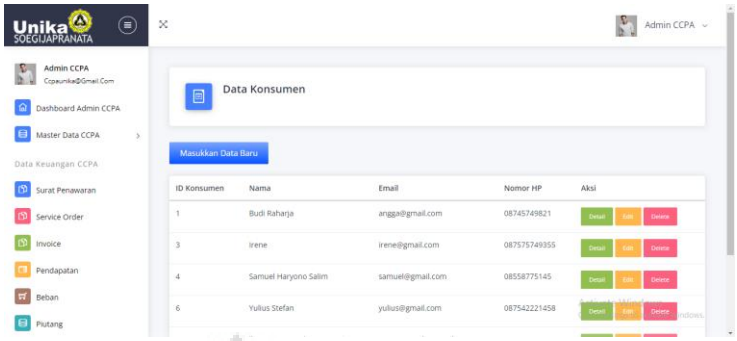
Tabel 4.10 Route dan Controler Edit dan Hapus Data Ujian

No	Gambar	Penjelasan
1.	`Route ::get('/updateujian&&{id_ujian}', [MasterCCPAController::class, 'updateujian']);`	Route untuk menampilkan halaman edit ujian
2.	`Route ::post('/editujian&&{id_ujian}', [MasterCCPAController::class, 'editujian']);`	Route untuk menyimpan perubahan data ujian
3.	`Route ::get('/updaterinciujian&&{id_ujian}', [MasterCCPAController::class, 'updaterinciujian']);`	Route untuk menampilkan halaman edit rincian ujian
4.	`Route ::post('/editrinciujian&&{id_ujian}', [MasterCCPAController::class, 'editrinciujian']);`	Route untuk menyimpan perubahan data rincian ujian
5.	`Route ::get('/deleteujian&&{id_ujian}', [MasterCCPAController::class, 'deleteujian']);`	Route untuk menghapus data ujian
6.	`public function updateujian($id_ujian)` `{` `  $data = [` `  'ujian'=> $this->MasterCCPAModel->DataUjian($id_ujian),` `  ];` `  return view('MasterCCPA.updateujian', $data);` `}`	Controller untuk menampilkan halaman edit ujian
7.	`public function editujian($id_ujian)` `{` `  Request()->validate([` `  'nama_ujian' => 'required|max:255',` `  ],[` `  'nama_ujian.required'=>'Nama wajib diisi !!',` `  ]);` `  $data = [` `  'nama_ujian' => Request()->nama_ujian,` `  ];` `  $this->MasterCCPAModel->editujian($id_ujian, $data);` `  return redirect('updaterinciujian&&'.$id_ujian);` `}`	Controller untuk menyimpan perubahan data ujian

8.		Controller untuk menampilkan halaman edit rincian ujian
9.		Controller untuk menyimpan perubahan data rincian ujian
10.		Controller untuk menghapus data ujian

f. Halaman Master Data Pelanggan

Gambar di bawah ini merupakan implementasi untuk halaman awal untuk master data pelanggan. Di halaman ini user dapat melihat daftar data pelanggan yang telah diinputkan sebelumnya. Melalui halaman ini, user dapat mulai menambahkan data pelanggan baru, melihat rincian data, mengedit ataupun menghapus data pelanggan yang sudah ada.



Gambar 4.111 Halaman Master Data Pelanggan

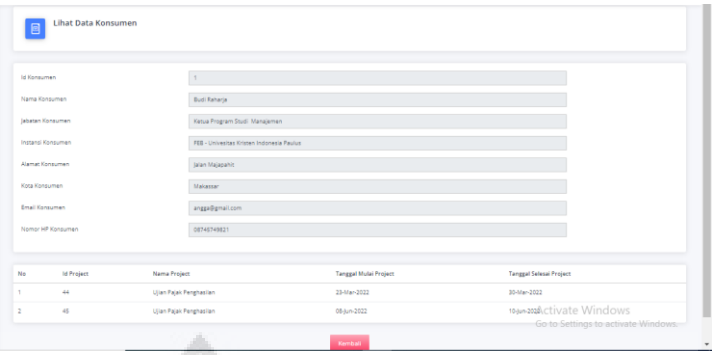
Gambar di bawah ini adalah gambar dari *route* dan juga *controller* yang menunjang berdirinya halaman awal untuk master data pelanggan dalam sistem ini.

```
//Route Konsumen
Route ::get('/datakonsumen', [KonsumenController::class, 'index']);
```

```
public function index()
{
    $data = [
        'konsumen'=> $this->KonsumenModel->allData(),
    ];
    return view ('konsumen.datakonsumen', $data);
}
```

Gambar 4.112 Route dan Controller Master Data Pelanggan

Gambar di bawah ini merupakan implementasi untuk halaman yang ditujukan untuk menampilkan informasi rinci terkait dengan seorang pelanggan. Di halaman ini user dapat melihat id pelanggan, nama pelanggan, jabatan, instansi tempat bekerja, alamat, kota, *email*, nomor hp, serta daftar project-project yang pernah diikuti.



Gambar 4.113 Halaman Lihat Data Pelanggan

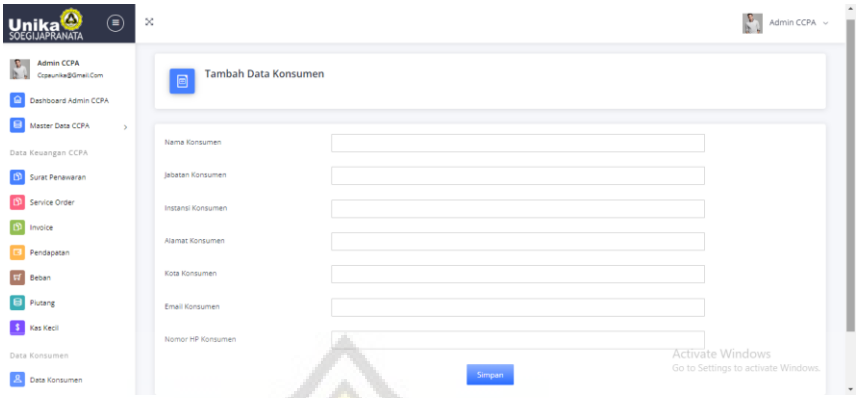
Gambar di bawah ini adalah gambar dari *route* dan juga *controller* yang menunjang berdirinya halaman lihat untuk master data pelanggan dalam sistem ini.

```
Route ::get('/lihatkonsumen&&{id_konsumen}', [KonsumenController::class, 'lihat']);
```

```
public function lihat($id_konsumen)
{
    $lihat = [
        'lihat'=> $this->KonsumenModel->LihatKonsumen($id_konsumen),
        'datarincian'=> $this->KonsumenModel->DataRincian($id_konsumen),
    ];
    return view('Konsumen.lihatkonsumen', $lihat);
}
```

Gambar 4.114 Route dan Controller Lihat Data Pelanggan

Gambar di bawah ini merupakan hasil implementasi akhir untuk halaman tambah data pelanggan baru. Halaman ini akan muncul setelah user menekan tombol tambah data pelanggan. Di halaman ini user diminta untuk menginputkan nama pelanggan, nama pelanggan, jabatan, instansi tempat bekerja, alamat, kota, *email*, dan nomor hp.



Gambar 4.115 Halaman Tambah Data Pelanggan

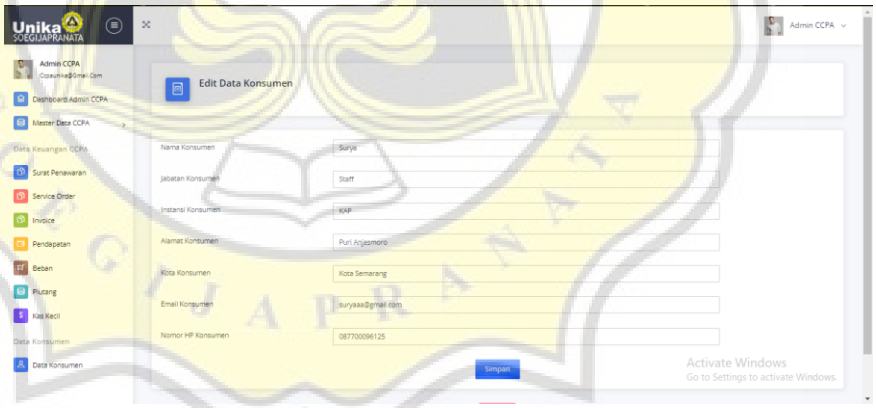
Gambar di bawah ini adalah gambar dari *route* dan juga *controller* yang mendukung proses pembuatan data pelanggan baru dalam sistem ini.

Tabel 4.11 Route dan Controller Tambah Data Pelanggan

No	Gambar	Penjelasn
1.	`Route ::get('/inputkonsumen', [KonsumenController::class, 'input']);`	Route untuk menampilkan halaman input pelanggan
2.	`Route ::post('/insertkonsumen', [KonsumenController::class, 'insert']);`	Route untuk menyimpan data pelanggan
3.	`public function input()` `{` `  return view('Konsumen.inputkonsumen');` `}`	Controller untuk menampilkan halaman input pelanggan

4.		Controller untuk menyimpan data pelanggan

Gambar di bawah ini merupakan hasil implementasi untuk halaman yang bertujuan untuk mengubah data pelanggan yang sudah ada. Halaman ini akan muncul setelah user menekan tombol edit di salah satu kolom data pelanggan yang sudah dipilih. Di halaman ini user dapat mengubah nama pelanggan, nama pelanggan, jabatan, instansi tempat bekerja, alamat, kota, *email*, dan nomor hp.



Gambar 4.116 Halaman Edit Data Pelanggan

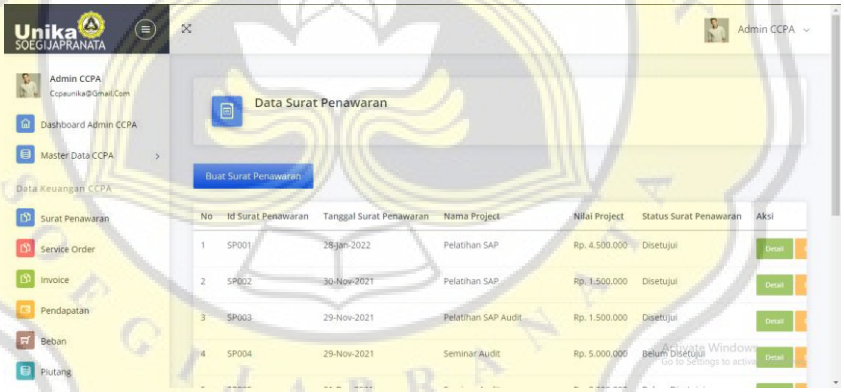
Gambar di bawah ini adalah gambar dari *route* dan juga *controller* yang mendukung proses pengeditan dan penghapusan data pelanggan yang sudah ada dalam sistem ini.

Tabel 4. 12 Route dan Controller Edit dan Hapus Data Pelanggan

No	Gambar	Penjelasan
1.	`Route ::get('/updatekonsumen&&{id_konsumen}', [KonsumenController::class, 'update']);`	Route untuk menampilkan halaman edit data pelanggan
2.	`Route ::post('/updatekonsumen2&&{id_konsumen}', [KonsumenController::class, 'update2']);`	Route untuk menyimpan perubahan data pelanggan
3.	`Route ::get('/deletekonsumen&&{id_konsumen}', [KonsumenController::class, 'delete']);`	Route untuk menghapus data pelanggan
4.	`public function update($id_konsumen)` `{` `$updatekonsumen = [` `'updatekonsumen'=> $this->KonsumenModel->UpdateKonsumen($id_konsumen),` `];` `return view('Konsumen.updatekonsumen', $updatekonsumen);` `}`	Controller untuk menampilkan halaman edit data pelanggan
5.	`public function update2($id_konsumen)` `{` `Request()->validate([` `'nama_konsumen' => 'required\|max:255',` `'jabatan_konsumen' => 'required',` `'instansi_konsumen' => 'required',` `'alamat_konsumen' => 'required',` `'kota_konsumen' => 'required',` `'email_konsumen' => 'required',` `'nomorhp_konsumen' => 'required',` `],[` `'nama_konsumen.required'=>'Nama Konsumen wajib diisi !!',` `'jabatan_konsumen.required'=>'Jabatan Konsumen wajib diisi !!',` `'instansi_konsumen.required'=>'Instansi Konsumen wajib diisi !!',` `'alamat_konsumen.required'=>'Alamat Konsumen wajib diisi !!',` `'kota_konsumen.required'=>'Kota Konsumen wajib diisi !!',` `'email_konsumen.required'=>'Email Konsumen wajib diisi !!',` `'nomorhp_konsumen.required'=>'Nomor Hp Konsumen wajib diisi !!',` `]);` `$data = [` `'nama_konsumen' => Request()->nama_konsumen,` `'jabatan_konsumen' => Request()->jabatan_konsumen,` `'instansi_konsumen' => Request()->instansi_konsumen,` `'alamat_konsumen' => Request()->alamat_konsumen,` `'kota_konsumen' => Request()->kota_konsumen,` `'email_konsumen' => Request()->email_konsumen,` `'nomorhp_konsumen' => Request()->nomorhp_konsumen,` `];` `$this->KonsumenModel->updateData($id_konsumen, $data);`	Controller untuk menyimpan perubahan data pelanggan
6.	`public function delete($id_konsumen)` `{` `$this->KonsumenModel->deleteData($id_konsumen);` `return redirect('datakonsumen')->with('pesan', 'Data Berhasil Dihapus !!');` `}`	Controller untuk menghapus data pelanggan

g. Halaman Surat Penawaran

Gambar di bawah ini merupakan hasil implementasi untuk halaman awal fitur surat penawaran, Di halaman awal ini, dapat melihat daftar surat penawaran yang sudah dibuat sebelumnya. Melalui halaman ini, user dapat mulai membuat data surat penawaran baru dengan cara menekan tombol buat surat penawaran. Tidak hanya itu saja di melalui halaman ini pula, apabila user menekan tombol detail yang ada di dalam tabel daftar surat penawaran maka akan mengarah kan user ke halaman khusus yang berisi data rinci tentang surat penawaran yang dipilih, namun apabila user menekan tombol edit maka user akan diarahkan ke halaman untuk mengubah data surat penawaran yang dipilih dan apabila user menekan tombol delete maka user dapat menghapus data surat penawaran yang dipilih.



Gambar 4.117 Halaman Surat Penawaran

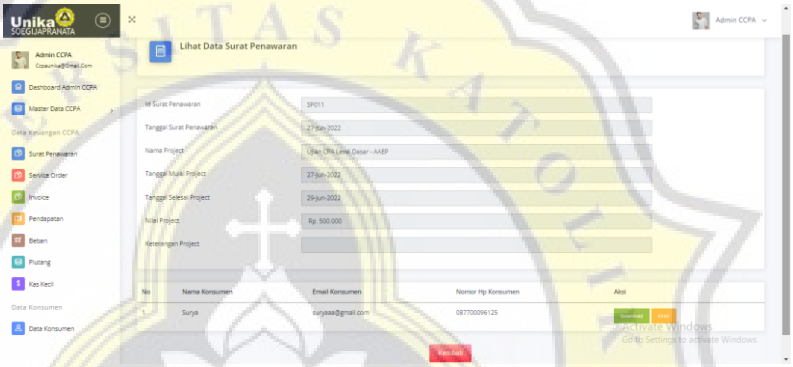
Gambar di bawah ini adalah gambar dari *route* dan juga *controller* yang menunjang berdirinya halaman awal untuk fitur surat penawaran dalam sistem ini.

```
Route ::get('/suratpenawaranccpa', [DokumenController::class, 'index']);
```

```
public function index()
{
    $data = [
        'suratpenawaran'=> $this->DokumenModel->allData(),
    ];
    return view ('DokumenCCPA.indexsp', $data);
}
```

Gambar 4.118 Route dan Controller Halaman Surat Penawaran

Gambar di bawah ini merupakan implementasi untuk halaman lihat surat penawaran. Di halaman ini user dapat melihat data detail dari suatu surat penawaran. Di halaman ini tersedia informasi tentang id surat penawaran, tanggal surat penawaran, nama project, tanggal mulai dan selesai project, nilai project, keterangan surat penawaran serta daftar pelanggan yang terlibat dalam project tersebut.



Gambar 4.119 Halaman Lihat Surat Penawaran

Gambar di bawah ini merupakan *route* dan juga *controller* yang mendukung berjalannya halaman lihat surat penawaran.

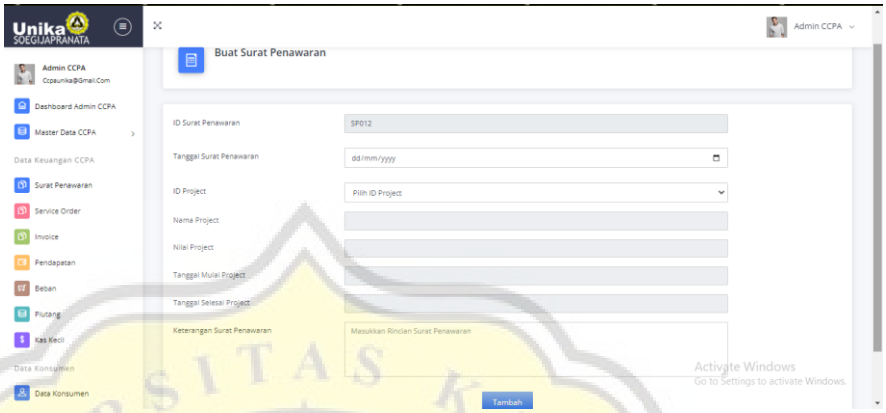
```
Route ::get('/lihatsuratpenawaran&&{id_suratpenawaran}', [DokumenController::class, 'lihat']);
```

```
public function lihat($id_suratpenawaran)
{
    $lihat = [
        'lihatsp'=> $this->DokumenModel->LihatSP($id_suratpenawaran),
    ];
    $konsumen = [
        'konsumen'=> $this->DokumenModel->LihatKonsumen($id_suratpenawaran),
    ];
    //DD($lihat);
    return view ('DokumenCCPA.lihatsp', $lihat, $konsumen);
}
```

Gambar 4.120 Route dan Controller Lihat Surat Penawaran

Gambar di bawah ini merupakan hasil akhir dari halaman untuk membuat surat penawaran baru. Untuk membuat suatu surat penawaran baru, user diharuskan untuk mengisi tanggal surat penawaran, memilih id project yang akan dibuatkan surat

penawarannya dan keterangan surat penawaran bila ada. Untuk Id surat penawaran akan dibuatkan oleh sistem secara otomatis.

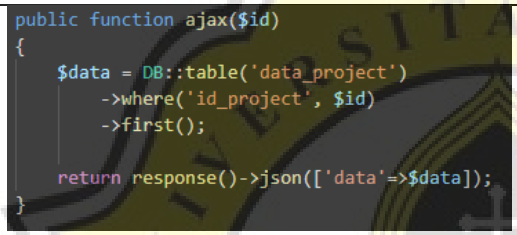
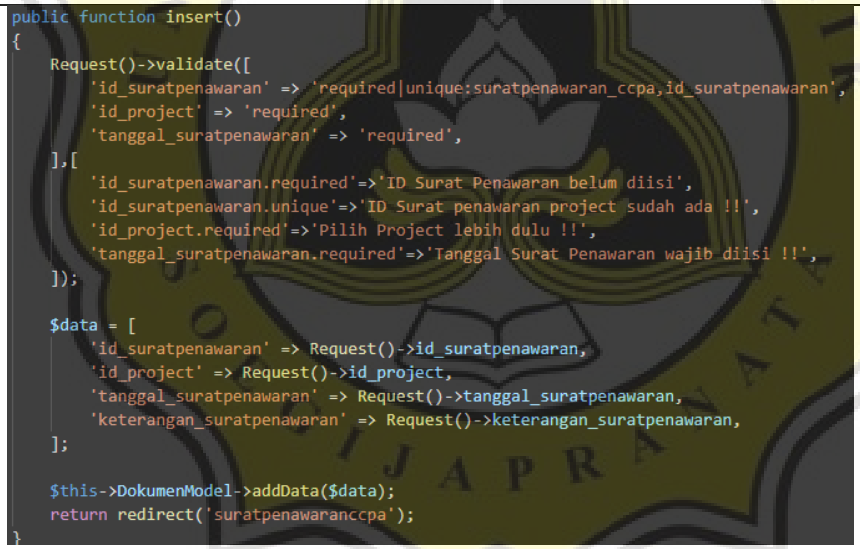


Gambar 4.121 Halaman Tambah Surat Penawaran

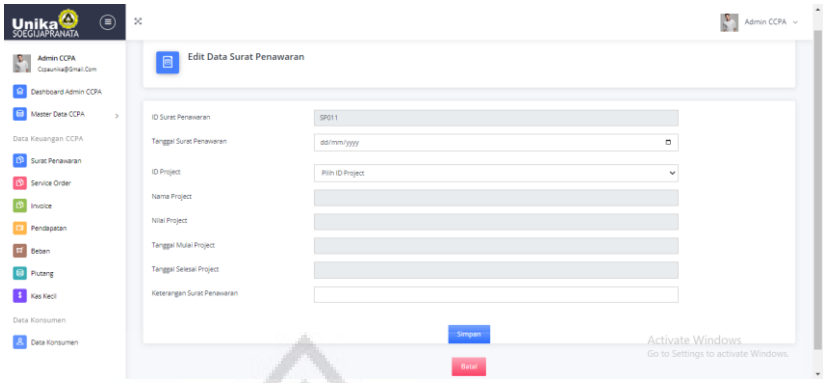
Gambar di bawah ini adalah gambar dari *route* dan juga *controller* yang mendukung proses pembuatan surat penawaran baru dalam sistem ini.

Tabel 4.13 Route dan Controller Tambah Surat Penawaran

No	Gambar	Penjelasan
1.	`Route ::get('/inputsp', [DokumenController::class, 'input']);`	Route untuk menampilkan halaman input SP
2.	`Route ::get('/inputsp/ajax/{id}', [DokumenController::class, 'ajax']);`	Route untuk memproses ajax
3.	`Route ::post('/insertsp', [DokumenController::class, 'insert']);`	Route untuk menyimpan data SP

4.		Controller untuk menampilkan halaman input SP
5.		Controller untuk memproses ajax
6.		Controller untuk menyimpan data SP

Gambar di bawah ini adalah implementasi untuk halaman edit surat penawaran. Di halaman ini, user dapat mengubah data surat penawaran apabila terjadi kesalahan input.

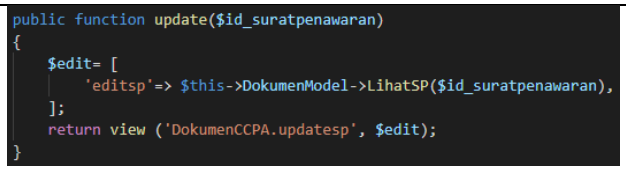
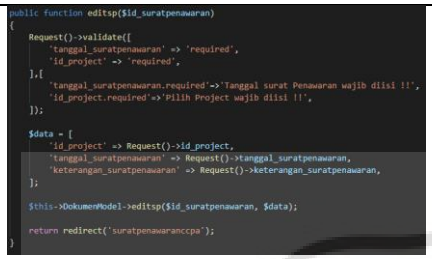
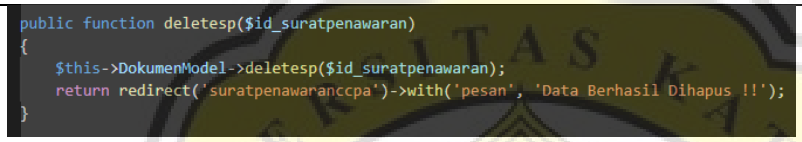


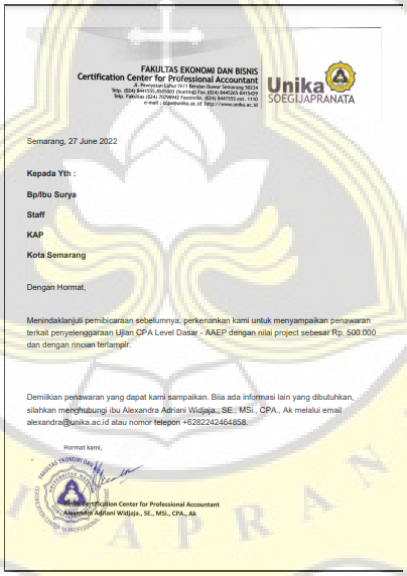
Gambar 4.122 Halaman Edit Surat Penawaran

Gambar di bawah ini adalah gambar dari *route* dan juga *controller* yang mendukung proses pengeditan, penghapusan, dan pengunduhan data surat penawaran yang sudah ada dalam sistem ini.

Tabel 4.14 Route dan Controller Edit dan Hapus Surat Penawaran

No	Gambar	Penjelasan
1.	`Route ::get('/downloadsuratpenawaran&&{id_suratpenawaran}&&{id_konsumen}', [DokumenController::class, 'downloac`	Route untuk mengunduh SP
2.	`Route ::get('/updatesuratpenawaran&&{id_suratpenawaran}', [DokumenController::class, 'update']);`	Route untuk menampilkan halaman edit SP
3.	`Route ::post('/editsp&&{id_suratpenawaran}', [DokumenController::class, 'editsp']);`	Route untuk menyimpan perubahan pada data SP
4.	`Route ::get('/deletesuratpenawaran&&{id_suratpenawaran}', [DokumenController::class, 'deletesp']);`	Route untuk menghapus SP
5.	`//DOWNLOAD DOKUMEN` `public function downloadsp($id_suratpenawaran, $id_konsumen)` `{` `$data = [` `'suratpenawaran'=> $this->DokumenModel->DownloadSP($id_suratpenawaran, $id_konsumen),` `'konsumen'=> $this->DokumenModel->konsumendownloadsp($id_konsumen),` `];` `$id = ($data['suratpenawaran']);` `$id2 = ($data['konsumen']);` `$suratpenawaran = $id[0];` `$konsumen = $id2[0];` `//DD ($suratpenawaran, $konsumen);` `//return view ('suratpenawaran_pdf', ['suratpenawaran' => $suratpenawaran]);` `$pdf = PDF::loadview('suratpenawaran_pdf',['suratpenawaran'=>$id[0], 'konsumen'=>$id2[0]]);` `return $pdf->stream();` `}`	Controller untuk mengunduh SP

6.		Controller untuk menampilkan halaman edit SP
7.		Controller untuk menyimpan perubahan pada data SP
8.		Controller untuk menghapus SP


FAKULTAS EKONOMI DAN BISNIS
Certification Center for Professional Accountant
Unika SOEGIJAPRANATA

Semarang, 27 June 2022

Kepada Yth :
Bp/Ibu Surya
Staff
KAP
Kota Semarang

Dengan Hormat,

Menindaklanjuti pembicaraan sebelumnya, perkenankan kami untuk menyampaikan penawaran terkait penyelenggaraan Ujian CPA Level Dasar - AAEP dengan nilai project sebesar Rp. 500.000 dan dengan rincian terlampir.

Demikian penawaran yang dapat kami sampaikan. Bila ada informasi lain yang dibutuhkan, silahkan menghubungi ibu Alexandra Adriani Widjaja., SE., MSi., CPA., Ak melalui email alexandra@unika.ac.id atau nomor telepon +6285224484858.

Hormat kami,

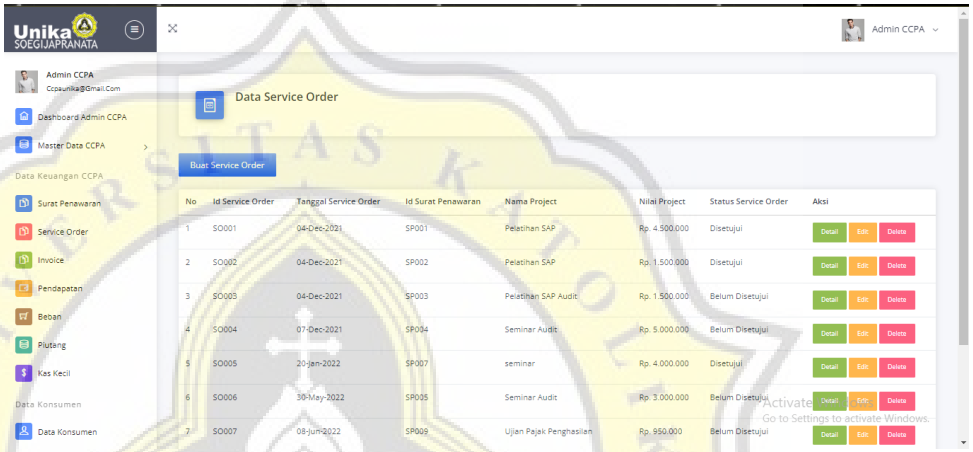
Certification Center for Professional Accountant
Alexandra Adriani Widjaja., SE., MSi., CPA., Ak

Gambar 4.123 Contoh Output Surat Penawaran

h. Halaman *Service Order*

Gambar di bawah ini merupakan hasil implementasi untuk halaman awal fitur *service order*. Di halaman awal ini, dapat melihat daftar *service order* yang sudah dibuat sebelumnya. Melalui halaman ini, user dapat mulai membuat data *service order* baru dengan cara menekan tombol buat *service order*. Tidak hanya itu saja di melalui halaman ini pula, apabila user menekan tombol detail

yang ada di dalam tabel daftar *service order* maka akan mengarah kan user ke halaman khusus yang berisi data rinci tentang *service order* yang dipilih, namun apabila user menekan tombol edit maka user akan diarahkan ke halaman untuk mengubah data *service order* yang dipilih dan apabila user menekan tombol delete maka user dapat menghapus data *service order* yang dipilih.



Gambar 4.124 Halaman Service Order

Gambar di bawah ini adalah gambar dari *route* dan juga *controller* yang menunjang berdirinya halaman awal untuk fitur *service order* dalam sistem ini.

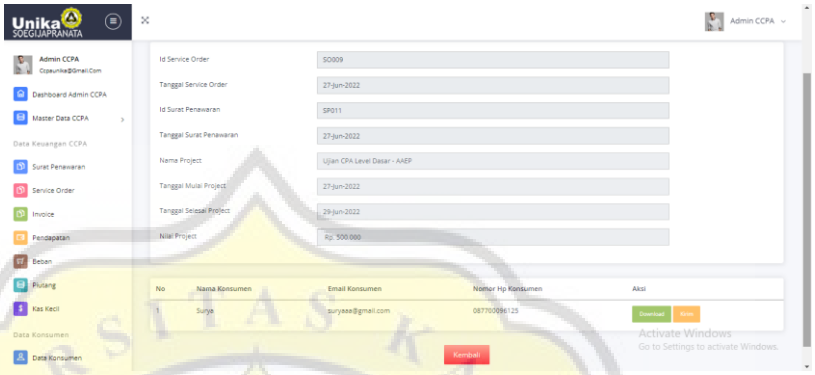
```
Route ::get('/serviceorderccpa', [DokumenController::class, 'indexso']);
```

```
public function indexso()
{
    $data = [
        'serviceorder'=> $this->DokumenModel->dataso(),
    ];
    return view ('DokumenCCPA.indexso', $data);
}
```

Gambar 4.125 Route dan Controller Halaman Service Order

Gambar di bawah ini merupakan implementasi untuk halaman lihat *service order*. Di halaman ini user dapat melihat data detail dari suatu *service order*. Di halaman ini tersedia informasi tentang id *service order*, tanggal *service order*, Id surat penawaran,

nama project, tanggal mulai dan selesai project, nilai project, keterangan *service order* serta daftar pelanggan yang terlibat dalam project tersebut.



Gambar 4.126 Halaman Lihat Service Order

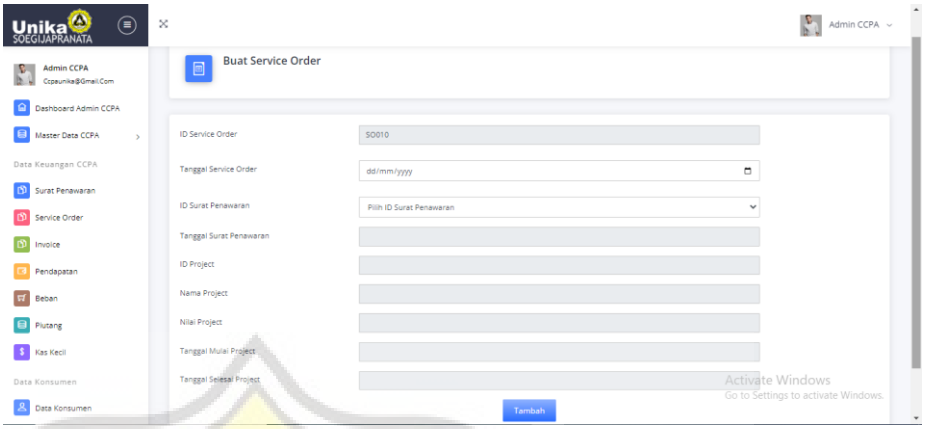
Gambar di bawah ini merupakan *route* dan juga *controller* yang mendukung berjalannya halaman lihat *service order*.

```
Route ::get('/lihatserviceorder&&{id_serviceorder}', [DokumenController::class, 'lihatso']);
```

```
public function lihatso($id_serviceorder)
{
    $lihat = [
        'lihatso'=> $this->DokumenModel->LihatSO($id_serviceorder),
    ];
    $konsumen = [
        'konsumen'=> $this->DokumenModel->LihatKonsumenSO($id_serviceorder),
    ];
    //DD($lihat);
    return view ('DokumenCCPA.lihatso', $lihat, $konsumen);
}
```

Gambar 4.127 Route dan Controller Lihat Service Order

Gambar di bawah ini merupakan hasil akhir dari halaman untuk membuat *service order* baru. Untuk membuat suatu *service order* baru, user diharuskan untuk mengisi tanggal *service order*, memilih surat penawaran yang akan dibuatkan *service order* nya dan keterangan *service order* bila ada. Untuk Id *service order* akan dibuatkan oleh sistem secara otomatis.



Gambar 4.128 Halaman Tambah Service Order

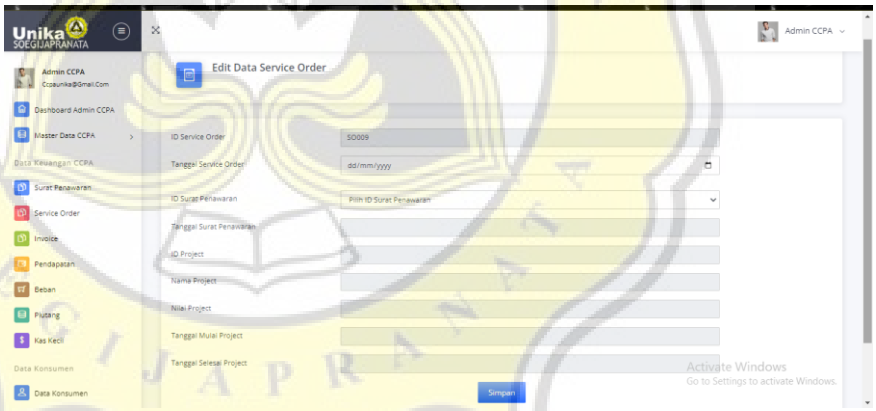
Gambar di bawah ini adalah gambar dari *route* dan juga *controller* yang mendukung proses pembuatan *service order* baru dalam sistem ini.

Tabel 4.15 Route dan Controller Tambah Service Order

No	Gambar	Penjelasan
1.	`Route ::get('/inputso', [DokumenController::class, 'inputso']);`	Route untuk menampilkan halaman input SO
2.	`Route ::get('/inputso/ajax/{id}', [DokumenController::class, 'ajaxso']);`	Route untuk memproses ajax SO
3.	`Route ::post('/insertso', [DokumenController::class, 'insertso']);`	Route untuk menyimpan data SO
4.	`public function inputso()` `{` `$data = [` `'suratpenawaran'=>$this->DokumenModel->datasp(),` `'idterakhir'=>$this->DokumenModel->idso(),` `];` `$ambilid = ($data['idterakhir']);` `if($ambilid == null){` `$id = 'SO001';` `}` `else{` `$id = (($data['idterakhir'])->id_serviceorder);` `$id = substr($id,2);` `$id = (int)$id + 1;` `$id = sprintf("%03d",$id);` `$id = 'SO'.$id;` `}` `$apa = [` `'id'=>$id,` `];` `return view ('DokumenCCPA.inputso', $data, $apa);` `}`	Controller untuk menampilkan halaman input SO

5.		Controller untuk memproses ajax SO
6.		Controller untuk menyimpan data SO

Gambar di bawah ini adalah implementasi untuk halaman edit *service order*. Di halaman ini, user dapat mengubah data *service order* apabila terjadi kesalahan input.



Gambar 4.129 Halaman Edit Service Order

Gambar di bawah ini adalah gambar dari *route* dan juga *controller* yang mendukung proses pengeditan, penghapusan, dan pengunduhan data *service order* yang sudah ada dalam sistem ini.

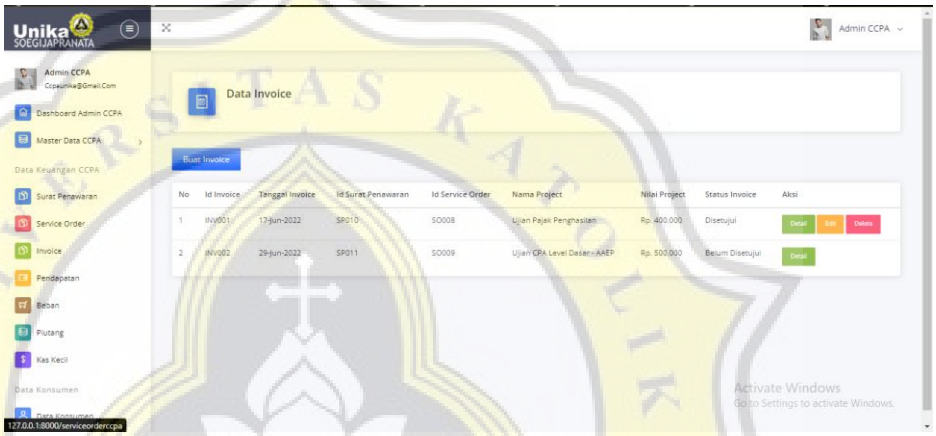
Tabel 4.16 Route dan Controller Edit dan Hapus Service Order

No	Gambar	Penjelasan
1.	`Route ::get('/updateserviceorder&&{id_serviceorder}', [DokumenController::class, 'updateso']);`	Route untuk menampilkan halaman edit SO
2.	`Route ::post('/editso&&{id_serviceorder}', [DokumenController::class, 'editso']);`	Route untuk menyimpan perubahan data SO
3.	`Route ::get('/deleteserviceorder&&{id_serviceorder}', [DokumenController::class, 'deleteso']);`	Route untuk menghapus data SO
4.	`public function updateso($id_serviceorder) { $edit= [ 'editso'=> $this->DokumenModel->LihatSO($id_serviceorder), ]; return view ('DokumenCCPA.updateso', $edit); }`	Controller untuk menampilkan halaman edit SO
5.	`public function editso($id_serviceorder) { Request()->validate(['tanggal_serviceorder' => 'required', 'id_suratpenawaran' => 'required',],['tanggal_serviceorder.required'=>'Tanggal Service Order wajib diisi !!', 'id_suratpenawaran.required'=>'Pilih Surat Penawaran wajib diisi !!',]); $data = ['id_suratpenawaran' => Request()->id_suratpenawaran, 'tanggal_serviceorder' => Request()->tanggal_serviceorder,]; $this->DokumenModel->editso($id_serviceorder, $data); return redirect('serviceorderccpa'); }`	Controller untuk menyimpan perubahan data SO
6.	`public function deleteso($id_serviceorder) { $this->DokumenModel->deleteso($id_serviceorder); return redirect('serviceorderccpa')->with('pesan', 'Data Berhasil Dihapus !!'); }`	Controller untuk menghapus data SO

i. Halaman *Invoice*

Gambar di bawah ini merupakan hasil implementasi untuk halaman awal fitur *invoice.* Di halaman awal ini, dapat melihat daftar *invoice* yang sudah dibuat sebelumnya. Melalui halaman ini, user dapat mulai membuat data *invoice* baru dengan cara menekan

tombol buat *invoice*. Tidak hanya itu saja di melalui halaman ini pula, apabila user menekan tombol detail yang ada di dalam tabel daftar *invoice* maka akan mengarah kan user ke halaman khusus yang berisi data rinci tentang *invoice* yang dipilih, namun apabila user menekan tombol edit maka user akan diarahkan ke halaman untuk mengubah data *invoice* yang dipilih dan apabila user menekan tombol delete maka user dapat menghapus data *invoice* yang dipilih.



Gambar 4.130 Halaman Invoice

Gambar di bawah ini adalah gambar dari *route* dan juga *controller* yang menunjang berdirinya halaman awal untuk fitur *invoice* dalam sistem ini.

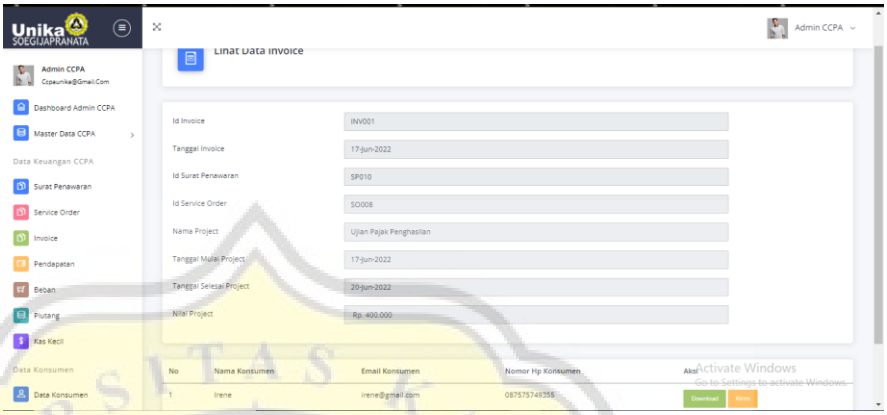
```
Route ::get('/invoiceccpa', [DokumenController::class, 'indexinv']);
```

```
public function indexinv()
{
    $data = [
        'invoice'=> $this->DokumenModel->datainv(),
    ];
    return view ('DokumenCCPA.indexinv', $data);
}
```

Gambar 4.131 Route dan Controller Halaman Invoice

Gambar di bawah ini merupakan implementasi untuk halaman lihat *invoice*. Di halaman ini user dapat melihat data detail dari suatu *invoice*. Di halaman ini tersedia informasi tentang id *invoice*, tanggal *invoice*, Id *service order*, nama project, tanggal

mulai dan selesai project, nilai project, keterangan *invoice* serta daftar pelanggan yang terlibat dalam project tersebut.



Gambar 4.132 Halaman Lihat Invoice

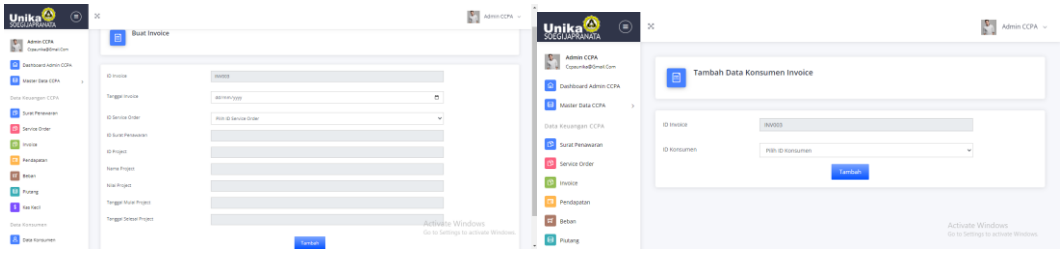
Gambar di bawah ini merupakan *route* dan juga *controller* yang mendukung berjalannya halaman lihat *invoice.*

```
Route ::get('/lihatinvoice&&{id_invoice}', [DokumenController::class, 'lihatinv']);
```

```
public function lihatinv($id_invoice)
{
    $lihat = [
        'lihatinv'=> $this->DokumenModel->LihatInv($id_invoice),
    ];
    $konsumen = [
        'konsumen'=> $this->DokumenModel->LihatKonsumenInv($id_invoice),
    ];
    //DD($lihat);
    return view ('DokumenCCPA.lihatinv', $lihat, $konsumen);
}
```

Gambar 4.133 Route dan Controller Lihat Invoice

Gambar di bawah ini merupakan hasil akhir dari halaman untuk membuat *invoice* baru. Proses pembuatan *invoice* baru terbagi menjadi dua bagian. Di bagian pertama, user diharuskan untuk mengisi tanggal *invoice*, memilih s*ervice order* yang akan dibuatkan *invoice* nya dan keterangan *invoice* bila ada. Untuk Id *invoice* akan dibuatkan oleh sistem secara otomatis. Lalu di bagian kedua user diminta untuk memilih kepada pelanggan siapa *invoice* tersebut dibuat.

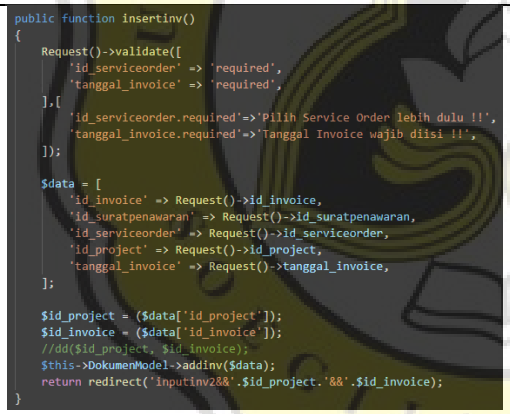
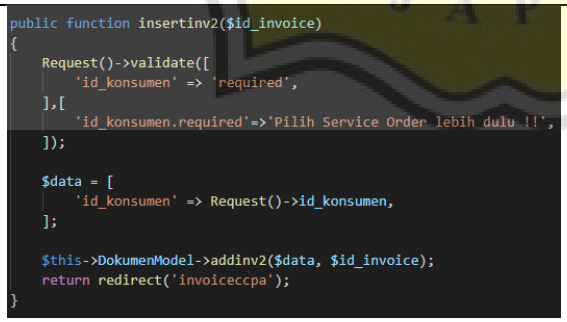


Gambar 4.134 Halaman Tambah Invoice

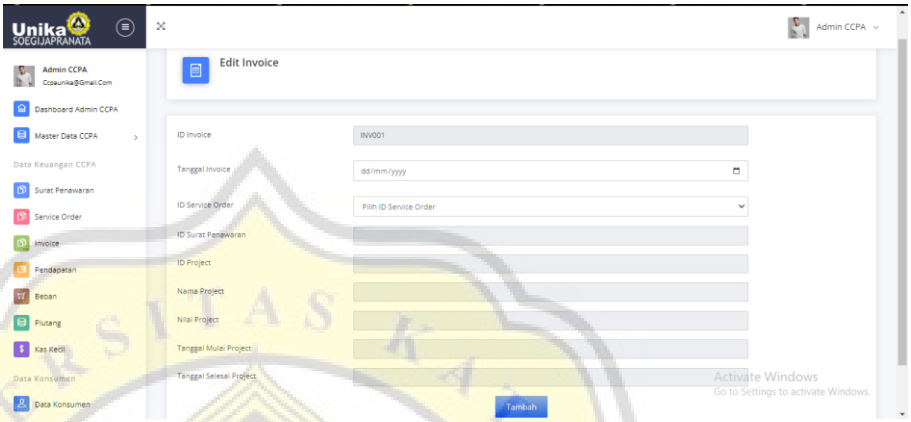
Gambar di bawah ini adalah gambar dari *route* dan juga *controller* yang mendukung proses pembuatan *invoice* baru dalam sistem ini.

Tabel 4.17 Route dan Controller Tambah Invoice

No	Gambar	Penjelasan
1.	`Route ::get('/inputinv', [DokumenController::class, 'inputinv']);`	Route untuk menampilkan halaman input Inv
2.	`Route ::get('/inputinv2&&{id_project}&&{id_invoice}', [DokumenController::class, 'inputinv2']);`	Route untuk menampilkan halaman input detail Inv
3.	`Route ::get('/inputinv/ajax/{id}', [DokumenController::class, 'ajaxinv']);`	Route untuk memproses ajax Inv
4.	`Route ::post('/insertinv', [DokumenController::class, 'insertinv']);`	Route untuk menyimpan data Inv

5.		Controller untuk menampilkan halaman input Inv
6.		Controller untuk menampilkan halaman input detail Inv
7.		Controller untuk memproses ajax Inv
8.		Controller untuk menyimpan data Inv

Gambar di bawah ini adalah implementasi untuk halaman edit *invoice*. Di halaman ini, user dapat mengubah data *invoice* apabila terjadi kesalahan input.

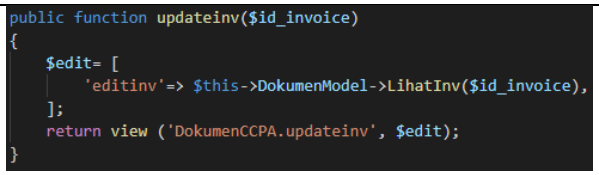
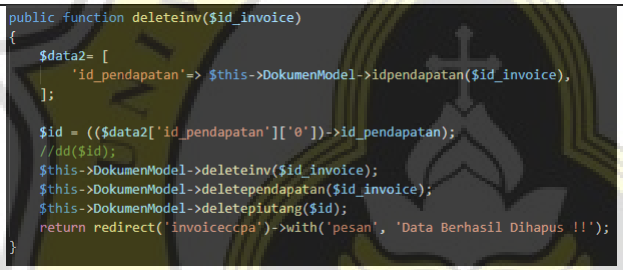


Gambar 4.135 Halaman Edit Invoice

Gambar di bawah ini adalah gambar dari *route* dan juga *controller* yang mendukung proses pengeditan, penghapusan, dan pengunduhan data *invoice* yang sudah ada dalam sistem ini.

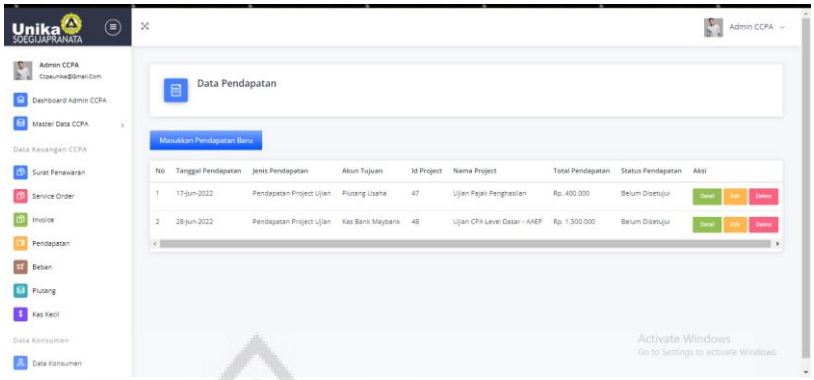
Tabel 4.18 Route dan Controller Edit dan Hapus Invoice

No	Gambar	Penjelasan
1.	`Route ::get('/updateinvoice&&{id_invoice}', [DokumenController::class, 'updateinv']);`	Route untuk menampilkan halaman edit Inv
2.	`Route ::post('/editinv&&{id_invoice}', [DokumenController::class, 'editinv'])`	Route untuk menyimpan perubahan data Inv
3.	`Route ::get('/deleteinvoice&&{id_invoice}', [DokumenController::class, 'deleteinv']);`	Route untuk menghapus data Inv

4.		Controller untuk menampilkan halaman edit Inv
5.		Controller untuk menyimpan perubahan data Inv
6.		Controller untuk menghapus data Inv

j. Halaman Pendapatan

Gambar di bawah ini merupakan hasil akhir dari halaman awal untuk fitur pendapatan. Di halaman awal ini, user dapat melihat daftar transaksi pendapatan yang telah diinputkan ke sistem. Melalui halaman ini, user dapat menekan tombol tambah data pendapatan yang nantinya akan mengarahkan user ke halaman khusus untuk menginputkan transaksi pendapatan baru. Ataupun user dapat menekan tombol detail untuk melihat detail data suatu transaksi pendapatan yang dipilih, user pun dapat menekan tombol edit untuk mengubah data dari suatu transaksi pendapatan yang telah ada atau menekan tombol hapus untuk menghapus data transaksi pendapatan yang sudah ada.



Gambar 4.136 Halaman Pendapatan

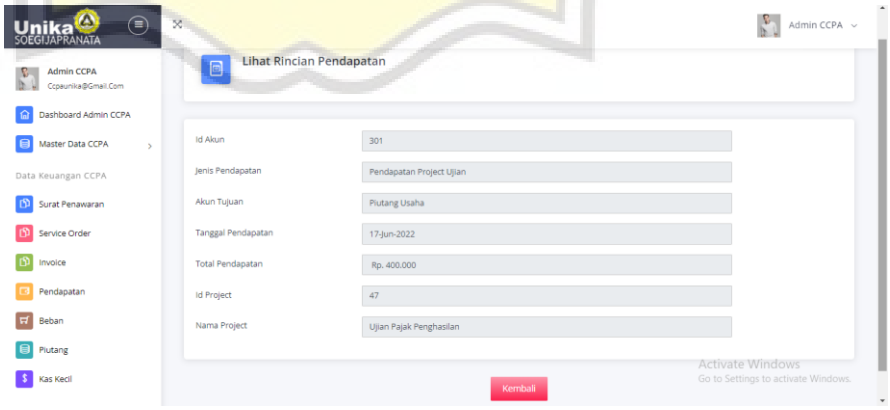
Gambar di bawah ini adalah gambar dari *route* dan juga *controller* yang menunjang berdirinya halaman awal untuk fitur pendaptan dalam sistem ini.

```
Route ::get('/pendapatanccpa', [KeuanganCCPAController::class, 'indexpendapatan']);
```

```
public function indexpendapatan()
{
    $data = [
        'pendapatan'=> $this->KeuanganCCPAModel->allDataPendapatan(),
    ];
    return view ('CCPA.pendapatan', $data);
}
```

Gambar 4.137 Route dan Controller Halaman Pendapatan

Gambar di bawah ini merupakan hasil akhir implementasi untuk halaman lihat pendapatan. Di halaman ini user dapat melihat rincian data dari suatu transaksi pendapatan.



Gambar 4.138 Halaman Lihat Pendapatan

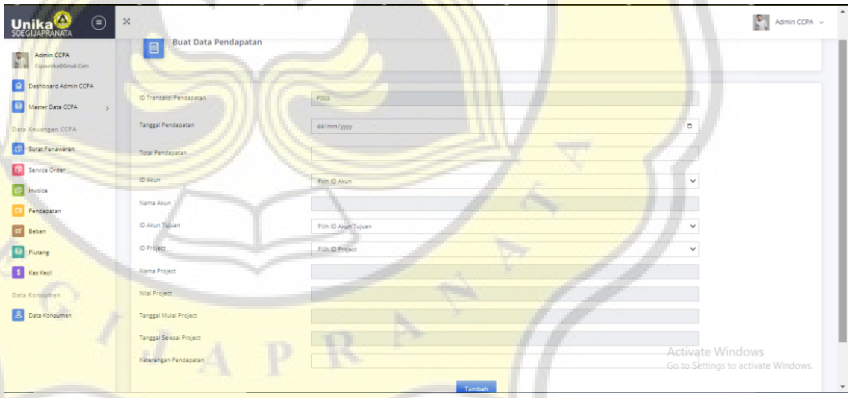
Gambar di bawah ini merupakan *route* dan juga *controller* yang mendukung berjalannya halaman lihat pendapatan.

```
Route ::get('/lihatpendapatan&&{id_pendapatan}', [KeuanganCCPAController::class, 'lihatpendapatan']);
```

```
public function lihatpendapatan($id_pendapatan)
{
    $lihat = [
        'lihatpendapatan'=> $this->KeuanganCCPAModel->LihatPendapatan($id_pendapatan),
    ];
    //DD($lihat);
    $id = ($lihat['lihatpendapatan']['0']->debit_pendapatan);
    $tujuan = [
        'debit_pendapatan'=> $this->KeuanganCCPAModel->tujuanpendapatan($id),
    ];
    return view ('CCPA.lihatpendapatan', $lihat, $tujuan);
}
```

Gambar 4.139 Route dan Controller Lihat Pendapatan

Gambar di bawah ini merupakan implementasi akhir untuk halaman tambah data transaksi pendapatan. Melalui halaman inilah nantinya user dapat menambahkan data transkasi pendapatan yang baru.



Gambar 4.140 Halaman Tambah Data Pendapatan

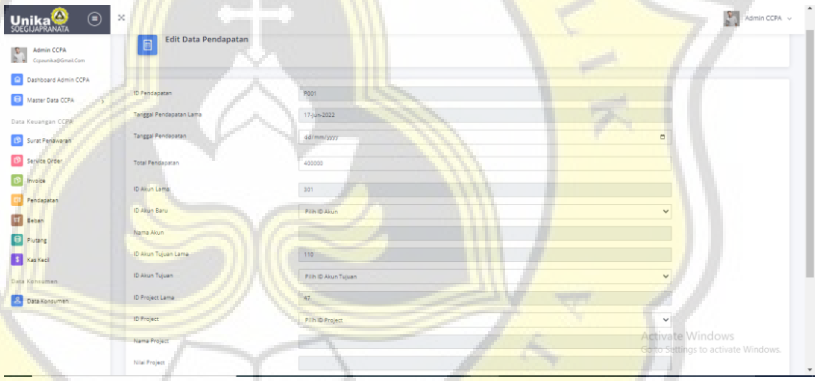
Gambar di bawah ini merupakan *route* dan juga *controller* yang mendukung berjalannya proses penambahan data transaksi pendapatan baru.

```
Route ::get('/inputpendapatan', [KeuanganCCPAController::class, 'inputpendapatan']);
Route ::get('/inputpendapatan/ajax/{id}', [KeuanganCCPAController::class, 'ajax']);
Route ::get('/inputpendapatan/ajaxakun/{id}', [KeuanganCCPAController::class, 'ajaxakun']);
Route ::post('/insertpendapatan', [KeuanganCCPAController::class, 'insertpendapatan']);
```

```php
public function inputpendapatan()
{
    $data = [
        'project'=>$this->KeuanganCCPAModel->dataproject(),
        'akunpendapatan'=>$this->KeuanganCCPAModel->dataakunpendapatan(),
        'akunkas'=>$this->KeuanganCCPAModel->dataakunkas(),
        'idpendapatanterakhir'=>$this->KeuanganCCPAModel->PendapatanTerakhir(),
    ];
    $ambilid = ($data['idpendapatanterakhir']);
    if($ambilid == null){
        $id = 'P001';
    }
    else{
    $id = (($data['idpendapatanterakhir'])->id_pendapatan);
    $id = substr($id,1);
    $id = (int)$id + 1;
    $id = sprintf("%03d",$id);
    $id = 'P'.$id;
    }
    $apa = [
        'id'=>$id,
    ];
    //DD($apa);
    return view ('CCPA.inputpendapatan', $data, $apa);
}
```

Gambar 4.141 Route dan Controller Tambah Data Pendapatan

Gambar di bawah ini merupakan halaman yang akan digunakan user ketika akan mengubah data dari suatu transaksi pendapatan yang telah diinputkan ke dalam sistem.

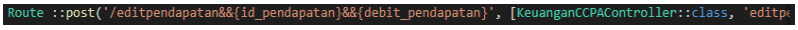
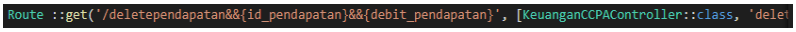
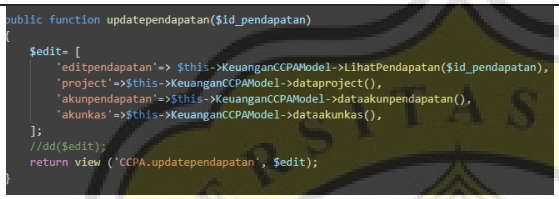
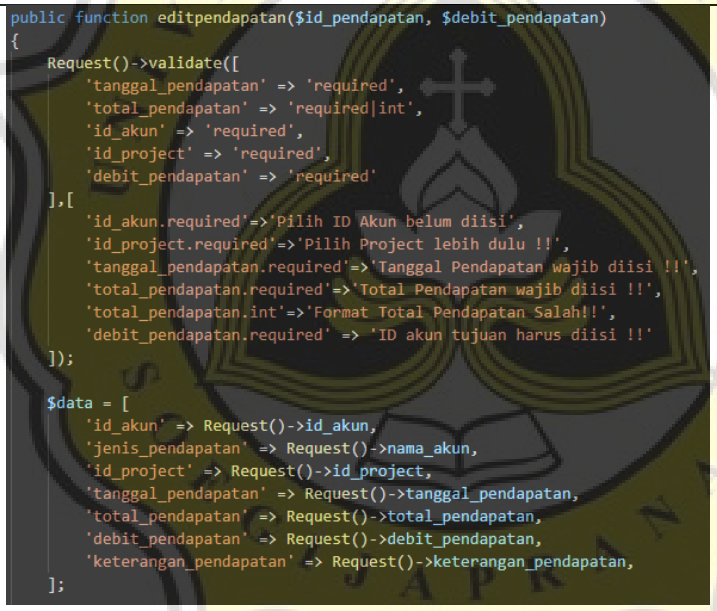
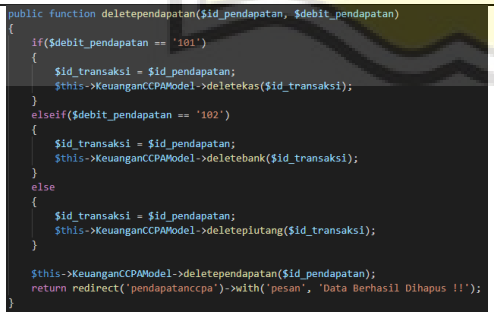


Gambar 4.142 Halaman Edit Data Pendapatan

Gambar di bawah ini merupakan *route* dan juga *controller* yang mendukung berjalannya halaman edit dan hapus pendapatan.

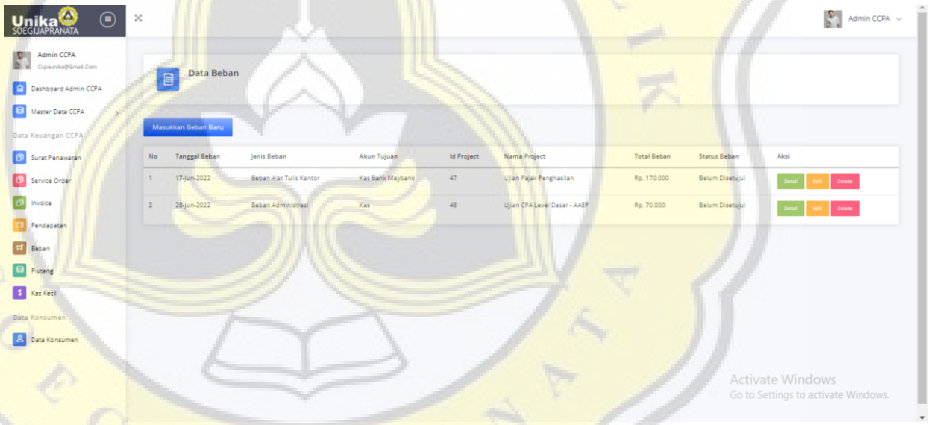
Tabel 4.19 Route dan Controller Edit dan Hapus Data Pendapatan

No	Gambar	Penjelasan
1.	`Route ::get('/updatependapatan&&{id_pendapatan}', [KeuanganCCPAController::class, 'updatependapatan']);`	Route untuk menampilkan halaman edit Pendapatan

2.		Route untuk menyimpan perubahan data Pendapatan
3.		Route untuk menghapus data Pendapatan
4.		Controller untuk menampilkan halaman edit Pendapatan
5.		Controller untuk menyimpan perubahan data Pendapatan
6.		Controller untuk menghapus data Pendapatan

k. Halaman Beban

Gambar di bawah ini merupakan hasil akhir dari halaman awal untuk fitur beban. Di halaman awal ini, user dapat melihat daftar transaksi beban yang telah diinputkan ke sistem. Melalui halaman ini, user dapat menekan tombol tambah data beban yang nantinya akan mengarahkan user ke halaman khusus untuk menginputkan transaksi beban baru. Ataupun user dapat menekan tombol detail untuk melihat detail data suatu transaksi beban yang dipilih, user pun dapat menekan tombol edit untuk mengubah data dari suatu transaksi beban yang telah ada atau menekan tombol hapus untuk menghapus data transaksi beban yang sudah ada.



Gambar 4.143 Halaman Data Beban

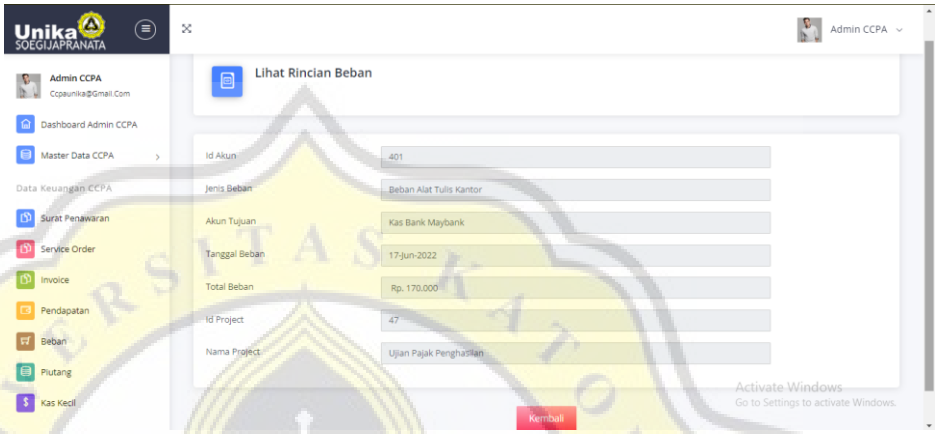
Gambar di bawah ini adalah gambar dari *route* dan juga *controller* yang menunjang berdirinya halaman awal untuk fitur beban dalam sistem ini.

```
Route ::get('/bebanccpa', [KeuanganCCPAController::class, 'indexbeban']);
```

```
public function indexbeban()
{
    $data = [
        'beban'=> $this->KeuanganCCPAModel->allDataBeban(),
    ];
    return view ('CCPA.beban', $data);
}
```

Gambar 4.144 Route dan Controller Halaman Beban

Gambar di bawah ini merupakan hasil akhir implementasi untuk halaman lihat beban. Di halaman ini user dapat melihat rincian data dari suatu transaksi beban.



Gambar 4.145 Halaman Lihat Data Beban

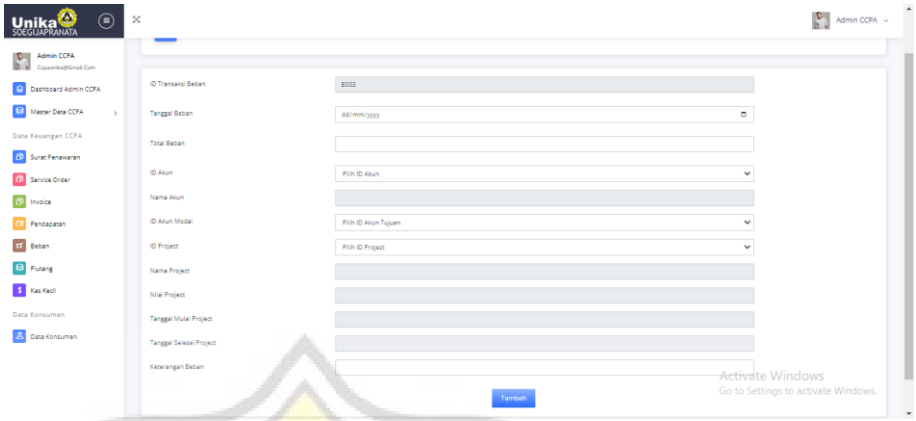
Gambar di bawah ini merupakan *route* dan juga *controller* yang mendukung berjalannya halaman lihat beban.

```
Route ::get('/lihatbeban&&{id_beban}', [KeuanganCCPAController::class, 'lihatbeban']);
```

```
public function lihatbeban($id_beban)
{
    $lihat = [
        'lihatbeban'=> $this->KeuanganCCPAModel->LihatBeban($id_beban),
    ];
    $id = ($lihat['lihatbeban']['0']->kredit_beban);
    $tujuan = [
        'kredit_beban'=> $this->KeuanganCCPAModel->tujuanbeban($id),
    ];
    return view ('CCPA.lihatbeban', $lihat, $tujuan);
}
```

Gambar 4.146 Route dan Controller Lihat Data Beban

Gambar di bawah ini merupakan implementasi akhir untuk halaman tambah data transaksi beban. Melalui halaman inilah nantinya user dapat menambahkan data transkasi beban yang baru.



Gambar 4.147 Halaman Edit Data Beban

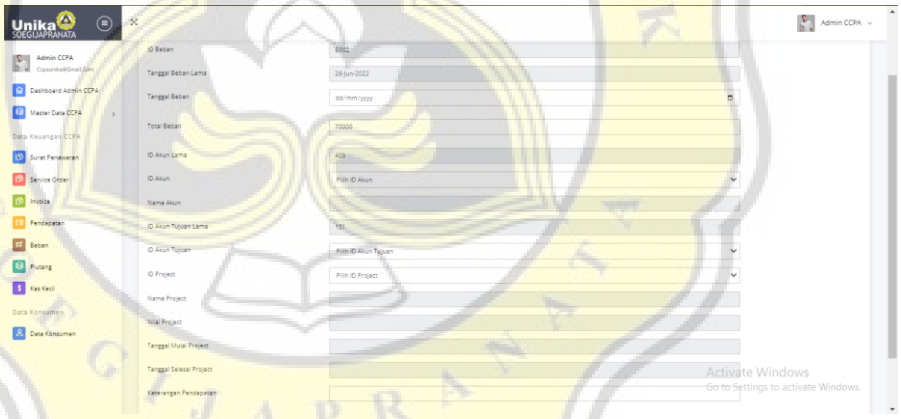
Gambar di bawah ini merupakan *route* dan juga *controller* yang mendukung berjalannya halaman tambah beban.

Tabel 4.20 Route dan Controller Tambah Data Beban

No	Gambar	Penjelasan
1.	`Route ::get('/inputbeban', [KeuanganCCPAController::class, 'inputbeban']);`	Route untuk menampilkan halaman input beban
2.	`Route ::post('/insertbeban', [KeuanganCCPAController::class, 'insertbeban']);`	Route untuk menyimpan data beban
3.	`public function inputbeban()` `{` `$data = [` `'project'=>$this->KeuanganCCPAModel->dataproject(),` `'akunbeban'=>$this->KeuanganCCPAModel->dataakunbeban(),` `'akunkas'=>$this->KeuanganCCPAModel->dataakunkas(),` `'idbebanterakhir'=>$this->KeuanganCCPAModel->BebanTerakhir(),` `];` `$ambilid = ($data['idbebanterakhir']);` `if($ambilid == null){` `$id = 'B001';` `}` `else{` `$id = (($data['idbebanterakhir'])->id_beban);` `$id = substr($id,1);` `$id = (int)$id + 1;` `$id = sprintf("%03d",$id);` `$id = 'B'.$id;` `}` `$apa = [` `'id'=>$id,` `];` `return view ('CCPA.inputbeban', $data, $apa);` `}`	Controller untuk menampilkan halaman input beban

4.		Controller untuk menyimpan data beban

Gambar di bawah ini merupakan halaman yang akan digunakan user ketika akan mengubah data dari suatu transaksi beban yang telah diinputkan ke dalam sistem.

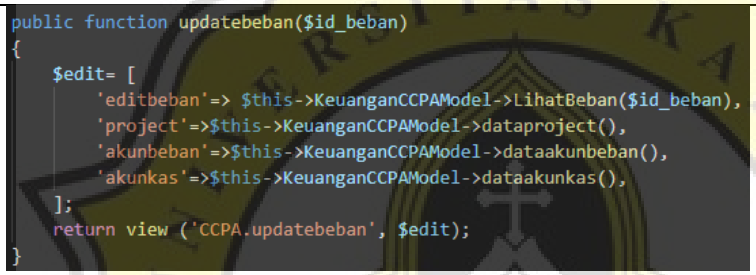


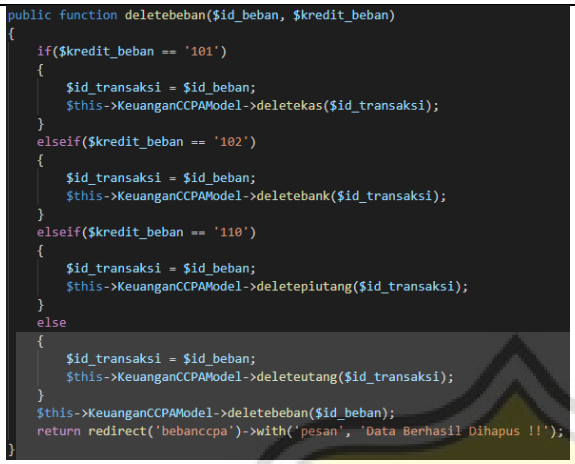
Gambar 4.148 Halaman Edit Data Beban

Gambar di bawah ini merupakan *route* dan juga *controller* yang mendukung berjalannya halaman edit dan hapus beban

Tabel 4.21 Route dan Controller Edit dan Hapus Data Beban

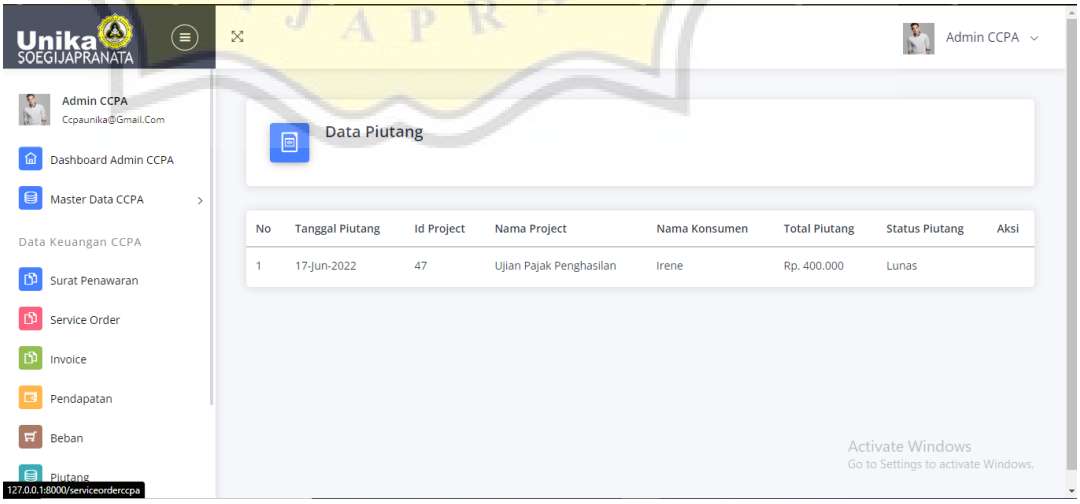
No	Gambar	Penjelasan
1.	`Route ::get('/updatebeban&&{id_beban}', [KeuanganCCPAController::class, 'updatebeban']);`	Route untuk menampilkan

		halaman edit data beban
2.	`Route ::post('/editbeban&&{id_beban}&&{kredit_beban}', [KeuanganCCPAController::class, 'editbeban']);`	Route untuk menyimpan perubahan data beban
3.	`Route ::get('/deletebeban&&{id_beban}&&{kredit_beban}', [KeuanganCCPAController::class, 'deletebeban']);`	Route untuk menghapus data beban
4.		Controller untuk menampilkan halaman edit data beban
5.		Controller untuk menyimpan perubahan data beban

6.			Controller untuk menghapus data beban

l. Halaman Piutang

Gambar di bawah ini adalah hasil akhir dari halaman piutang. Di halaman ini, user dapat melihat daftar piutang yang masih belum lunas maupun yang sudah lunas. Di halaman ini juga pun, user dapat menekan tombol pelunasan piutang yang nantinya akan membuat status piutang yang sebelumnya belum lunas menjadi lunas.Namun ketika akan melunasi suatu piutang, user diharapkan benar-benar memastikan apakah kas sudah masuk dan tanggal pelunasannya. Hal ini dikarenakan ketika akan melunasi suatu piutang user diminta untuk mengisi tanggal pelunasan dan metode pelunasan (baik itu dengan tunai maupun dengan transfer bank).

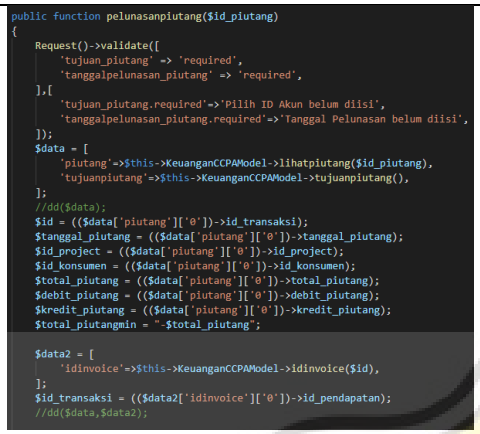


Gambar 4.149 Halaman Piutang

Gambar di bawah ini merupakan *route* dan juga *controller* yang mendukung berjalannya halaman piutang.

Tabel 4.22 Route dan Controller Piutang

No	Gambar	Penjelasan
1.	`Route ::get('/piutangccpa', [KeuanganCCPAController::class, 'piutangccpa']);`	Route untuk menampilkan halaman piutang
2.	`Route ::get('/lihatpiutang&&{id_piutang}', [KeuanganCCPAController::class, 'lihatpiutang']);`	Route untuk menampilkan halaman detail data piutang
3.	`Route ::post('/pelunasanpiutang&&{id_piutang}', [KeuanganCCPAController::class, 'pelunasanpiutang']);`	Route untuk menyimpan data piutang
4.	`//PIUTANG CCPA` `public function piutangccpa()` `{` `$data = [` `'piutang'=>$this->KeuanganCCPAModel->piutangccpa(),` `];` `//dd($data);` `return view ('CCPA.piutangccpa', $data);` `}`	Controller untuk menampilkan halaman piutang
5.	`public function lihatpiutang($id_piutang)` `{` `$data = [` `'piutang'=>$this->KeuanganCCPAModel->lihatpiutang($id_piutang),` `'tujuanpiutang'=>$this->KeuanganCCPAModel->tujuanpiutang(),` `];` `$id = (($data['piutang']['0'])->id_transaksi);` `$data2 = [` `'idinvoice'=>$this->KeuanganCCPAModel->idinvoice($id),` `];` `//dd($data,$data2);` `return view ('CCPA.lihatpiutang', $data, $data2);` `}`	Controller untuk menampilkan halaman detail data piutang

6.		Controller untuk menyimpan data piutang

m. Halaman Kas Kecil

Gambar di bawah ini adalah hasil akhir untuk halaman kas kecil. Dimana di halaman ini user dapat melihat arus keluar masuk dari kas beserta dengan detail transaksinya.

Gambar 4.150 Halaman Kas Kecil

Gambar di bawah ini merupakan *route* dan juga *controller* yang mendukung berjalannya halaman kas kecil.

```
//Route Kas Kecil CCPA
Route ::get('/kaskecilccpa', [KeuanganCCPAController::class, 'kaskecilccpa']);
```

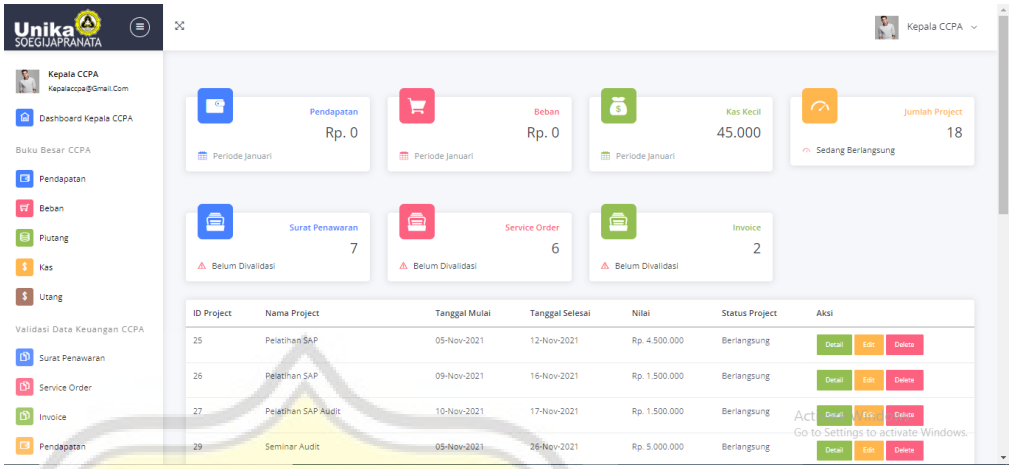
```
//KAS KECIL CCPA
public function kaskecilccpa()
{
    $data = [
        'kasmasuk'=>$this->KeuanganCCPAModel->kasmasuk(),
        'kaskeluar'=>$this->KeuanganCCPAModel->kaskeluar(),
    ];
    return view ('CCPA.kaskecilccpa', $data);
}
```

Gambar 4.151 Route dan Controller Kas Kecil

3. Halaman Kepala

a. Halaman *Dashboard Kepala*

Gambar di bawah ini merupakan hasil dari implementasi untuk halaman *dashboard* kepala. Di halaman ini kepala dapat melihat beberapa informasi penting diantaranya jumlah surat penawaran, *service order*¸dan *invoice* yang belum divalidasi serta jumlah project yang berlangsung. Selain itu di halaman ini, kepala dapat melihat saldo pendapatan, beban dan juga kas dalam suatu periode. Di halaman ini pula user dapat melihat tabel yang berisikan daftar project yang ada. Dimana di tiap projectnya user dapat melihat data detailnya, mengedit datanya apabila terjadi kesalahan, serta menghapus data project yang sudah ada. Dibagian kiri dari halaman ini terdapat *navbar* yang nantinya memudahkan user untuk mengakses halaman lainnya, dan di bagian atas dari halaman ini terdapat fitur *logout* yang akan muncul ketika user menekan nama akunnya.



Gambar 4.152 Halaman Dashboard Kepala

Gambar di bawah ini merupakan gambar *route* dan *controller* untuk halaman *dashboard* kepala.

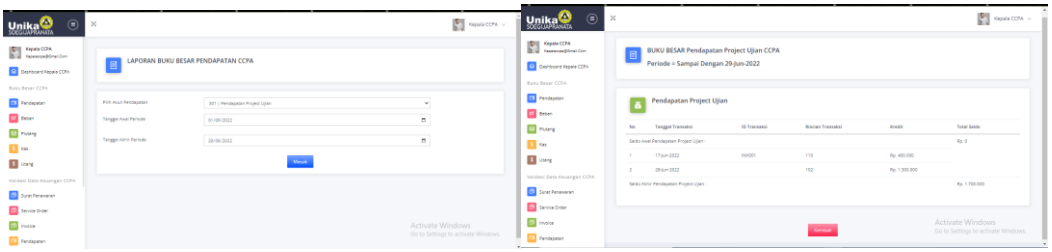
```
Route ::get('/dashboardkepalaCCPA', [ProjectController::class, 'indexkepala']);
```

```
public function indexkepala()
{
    $data = [
        'project'=> $this->ProjectModel->allData(),
    ];
    $jumlah = [
        'jumlahproject'=> $this->ProjectModel->jumlahproject(),
        'jumlahsp'=> $this->ProjectModel->jumlahsp(),
        'jumlahso'=> $this->ProjectModel->jumlahso(),
        'jumlahinvoice'=> $this->ProjectModel->jumlahinvoice(),
        'jumlahpendapatan'=> $this->ProjectModel->jumlahpendapatan(),
        'jumlahbeban'=> $this->ProjectModel->jumlahbeban(),
    ];
    return view ('CCPA.dashboardkepala', $data, $jumlah);
}
```

Gambar 4.153 Route dan Controller Halaman Dashboard Kepala

b. Halaman Buku Besar Pendapatan

Gambar di bawah ini merupakan hasil akhir untuk halaman fitur buku besar pendapatan. Di halaman awal ini user diminta untuk memilih jenis pendapatan yang akan dilihat buku besarnya dan menginputkan tanggal awal dan tanggal akhir periode laporan. Setelah itu user diminta untuk menekan tombol masuk, yang kemudian akan mengarahkan user ke halaman selanjutnya dimana user dapat melihat detail buku besar untuk akun pendapatan yang dipilihnya.



Gambar 4.154 Halaman Buku Besar Pendapatan

Gambar di bawah ini merupakan *route* dan *controller* yang mendasari fitur buku besar untuk akun pendapatan ini.

```
Route ::get('/bbpendapatanccpa', [KeuanganCCPAController::class, 'bbpendapatanccpa']);
Route ::post('/bbpendapatanccpa2', [KeuanganCCPAController::class, 'insertbbpendapatanccpa']);
```

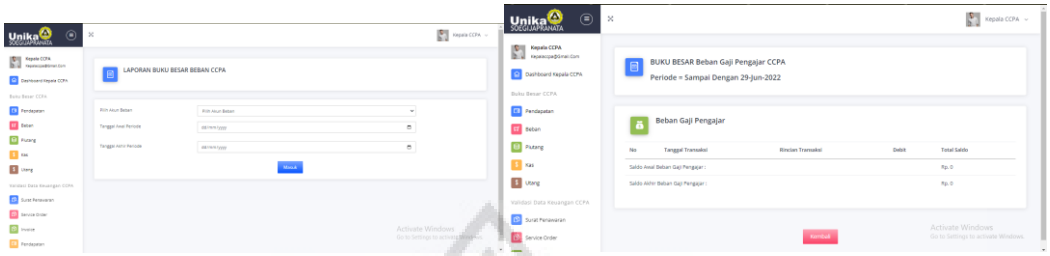
```
public function bbpendapatanccpa()
{
    $data = [
        'jenispendapatan'=>$this->KeuanganCCPAModel->dataakunpendapatan(),
    ];
    return view ('CCPA.bbpendapatanccpa', $data);
}

public function insertbbpendapatanccpa()
{
    Request()->validate([
        'tanggal_akhir' => 'required',
        'tanggal_awal' => 'required',
        'nama_akun' => 'required',
    ],[
        'tanggal_akhir.required'=>'Tanggal Akhir Belum diisi',
        'tanggal_awal.required'=>'Tanggal Akhir Belum diisi',
        'nama_akun.required'=>'Nama Akun Belum diisi',
    ]);

    $tanggal_awal = Request()->tanggal_awal;
    $tanggal_akhir = Request()->tanggal_akhir;
    $nama_akun = Request()->nama_akun;
    //DD($nama_akun);
    $data = [
        'saldoawal_bbpendapatan'=>$this->KeuanganCCPAModel->saldoawal_bbpendapatan($tanggal_awal, $nama_aku
        'bbpendapatanccpa'=>$this->KeuanganCCPAModel->alldatabbpendapatan($tanggal_awal, $tanggal_akhir, $
        'saldoakhir_bbpendapatan'=>$this->KeuanganCCPAModel->saldoakhir_bbpendapatan($tanggal_akhir, $nama
        'tanggal_akhir'=>$tanggal_akhir,
        'nama_akun'=>$nama_akun,
```

Gambar 4.155 Route dan Controller Buku Besar Pendapatan

c. Halaman Buku Besar Beban

Gambar di bawah ini merupakan hasil akhir untuk halaman fitur buku besar beban. Di halaman awal ini user diminta untuk memilih jenis beban yang akan dilihat buku besarnya dan menginputkan tanggal awal dan tanggal akhir periode laporan. Setelah itu user diminta untuk menekan tombol masuk, yang kemudian akan mengarahkan user ke halaman selanjutnya dimana

user dapat melihat detail buku besar untuk akun beban yang dipilihnya.



Gambar 4.156 Halaman Buku Besar Beban

Gambar di bawah ini merupakan *route* dan *controller* yang mendasari fitur buku besar untuk akun beban ini.

```
Route ::get('/bbbebanccpa', [KeuanganCCPAController::class, 'bbbebanccpa']);
Route ::post('/bbbebanccpa2', [KeuanganCCPAController::class, 'insertbbbebanccpa']);
```

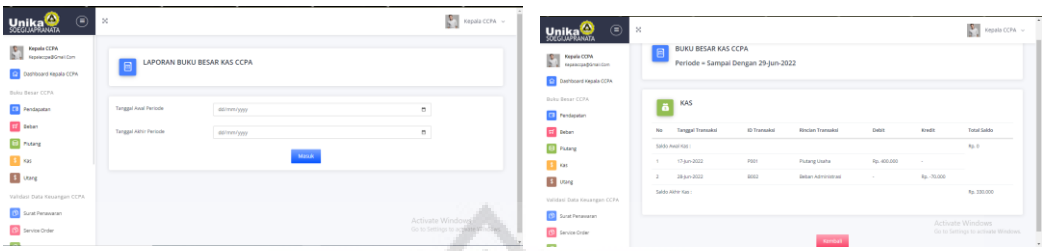
```
public function bbbebanccpa()
{
    $data = [
        'jenisbeban'=>$this->KeuanganCCPAModel->dataakunbeban(),
    ];
    return view ('CCPA.bbbebanccpa', $data);
}

public function insertbbbebanccpa()
{
    Request()->validate([
        'tanggal_akhir' => 'required',
        'tanggal_awal' => 'required',
        'nama_akun' => 'required',
    ],[
        'tanggal_akhir.required'=>'Tanggal Akhir Belum diisi',
        'tanggal_awal.required'=>'Tanggal Akhir Belum diisi',
        'nama_akun.required'=>'Nama Akun Belum diisi',
    ]);

    $tanggal_awal = Request()->tanggal_awal;
    $tanggal_akhir = Request()->tanggal_akhir;
    $nama_akun = Request()->nama_akun;
    //DD($nama_akun);
    $data = [
        'saldoawal_bbbeban'=>$this->KeuanganCCPAModel->saldoawal_bbbeban($tanggal_awal, $nama_akun),
        'bbbebanccpa'=>$this->KeuanganCCPAModel->alldatabbbeban($tanggal_awal, $tanggal_akhir, $nama_akun),
        'saldoakhir_bbbeban'=>$this->KeuanganCCPAModel->saldoakhir_bbbeban($tanggal_akhir, $nama_akun),
        'tanggal_akhir'=>$tanggal_akhir,
        'nama_akun'=>$nama_akun,
    ];
```

Gambar 4.157 Route dan Controller Buku Besar Beban

d. Halaman Buku Besar Kas

Gambar di bawah ini merupakan hasil akhir untuk halaman fitur buku besar kas. Di halaman awal ini user diminta untuk menginputkan tanggal awal dan tanggal akhir periode laporan. Setelah itu user diminta untuk menekan tombol masuk, yang

kemudian akan mengarahkan user ke halaman selanjutnya dimana user dapat melihat detail buku besar untuk akun kas.



Gambar 4.158 Halaman Buku Besar Kas

Gambar di bawah ini merupakan *route* dan *controller* yang mendasari fitur buku besar untuk akun kas ini.

```
Route ::get('/bbkasccpa', [KeuanganCCPAController::class, 'bbkasccpa']);
Route ::post('/bbkasccpa2', [KeuanganCCPAController::class, 'insertbbkasccpa']);
```

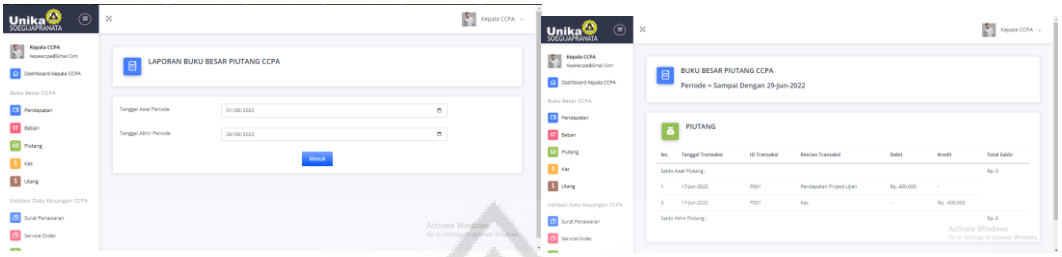
```
public function bbkasccpa()
{
    return view ('CCPA.bbkasccpa');
}

public function insertbbkasccpa()
{
    Request()->validate([
        'tanggal_akhir' => 'required',
        'tanggal_awal' => 'required',
    ],[
        'tanggal_akhir.required'=>'Tanggal Akhir Belum diisi',
        'tanggal_awal.required'=>'Tanggal Akhir Belum diisi',
    ]);

    $tanggal_awal = Request()->tanggal_awal;
    $tanggal_akhir = Request()->tanggal_akhir;
    $data = [
        'saldoawal_bbkas'=>$this->KeuanganCCPAModel->saldoawal_bbkas($tanggal_awal),
        'bbkasccpa'=>$this->KeuanganCCPAModel->alldatakas($tanggal_awal, $tanggal_akhir),
        'saldoakhir_bbkas'=>$this->KeuanganCCPAModel->saldoakhir_bbkas($tanggal_akhir),
        'tanggal_akhir'=>$tanggal_akhir,
    ];
    //DD($data);
    return view ('CCPA.bbkasccpa2', $data);
}
```

Gambar 4.159 Route dan Controller Buku Besar Kas

e. Halaman Buku Besar Piutang

Gambar di bawah ini merupakan hasil akhir untuk halaman fitur buku besar piutang. Di halaman awal ini user diminta untuk menginputkan tanggal awal dan tanggal akhir periode laporan. Setelah itu user diminta untuk menekan tombol masuk, yang

kemudian akan mengarahkan user ke halaman selanjutnya dimana user dapat melihat detail buku besar untuk akun piutang.



Gambar 4.160 Halaman Buku Besar Piutang

Gambar di bawah ini merupakan *route* dan *controller* yang mendasari fitur buku besar untuk akun piutang ini.

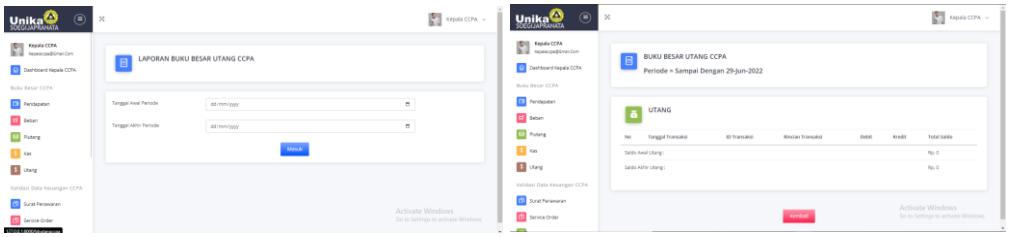
```
Route ::get('/bbpiutangccpa', [KeuanganCCPAController::class, 'bbpiutangccpa']);
Route ::post('/bbpiutangccpa2', [KeuanganCCPAController::class, 'insertbbpiutangccpa']);
```

```
public function bbpiutangccpa()
{
    return view ('CCPA.bbpiutangccpa');
}

public function insertbbpiutangccpa()
{
    Request()->validate([
        'tanggal_akhir' => 'required',
        'tanggal_awal' => 'required',
    ],[
        'tanggal_akhir.required'=>'Tanggal Akhir Belum diisi',
        'tanggal_awal.required'=>'Tanggal Akhir Belum diisi',
    ]);

    $tanggal_awal = Request()->tanggal_awal;
    $tanggal_akhir = Request()->tanggal_akhir;
    $data = [
        'saldoawal_bbpiutang'=>$this->KeuanganCCPAModel->saldoawal_bbpiutang($tanggal_awal),
        'bbpiutangccpa'=>$this->KeuanganCCPAModel->alldatapiutang($tanggal_awal, $tanggal_akhir),
        'saldoakhir_bbpiutang'=>$this->KeuanganCCPAModel->saldoakhir_bbpiutang($tanggal_akhir),
        'tanggal_akhir'=>$tanggal_akhir,
    ];
    //DD($data);
    return view ('CCPA.bbpiutangccpa2', $data);
}
```

Gambar 4.161 Route dan Controller Buku Besar Piutang

f. Halaman Buku Besar Utang

Gambar di bawah ini merupakan hasil akhir untuk halaman fitur buku besar utang. Di halaman awal ini user diminta untuk menginputkan tanggal awal dan tanggal akhir periode laporan. Setelah itu user diminta untuk menekan tombol masuk, yang kemudian akan mengarahkan user ke halaman selanjutnya dimana user dapat melihat detail buku besar untuk akun utang.



Gambar 4.162 Halaman Buku Besar Utang

Gambar di bawah ini merupakan *route* dan *controller* yang mendasari fitur buku besar untuk akun utang ini.

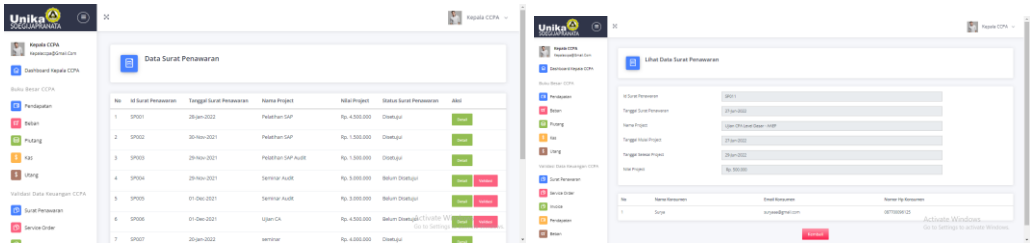
```
Route ::get('/bbutangccpa', [KeuanganCCPAController::class, 'bbutangccpa']);
Route ::post('/bbutangccpa2', [KeuanganCCPAController::class, 'insertbbutangccpa']);
```

```
public function bbutangccpa()
{
    return view ('CCPA.bbutangccpa');
}

public function insertbbutangccpa()
{
    Request()->validate([
        'tanggal_akhir' => 'required',
        'tanggal_awal' => 'required',
    ],[
        'tanggal_akhir.required'=>'Tanggal Akhir Belum diisi',
        'tanggal_awal.required'=>'Tanggal Akhir Belum diisi',
    ]);

    $tanggal_awal = Request()->tanggal_awal;
    $tanggal_akhir = Request()->tanggal_akhir;
    $data = [
        'saldoawal_bbutang'=>$this->KeuanganCCPAModel->saldoawal_bbutang($tanggal_awal),
        'bbutangccpa'=>$this->KeuanganCCPAModel->alldatautang($tanggal_awal, $tanggal_akhir),
        'saldoakhir_bbutang'=>$this->KeuanganCCPAModel->saldoakhir_bbutang($tanggal_akhir),
        'tanggal_akhir'=>$tanggal_akhir,
    ];
    //DD($data);
    return view ('CCPA.bbutangccpa2', $data);
}
```

Gambar 4.163 Route dan Controller Buku Besar Utang

g. Halaman Validasi Surat Penawaran

Gambar di bawah ini merupakan hasil akhir untuk halaman validasi surat penawaran. Di halaman ini user dapat melakukan validasi surat penawaran dan melihat rincian data dari suatu surat penawaran.



Gambar 4.164 Halaman Validasi Surat Penawaran

Gambar di bawah ini merupakan *route* dan *controller* yang mendasari fitur validasi surat penawaran ini.

```
Route ::get('/validasispccpa', [DokumenController::class, 'validasispccpa']);
Route ::get('/lihatsp&&{id_suratpenawaran}', [DokumenController::class, 'lihatspvalid']);
Route ::get('/validasisp&&{id_suratpenawaran}', [DokumenController::class, 'validasisp']);
```

```
public function validasispccpa()
{
    $data = [
        'suratpenawaran'=> $this->DokumenModel->allData(),
    ];
    return view ('DokumenCCPA.validasispccpa', $data);
}
```

```
public function lihatspvalid($id_suratpenawaran)
{
    $lihat = [
        'lihatsp'=> $this->DokumenModel->LihatSP($id_suratpenawaran),
    ];
    $konsumen = [
        'konsumen'=> $this->DokumenModel->LihatKonsumen($id_suratpenawaran),
    ];
    //DD($lihat);
    return view ('DokumenCCPA.lihatspvalid', $lihat, $konsumen);
}
```

```
public function validasisp($id_suratpenawaran)
{
    $data = [
        'status_suratpenawaran' => 'Disetujui',
    ];
    $this->DokumenModel->validasisp($id_suratpenawaran, $data);
    return redirect('validasispccpa')->with('pesan', 'Data Berhasil Divalidasi !!');
}
```

Gambar 4.165 Route dan Controller Validasi Surat Penawaran

h. Halaman Validasi *Service Order*

Gambar di bawah ini merupakan hasil akhir untuk halaman validasi *service order*. Di halaman ini user dapat melakukan validasi *service order* dan melihat rincian data dari suatu *service order*.

Gambar 4.166 Halaman Validasi Service Order

Gambar di bawah ini merupakan *route* dan *controller* yang mendasari fitur validasi *service order* ini.

```
Route ::get('/validasisoccpa', [DokumenController::class, 'validasisoccpa']);
Route ::get('/lihatso&&{id_serviceorder}', [DokumenController::class, 'lihatsovalid']);
Route ::get('/validasiso&&{id_serviceorder}', [DokumenController::class, 'validasiso']);
```

```php
public function validasisoccpa()
{
    $data = [
        'serviceorder'=> $this->DokumenModel->dataso(),
    ];
    return view ('DokumenCCPA.validasisoccpa', $data);
}
```

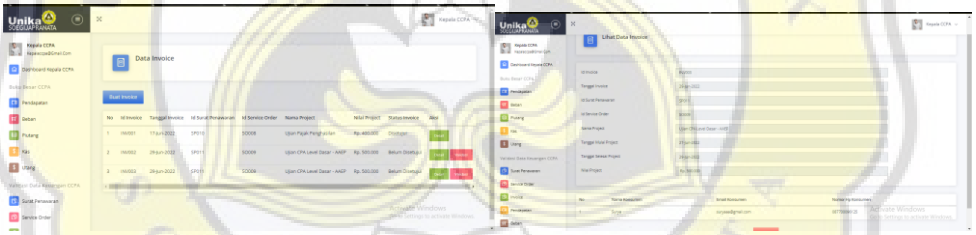
```php
public function lihatsovalid($id_serviceorder)
{
    $lihat = [
        'lihatso'=> $this->DokumenModel->LihatSO($id_serviceorder),
    ];
    $konsumen = [
        'konsumen'=> $this->DokumenModel->LihatKonsumenSO($id_serviceorder),
    ];
    //DD($lihat);
    return view ('DokumenCCPA.lihatsovalid', $lihat, $konsumen);
}
```

```php
public function validasiso($id_serviceorder)
{
    $data = [
        'status_serviceorder' => 'Disetujui',
    ];
    $this->DokumenModel->validasiso($id_serviceorder, $data);
    return redirect('validasisoccpa')->with('pesan', 'Data Berhasil Divalidasi !!');
}
```

Gambar 4.167 Route dan Controller Validasi Service Order

i. Halaman Validasi *Invoice*

Gambar di bawah ini merupakan hasil akhir untuk halaman validasi *invoice*. Di halaman ini user dapat melakukan validasi *invoice* dan melihat rincian data dari suatu *invoice*.



Gambar 4.168 Halaman Validasi Invoice

Gambar di bawah ini merupakan *route* dan *controller* yang mendasari fitur validasi *invoice* ini.

```php
Route ::get('/validasiinvccpa', [DokumenController::class, 'validasiinvccpa']);
Route ::get('/lihatinv&&{id_invoice}', [DokumenController::class, 'lihatinvvalid']);
Route ::get('/validasiinv&&{id_invoice}', [DokumenController::class, 'validasiinv']);
```

```php
public function validasiinvccpa()
{
    $data = [
        'invoice'=> $this->DokumenModel->datainv(),
    ];
    return view ('DokumenCCPA.validasiinvccpa', $data);
}
```

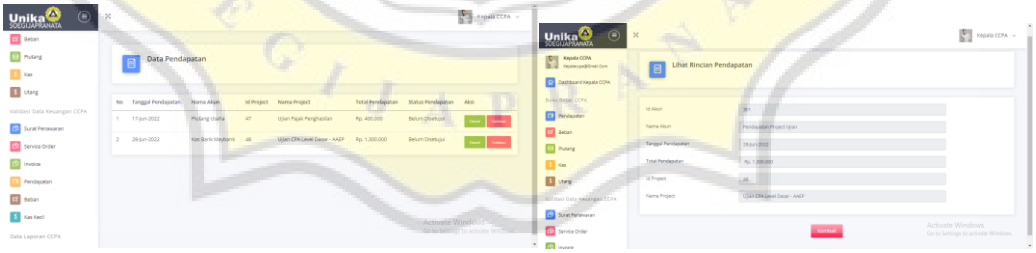
```php
public function lihatinvvalid($id_invoice)
{
    $lihat = [
        'lihatinv'=> $this->DokumenModel->LihatInv($id_invoice),
    ];
    $konsumen = [
        'konsumen'=> $this->DokumenModel->LihatKonsumenInv($id_invoice),
    ];
    //DD($lihat);
    return view ('DokumenCCPA.lihatinvvalid', $lihat, $konsumen);
}
```

```php
public function validasiinv($id_invoice)
{
    $data = [
        'status_invoice' => 'Disetujui',
    ];

    $data3 = [
        'datainvoice'=>$this->DokumenModel->datainvoice($id_invoice),
        'idpendapatanterakhir'=>$this->DokumenModel->PendapatanTerakhir(),
    ];
    //dd($data3);
    $ambilidpendapatan = ($data3['idpendapatanterakhir']);
    if($ambilidpendapatan == null){
        $id2 = 'P001';
    }
    else{
    $id2 = (($data3['idpendapatanterakhir'])->id_pendapatan);
    $id2 = substr($id2,1);
    $id2 = (int)$id2 + 1;
    $id2 = sprintf("%03d",$id2);
    $id2 = 'P'.$id2;
    }

    $id_invoice = ($data3['datainvoice']['0']->id_invoice);
    $id_project = ($data3['datainvoice']['0']->id_project);
    $id_konsumen = ($data3['datainvoice']['0']->id_konsumen);
    $tanggal = ($data3['datainvoice']['0']->tanggal_invoice);
    $data4 = [
        'dataproject'=>$this->DokumenModel->lihatproject($id_project),
    ];
    //dd($data4);
    $total = ($data4['dataproject']['0']->nilai_project);
```

Gambar 4.169 Route dan Controller Validasi Invoice

j. Halaman Validasi Pendapatan

Gambar di bawah ini merupakan hasil akhir untuk halaman validasi pendapatan. Di halaman ini user dapat melakukan validasi pendapatan dan melihat rincian data dari suatu pendapatan.



Gambar 4.170 Halaman Validasi Pendapatan

Gambar di bawah ini merupakan *route* dan *controller* yang mendasari fitur validasi pendapatan ini.

```php
Route ::get('/validasipendapatan', [KeuanganCCPAController::class, 'validasipendapatan']);
Route ::get('/lihatvalidasipendapatan&&{id_pendapatan}', [KeuanganCCPAController::class, 'lihatpendapatanva
```

```php
public function validasipendapatan()
{
    $data = [
        'pendapatan'=> $this->KeuanganCCPAModel->allDataPendapatan(),
    ];
    return view ('CCPA.validasipendapatan', $data);
}
```

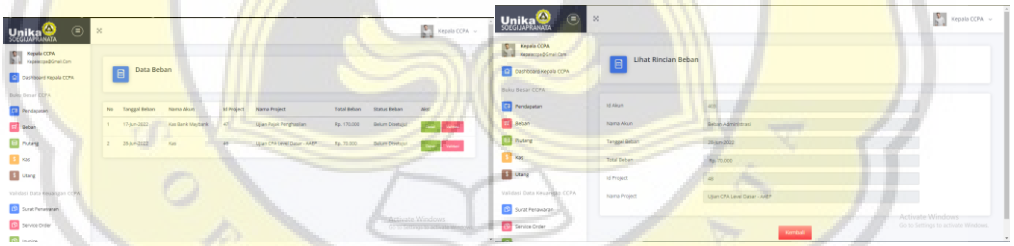
```php
public function validasipendapatan2($id_pendapatan)
{
    $data = [
        'status_pendapatan' => 'Disetujui',
    ];
    $this->KeuanganCCPAModel->validasipendapatan($id_pendapatan, $data);
    return redirect('validasipendapatan')->with('pesan', 'Data Berhasil Divalidasi !!');
}
```

```php
public function lihatpendapatanvalid($id_pendapatan)
{
    $lihat = [
        'lihatpendapatan'=> $this->KeuanganCCPAModel->LihatPendapatan($id_pendapatan),
    ];
    //DD($lihat);
    return view ('CCPA.lihatpendapatanvalid', $lihat);
}
```

Gambar 4.171 Route dan Controller Validasi Pendapatan

k. Halaman Validasi Beban

Gambar di bawah ini merupakan hasil akhir untuk halaman validasi beban. Di halaman ini user dapat melakukan validasi beban dan melihat rincian data dari suatu beban.



Gambar 4.172 Halaman Validasi Beban

Gambar di bawah ini merupakan *route* dan *controller* yang mendasari fitur validasi beban ini.

```php
Route ::get('/validasibeban', [KeuanganCCPAController::class, 'validasibeban']);
Route ::get('/lihatvalidasibeban&&{id_beban}', [KeuanganCCPAController::class, 'lihatbebanvalid']);
Route ::get('/validasibeban&&{id_beban}', [KeuanganCCPAController::class, 'validasibeban2']);
```

```php
public function validasibeban()
{
    $data = [
        'beban'=> $this->KeuanganCCPAModel->allDataBeban(),
    ];
    return view ('CCPA.validasibeban', $data);
}
```

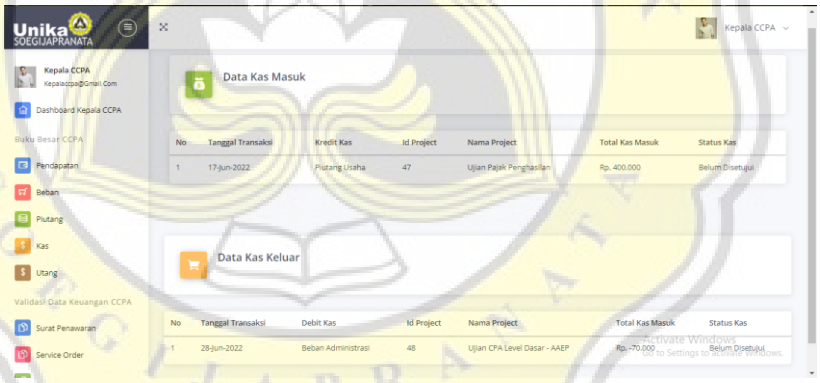
```php
public function validasibeban2($id_beban)
{
    $data = [
        'status_beban' => 'Disetujui',
    ];
    $this->KeuanganCCPAModel->validasibeban($id_beban, $data);
    return redirect('validasibeban')->with('pesan', 'Data Berhasil Divalidasi !!');
}
```

```php
public function lihatbebanvalid($id_beban)
{
    $lihat = [
        'lihatbeban'=> $this->KeuanganCCPAModel->LihatBeban($id_beban),
    ];
    //DD($lihat);
    return view ('CCPA.lihatbebanvalid', $lihat);
}
```

Gambar 4.173 Route dan Controller Validasi Beban

l. Halaman Validasi Kas Kecil

Gambar di bawah ini merupakan hasil akhir untuk halaman validasi kas kecil. Di halaman ini user dapat melakukan validasi kas kecil dan melihat rincian data dari suatu kas kecil.



Gambar 4.174 Halaman Validasi Kas Kecil

Gambar di bawah ini merupakan *route* dan *controller* yang mendasari fitur validasi kas kecil ini.

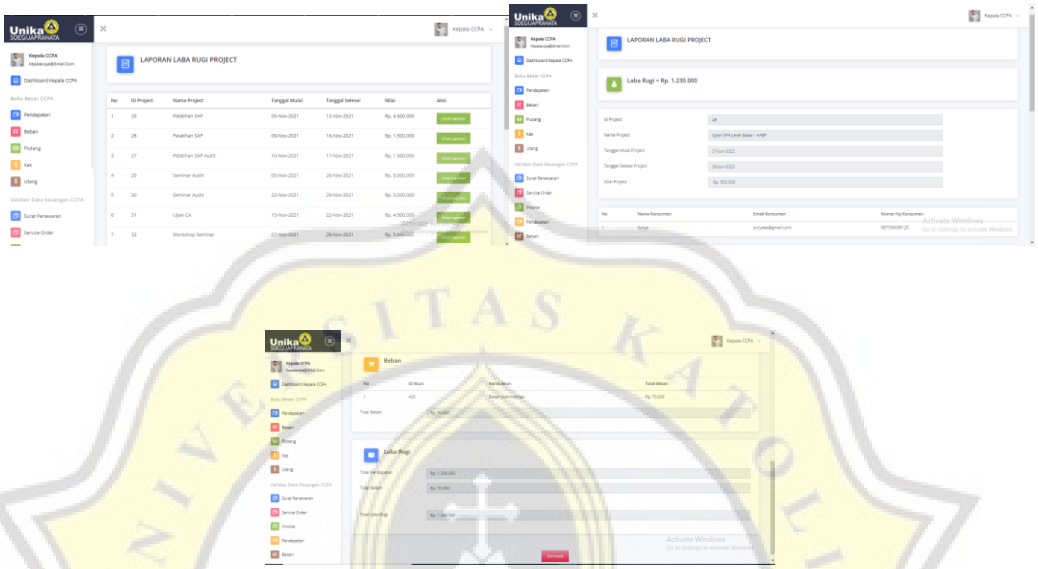
```php
Route ::get('/validkaskecilccpa', [KeuanganCCPAController::class, 'kaskecilccpa']);
```

Gambar 4.175 Route dan Controller Validasi Kas Kecil

m. Halaman Laporan Laba Rugi Project

Gambar di bawah ini merupakan hasil akhir dari halaman awal fitur laporan laba rugi project. Di halaman awal ini disajikan

tabel yang berisikan data project, user dapat memilih salah satu project untuk melihat laporan laba ruginya dengan menekan tombol lihat laporan.



Gambar 4.176 Halaman Laporan Laba Rugi Project

Gambar di bawah ini merupakan *route* dan *controller* yang mendasari fitur laporan laba rugi project ini.

```
Route ::get('/labarugiproject', [KeuanganCCPAController::class, 'labarugiproject']);
Route ::get('/lihatlabarugiproject&&{id_project}', [KeuanganCCPAController::class, 'laporanlrproject']);
```

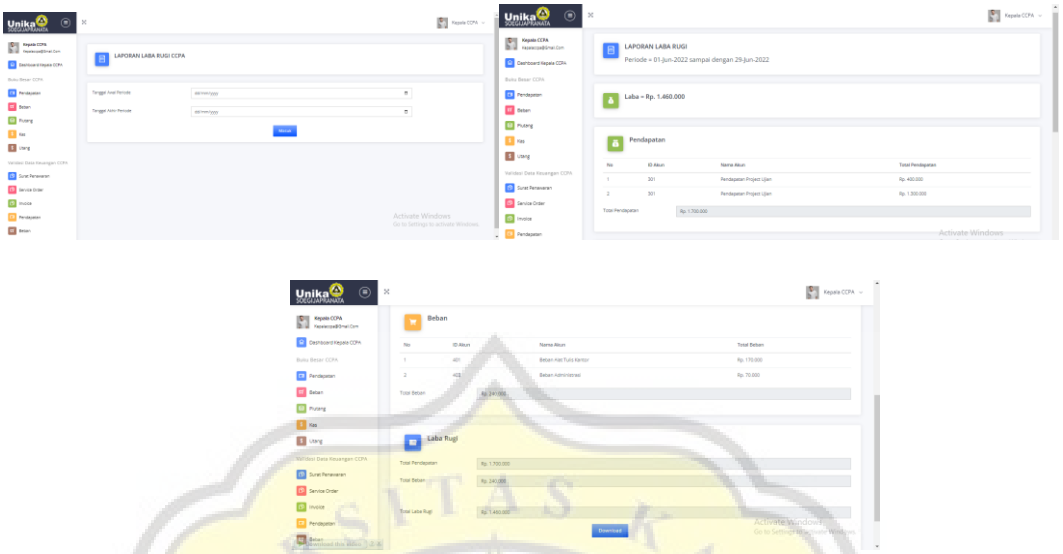
```
public function laporanlrproject($id_project)
{
    $data = [
        'project'=>$this->KeuanganCCPAModel->datarinciproject($id_project),
        'konsumen'=>$this->KeuanganCCPAModel->datarinci1konsumen($id_project),
        'pendapatan'=>$this->KeuanganCCPAModel->laporanpendapatan($id_project),
        'total_pendapatan'=>$this->KeuanganCCPAModel->totalpendapatan($id_project),
        'beban'=>$this->KeuanganCCPAModel->laporanbeban($id_project),
        'total_beban'=>$this->KeuanganCCPAModel->totalbeban($id_project),
    ];
    $labarugi = ($data['total_pendapatan'])-($data['total_beban']);
    $datakeuangan = [
        'labarugi'=>$labarugi,
    ];
    return view ('CCPA.laporanlrproject', $data, $datakeuangan);
}
```

```
public function labarugiproject()
{
    $data = [
        'project'=>$this->KeuanganCCPAModel->dataproject(),
    ];
    return view ('CCPA.labarugiproject', $data);
}
```

Gambar 4.177 Route dan Controller Halaman Laporan Laba Rugi Project

n. Halaman Laporan Laba Rugi

Gambar di bawah ini merupakan hasil akhir dari halaman laporan laba rugi. Di halaman ini user diminta untuk mengisi tanggal awal dan tanggal akhir untuk periode laporannya.



Gambar 4.178 Halaman Laporan Laba Rugi

Gambar di bawah ini adalah *route* dan *controller* untuk fitur laporan laba rugi pada sistem ini.

```
Route ::get('/laporanlabarugiccpa', [KeuanganCCPAController::class, 'labarugiccpa']);
Route ::post('/laporanlabarugiccpa2', [KeuanganCCPAController::class, 'insertlabarugi']);
```

```
public function labarugiccpa()
{
    return view ('CCPA.labarugiccpa');
}
```

```
public function insertlabarugi()
{
    Request()->validate([
        'tanggal_awal' => 'required',
        'tanggal_akhir' => 'required',
    ],[
        'tanggal_awal.required'=>'Tanggal Awal Belum diisi',
        'tanggal_akhir.required'=>'Tanggal Akhir Belum diisi',
    ]);

    $tanggal_awal = Request()->tanggal_awal;
    $tanggal_akhir = Request()->tanggal_akhir;
    $data = [
        'pendapatan'=>$this->KeuanganCCPAModel->pendapatanlabarugi($tanggal_awal, $tanggal_akhir),
        'beban'=>$this->KeuanganCCPAModel->bebanlabarugi($tanggal_awal, $tanggal_akhir),
        'total_pendapatanlr'=>$this->KeuanganCCPAModel->totalpendapatanlr($tanggal_awal, $tanggal_akhir),
        'total_bebanlr'=>$this->KeuanganCCPAModel->totalbebanlr($tanggal_awal, $tanggal_akhir),
    ];
    //dd($data);
    $labarugi = ($data['total_pendapatanlr'])-($data['total_bebanlr']);
    if( ($data['total_pendapatanlr']) > ($data['total_bebanlr']))
    {
        $label = "Laba";
    }
    else
    {
        $label = "Rugi";
    }

    $datakeuangan = [
        'labarugi'=>$labarugi,
```

Gambar 4.179 Route dan Controller Laporan Laba Rugi

PERIODE : 01-Jun-2022 SAMPAI DENGAN 03-Jul-2022

Pendapatan

No	Tanggal Transaksi	Nama Akun	ID Project	Nama Project	Total Pendapatan
1	17-Jun-2022	Pendapatan Project Ujian	47	Ujian Pajak Penghasilan	Rp. 400.000
2	28-Jun-2022	Pendapatan Project Ujian	48	Ujian CPA Level Dasar - AAEP	Rp. 1.300.000
Total Pendapatan :					Rp. 1.700.000

Beban

No	Tanggal Transaksi	Nama Akun	ID Project	Nama Project	Total Beban
1	17-Jun-2022	Beban Alat Tulis Kantor	47	Ujian Pajak Penghasilan	Rp. 170.000
2	28-Jun-2022	Beban Administrasi	48	Ujian CPA Level Dasar - AAEP	Rp. 70.000
Total Beban :					Rp. 240.000

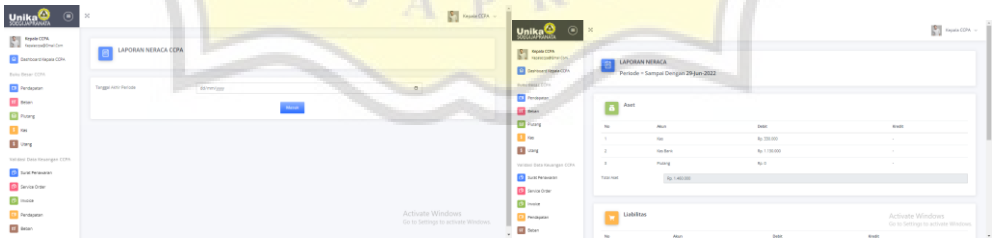
Laba Rugi

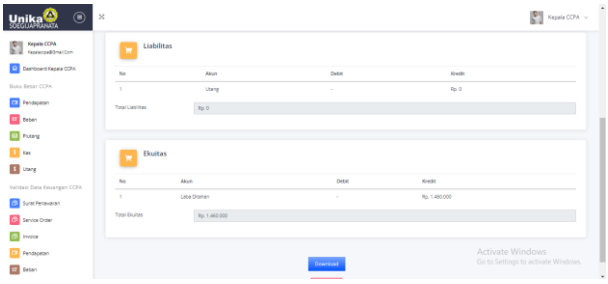
Perhitungan Laba Rugi	
Total Pendapatan :	Rp. 1.700.000
Total Beban :	Rp. 240.000
Laba :	Rp. 1.460.000

Gambar 4.180 Contoh Output Laporan Laba Rugi

o. Halaman Laporan Neraca

Gambar di bawah ini merupakan hasil akhir dari halaman laporan neraca. Di halaman ini user diminta untuk mengisi tanggal akhir untuk periode laporannya. Setelah itu user diminta menekan tombol masuk dan lalu user akan diarahkan ke halaman yang menampilkan informasi tentang laporan laba rugi.





Gambar 4.181 Halaman Laporan Neraca

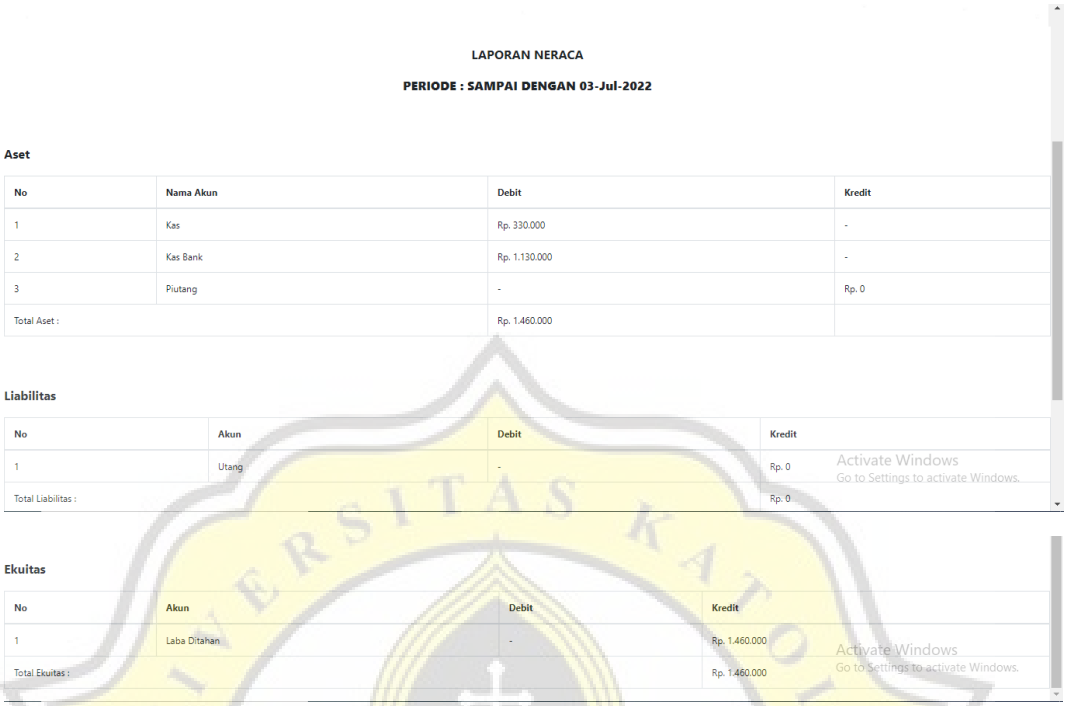
Gambar di bawah ini adalah *route* dan *controller* untuk fitur laporan neraca pada sistem ini.

```
Route ::get('/laporanneracaccpa', [KeuanganCCPAController::class, 'neracaccpa']);
Route ::post('/laporanneracaccpa2', [KeuanganCCPAController::class, 'insertneraca']);
```

```
public function neracaccpa()
{
    return view ('CCPA.neracaccpa');
}
```

```
public function insertneraca()
{
    Request()->validate([
        'tanggal_akhir' => 'required',
    ],[
        'tanggal_akhir.required'=>'Tanggal Akhir Belum diisi',
    ]);

    $tanggal_akhir = Request()->tanggal_akhir;
    $data = [
        'total_pendapatanneraca'=>$this->KeuanganCCPAModel->totalpendapatanneraca($tanggal_akhir),
        'total_bebanneraca'=>$this->KeuanganCCPAModel->totalbebanneraca($tanggal_akhir),
    ];
    //dd($data);
    $labarugi = ($data['total_pendapatanneraca'])-($data['total_bebanneraca']);
    $data2 = [
        'kas'=>$this->KeuanganCCPAModel->totalkas($tanggal_akhir),
        'kas_bank'=>$this->KeuanganCCPAModel->totalkasbank($tanggal_akhir),
        'piutang'=>$this->KeuanganCCPAModel->totalpiutang($tanggal_akhir),
        'utang'=>$this->KeuanganCCPAModel->totalutang($tanggal_akhir),
        'labarugi'=>$labarugi,
    ];
    $aset = ($data2['kas'])+($data2['kas_bank'])+($data2['piutang']);
    $liabilitas = ($data2['utang']);
    $ekuitas = ($data2['labarugi']);
    //dd($aset, $liabilitas);

    $datakeuangan = [
        'aset'=>$aset,
        'liabilitas'=>$liabilitas,
        'ekuitas'=>$ekuitas,
```

Gambar 4.182 Route dan Controller Laporan Neraca



LAPORAN NERACA

PERIODE : SAMPAI DENGAN 03-Jul-2022

Aset

No	Nama Akun	Debit	Kredit
1	Kas	Rp. 330.000	-
2	Kas Bank	Rp. 1.130.000	-
3	Piutang	-	Rp. 0
Total Aset :		Rp. 1.460.000	

Liabilitas

No	Akun	Debit	Kredit
1	Utang	-	Rp. 0
Total Liabilitas :			Rp. 0

Ekuitas

No	Akun	Debit	Kredit
1	Laba Ditahan	-	Rp. 1.460.000
Total Ekuitas :			Rp. 1.460.000

Gambar 4.183 Contoh Output Laporan Neraca

4.4 Pengujian Sistem

4.4.1 Hasil Pengujian Fungsionalitas

Pengujian fungsionalitas ini dilakukan dengan tujuan untuk memastikan seluruh komponen dan fitur yang ada di dalam sistem ini dapat bekerja dengan baik tanpa adanya masalah. Pengujian fungsionalitas ini dilakukan berdasarkan akses dan fungsi dari setiap fiturnya. Hasil pengujian fungsionalitas dari sistem informasi keuangan dan pelanggan adalah sebagai berikut.

Tabel 4.23 Hasil Uji Fungsionalitas

Modul	Pengujian	Hasil Pengujian
Login	Memastikan email dan password yang diinputkan telah sesuai dengan database,	Berjalan dengan baik

	Pengguna dapat masuk sesuai dengan hak aksesnya	
Register	Pengguna dapat membuat akun baru	Berjalan dengan baik
Logout	User dapat keluar dari sistem	Berjalan dengan baik
ADMIN		
Dashboard	Memastikan semua informasi (daftar project, jumlah dokumen yang belum divalidasi dan jumlah project berlangsung) yang dibutuhkan admin tersedia di *dashboard*	Berjalan dengan baik
Project	User dapat menambah, mengedit serta menghapus data project	Berjalan dengan baik
Master data akun	User dapat menambah dan mengedit master data akun	Berjalan dengan baik
Master data SDM	User dapat menambah, mengedit serta menghapus data sdm	Berjalan dengan baik
Master data ujian	User dapat menambah, mengedit serta menghapus data ujian	Berjalan dengan baik
Master data pelanggan	User dapat menambah, mengedit serta menghapus data pelanggan	Berjalan dengan baik
Surat penawaran	User dapat menambah, mengedit serta menghapus dan mengunduh surat penawaran	Berjalan dengan baik
Service Order	User dapat menambah, mengedit serta menghapus dan mengunduh *service order*	Berjalan dengan baik
Invoice	User dapat menambah, mengedit serta menghapus dan mengunduh *invoice*	Berjalan dengan baik
Pendapatan	User dapat menambah, mengedit serta menghapus data pendapatan	Berjalan dengan baik

Beban	User dapat menambah, mengedit serta menghapus data beban	Berjalan dengan baik
Piutang	User dapat melihat daftar piutang yang ada serta dapat melunasi piutang	Berjalan dengan baik
Kas Kecil	User dapat melihat informasi tentang arus keluar masuk kas	Berjalan dengan baik
KEPALA		
Dashboard	Memastikan semua informasi (daftar project, jumlah dokumen yang belum divalidasi, saldo pendapatan, beban dan kas serta jumlah project berlangsung) yang dibutuhkan admin tersedia di *dashboard*	Berjalan dengan baik
Buku Besar Pendapatan	User dapat melihat informasi tentang buku besar akun pendapatan	Berjalan dengan baik
Buku Besar Beban	User dapat melihat informasi tentang buku besar akun beban	Berjalan dengan baik
Buku Besar Kas	User dapat melihat informasi tentang buku besar akun kas	Berjalan dengan baik
Buku Besar Piutang	User dapat melihat informasi tentang buku besar akun piutang	Berjalan dengan baik
Buku Besar Utang	User dapat melihat informasi tentang buku besar akun utang	Berjalan dengan baik
Validasi Surat Penawaran	User dapat melihat detail surat penawaran dan mevalidasi surat penawaran	Berjalan dengan baik
Validasi *Sevice Order*	User dapat melihat detail *service order* dan mevalidasi *service order*	Berjalan dengan baik
Validasi *Invoice*	User dapat melihat detail *invoice* dan mevalidasi *invoice*	Berjalan dengan baik

Validasi Pendapatan	User dapat melihat detail pendapatan dan mevalidasi pendapatan	Berjalan dengan baik
Validasi Beban	User dapat melihat detail beban dan mevalidasi beban	Berjalan dengan baik
Validasi Kas	User dapat melihat detail kas dan mevalidasi kas	Berjalan dengan baik
Validasi Piutang	User dapat melihat detail piutang dan mevalidasi piutang	Berjalan dengan baik
Laporan Laba Rugi Project	User dapat melihat serta mengunduh laporan laba rugi per project	Berjalan dengan baik
Laporan Laba Rugi	User dapat melihat serta mengunduh laporan laba rugi dalam suatu periode	Berjalan dengan baik
Laporan Neraca	User dapat melihat serta mengunduh laporan neraca dalam suatu peride	Berjalan dengan baik

4.4.2 Hasil Wawancara

Kegiatan wawancara ini dilakukan dengan tujuan untuk memastika apakah sistem yang dikembangkan sudah sesuai dengan kebutuhan instansi dan apakah sistem yang dikembangkan ini mudah dipahami dan digunakan oleh user. Proses wawancara ini dilakukan dengan dua orang yakni admin CCPA dan kepala CCPA. Berikut ini hasil dari wawancara :

Tabel 4.24 Tabel Hasil Wawancara

No.	**Pertanyaan**	**Jawaban**	
		Admin	**CCPA**
PERFORMENT EXPECTANCY			

1.	Apakah dengan sistem ini anda dapat mudah melakukan perintah penginputan, pengeditan dan penghapusan data (baik untuk data dokumen (SP, SO, dan Invoice) maupun untuk data keuangan serta data pelanggan ? Jelaskan?	Iya, mudah. Tetapi untuk tombol penghapusan data jangan diberikan ke semua pihak alias satu pihak saja yang memiliki akses tersebut	Ya. Semua menu transaksi yg dibutuhkan sudah tersedia dan dibuat sesuai kebutuhan unik unit.
2.	Apakah dengan adanya penggunaan sistem membantu kinerja anda dalam hal pencatatan data keuangan, data pelanggan, dan data dokumen keuangan yang ada di CCPA dan P3A? Jelaskan?	Iya, membantu. Sebab akan terekam di dalam sistem daripada melalui manual yang bisa saja hilang	Ya. Fitur penting selain pencatatan keuangan adalah adanya data pelanggan dan proyek.
3.	Menurut anda, apakah sistem dapat memproses data-data yang telah diinputkan secara cepat dan tepat? Jelaskan?	Iya, bagi saya cepat. Karena bisa langsung menemukan data yang dicari, daripada menggunakan cara manual	Dapat menyimpan data dengan tepat dan menghasilkan laporan lebih cepat. Tetapi dari sisi input, masih bisa ditingkatkan.
4.	Menurut anda, Apakah sistem dapat terhubung secara langsung dan realtime ke jurnal, neraca saldo, dan laporan laba rugi (modul akuntansi)dengan tepat dan benar? Jelaskan?	Iya, bagi saya cepat. Karena bisa langsung menemukan data yang dicari, daripada menggunakan cara manual	Ya. Penjurnalan otomatis sudah sesuai dengan kaidah akuntansi.
5.	Menurut anda, Apakah sistem ini dapat langsung terhubung secara langsung dan realtime untuk mengakses data pelanggan (modul pelanggan)?Jelaskan?	Iya, bagi saya cepat. Karena bisa langsung menemukan data yang dicari, daripada menggunakan cara manual	Ya. Penelusuran per project sangat membantu
6.	Menurut anda, apakah dengan adanya penggunaan sistem dapat menghemat waktu, tenaga dan pengeluaran yang ada sebelumnya? Jelaskan?	Iya, menghemat waktu kami	Menghemat waktu dan tenaga untuk membuat laporan.
EFFORT EXPECTANCY			
1.	Apakah tampilan layar(user interface) membuat anda nyaman untuk bekerja menggunakan sistem ini? Jelaskan?	Iya, nyaman	Cukup nyaman, tp peletakan menu transaksi masih bisa disimplifikasi.

2.	Menurut anda, apakah alur penggunaan sistem mudah dipahami dan dipelajari? Jelaskan?	Iya, mudah untuk dipahami	Ya.
3.	Menurut anda, apakah bahasa yang digunakan di sistem mudah dimengerti? Jelaskan?	Iya, mudah dimengerti	Ya.
SOCIAL INFLUENCE			
1.	Menurut anda, apakah sistem yang ada telah sesuai dengan standar sistem-sistem keuangan yang ada saat ini? Jelaskan?	Iya, telah sesuai	Secara fungsional sudah sesuai
2.	Menurut anda,apakah orang-orang di sekitar anda mendukung untuk menggunakan sistem ini untuk menunjang kinerja? Jelaskan?	Iya, mendukung	Ya. Sistem ini layak untuk digunakan secara riil.
3.	Menurut anda, apakah orang-orang sekitar anda akan membantu, ketika anda mengalami kesulitan dalam menggunakan sistem ini? Jelaskan?	Kurang tahu	Ya
FACILITATING CONDITIONS			
1.	Apakah anda mempunyai pengalaman dan kemampuan dalam menggunakan sistem keuangan? Jelaskan?	Iya, saya memiliki kemampuan itu	Ya. Memiliki pengalaman dengan beberapa software ERP
2.	Apakah sarana (laptop atau komputer dan jaringan internet) yang ada di CCPA dapat menunjang penggunaan sistem ini? Jelaskan?	Iya, dapat menunjang penggunaan sistem	Ya
3.	Apakah ada tenaga ahli yang dapat membantu anda ketika anda mengalami kesulitan dalam menggunakan sistem ini? Jelaskan?	Kurang tahu	Ya
4.	Apakah peraturan dan kebiasaan kerja yang ada di CCPA dapat menunjang penggunaan sistem ini? Jelaskan?	Iya, menunjang	Ya. CCPA memiliki keharusan melakukan pelaporan keuangan dan butuh melakukan analisa statistik pelanggan.

HEDONIC MOTIVATION			
1.	Menurut Anda, apakah anda merasa senang setelah menggunakan sistem ini? Jelaskan?	Iya, saya merasa senang	Ya. Sangat membantu merapikan pekerjaan administratif.
2.	Menurut anda, apakah sistem ini membantu anda bekerja? Jelaskan?	Bagi saya sangat membantu	Ya. Fitur pelanggan dan project sangat sesuai dengan kebutuhan
BEHAVIOR INTENTION			
1.	Apakah anda ingin menggunakan sistem ini secara terus menerus untuk ke depannya? Jelaskan?	Iya, saya akan menggunakan sistem ini terus-menerus	Ya
2.	Apakah sistem ini sudah berjalan sesuai dengan yang anda harapkan? Jelaskan?	Iya, walaupun ada sedikit yang perlu diperbaiki	Secara fungsi sudah.
3.	Apakah anda lebih memilih menggunakan sistem ini daripada menggunakan sistem manual? Jelaskan alasannya?	Iya, karena memudahkan pekerjaan saya	Ya. Data tersimpan lebih rapi dan terpusat. Lebih cepat menghasilkan laporan.

Berdasarkan hasil wawancara diatas dapat disimpulkan bahwa sistem informasi manajemen keuangan dan pelanggan ini, telah sesuai dengan kebutuhan CCPA. Hal ini dapat dibuktikan dengan merasa terbantunya admin dan kepala semenjak adanya penggunaan sistem ini. Dengan adanya sistem ini dapat membantu admin dan juga kepala dalam mengatur data keuangan dan pelanggan dengan lebih baik lagi. Dari hasil wawancara diatas juga dapat kita simpulkan bahwa sistem ini memiliki alur dan tampilan yang mudah dipahami oleh user.