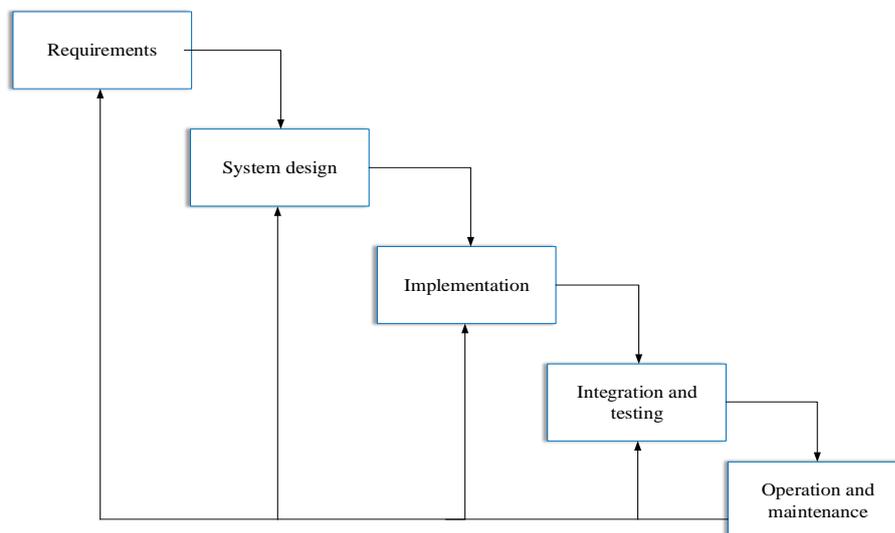


BAB III

METODE PENELITIAN

3.1 Arsitektur Aplikasi

Pada bab ini akan digambarkan rancangan web administrasi personalia. Rancangan web administrasi personalia dirancang menggunakan metode *waterfall*. Metode ini memungkinkan untuk departementalisasi dan kontrol. Proses pengembangan model fase *one by one*, sehingga meminimalisir kesalahan yang mungkin akan terjadi. Pengembangan bergerak dari konsep, yaitu melalui desain, implementasi, pengujian, instalasi, penyelesaian masalah, dan berakhir di operasi dan pemeliharaan. Tahapan metode *waterfall* dapat dilihat pada gambar 3.1.



Gambar 3.1 Tahapan metode *waterfall*

Dalam pengembangannya metode *waterfall* memiliki beberapa tahapan yang berurut yaitu: *requirement* (analisis kebutuhan), *design system* (desain sistem), *Coding* (pengkodean) & *Testing* (pengujian), Penerapan Program, pemeliharaan. Tahapan tahapan dari metode *waterfall* adalah sebagai berikut :

1. *Requirement* Analisis

Pada tahap ini dilakukan analisa kebutuhan yang bertujuan untuk memahami kebutuhan pengguna dan batasan perangkat lunak tersebut. Informasi ini biasanya dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang

dibutuhkan oleh pengguna. Analisa kebutuhan akan dijelaskan pada sub bab 3.3.

2. *System design*

System design digunakan untuk mengubah analisa kebutuhan menjadi representasi ke dalam bentuk *blueprint software* sebelum implementasi dimulai. Tahap *system design* menyangkut perancangan sistem dimana kita akan memberikan solusi dari masalah yang muncul pada tahap analisis. Perancangan menggunakan *Use case diagram*, *Activity diagram*, *ER diagram*, *Sequence diagram*, *Class diagram* dan perancangan antarmuka. Tahap *system design* akan dijelaskan pada sub bab 3.4.

3. *Implementation*

Pada tahap ini, keseluruhan desain diubah menjadi kode-kode program. kode program yang dihasilkan masih berupa modul-modul yang selanjutnya akan diintegrasikan menjadi program yang lengkap untuk meyakinkan bahwa persyaratan perangkat lunak telah dipenuhi. Tahap *implemetation* akan dijelaskan pada sub bab 4.1.

4. *Integration & Testing*

Seluruh *unit* yang dikembangkan pada tahap implementasi akan diintegrasikan ke dalam sistem. Setelah integrasi, seluruh sistem diuji untuk mengecek setiap kegagalan maupun kesalahan. Tahap *testing* akan dijelaskan pada sub bab 4.1.3.

5. *Operation & Maintenance*

Pada tahap ini perangkat lunak yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada tahap *testing* karena perangkat lunak harus menyesuaikan dengan sistem operasi dan perangkat baru. *Operation* dan *maintenance* dilakukan untuk perkembangan fungsionalitas.

3.2 Analisis Kebutuhan

Analisis kebutuhan didapat dari observasi dan wawancara. Berikut merupakan analisis kebutuhan pada web :

1. *Login* untuk dapat menggunakan web.

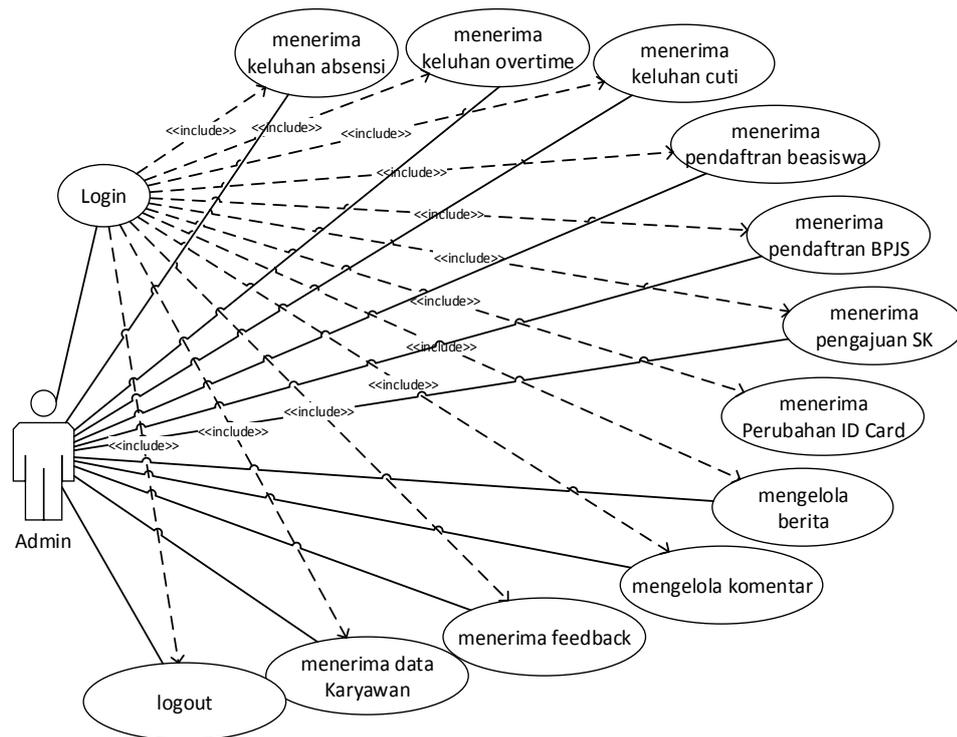
2. Dapat melihat, merubah dan menghapus keluhan absensi
3. Dapat melihat, merubah dan menghapus keluhan *overtime*
4. Dapat melihat, merubah dan menghapus keluhan cuti
5. Dapat melihat, merubah dan menghapus pendaftaran beasiswa
6. Dapat melihat, merubah dan menghapus pendaftaran BPJS
7. Dapat melihat, merubah dan menghapus pengajuan surat keterangan
8. Dapat melihat, merubah dan menghapus perubahan *id card*
9. Dapat melihat, membuat dan menghapus *newsfeed*
10. Dapat melihat, membuat dan menghapus komentar
11. Dapat melihat dan menghapus *feedback*
12. Dapat melihat, membuat, merubah dan menghapus karyawan
13. *Logout* untuk keluar dari web

3.3 Rancangan

Dalam perancangan *logic* aplikasi ini menggunakan *United Markup Language* (UML). Model UML yang digunakan dalam pengembangan web admini yaitu *use case diagram*, *sequence diagram*, *activity diagram*, *Entity Relationship Diagram* dan *class diagram*.

3.3.1 Use Case Diagram

Berdasarkan analisis kebutuhan pada sub bab 3.2, maka dibuatlah *use case diagram* untuk membantu perancangan *logic* pada web. Pada *use case diagram* memiliki 1 aktor yaitu admin. Gambaran *use case diagram* dapat dilihat pada gambar 3.2.



Gambar 3.2 Use Case Diagram Admin

Berikut adalah penjelasan tentang gambar 3.2 :

1. *Login* : memungkinkan admin untuk menggunakan web admin
2. Menerima keluhan absensi : memungkinkan admin unutkan dapat melihat, menghapus dan *update* mengenai keluhan absensi
3. Menerima keluhan *overtime* : memungkinkan admin unutkan dapat melihat, menghapus dan *update* mengenai keluhan *overtime*
4. Menerima keluhan cuti : memungkinkan admin unutkan dapat melihat, menghapus dan *update* mengenai keluhan cuti
5. Menerima pendaftaran beasiswa : memungkinkan admin unutkan dapat melihat, menghapus dan *update* mengenai pendaftaran beasiswa
6. Menerima pendaftaran BPJS : memungkinkan admin unutkan dapat melihat, menghapus dan *update* mengenai pendaftaran BPJS
7. Menerima pengajuan surat keterangan: memungkinkan admin unutkan dapat melihat, menghapus dan *update* mengenai pengajuan surat keterangan
8. Menerima perubahan *id card* : memungkinkan admin unutkan dapat melihat, menghapus dan *update* mengenai perubahan *id card*

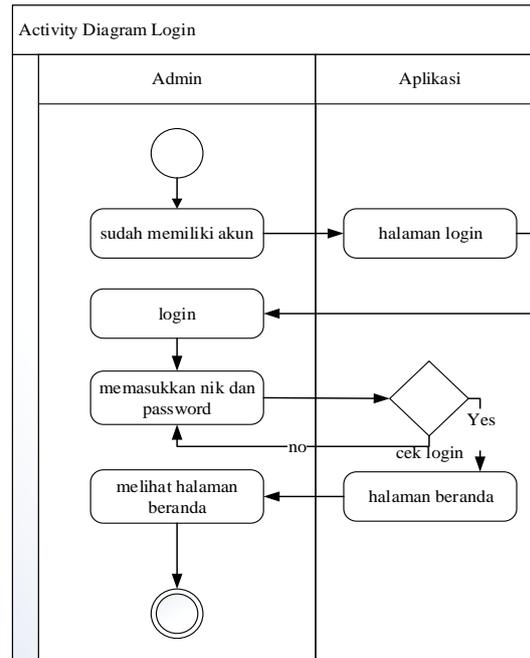
9. Mengelola *newfeed* : memungkinkan admin unutup dapat melihat, menghapus dan menambah mengenai *newsfeed*
10. Mengelola komentar : memungkinkan admin unutup dapat melihat, menghapus dan *update* mengenai komentar
11. Menerima *feedback* : memungkinkan admin unutup dapat melihat, menghapus dan *update* mengenai *feedback*
12. Menerima data karyawan : memungkinkan admin unutup dapat melihat, menghapus dan *update* mengenai karyawan
13. Logout : memungkinkan admin untuk tidak lagi menggunakan web

4.3.2 Activity Diagram

Berdasarkan *use case* diagram pada gambar 3.2, maka diperoleh *activity* diagram berdasarkan aktor yang terlibat dalam *use case* diagram. *Activity* diagram dibagi menjadi beberapa bagian, yaitu :

1. Activity Diagram Login

Activity diagram *login* dirancang berdasarkan *case login* pada *use case* diagram pada gambar 3.2. Pada *activity* diagram *login* yang akan menjelaskan proses saat admin akan melakukan *login*. Untuk melakukan *login* admin akan memasukkan nik dan kata sandi yang telah dimiliki admin. Setelah admin memasukkan nik dan kata sandi, aplikasi akan melakukan validasi nik dan password admin benar atau tidak. Jika salah admin dapat memasukkan kembali nik dan kata sandi. Jika nik dan kata sandi benar aplikasi akan menampilkan halaman beranda. *Activity* diagram *login* dapat dilihat pada gambar 3.3.



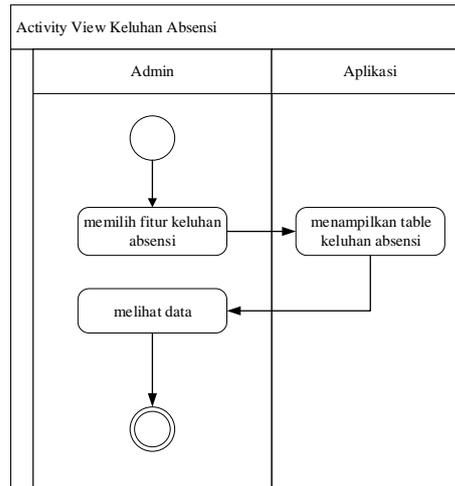
Gambar 3.3 Activity diagram login

2. Activity Diagram Menerima Keluhan Absensi

Activity diagram menerima keluhan absensi dirancang berdasarkan case menerima keluhan absensi pada use case diagram gambar 3.2 yang dibagi menjadi 3 bagian, yaitu:

Activity diagram view keluhan absensi

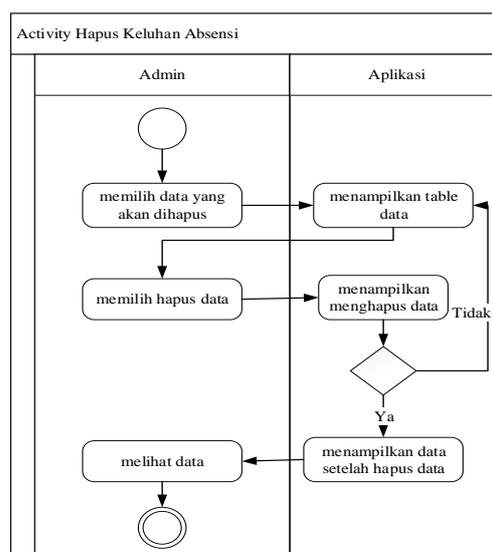
Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melihat data keluhan absensi yang masuk pada aplikasi. Pertama-tama admin memilih fitur keluhan absensi maka admin langsung bisa menuju ke halaman keluhan absensi dan langsung dapat melihat data yang masuk. *Activity* diagram dapat di lihat pada gambar 3.4.



Gambar 3.4 Activity diagram view keluhan absensi

Activity Diagram Hapus Keluhan Absensi

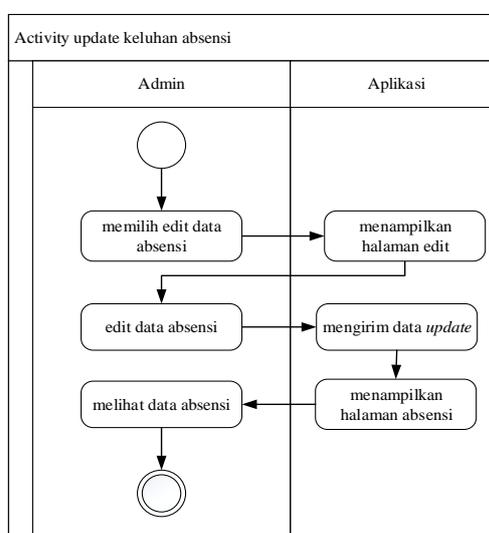
Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melakukan penghapusan keluhan absensi yang masuk pada aplikasi. Pertama-tama admin memilih fitur keluhan absensi maka admin langsung bisa menuju ke halaman keluhan absensi dan setelah itu admin memilih untuk menghapus data yang ada, disini akan ada pilihan seperti jika ingin menghapus maka admin akan menuju halaman keluhan absensi yang terbaru dan jika admin mengurungkan untuk menghapus maka admin akan tetap di halaman . *Activity* diagram dapat di lihat pada gambar 3.5.



Gambar 3.5 Activity diagram hapus keluhan absensi

Activity Diagram Update Keluhan Absensi

Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melakukan *update* keluhan absensi pada aplikasi yang sudah ada. Pertama-tama admin memilih fitur keluhan absensi, setelah itu admin langsung bisa menuju ke halaman keluhan absensi dan setelah itu admin memilih data untuk *update*. Lalu admin mengirim keluhan, setelah mengirim admin langsung menuju ke halaman keluhan absensi dari aplikasi. *Activity* diagram dapat di lihat pada gambar 3.6.



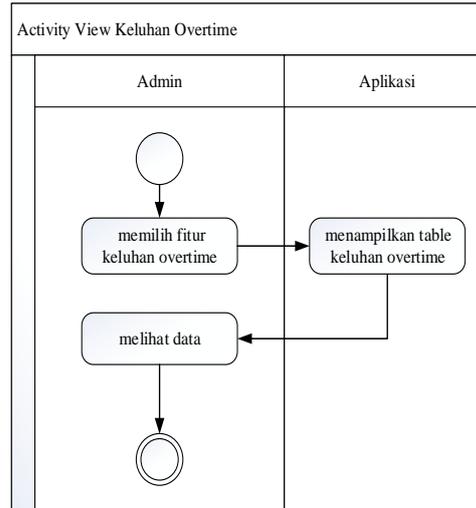
Gambar 3.6 Activity diagram *update* keluhan absensi

3. Activity Diagram Menerima Keluhan Overtime

Activity diagram menerima keluhan overtime dirancang berdasarkan case menerima keluhan overtime pada use case diagram gambar 3.2 yang dibagi menjadi 3 bagian, yaitu:

Activity Diagram View Keluhan Overtime

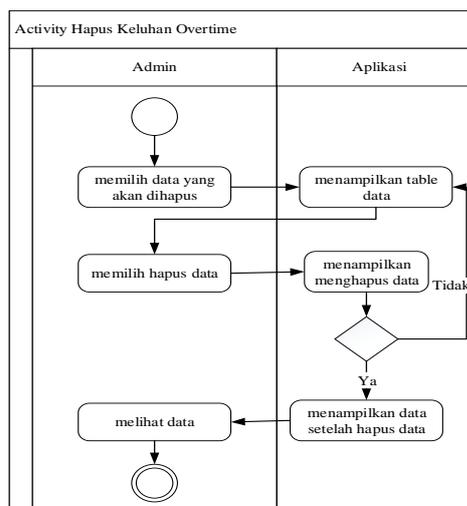
Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melihat data keluhan overtime yang masuk pada aplikasi. Pertama-tama admin memilih fitur keluhan overtime, maka admin langsung bisa menuju ke halaman keluhan absensi dan langsung dapat melihat keluhan yang masuk. *Activity* diagram dapat di lihat pada gambar 3.7.



Gambar 3.7 Activity diagram view keluhan overtime

Activity Diagram Hapus Keluhan Overtime

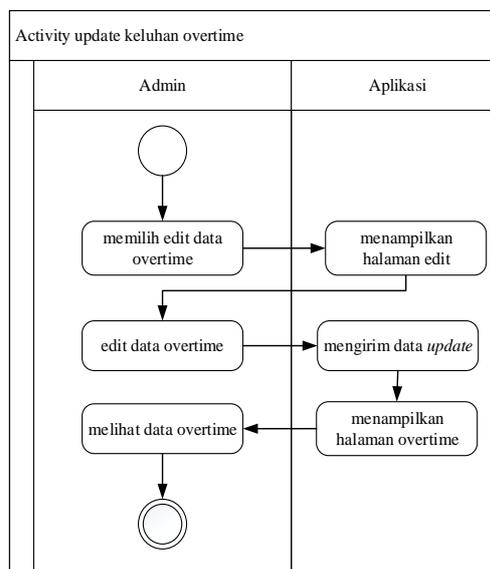
Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melakukan penghapusan data keluhan overetime yang masuk pada aplikasi. Pertama-tama admin memilih fitur keluhan overtime maka admin langsung bisa menuju ke halaman keluhan overtime dan setelah itu admin meilih untuk menghapus keluhan yang ada, disini akan ada pilihan seperti jika ingin menghapus maka admin akan menuju halaman keluhan overtime yang terbaru dan jika admin mengurungkan untuk menghapus maka admin akan tetap di halaman . *Activity* diagram dapat di lihat pada gambar 3.8.



Gambar 3.8 Activity diagram hapus keluhan overtime

Activity Diagram *Update* Keluhan Overtime

Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melakukan *update* data keluhan overtime pada aplikasi yang sudah ada. Pertama-tama admin memilih fitur keluhan overtime, setelah itu admin langsung bisa menuju ke halaman keluhan overtime dan setelah itu admin memilih keluhan untuk *update*. Lalu admin mengirim, setelah mengirim admin langsung menuju ke halaman keluhan overtime dari aplikasi. *Activity* diagram dapat di lihat pada gambar 3.9.



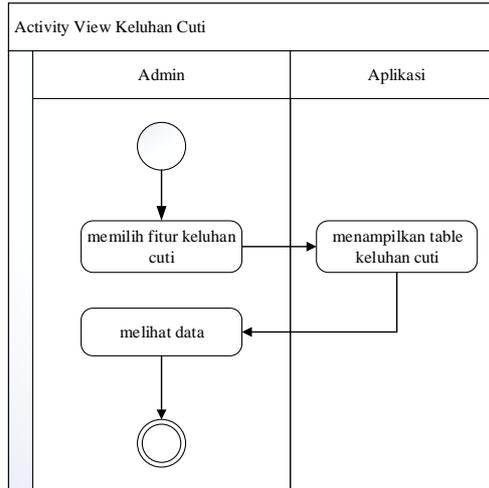
Gambar 3.9 Activity diagram *update* keluhan overtime

4. Activity Diagram Menerima Keluhan Cuti

Activity diagram menerima keluhan cuti dirancang berdasarkan case menerima keluhan cuti pada use case diagram gambar 3.2 yang dibagi menjadi 3 bagian, yaitu:

Activity Diagram *View* Keluhan Cuti

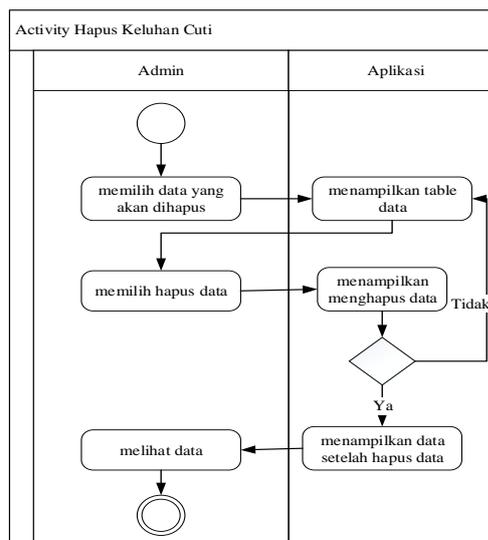
Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melihat data keluhan overtime yang masuk pada aplikasi. Pertama-tama admin memilih fitur keluhan overtime, maka admin langsung bisa menuju ke halaman keluhan absensi dan langsung dapat melihat keluhan yang masuk. *Activity* diagram dapat di lihat pada gambar 3.10.



Gambar 3.10 Activity diagram *view* keluhan cuti.

Activity Diagram Hapus Cuti

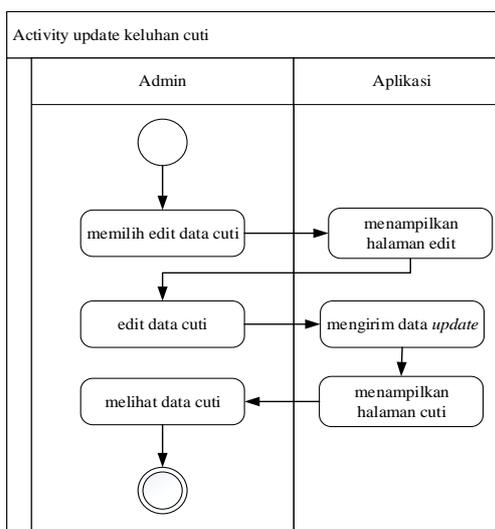
Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melakukan penghapusan data keluhan cuti yang masuk pada aplikasi. Pertama-tama admin memilih fitur keluhan cuti maka admin langsung bisa menuju ke halaman keluhan cuti dan setelah itu admin memilih untuk menghapus keluhan yang ada, disini akan ada pilihan seperti jika ingin menghapus maka admin akan menuju halaman keluhan cuti yang terbaru dan jika admin mengurungkan untuk menghapus keluhan maka admin akan tetap di halaman. *Activity* diagram dapat di lihat pada gambar 3.11.



Gambar 3.11 Activity diagram hapus keluhan cuti

Activity Diagram *Update* Keluhan Cuti

Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melakukan *update* keluhan cuti pada aplikasi yang sudah ada. Pertama-tama admin memilih fitur keluhan cuti, setelah itu admin langsung bisa menuju ke halaman keluhan cuti dan setelah itu admin memilih untuk *update*. Lalu admin mengirim, setelah mengirim data admin langsung menuju ke halaman keluhan cuti dari aplikasi. *Activity* diagram dapat di lihat pada gambar 3.12.



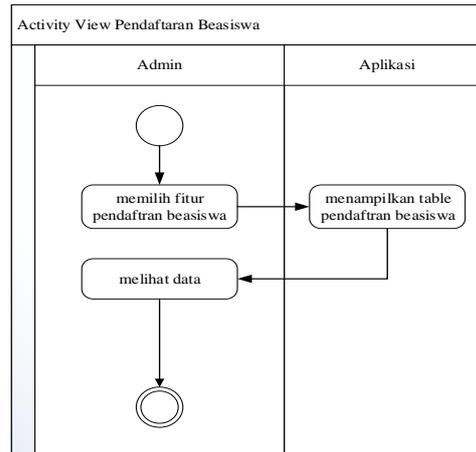
Gambar 3.12 Activity diagram *update* keluhan cuti

5. Activity Diagram Menerima Pendaftaran beasiswa

Activity diagram pendaftaran beasiswa dirancang berdasarkan case menerima keluhan cuti pada use case diagram gambar 3.2 yang dibagi menjadi 3 bagian, yaitu:

Activity Diagram *View* Beasiswa

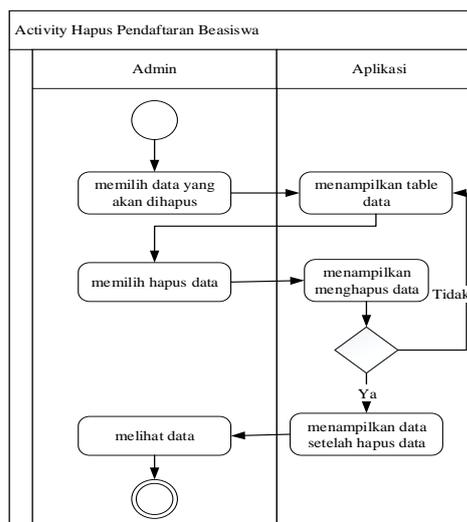
Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melihat pengajuan beasiswa yang masuk pada aplikasi. Pertama-tama admin memilih fitur pengajuan beasiswa, maka admin langsung bisa menuju ke halaman pengajuan beasiswa dan langsung dapat melihat keluhan yang masuk. *Activity* diagram dapat di lihat pada gambar 3.13.



Gambar 3.13 Activity diagram *View* beasiswa

Activity Diagram Hapus Pendaftaran Beasiswa

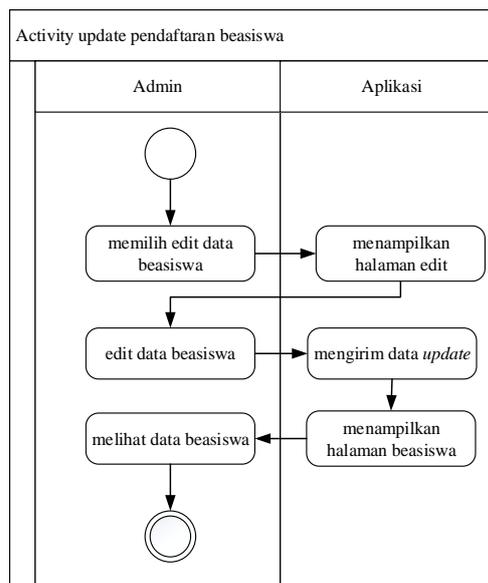
Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melakukan penghapusan data pengajuan beasiswa yang masuk pada aplikasi. Pertama-tama admin memilih fitur pengajuan beasiswa maka admin langsung bisa menuju ke halaman pengajuan beasiswa dan setelah itu admin memilih untuk menghapus keluhan yang ada, disini akan ada pilihan seperti jika ingin menghapus maka admin akan menuju halaman pengajuan beasiswa yang terbaru dan jika admin mengurungkan untuk menghapus maka admin akan tetap di halaman. *Activity* diagram dapat di lihat pada gambar 3.14.



Gambar 3.14 Activity diagram hapus beasiswa

Activity Diagram *Update* Beasiswa

Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melakukan pengupdatean beasiswa pada aplikasi yang sudah ada. Pertama-tama admin memilih fitur beasiswa, setelah itu admin langsung bisa menuju ke halaman beasiswa dan setelah itu admin memilih untuk diupdate. Lalu admin mengirim data, setelah mengirim admin langsung menuju ke halaman beasiswa dari aplikasi. *Activity* diagram dapat di lihat pada gambar 3.15.



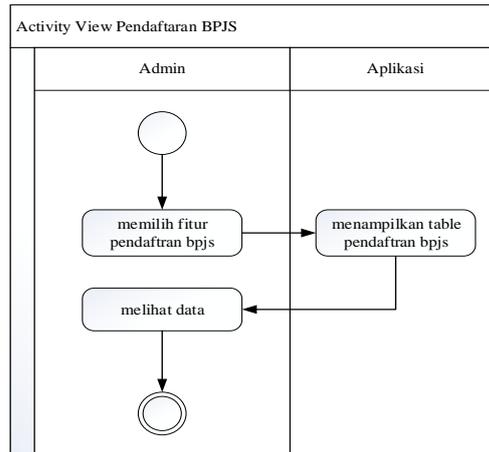
Gambar 3.15 Activity diagram *update* beasiswa

6. Activity Diagram Menerima Pendaftaran BPJS

Activity diagram menerima pendaftaran BPJS dirancang berdasarkan case menerima pendaftaran BPJS pada use case diagram gambar 3.2 yang dibagi menjadi 3 bagian, yaitu:

Activity Diagram *View* Pendaftaran BPJS

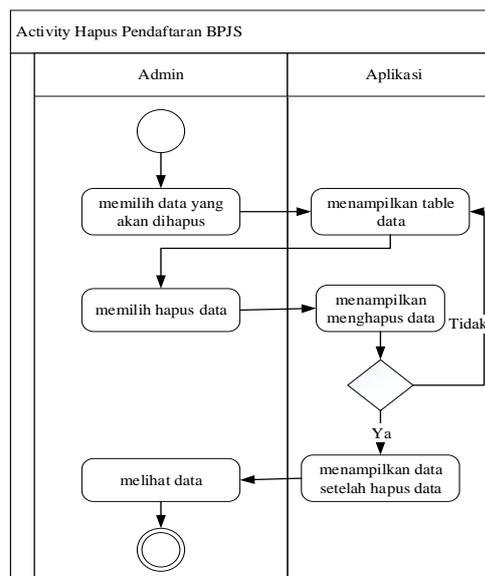
Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melihat data pengajuan BPJS yang masuk pada aplikasi. Pertama-tama admin memilih fitur pengajuan BPJS, maka admin langsung bisa menuju ke halaman pengajuan BPJS dan langsung dapat melihat yang ada. *Activity* diagram dapat di lihat pada gambar 3.16.



Gambar 3.16 Activity diagram *view* BPJS

Activity Diagram Hapus Pendaftaran BPJS

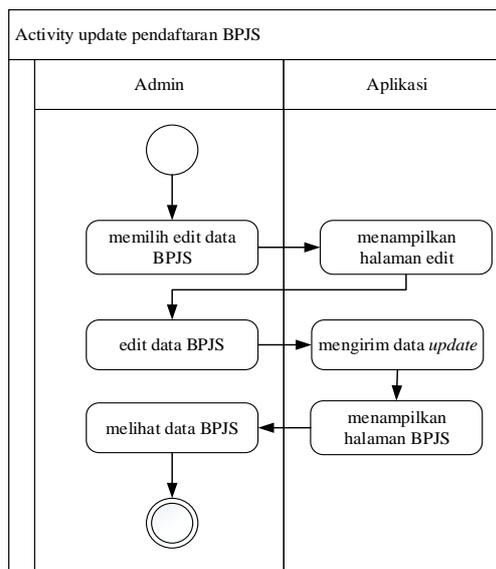
Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melakukan penghapusan data pengajuan BPJS yang masuk pada aplikasi. Pertama-tama admin memilih fitur pengajuan beasiswa, setelah itu admin langsung bisa menuju ke halaman pengajuan BPJS dan setelah itu admin melihat untuk menghapus, disini akan ada pilihan seperti jika ingin menghapus maka admin akan menuju halaman pengajuan BPJS yang terbaru dan jika admin mengurungkan untuk menghapus maka admin akan tetap di halaman. *Activity* diagram dapat di lihat pada gambar 3.17.



Gambar 3.17 Activity diagram hapus BPJS

Activity Diagram *Update* BPJS

Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melakukan *update* BPJS pada aplikasi yang sudah ada. Pertama-tama admin memilih fitur BPJS, setelah itu admin langsung bisa menuju ke halaman BPJS dan setelah itu admin memilih untuk *update*. Lalu admin mengirim, setelah mengirim data admin langsung menuju ke halaman BPJS dari aplikasi. *Activity* diagram dapat dilihat pada gambar 3.18.



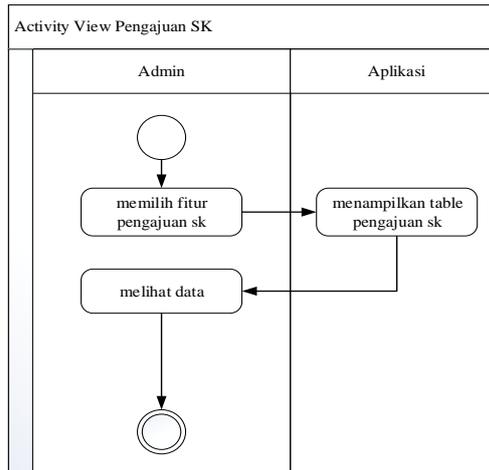
Gambar 3.18 Activity diagram *update* BPJS

7. Activity Diagram Menerima Pengajuan Surat Keterangan

Activity diagram menerima pengajuan surat keterangan dirancang berdasarkan case menerima pengajuan surat keterangan pada use case diagram gambar 3.2 yang dibagi menjadi 3 bagian, yaitu:

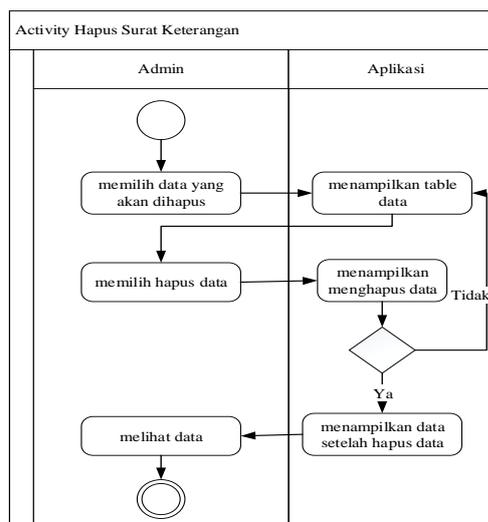
Activity Diagram *View* Pengajuan Surat Keterangan

Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melihat data pengajuan surat keterangan yang masuk pada aplikasi. Pertama-tama admin memilih fitur pengajuan surat keterangan, maka admin langsung bisa menuju ke halaman pengajuan surat keterangan dan langsung dapat melihat. *Activity* diagram dapat dilihat pada gambar 3.19.



Gambar 3.19 Activity diagram *view* pengajuan surat keterangan
Activity Diagram Hapus Pengajuan Surat Keterangan (SK)

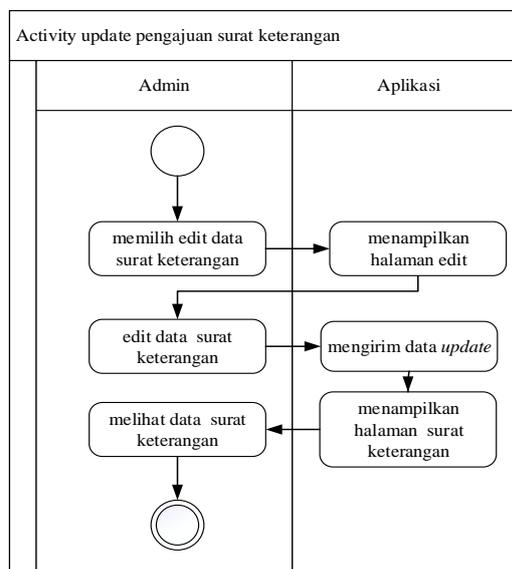
Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melakukan penghapusan pengajuan surat keterangan yang masuk pada aplikasi. Pertama-tama admin memilih fitur pengajuan surat keterangan, setelah itu admin langsung bisa menuju ke halaman pengajuan SK dan setelah itu admin memilih untuk menghapus yang ada, disini akan ada pilihan seperti jika ingin menghapus maka admin akan menuju halaman pengajuan surat keterangan yang terbaru dan jika admin mengurungkan untuk menghapus maka admin akan tetap di halaman. *Activity* diagram dapat di lihat pada gambar 3.20.



Gambar 3.20 Activity diagram hapus pengajuan surat keterangan

Activity Diagram *Update* Pengajuan Surat Keterangan

Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melakukan *update* data pengajuan surat keterangan pada aplikasi yang sudah ada. Pertama-tama admin memilih fitur pengajuan surat keterangan, setelah itu admin langsung bisa menuju ke halaman pengajuan surat keterangan dan setelah itu admin memilih data untuk *update*. Lalu admin mengirim data, setelah mengirim data admin langsung menuju ke halaman pengajuan surat keterangan dari aplikasi. *Activity* diagram dapat di lihat pada gambar 3.21.



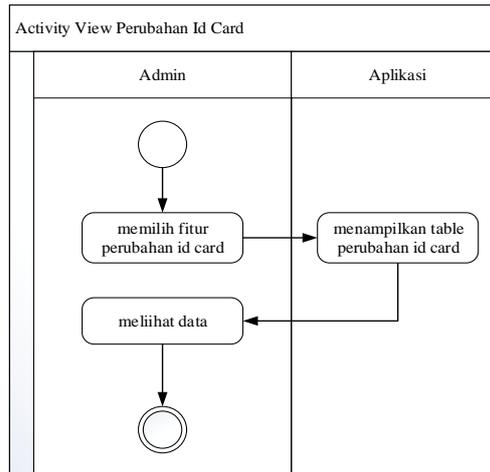
Gambar 3.21 Activity diagram *update* pengajuan surat keterangan

8. Activity Diagram Menerima Perubahan Id Card

Activity diagram menerima perubahan id card dirancang berdasarkan case menerima perubahan id card pada use case diagram gambar 3.2 yang dibagi menjadi 3 bagian, yaitu:

Activity Diagram *View* Perubahan Id Card

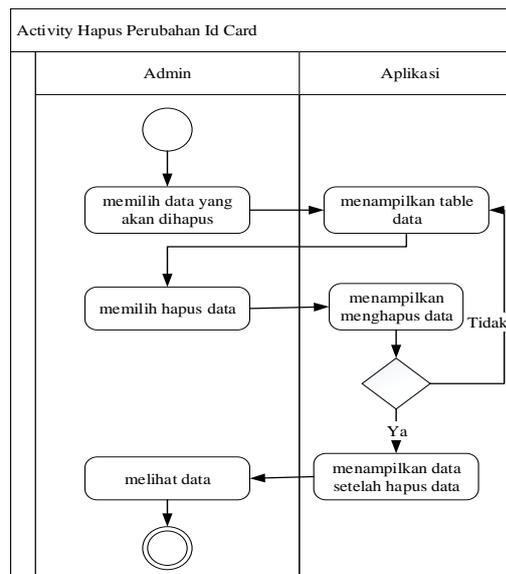
Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melihat data perubahan id card yang masuk pada aplikasi. Pertama-tama admin memilih fitur perubahan id card, maka admin langsung bisa menuju ke halaman perubahan id card dan langsung dapat melihat perubahan id card. *Activity* diagram dapat di lihat pada gambar 3.22.



Gambar 3.22 Activity diagram *view* perubahan id card

Activity Diagram Hapus Perubahan Id Card

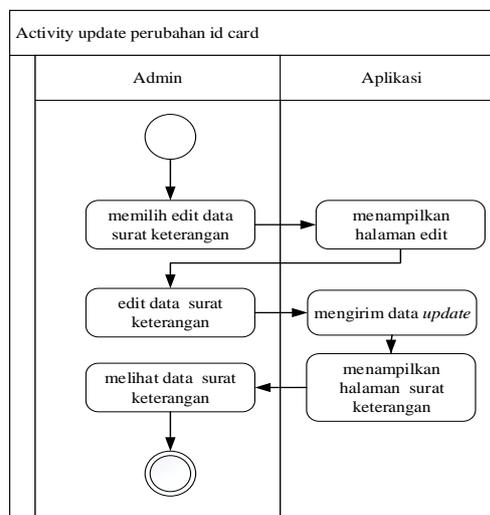
Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melakukan penghapusan perubahan id card yang masuk pada aplikasi. Pertama-tama admin memilih fitur perubahan id card, setelah itu admin langsung bisa menuju ke halaman perubahan id card dan setelah itu admin memilih untuk menghapus, disini akan ada pilihan seperti jika ingin menghapus maka admin akan menuju halaman perubahan id card yang terbaru dan jika admin mengurungkan untuk menghapus maka admin akan tetap di halaman . *Activity* diagram dapat di lihat pada gambar 3.23.



Gambar 3.23 Activity diagram hapus perubahan id card

Activity Diagram *Update* Perubahan Id Card

Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melakukan *update* perubahan id card pada aplikasi yang sudah ada. Pertama-tama admin memilih fitur perubahan id card, setelah itu admin langsung bisa menuju ke halaman keluhan cuti dan setelah itu admin memilih data untuk *update*. Lalu admin mengirim, setelah mengiri data admin langsung menuju ke halaman perubahan id card dari aplikasi. *Activity* diagram dapat di lihat pada gambar 3.24.



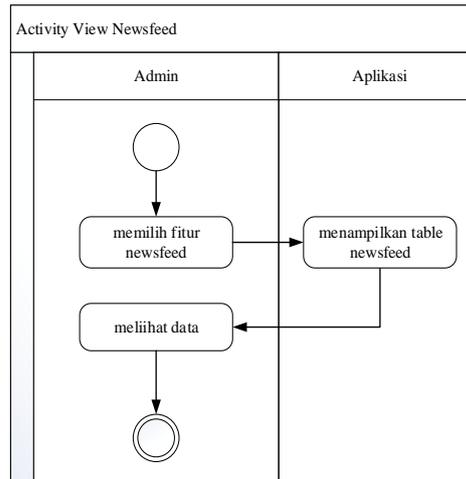
Gambar 3.24 Activity diagram *update* perubahan id card

9. Activity Diagram Mengelola Newsfeed

Activity diagram mengelola newsfeed dirancang berdasarkan case mengelola newsfeed pada use case diagram gambar 3.2 yang dibagi menjadi 3 bagian, yaitu:

Activity Diagram *View Newsfeed*

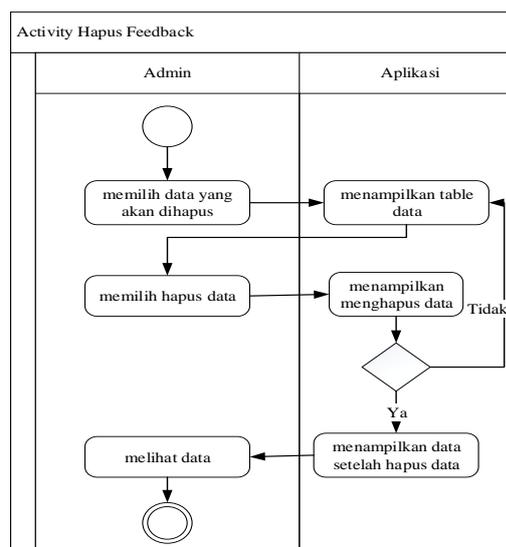
Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melihat data *newsfeed* yang ada pada aplikasi. Pertama-tama admin memilih fitur *newsfeed*, maka admin langsung bisa menuju ke halaman *newsfeed* dan langsung dapat melihat data yang masuk. *Activity* diagram dapat di lihat pada gambar 3.25.



Gambar 3.25 Activity diagram *view newsfeed*

Activity Diagram Hapus Newsfeed

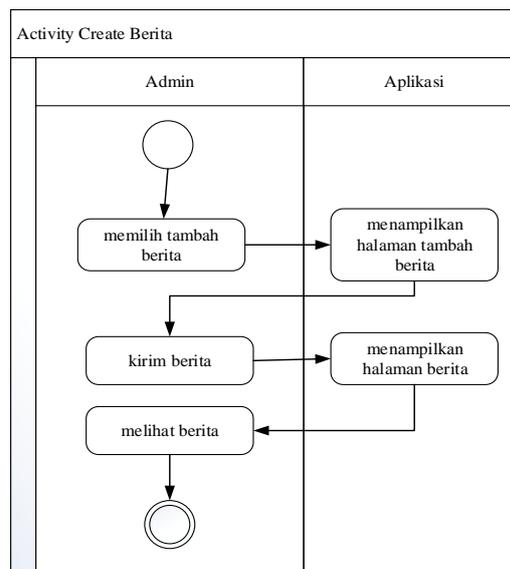
Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melakukan penghapusan *newsfeed* yang sudah pada aplikasi. Pertama-tama admin memilih fitur *newsfeed*, setelah itu admin langsung bisa menuju ke halaman *newsfeed* dan setelah itu admin memilih untuk menghapus berita, disini akan ada pilihan seperti jika ingin menghapus maka admin akan menuju halaman *newsfeed* yang terbaru dan jika admin mengurungkan untuk menghapus maka admin akan tetap di halaman. *Activity* diagram dapat di lihat pada gambar 3.26.



Gambar 3.26 Activity diagram hapus *newsfeed*

Activity Diagram Create Newsfeed

Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melakukan pembuatan atau *create* data *newsfeed* pada aplikasi. Pertama-tama admin memilih fitur *newsfeed*, setelah itu admin langsung bisa menuju ke halaman *newsfeed* dan setelah itu admin memilih untuk membuat berita baru pada aplikasi. Setelah itu admin mengirim, maka admin langsung menuju ke halaman *newsfeed* dari aplikasi. *Activity* diagram dapat di lihat pada gambar 3.27.



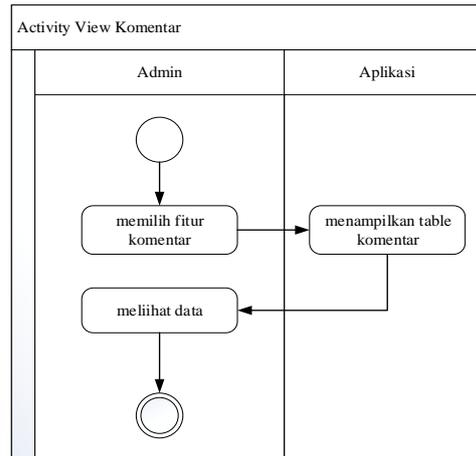
Gambar 3.27 Activity diagram *create newsfeed*

10. Activity Diagram Mengelola Komentar

Activity diagram mengelola komentar dirancang berdasarkan case mengelola komentar pada use case diagram gambar 3.2 yang dibagi menjadi 3 bagian, yaitu:

Activity Diagram View Komentar

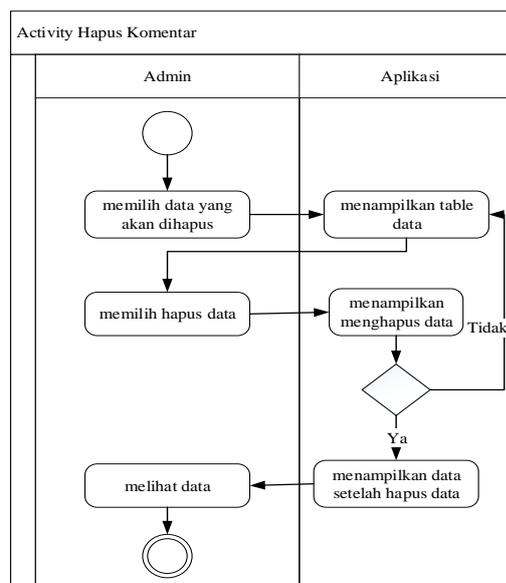
Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melihat data komentar yang ada pada aplikasi. Pertama-tama admin memilih fitur komentar, maka admin langsung bisa menuju ke halaman komentar dan langsung dapat melihat komentar yang masuk. *Activity* diagram dapat di lihat pada gambar 3.28.



Gambar 3.28 Activity diagram *view* komentar

Activity Diagram Hapus Komentar

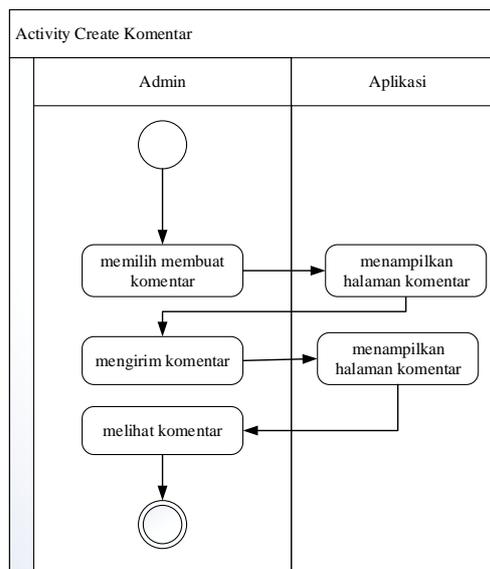
Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melakukan penghapusan data komentar yang sudah pada aplikasi. Pertama-tama admin memilih fitur komentar, setelah itu admin langsung bisa menuju ke halaman komentar dan setelah itu admin memilih untuk menghapus komentar yang sudah dipilih, disini akan ada pilihan seperti jika ingin menghapus maka admin akan menuju halaman komentar yang terbaru dan jika admin mengurungkan untuk menghapus maka admin akan tetap di halaman . *Activity* diagram dapat di lihat pada gambar 3.29.



Gambar 3.29 Activity diagram hapus komentar

Activity Diagram *Create* Komentar

Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melakukan pembuatan atau *create* komentar pada aplikasi. Pertama-tama admin memilih fitur *newsfeed*, setelah itu admin langsung bisa menuju ke halaman komentar dan setelah itu admin memilih untuk membuat komentar baru pada aplikasi. Setelah itu admin mengirim, setelah itu admin langsung menuju ke halaman komentar dari aplikasi. *Activity* diagram dapat di lihat pada gambar 3.30.



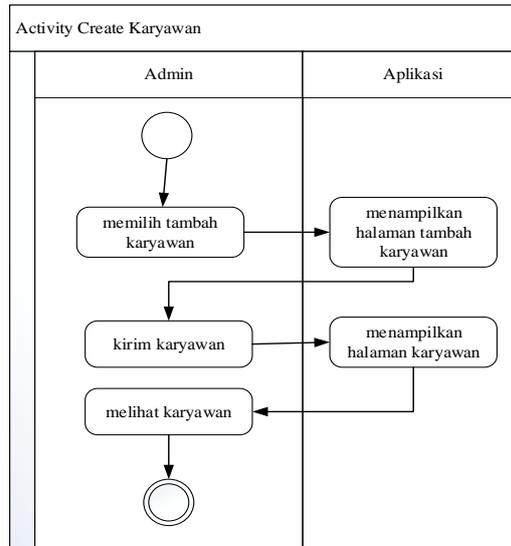
Gambar 3.30 Activity diagram *create* komentar

11. Activity Diagram Menerima Data Karyawan

Activity diagram menerima data karyawan dirancang berdasarkan case menerima newsfeed pada use case diagram gambar 3.2 yang dibagi menjadi 4 bagian, yaitu:

Activity Diagram Tambah Data Karyawan

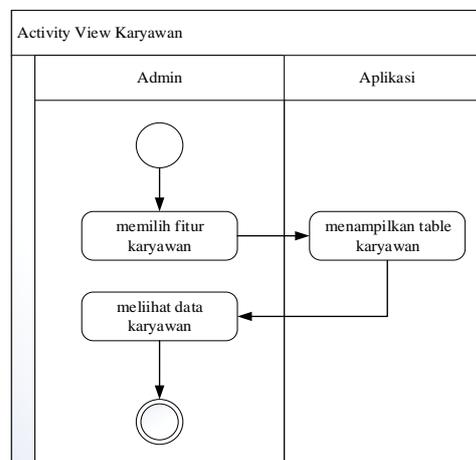
Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melakukan pembuatan atau *create* data karyawan pada aplikasi. Pertama-tama admin memilih fitur karyawan, setelah itu admin langsung bisa menuju ke halaman karyawan dan setelah itu admin memilih untuk membuat data baru pada aplikasi. Setelah itu admin mengirim, maka admin langsung menuju ke halaman karyawan dari aplikasi. *Activity* diagram dapat di lihat pada gambar 3.31.



Gambar 3.31 Activity diagram *create* karyawan

Activity Diagram View Karyawan

Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melihat data karyawan yang masuk pada aplikasi. Pertama-tama admin memilih fitur karyawan maka admin langsung bisa menuju ke halaman karyawan dan langsung dapat melihat data karyawan yang masuk. *Activity* diagram dapat di lihat pada gambar 3.32.

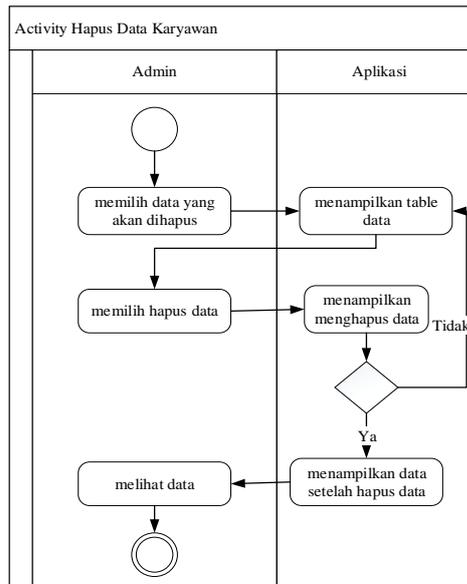


Gambar 3.32 Activity diagram *view* karyawan

Activity Diagram Hapus Karyawan

Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melakukan penghapusan data karyawan yang masuk pada aplikasi. Pertama-tama admin memilih fitur

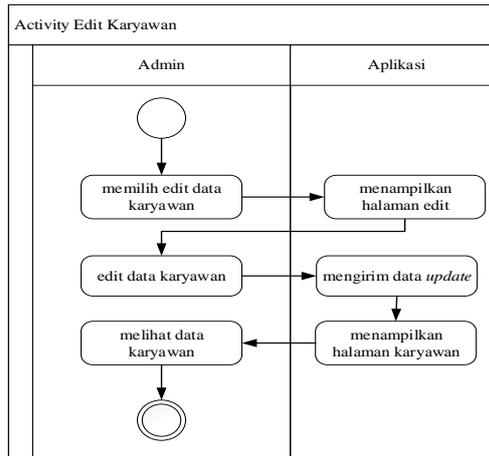
karyawan, setelah itu admin langsung bisa menuju ke halaman karyawan dan setelah itu admin memilih untuk menghapus data yang ada, disini akan ada pilihan seperti jika ingin menghapus data maka admin akan menuju halaman karyawan yang terbaru dan jika admin mengurungkan untuk menghapus data maka admin akan tetap di halaman . *Activity* diagram dapat di lihat pada gambar 3.33.



Gambar 3.33 Activity diagram hapus karyawan

Activity Diagram Update Karyawan

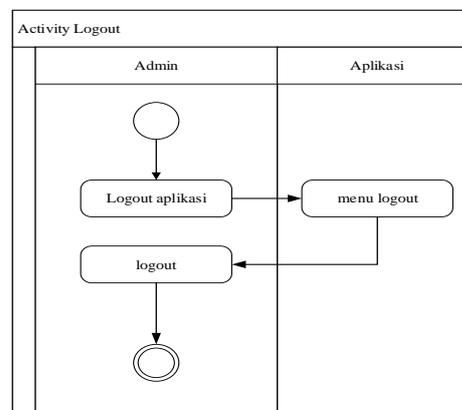
Pada *activity* diagram ini admin sudah sukses melakukan login, disini menjelaskan proses saat admin melakukan *update* data karyawan pada aplikasi yang sudah ada. Pertama-tama admin memilih fitur karyawan, setelah itu admin langsung bisa menuju ke halaman karyawan dan setelah itu admin memilih data untuk *update*. Lalu admin mengirim data, setelah mengirim data admin langsung menuju ke halaman karyawan dari aplikasi. *Activity* diagram dapat di lihat pada gambar 3.34.



Gambar 3.34 Activity diagram *update* karyawan

12. Activity Diagram Logout

Berdasarkan case logout pada use case diagram 3.2 maka dibuat lah class diagram logout yang akan menjelaskan proses admin dalam melakukan logout dari aplikasi. Saat admin akan melakukan logout admin akan memilih menu logout dari aplikasi. Proses logout admin dapat dilihat pada gambar 3.35.

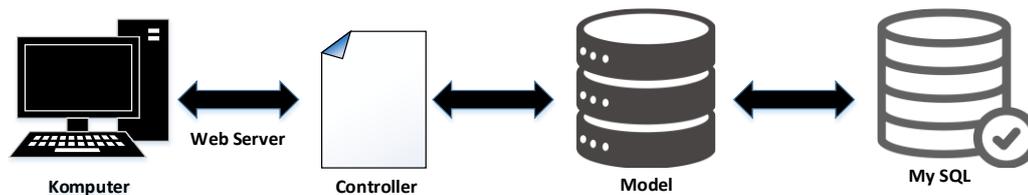


Gambar 3.35 Activity diagram logout

3.3.3 Arsitektur web

Berdasarkan proses pada *activity* diagram maka dirancanglah arsitektur web. Arsitektur web menggambarkan pengembang web berdasarkan komponen utama yang membangun sebuah web seperti manipulasi data, antarmuka pengguna dan bagian yang menjadi control dalam sebuah web

Arsitektur web yang digunakan dapat dilihat pada gambar 3.36.



Gambar 3.36 Arsitektur web admin

Penjelasan tentang gambar arsitektur pada gambar 3.36 adalah sebagai berikut:

1. Pada komputer dan controller terdapat web server, Fungsi utama dari sebuah web server adalah memproses berkas-berkas yang diminta oleh klien dan kemudian memberikan respon berupa halaman web yang terdiri dari teks, gambar, dokumen, dan sebagainya. Jadi untuk mengatur bagaimana data yang diperoleh dari *controller* di tampilkan pada admin.
2. *Controller* yang bertugas untuk mengirim perintah ke model untuk mendapatkan data yang diinginkan. *Controller* tidak mengetahui bagaimana data diambil dari database, karena *controller* tidak berisi kode perintah SQL. *Controller* mengolah data dari *inputan* admin dan data dari model kemudian data olahan tersebut dikirim ke *user interface* yang berada di komputer untuk ditampilkan sesuai dengan aturan *controller*. *Controller* menghubungkan antara model dengan *user interface* dan mengatur hubungan tersebut.
3. Model yang bertugas mengambil data dari *database* dan juga memasukkan data ke *database*. Isi utamanya berupa perintah SQL. Hasilnya dikirim ke *controller*.
4. *My SQL* berfungsi untuk menyimpan data.

Komponen arsitektur web

Komponen yang dibutuhkan dalam arsitektur web admin di PT Indocement Tunggal Prakarsa Tbk. terdiri dari perangkat keras dan perangkat lunak untuk mendukung berjalannya perancangan dan pembuatan web admin.

A. Perangkat keras

Daftar perangkat keras yang digunakan pengembang adalah sebagai berikut:

1. Laptop

Laptop berguna untuk pembuatan web admin memiliki spesifikasi sebagai berikut:

- a) Processor Intel® Core™ i5 CPU M 460 @2.53 GHz
- b) Memory 4,00 GB
- c) Hardisk 250 GB

B. Perangkat lunak

Perangkat lunak dibutuhkan untuk membuat aplikasi, diantaranya adalah sebagai berikut :

- a) Netbeans Ide 8.2

Digunakan untuk membuat web.

- b) *MySQL*

Digunakan untuk pengolahan *database* yang akan digunakan dalam web admin.

- c) *Apache Tomcat*

Digunakan sebagai web server pada we.

- d) *Browser Internet*

Digunakan untuk menjalankan web admin.

3.3.4 Class Diagram

Berdasarkan proses pada *sequence* diagram maka dirancanglah *class* diagram. *Class* diagram menggambarkan rancangan *class* dalam pembangunan web admin.

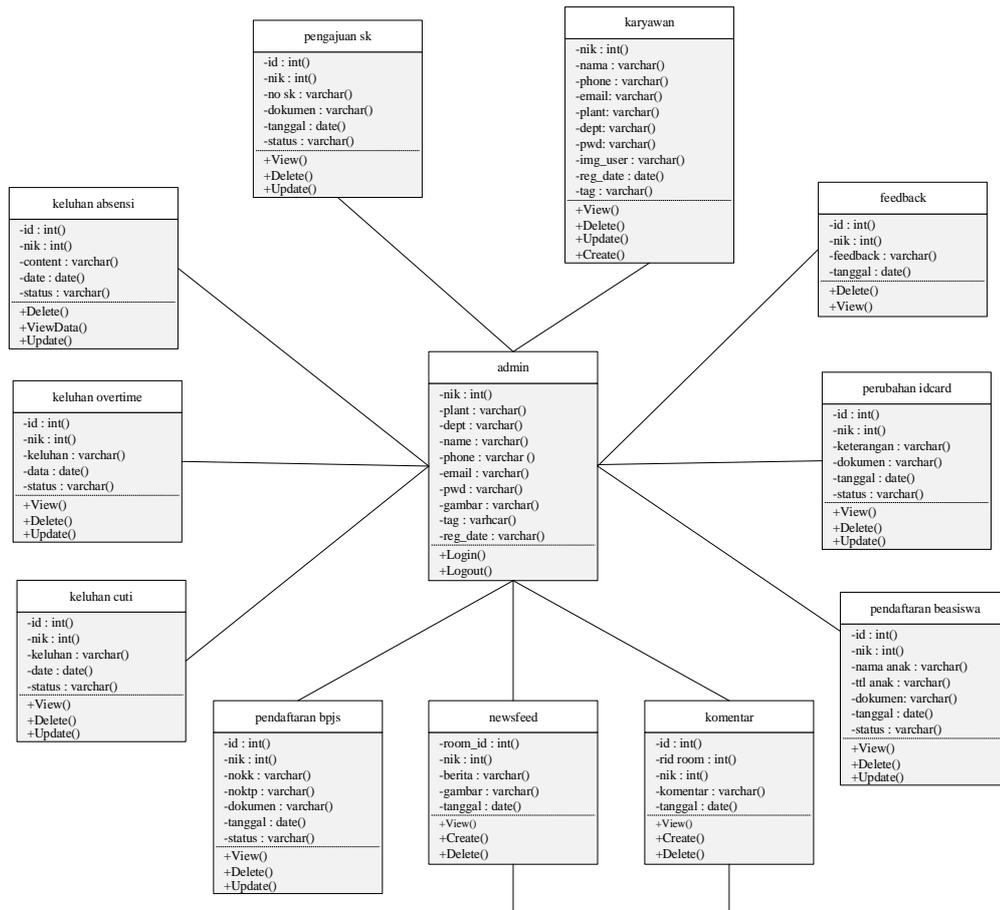
Class Diagram yang digunakan dapat dilihat pada gambar 3.37.

Berikut penjelasan *class diagram* pada gambar 3.37:

1. Plant memiliki *composition* dengan class user. Artinya user tidak dapat mendaftarkan akun jika tidak memiliki lokasi plant pekerjaan.
2. Dept memiliki *composition* dengan class user. Artinya user tidak dapat mengakses akun jika tidak memiliki departemen bagian pekerjaan.
3. *Class* feedback berfungsi untuk pengelolaan data feedback yang masuk, yang mencakup proses lihat data dan hapus data.
4. *Class* keluhan absensi berfungsi untuk pengelolaan keluhan absensi yang masuk, yang mencakup proses lihat, *update*, dan hapus data.

5. *Class* keluhan cuti berfungsi untuk pengelolaan keluhan cuti yang masuk, yang mencakup proses lihat, *update* dan hapus data.
6. *Class* keluhan overtime berfungsi untuk pengelolaan keluhan overtime yang masuk, yang mencakup proses lihat, *update* dan hapus data.
7. *Class* pendaftaran beasiswa anak berfungsi pengelolaan pendafrtan beasiswa yang masuk, yang mencakup proses lihat, *update* dan hapus data.
8. *Class* pendaftaran BPJS. berfungsi pengelolaan pendafrtan BPJS yang masuk, yang mencakup proses lihat, *update* dan hapus data.
9. *Class* perubahan id card berfungsi pengelolaan perubahan is card yang masuk, yang mencakup proses lihat, *update* dan hapus data.
10. *Class* pengajuan sk berfungsi pengelolaan data pengajuan sk yang masuk, yang mencakup proses lihat, *update* dan hapus data.
11. *Class* newsfeed berfungsi sebagai pengelolaan berita *terupdate*, yang mencakup proses lihat data, membuat data data, menghapus data, dan mengirim data.
12. *Class* komentar berfungsi untuk pengelolaan komentar yang masuk dan keluar. Yang mencakup proses lihat data, membuat data, menghapus data, dan mengirim data.
13. *Class* karyawan berfungsi untuk pengelolaan data karyawan yang masuk dan keluar. Yang mencakup proses lihat data, membuat data, menghapus data, dan mengirim data.

Rancangan *Class* diagram yang akan digunakan dalam proses pembuatan web admin dapat dilihat pada gambar 3.37.



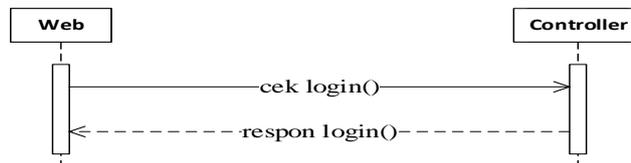
Gambar 3.37 Class Diagram untuk Admin

3.3.5 Sequence diagram

Berdasarkan *activity* diagram yang sudah dibuat pada sub bab sebelumnya, maka dirancanglah *sequence* diagram. *Sequence* diagram menggambarkan proses yang dilakukan oleh web dengan *controller*. Web dan *controller* melakukan proses mengirim dan menerima data. Pada web admin terdapat 2 proses yaitu *request* dan *response*. Proses *request* yaitu aplikasi melakukan pengiriman dan permintaan data ke *controller*. Proses *response* yaitu respon yang dikirimkan *controller* terhadap pengiriman dan permintaan data dari web. Dalam proses *request* dan *response* data antara web dan *controller*. Rancangan *sequence* diagram dibagi menjadi beberapa bagian yaitu sebagai berikut.

1. Sequence diagram login

Sequence diagram login menggambarkan rancangan arsitektur untuk proses *login*. *Sequence diagram login* dibuat berdasarkan *activity diagram login*. Rancangan *sequence diagram login* dapat dilihat pada gambar dibawah ini.



Gambar 3.38 *Sequence diagram* proses *login*

Berikut penjelasan *sequence diagram login* pada gambar 3.38 :

- Dalam proses *login*, aplikasi mengirimkan *request* data *nik* dan *password* sebagai validasi *login*. *Request* *nik* dan *password* dapat dilihat pada *list 3.1*.

```
String nik = request.getParameter("nik");
String pwd = request.getParameter("pwd");
```

List 3.1 *Request login admin*

- Jika data yang dikirimkan salah, *controller* akan mengirimkan *response* untuk tetap berada di halaman *login* seperti pada *list 3.2*

```
response.sendRedirect("login.jsp");
```

List 3.2 *Response login salah*

- Jika data yang dikirimkan benar, *controller* akan mengirimkan *response* untuk menampilkan halaman indek seperti pada *list 3.3*

```
response.sendRedirect("index.jsp");
```

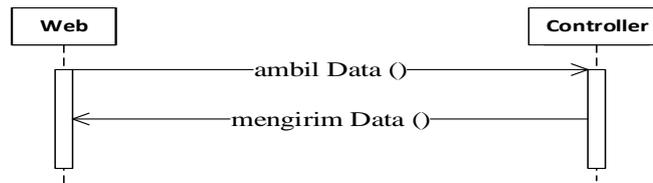
List 3.3 *Response login benar*

2. Sequence diagram menerima keluhan absensi

Sequence diagram menerima keluhan absensi dirancang berdasarkan *activity diagram* menerima keluhan absensi. *Sequence diagram* menerima keluhan absensi dibagi menjadi 3 bagian, yaitu :

Sequence diagram view keluhan absensi

Sequence diagram keluhan absensi menggambarkan rancangan arsitektur untuk proses keluhan absensi. *Sequence diagram* keluhan absensi dibuat berdasarkan *activity diagram* keluhan absensi. Rancangan *sequence diagram* keluhan absensi dapat dilihat pada gambar dibawah ini



Gambar 3.39 *Sequence view* diagram keluhan absensi

Berikut penjelasan *sequence* diagram keluhan absensi pada gambar 3.39 :

- a. Web mengirimkan *request* untuk mengambil data keluhan absensi. *Request* data keluhan absensi dapat dilihat pada *list* 3.4.

```
String dataAbsensi = request.getParameter("dataAbsensi");
```

List 3.4 *Request* keluhan absensi

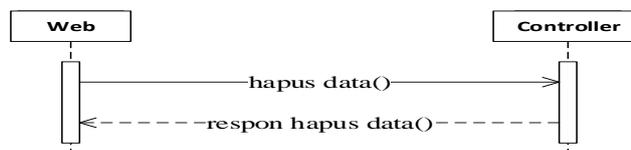
- b. Setelah *request* diterima oleh *controller*, *controller* mengirimkan *response* berisi data keluhan absensi yang akan ditampilkan di web. *Response* data keluhan absensi dapat dilihat pada *list* 3.5

```
response.setId(Integer.parseInt(request.getParameter("id")));
response.setContent(request.getParameter("content"));
response.setDate(request.getParameter("date"));
response.setStatus(request.getParameter("status"));
```

List 3.5 *Response* keluhan absensi

Sequence diagram hapus keluhan absensi

Sequence diagram hapus keluhan absensi menggambarkan rancangan arsitektur untuk proses hapus keluhan absensi. *Sequence* diagram hapus keluhan absensi dibuat berdasarkan *activity* diagram hapus keluhan absensi. Rancangan *sequence* diagram hapus keluhan absensi dapat dilihat pada gambar dibawah ini.



Gambar 3.40 *Sequence* diagram hapus keluhan absensi

Berikut penjelasan *sequence* diagram hapus keluhan absensi pada gambar 3.40:

- a. Web mengirimkan *request* untuk menghapus data keluhan absensi. *Request* data hapus keluhan absensi dapat dilihat pada *list* 3.6

```
String proses = request.getParameter("hapus-absensi");
```

List 3.6 *Request* hapus keluhan absensi

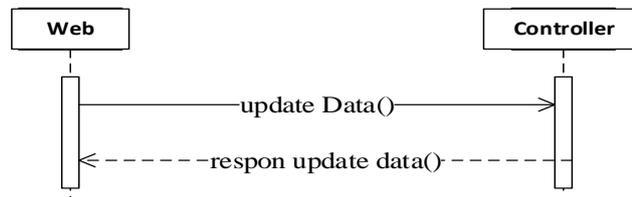
- b. Setelah *request* diterima oleh *controller*, *controller* memberikan *response* untuk menuju ke halaman keluhan absensi. *Response* hapus keluhan absensi dapat dilihat pada *list 3.7*

```
response.sendRedirect("keluAbsensi.jsp");
```

List 3.7 *Response* hapus keluhan absensi

Sequence diagram *update* keluhan absensi

Sequence diagram *update* keluhan absensi menggambarkan rancangan arsitektur untuk proses *update* keluhan absensi. *Sequence* diagram *update* keluhan absensi dibuat berdasarkan *activity* diagram *update* keluhan absensi. Rancangan *sequence* diagram *update* keluhan absensi dapat dilihat pada gambar dibawah ini.



Gambar 3.41 *Sequence* diagram *update* keluhan absensi

Berikut penjelasan *sequence* diagram *update* keluhan absensi pada gambar 3.41:

- a. Web mengirimkan *request* untuk meng*update* data keluhan absensi. *Request* data *update* keluhan absensi dapat dilihat pada *list 3.8*

```
String proses = request.getParameter("update-absensi");
```

List 3.8 *Request* *update* keluhan absensi

- b. Setelah *request* diterima oleh *controller*, *controller* memberikan *response* untuk meng*update* data keluhan absensi dan menuju halaman *update* absensi. *Response* *update* data keluhan absensi dapat dilihat pada *list 3.9*.

```
response.sendRedirect("update_absen.jsp?id=" +  
request.getParameter("id"));
```

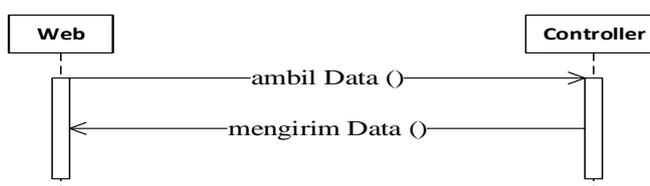
List 3.9 *Response* *update* keluhan absensi

3. Sequence diagram menerima keluhan overtime

Sequence diagram menerima keluhan absensi dirancang berdasarkan *activity* diagram menerima keluhan overtime. *Sequence* diagram menerima keluhan overtime dibagi menjadi 3 bagian, yaitu :

Sequence diagram *view* keluhan overtime

Sequence diagram keluhan overtime menggambarkan rancangan arsitektur untuk proses keluhan overtime. *Sequence* diagram keluhan overtime dibuat berdasarkan *activity* diagram keluhan overtime. Rancangan *sequence* diagram keluhan overtime dapat dilihat pada gambar dibawah ini



Gambar 3.42 *Sequence* diagram *view* keluhan overtime

Berikut penjelasan *sequence* diagram keluhan overtime pada gambar 3.42 :

- Web mengirimkan *request* data untuk mengirim data keluhan overtime. *Request* data keluhan overtime dapat dilihat pada list 3.10.

```
String dataOvertime = request.getParameter("dataOvertime");
```

List 3.10 *Request* keluhan overtime

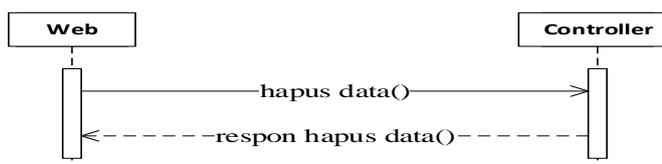
- Setelah *request* diterima oleh *controller*, *controller* mengirimkan *response* berisi data keluhan overtime yang akan ditampilkan di web. *Response* data keluhan overtime dapat dilihat pada list 3.11

```
response.setId(Integer.parseInt(request.getParameter("id")));
response.setContent(request.getParameter("content"));
response.setDate(request.getParameter("date"));
response.setStatus(request.getParameter("status"));
```

List 3.11 *Response* keluhan overtime

Sequence diagram hapus keluhan overtime

Sequence diagram hapus keluhan overtime menggambarkan rancangan arsitektur untuk proses hapus keluhan overtime. *Sequence* diagram hapus keluhan overtime dibuat berdasarkan *activity* diagram hapus keluhan overtime. Rancangan *sequence* diagram hapus keluhan overtime dapat dilihat pada gambar dibawah ini.



Gambar 3.43 *Sequence* diagram hapus keluhan overtime

Berikut penjelasan *sequence* diagram hapus keluhan *overtime* pada gambar 3.43:

- a. Web mengirimkan *request* untuk menghapus data keluhan *overtime*. *Request* data hapus keluhan *overtime* dapat dilihat pada list 3.12

```
String proses = request.getParameter("hapus-overtime");
```

List 3.12 Request hapus keluhan *overtime*

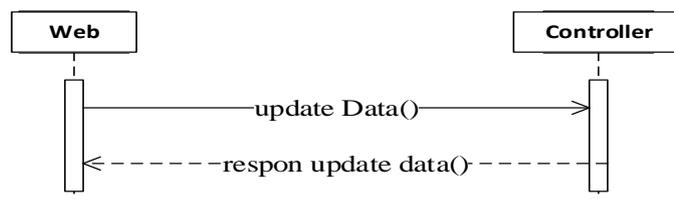
- b. Setelah *request* diterima oleh *controller*, *controller* memberikan *response* untuk menuju ke halaman keluhan *overtime*. *Response* hapus keluhan *overtime* dapat dilihat pada list 3.13

```
response.sendRedirect("keluOver.jsp");
```

List 3.13 Response hapus keluhan *overtime*

Sequence diagram update keluhan *overtime*

Sequence diagram *update* keluhan *overtime* menggambarkan rancangan arsitektur untuk proses *update* keluhan *overtime*. *Sequence* diagram *update* keluhan *overtime* dibuat berdasarkan *activity* diagram *update* keluhan *overtime*. Rancangan *sequence* diagram *update* keluhan *overtime* dapat dilihat pada gambar dibawah ini.



Gambar 3.44 *Sequence* diagram *update* keluhan *overtime*

Berikut penjelasan *sequence* diagram *update* keluhan *overtime* pada gambar 3.44:

- a. Web mengirimkan *request* untuk mengupdate data keluhan *overtime*. *Request* data *update* keluhan *overtime* dapat dilihat pada list 3.14

```
String proses = request.getParameter("update-overtime");
```

List 3.14 Request update keluhan *overtime*

- b. Setelah *request* diterima oleh *controller*, *controller* memberikan *response* untuk mengupdate data keluhan *overtime* dan menuju halaman *update overtime*. *Response update* data keluhan *overtime* dapat dilihat pada list 3.15.

```
response.sendRedirect("edit_overtime.jsp?id=" + request.getParameter("id"));
```

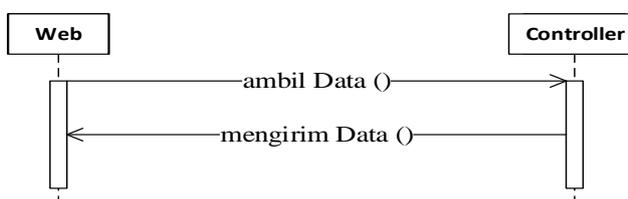
List 3.15 Response update keluhan *overtime*

4. Sequence diagram menerima keluhan cuti

Sequence diagram menerima keluhan cuti dirancang berdasarkan *activity* diagram menerima keluhan cuti. *Sequence* diagram menerima keluhan cuti dibagi menjadi 3 bagian, yaitu :

Sequence diagram keluhan cuti

Sequence diagram keluhan cuti menggambarkan rancangan arsitektur untuk proses keluhan cuti. *Sequence* diagram keluhan cuti dibuat berdasarkan *activity* diagram keluhan cuti. Rancangan *sequence* diagram keluhan cuti dapat dilihat pada gambar dibawah ini



Gambar 3.45 *Sequence* diagram keluhan cuti

Berikut penjelasan *sequence* diagram keluhan cuti pada gambar 3.45 :

- a. Web mengirimkan *request* data untuk mengirim data keluhan cuti. *Request* data keluhan cuti dapat dilihat pada *list* 3.16.

```
String dataCuti = request.getParameter("dataCuti");
```

List 3.16 *Request* keluhan cuti

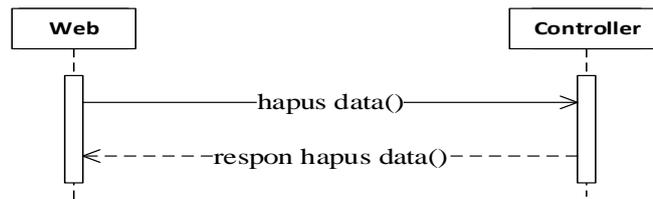
- b. Setelah *request* diterima oleh *controller*, *controller* mengirimkan *response* berisi data keluhan cuti yang akan ditampilkan di web. *Response* data keluhan cuti dapat dilihat pada *list* 3.17

```
response.setId(Integer.parseInt(request.getParameter("id")));
response.setContent(request.getParameter("content"));
response.setDate(request.getParameter("date"));
response.setStatus(request.getParameter("status"));
```

List 3.17 *Response* keluhan cuti

Sequence diagram hapus keluhan cuti

Sequence diagram hapus keluhan cuti menggambarkan rancangan arsitektur untuk proses hapus keluhan cuti. *Sequence* diagram hapus keluhan cuti dibuat berdasarkan *activity* diagram hapus keluhan cuti. Rancangan *sequence* diagram hapus keluhan cuti dapat dilihat pada gambar dibawah ini.



Gambar 3.46 *Sequence* diagram hapus keluhan cuti

Berikut penjelasan *sequence* diagram hapus keluhan cuti pada gambar 3.46 :

- Web mengirimkan *request* untuk menghapus data keluhan cuti. *Request* data hapus keluhan cuti dapat dilihat pada *list* 3.18.

```
String proses = request.getParameter("hapus-cuti");
```

List 3.18 *Request* hapus keluhan cuti

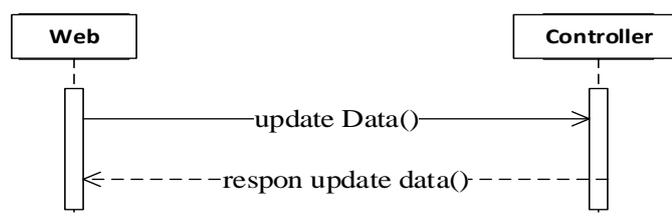
- Setelah *request* diterima oleh *controller*, *controller* memberikan *response* untuk menuju ke halaman keluhan cuti. *Response* hapus keluhan cuti dapat dilihat pada *list* 3.19

```
response.sendRedirect("keluhanCuti.jsp");
```

List 3.19 *Request* hapus keluhan cuti

Sequence diagram *update* keluhan cuti

Sequence diagram *update* keluhan cuti menggambarkan rancangan arsitektur untuk proses *update* keluhan cuti. *Sequence* diagram *update* keluhan cuti dibuat berdasarkan *activity* diagram *update* keluhan cuti. Rancangan *sequence* diagram *update* keluhan cuti dapat dilihat pada gambar dibawah ini.



Gambar 3.47 *Sequence* diagram *update* keluhan cuti

Berikut penjelasan *sequence* diagram *update* keluhan cuti pada gambar 3.47:

- Web mengirimkan *request* untuk meng*update* data keluhan cuti. *Request* data *update* keluhan cuti dapat dilihat pada *list* 3.20

```
String proses = request.getParameter("update-cuti");
```

List 3.20 *Request* *update* keluhan cuti

- b. Setelah *request* diterima oleh *controller*, *controller* memberikan *response* untuk mengupdate data keluhan cuti dan menuju halaman *update* cuti. *Response update* data keluhan cuti dapat dilihat pada list 3.21.

```
response.sendRedirect("update_cuti.jsp?id=" + request.getParameter("id"));
```

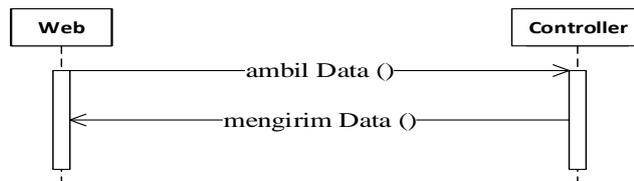
List 3.21 *Response update* keluhan cuti

5. Sequence diagram menerima pendaftaran beasiswa

Sequence diagram menerima pendaftaran beasiswa dirancang berdasarkan *activity* diagram menerima pendaftaran beasiswa. *Sequence* diagram menerima pendaftaran beasiswa dibagi menjadi 3 bagian, yaitu :

Sequence diagram view pendaftaran beasiswa

Sequence diagram pendaftaran beasiswa menggambarkan rancangan arsitektur untuk proses pendaftaran beasiswa. *Sequence* diagram pendaftaran beasiswa dibuat berdasarkan *activity* diagram pendaftaran beasiswa. Rancangan *sequence* diagram pendaftaran beasiswa dapat dilihat pada gambar dibawah ini.



Gambar 3.48 Sequence diagram view pendaftaran beasiswa

Berikut penjelasan *sequence* diagram pendaftaran beasiswa pada gambar 3.4 :

- a. Web mengirimkan *request* data untuk mengirim data pendaftaran beasiswa. *Request* data pendaftaran beasiswa dapat dilihat pada list 3.22.

```
String dataBeasiswa = request.getParameter("dataBeasiswa");
```

List 3.22 *Request* pendaftaran beasiswa

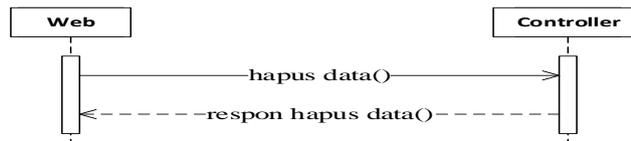
- b. Setelah *request* diterima oleh *controller*, *controller* mengirimkan *response* berisi data pendaftaran beasiswa yang akan ditampilkan di web. *Response* data pendaftaran beasiswa dapat dilihat pada list 3.22

```
response.setId(Integer.parseInt(request.getParameter("id")));
response.setDate(request.getParameter("date"));
response.setDoc(request.getParameter("doc"));
response.setNm_anak(request.getParameter("nm_anak"));
response.setStatus(request.getParameter("status"));
response.setTtl(request.getParameter("ttl"));
```

List 3.22 *Response* pendaftaran beasiswa

Sequence diagram hapus pendaftaran beasiswa

Sequence diagram hapus pendaftaran beasiswa menggambarkan rancangan arsitektur untuk proses hapus pendaftaran beasiswa. *Sequence* diagram hapus pendaftaran beasiswa dibuat berdasarkan *activity* diagram hapus pendaftaran beasiswa. Rancangan *sequence* diagram hapus pendaftaran beasiswa dapat dilihat pada gambar dibawah ini.



Gambar 3.49 *Sequence* diagram hapus pendaftaran beasiswa

Berikut penjelasan *sequence* diagram hapus pendaftaran beasiswa pada gambar 3.49:

- Web mengirimkan *request* untuk menghapus data pendaftaran beasiswa. *Request* data hapus pendaftaran beasiswa dapat dilihat pada list 3.23.

```
String proses = request.getParameter("hapus-beasiswa");
```

List 3.23 *Request* hapus pendaftaran beasiswa

- Setelah *request* diterima oleh *controller*, *controller* memberikan *response* untuk menuju ke halaman pendaftaran beasiswa. *Response* pendaftaran beasiswa dapat dilihat pada list 3.24

```
response.sendRedirect("beasiswa.jsp");
```

List 3.24 *Response* hapus pendaftaran beasiswa

Sequence diagram update pendaftaran beasiswa

Sequence diagram *update* pendaftaran beasiswa menggambarkan rancangan arsitektur untuk proses *update* pendaftaran beasiswa. *Sequence* diagram *update* pendaftaran beasiswa dibuat berdasarkan *activity* diagram *update* pendaftaran beasiswa. Rancangan *sequence* diagram *update* pendaftaran beasiswa dapat dilihat pada gambar dibawah ini.



Gambar 3.50 *Sequence* diagram *update* pendaftaran beasiswa

Berikut penjelasan *sequence* diagram *update* pendaftaran beasiswa pada gambar 3.50:

- a. Web mengirimkan *request* untuk mengupdate data pendaftaran beasiswa. *Request* data *update* pendaftaran beasiswa dapat dilihat pada *list* 3.25

```
String proses = request.getParameter("update-beasiswa");
```

List 3.25 *Request* update pendaftaran beasiswa

- b. Setelah *request* diterima oleh *controller*, *controller* memberikan *response* untuk mengupdate data pendaftaran beasiswa dan menuju halaman *update* pendaftaran beasiswa. *Response* *update* data pendaftaran beasiswa dapat dilihat pada *list* 3.26.

```
response.sendRedirect("update_beasiswa.jsp?id=" +
request.getParameter("id"));
```

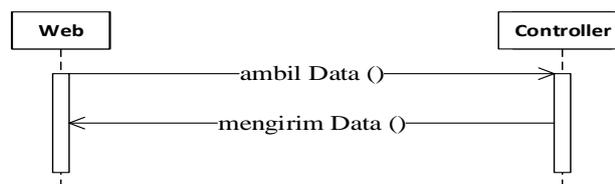
List 3.26 *Response* update keluhan absensi

6. Sequence diagram menerima pendaftaran BPJS

Sequence diagram menerima pendaftaran BPJS dirancang berdasarkan *activity* diagram menerima pendafran BPJS. *Sequence* diagram menerima pendafran BPJS dibagi menjadi 3 bagian, yaitu :

Sequence diagram view pendaftaran BPJS

Sequence diagram pendaftaran BPJS menggambarkan rancangan arsitektur untuk proses pendaftaran BPJS. *Sequence* diagram pendaftaran BPJS dibuat berdasarkan *activity* diagram pendaftaran BPJS. Rancangan *sequence* diagram pendaftaran BPJS dapat dilihat pada gambar dibawah ini.



Gambar 3.51 *Sequence* diagram pendaftaran BPJS

Berikut penjelasan *sequence* diagram pendaftaran BPJS pada gambar 3.51 :

- a. Web mengirimkan *request* data untuk mengirim data pendaftaran BPJS. *Request* data pendaftaran BPJS dapat dilihat pada *list* 3.27.

```
String dataBpjs = request.getParameter("dataBpjs ");
```

List 3.27 *Request* pendaftaran BPJS

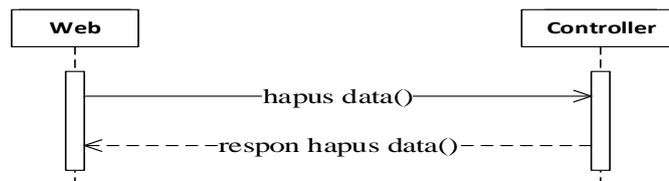
- b. Setelah *request* diterima oleh *controller*, *controller* mengirimkan *response* berisi data pendaftaran BPJS yang akan ditampilkan di web. *Response* data pendaftaran BPJS dapat dilihat pada list 3.28

```
response.setId(Integer.parseInt(request.getParameter("id")));
response.setDate(request.getParameter("date"));
response.setNokk(request.getParameter("nokk"));
response.setNoktp(request.getParameter("noktp"));
response.setDoc(request.getParameter("doc"));
response.setStatus(request.getParameter("status"));
```

List 3.28 *Response* pendaftaran BPJS

Sequence diagram hapus pendaftaran BPJS

Sequence diagram hapus pendaftaran BPJS menggambarkan rancangan arsitektur untuk proses hapus pendaftaran BPJS. *Sequence* diagram hapus pendaftaran BPJS dibuat berdasarkan *activity* diagram hapus pendaftaran BPJS. Rancangan *sequence* diagram hapus pendaftaran BPJS dapat dilihat pada gambar dibawah ini.



Gambar 3.52 *Sequence* diagram hapus pendaftaran BPJS

Berikut penjelasan *sequence* diagram hapus pendaftaran beasiswa pada gambar 3.52 :

- a. Web mengirimkan *request* untuk menghapus data pendaftaran BPJS. *Request* data hapus pendaftaran BPJS dapat dilihat pada list 3.28.

```
String proses = request.getParameter("hapus-bpjs");
```

List 3.28 *Request* hapus pendaftaran BPJS

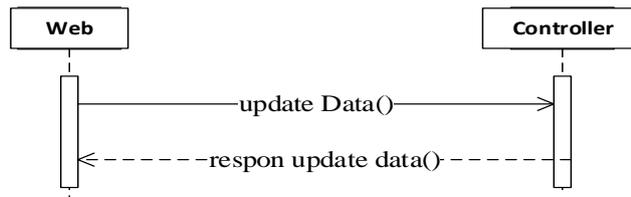
- b. Setelah *request* diterima oleh *controller*, *controller* memberikan *response* untuk menuju ke halaman pendaftaran BPJS. *Response* pendaftaran BPJS dapat dilihat pada list 3.29

```
response.sendRedirect("bpjs.jsp");
```

List 3.29 *Response* hapus pendaftaran BPJS

Sequence diagram update pendaftaran BPJS

Sequence diagram update pendaftaran BPJS menggambarkan rancangan arsitektur untuk proses *update* pendaftaran BPJS. *Sequence diagram update* pendaftaran BPJS dibuat berdasarkan *activity diagram update* pendaftaran BPJS. Rancangan *sequence diagram update* pendaftaran beasi BPJS swa dapat dilihat pada gambar dibawah ini.



Gambar 3.53 *Sequence diagram update* pendaftaran BPJS

Berikut penjelasan *sequence diagram update* pendaftaran BPJS pada gambar 3.53:

- a. Web mengirimkan *request* untuk meng*update* data pendaftaran BPJS. *Request data update* pendaftaran BPJS dapat dilihat pada *list 3.30*

```
String proses = request.getParameter("update-bpjs");
```

List 3.30 *Request update* pendaftaran BPJS

- b. Setelah *request* diterima oleh *controller*, *controller* memberikan *response* untuk meng*update* data pendaftaran BPJS dan menuju halaman *update* pendaftaran BPJS. *Response update* data pendaftaran BPJS dapat dilihat pada *list 3.31*.

```
response.sendRedirect("edit bpjs.jsp?id=" + request.getParameter("id"));
```

List 3.31 *Response update* pendaftaran BPJS

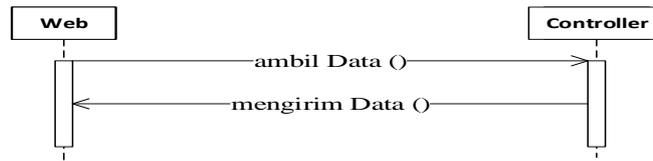
7. Sequence diagram menerima pengajuan surat keterangan

Sequence diagram menerima pengajuan surat keterangan dirancang berdasarkan *activity diagram* menerima pengajuan surat keterangan. *Sequence diagram* menerima pengajuan surat keterangan dibagi menjadi 3 bagian, yaitu :

Sequence diagram view pengajuan surat keterangan

Sequence diagram pengajuan surat keterangan menggambarkan rancangan arsitektur untuk proses pengajuan surat keterangan. *Sequence diagram* pengajuan surat keterangan dibuat berdasarkan *activity diagram* pengajuan surat keterangan.

Rancangan *sequence* diagram pengajuan surat keterangan dapat dilihat pada gambar dibawah ini.



Gambar 3.54 *Sequence* diagram pengajuan surat keterangan

Berikut penjelasan *sequence* diagram pengajuan surat keterangan pada gambar 3.54 :

- a. Web mengirimkan *request* data untuk mengirim data pengajuan surat keterangan. *Request* data pengajuan surat keterangan dapat dilihat pada *list* 3.32.

```
String dataSuratketerangan = request.getParameter("dataSuratketerangan");
```

List 3.32 Request view pengajuan surat keterangan

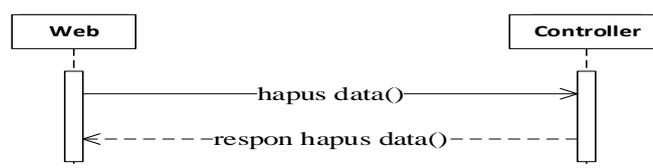
- b. Setelah *request* diterima oleh *controller*, *controller* mengirimkan *response* berisi data pengajuan surat keterangan yang akan ditampilkan di web. *Response* data pengajuan surat keterangan dapat dilihat pada *list* 3.33

```
response.setId(Integer.parseInt(request.getParameter("id")));
response.setJnssurat(request.getParameter("jnssurat"));
response.setDate(request.getParameter("date"));
response.setDoc(request.getParameter("doc"));
response.setStatus(request.getParameter("status"));
response.setNik(Integer.parseInt(request.getParameter("nik")));;
```

List 3.33 Response view pengajuan surat keterangan

Sequence diagram hapus pengajuan surat keterangan

Sequence diagram hapus pengajuan surat keterangan menggambarkan rancangan arsitektur untuk proses hapus pengajuan surat keterangan. *Sequence* diagram hapus pengajuan surat keterangan dibuat berdasarkan *activity* diagram hapus pengajuan surat keterangan. Rancangan *sequence* diagram hapus pengajuan surat keterangan dapat dilihat pada gambar dibawah ini.



Gambar 3.55 *Sequence* diagram hapus pengajuan surat keterangan

Berikut penjelasan *sequence* diagram hapus pengajuan surat keterangan pada gambar 3.55 :

- a. Web mengirimkan *request* untuk menghapus data pengajuan surat keterangan. *Request* data hapus pengajuan surat keterangan dapat dilihat pada *list* 3.34.

```
String proses = request.getParameter("hapus- sketerangan");
```

List 3.34 *Request* hapus pengajuan surat keterangan

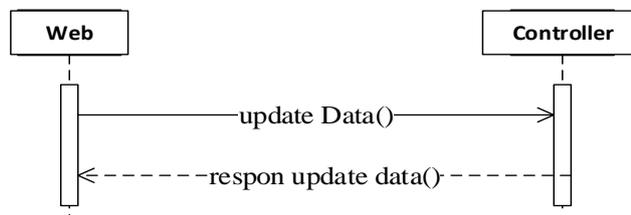
- b. Setelah *request* diterima oleh *controller*, *controller* memberikan *response* untuk menuju ke halaman pengajuan surat keterangan. *Response* hapus pengajuan surat keterangan dapat dilihat pada *list* 3.35

```
response.sendRedirect("SuKeterangan.jsp");
```

List 3.35 *Response* hapus pengajuan surat keterangan

Sequence diagram *update* pengajuan surat keterangan

Sequence diagram *update* pengajuan surat keterangan menggambarkan rancangan arsitektur untuk proses *update* pengajuan surat keterangan. *Sequence* diagram *update* pengajuan surat keterangan dibuat berdasarkan *activity* diagram *update* pengajuan surat keterangan. Rancangan *sequence* diagram *update* pengajuan surat keterangan dapat dilihat pada gambar dibawah ini.



Gambar 3.56 *Sequence* diagram *update* pengajuan surat keterangan

Berikut penjelasan *sequence* diagram *update* pengajuan surat keterangan pada gambar 3.56 :

- a. Web mengirimkan *request* untuk meng*update* data pengajuan surat keterangan. *Request* data *update* pengajuan surat keterangan dapat dilihat pada *list* 3.36

```
String proses = request.getParameter("update-sketerangan");
```

List 3.36 *Request* *update* pengajuan surat keterangan

- b. Setelah *request* diterima oleh *controller*, *controller* memberikan *response* untuk meng*update* data pengajuan surat keterangan dan menuju halaman

update pengajuan surat keterangan. *Response update* data pengajuan surat keterangan dapat dilihat pada *list 3.37*.

```
response.sendRedirect("update_keterangan.jsp?id="+
request.getParameter("id"));
```

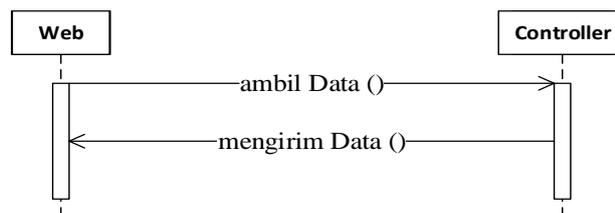
List 3.37 *Response update* pengajuan surat keterangan

8. Sequence diagram menerima perubahan id card

Sequence diagram menerima perubahan *id card* dirancang berdasarkan *activity* diagram menerima perubahan *id card*. *Sequence* diagram menerima perubahan *id card* dibagi menjadi 3 bagian, yaitu :

Sequence diagram view perubahan *id card*

Sequence diagram perubahan *id card* menggambarkan rancangan arsitektur untuk proses perubahan *id card*. *Sequence* diagram perubahan *id card* dibuat berdasarkan *activity* diagram perubahan *id card*. Rancangan *sequence* diagram perubahan *id card* dapat dilihat pada gambar dibawah ini.



Gambar 3.57 *Sequence* diagram perubahan *id card*

Berikut penjelasan *sequence* diagram perubahan *id card* pada gambar 3.57 :

- a. Web mengirimkan *request* data untuk mengirim data perubahan *id card*. *Request* data perubahan *id card* dapat dilihat pada *list 3.38*.

```
String dataIdcard = request.getParameter("dataIdcard ");
```

List 3.38 *Request* perubahan *id card*

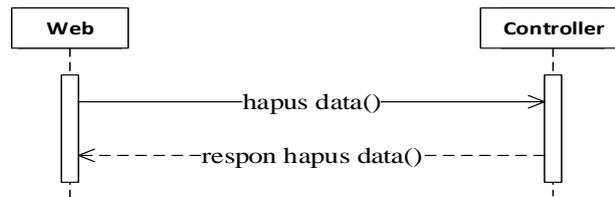
- b. Setelah *request* diterima oleh *controller*, *controller* mengirimkan *response* berisi data perubahan *id card* yang akan ditampilkan di web. *Response* data perubahan *id card* dapat dilihat pada *list 3.39*

```
response.setId(Integer.parseInt(request.getParameter("id")));
response.setContent(request.getParameter("content"));
response.setDate(request.getParameter("date"));
response.setDoc(request.getParameter("doc"));
response.setStatus(request.getParameter("status"));
response.setNik(Integer.parseInt(request.getParameter("nik")));
```

List 3.39 *Responses* view perubahan *id card*

Sequence diagram hapus perubahan id card

Sequence diagram hapus perubahan *id card* menggambarkan rancangan arsitektur untuk proses hapus perubahan *id card*. *Sequence* diagram hapus perubahan *id card* dibuat berdasarkan *activity* diagram hapus perubahan *id card*. Rancangan *sequence* diagram hapus perubahan *id card* dapat dilihat pada gambar dibawah ini.



Gambar 3.58 *Sequence* diagram hapus perubahan *id card*

Berikut penjelasan *sequence* diagram hapus perubahan *id card* pada gambar 3.58:

- a. Web mengirimkan *request* untuk menghapus data perubahan *id card*. *Request* data hapus perubahan *id card* dapat dilihat pada list 3.40.

```
String proses = request.getParameter("hapus-idcard");
```

List 3.40 *Request* hapus perubahan *id card*

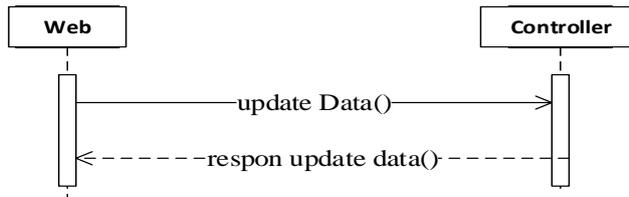
- b. Setelah *request* diterima oleh *controller*, *controller* memberikan *response* untuk menuju ke halaman perubahan *id card*. *Response* hapus perubahan *id card* dapat dilihat pada list 3.41

```
response.sendRedirect("ubahIdCard.jsp");
```

List 3.41 *Response* hapus perubahan *id card*

Sequence diagram *update* perubahan id card

Sequence diagram *update* perubahan *id card* menggambarkan rancangan arsitektur untuk proses *update* perubahan *id card*. *Sequence* diagram *update* perubahan *id card* dibuat berdasarkan *activity* diagram *update* perubahan *id card*. Rancangan *sequence* diagram *update* perubahan *id card* dapat dilihat pada gambar dibawah ini.



Gambar 3.59 *Sequence diagram update perubahan id card*

Berikut penjelasan *sequence diagram update perubahan id card* pada gambar 3.59 :

- a. Web mengirimkan *request* untuk mengupdate data perubahan *id card*. *Request data update* perubahan *id card* dapat dilihat pada list 3.42.

```
String proses = request.getParameter("update-idcard");
```

List 3.42 *Request data update perubahan id card*

- b. Setelah *request* diterima oleh *controller*, *controller* memberikan *response* untuk mengupdate data perubahan *id card* dan menuju halaman *update* perubahan *id card*. *Response update data* perubahan *id card* dapat dilihat pada list 3.43.

```
response.sendRedirect("update_idcard.jsp?id=" +
request.getParameter("id"));
```

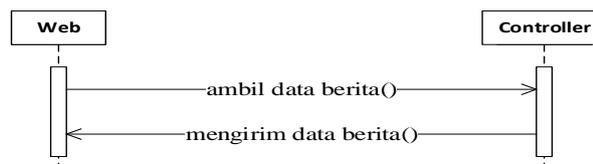
List 3.43 *Response update perubahan id card*

9. Sequence diagram mengelola berita

Sequence diagram mengelola berita dirancang berdasarkan *activity diagram* mengelola berita. *Sequence diagram* mengelola berita dibagi menjadi 3 bagian, yaitu :

Sequence diagram view berita

Sequence diagram berita menggambarkan rancangan arsitektur untuk proses berita. *Sequence diagram* berita dibuat berdasarkan *activity diagram* berita. Rancangan *sequence diagram* berita dapat dilihat pada gambar dibawah ini.



Gambar 3.60 *Sequence view diagram berita*

Berikut penjelasan *sequence* diagram berita pada gambar 3.60:

- a. Web mengirimkan *request* untuk mengambil data berita. *Request* data berita dapat dilihat pada *list* 3.44.

```
String dataBerita = request.getParameter("dataBerita ");
```

List 3.44 *Request* data berita

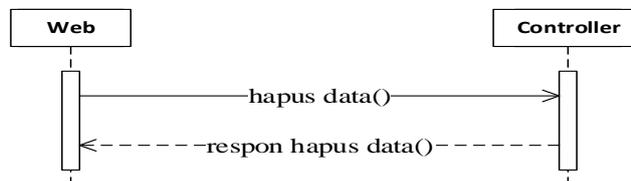
- b. Setelah *request* diterima oleh *controller*, *controller* mengirimkan *response* berisi data berita yang akan ditampilkan di web. *Response* data berita dapat dilihat pada *list* 3.45

```
dataBerita.setNik(Integer.parseInt(niikk));
dataBerita.setContent(feedContent);
dataBerita.setImg_content("feed/" + item.getName());
```

List 3.45 *Response* data berita

Sequence diagram hapus berita

Sequence diagram hapus berita menggambarkan rancangan arsitektur untuk proses hapus berita. *Sequence* diagram hapus berita dibuat berdasarkan *activity* diagram hapus berita. Rancangan *sequence* diagram hapus berita dapat dilihat pada gambar dibawah ini.



Gambar 3.61 *Sequence* diagram hapus berita

Berikut penjelasan *sequence* diagram hapus berita pada gambar 3.61:

- a. Web mengirimkan *request* untuk menghapus data berita. *Request* data hapus berita dapat dilihat pada *list* 3.46.

```
String proses = request.getParameter("hapus-berita");
```

List 3.46 *Request* hapus berita

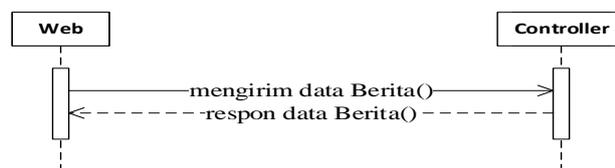
- b. Setelah *request* diterima oleh *controller*, *controller* memberikan *response* untuk menuju ke halaman berita. *Response* hapus berita dapat dilihat pada *list* 3.47

```
response.sendRedirect("index.jsp");
```

List 3.47 *Response* hapus berita

Sequence diagram tambah berita

Sequence diagram tambah berita menggambarkan rancangan arsitektur untuk proses tambah berita. *Sequence* diagram tambah berita dibuat berdasarkan *activity* diagram tambah berita. Rancangan *sequence* diagram tambah berita dapat dilihat pada gambar dibawah ini.



Gambar 3.62 *Sequence* diagram tambah berita

Berikut penjelasan *sequence* diagram tambah berita pada gambar 3.62:

- Web mengirimkan *request* untuk menambah data berita. *Request* data tambah berita dapat dilihat pada *list* 3.48.

```
String proses = request.getParameter("input-berita");
```

List 3.48 *Request* tambah berita

- Setelah *request* diterima oleh *controller*, *controller* memberikan *response* untuk menuju ke halaman berita. *Response* tambah berita dapat dilihat pada *list* 3.49

```
response.sendRedirect("insert_berita.jsp");
```

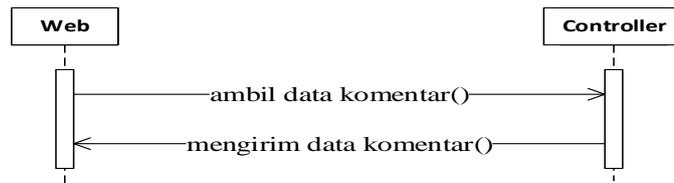
List 3.49 *Response* tambah berita

10. Sequence diagram mengelola komentar

Sequence diagram mengelola komentar dirancang berdasarkan *activity* diagram mengelola komentar. *Sequence* diagram mengelola komentar dibagi menjadi 3 bagian, yaitu :

Sequence diagram *view* komentar

Sequence diagram komentar menggambarkan rancangan arsitektur untuk proses komentar. *Sequence* diagram berita dibuat berdasarkan *activity* diagram komentar. Rancangan *sequence* diagram komentar dapat dilihat pada gambar dibawah ini.



Gambar 3.63 *Sequence* view diagram komentar

Berikut penjelasan *sequence* diagram komentar beranda pada gambar 3.63:

- Web mengirimkan *request* untuk mengambil data komentar. *Request* data komentar dapat dilihat pada *list* 3.50.

```
String dataKomentar = request.getParameter("dataKomentar");
```

List 3.50 *Request* data komentar

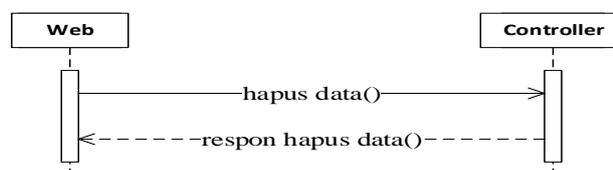
- Setelah *request* diterima oleh *controller*, *controller* mengirimkan *response* berisi data komentar yang akan ditampilkan di web. *Response* data komentar dapat dilihat pada *list* 3.51

```
response.setId(Integer.parseInt(request.getParameter("id")));
response.setCmtcontent(request.getParameter("cmtcontent"));
response.setCmtposdate(request.getParameter("cmtposdate"));
response.setRoom_id(Integer.parseInt(request.getParameter("room id")));
```

List 3.51 *Response* data komentar

Sequence diagram hapus komentar

Sequence diagram hapus komentar menggambarkan rancangan arsitektur untuk proses hapus komentar. *Sequence* diagram hapus komentar dibuat berdasarkan *activity* diagram hapus komentar. Rancangan *sequence* diagram hapus komentar dapat dilihat pada gambar dibawah ini.



Gambar 3.64 *Sequence* diagram hapus komentar

Berikut penjelasan *sequence* diagram hapus komentar pada gambar 3.64:

- Web mengirimkan *request* untuk menghapus data komentar. *Request* data hapus komentar dapat dilihat pada *list* 3.52.

```
String proses = request.getParameter("hapus-komrntar");
```

List 3.52 *Request* hapus komentar

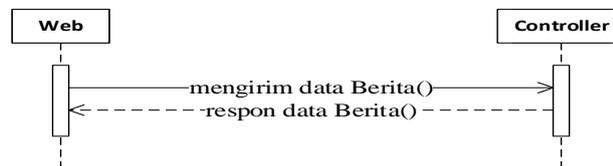
- b. Setelah *request* diterima oleh *controller*, *controller* memberikan *response* untuk menuju ke halaman komentar. *Response* hapus komentar dapat dilihat pada *list 3.53*

```
response.sendRedirect("index.jsp");
```

List 3.53 *Response* hapus komentar

Sequence diagram tambah komentar

Sequence diagram tambah komentar menggambarkan rancangan arsitektur untuk proses tambah komentar. *Sequence* diagram tambah komentar dibuat berdasarkan *activity* diagram tambah komentar. Rancangan *sequence* diagram tambah komentar dapat dilihat pada gambar dibawah ini.



Gambar 3.65 *Sequence* diagram tambah komentar

Berikut penjelasan *sequence* diagram tambah komentar pada gambar 3.65:

- a. Web mengirimkan *request* untuk menambah data komentar. *Request* data tambah komentar dapat dilihat pada *list 3.54*.

```
String proses = request.getParameter("input-komentar");
```

List 3.54 *Request* tambah komentar

- b. Setelah *request* diterima oleh *controller*, *controller* memberikan *response* untuk menuju ke halaman komentar. *Response* tambah komentar dapat dilihat pada *list 3.55*

```
response.sendRedirect("index.jsp");
```

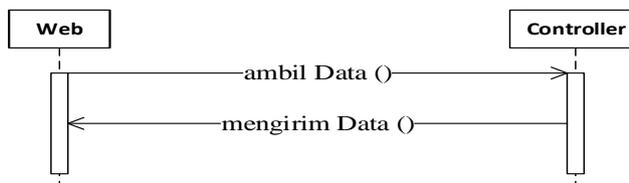
List 3.55 *Response* tambah komentar

11. Sequence diagram menerima feedback

Sequence diagram menerima *feedback* dirancang berdasarkan *activity* diagram menerima *feedback*. *Sequence* diagram menerima *feedback* dibagi menjadi 3 bagian, yaitu :

Sequence diagram *view feedback*

Sequence diagram feedback menggambarkan rancangan arsitektur untuk proses *feedback*. *Sequence diagram feedback* dibuat berdasarkan *activity diagram feedback*. Rancangan *sequence diagram feedback* dapat dilihat pada gambar dibawah ini



Gambar 3.66 *Sequence diagram view feedback*

Berikut penjelasan *sequence diagram feedback* pada gambar 3.66 :

- Web mengirimkan *request* data untuk mengirim data *feedback*. *Request* data *feedback* dapat dilihat pada *list 3.56*.

```
String dataFeedback = request.getParameter("dataFeedback ");
```

List 3.56 *Request feedback*

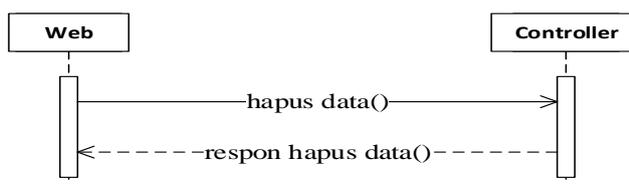
- Setelah *request* diterima oleh *controller*, *controller* mengirimkan *response* berisi data *feedback* yang akan ditampilkan di web. *Response* data *feedback* dapat dilihat pada *list 3.57*

```
response.setId(Integer.parseInt(request.getParameter("id")));
response.setContent(request.getParameter("content"));
response.setPosdate(request.getParameter("posdate"));
response.setNik(Integer.parseInt(request.getParameter("nik")));
```

List 3.57 *Response data feedback*

Sequence diagram *hapus feedback*

Sequence diagram hapus feedback menggambarkan rancangan arsitektur untuk proses *hapus feedback*. *Sequence diagram hapus feedback* dibuat berdasarkan *activity diagram hapus feedback*. Rancangan *sequence diagram hapus feedback* dapat dilihat pada gambar dibawah ini.



Gambar 3.67 *Sequence diagram hapus feedback*

Berikut penjelasan *sequence* diagram hapus *feedback* pada gambar 3.67 :

- a. Web mengirimkan *request* untuk menghapus data *feedback*. *Request* data hapus *feedback* dapat dilihat pada *list* 3.58

```
String proses = request.getParameter("hapus-feedback");
```

List 3.58 *Request* hapus *feedback*

- b. Setelah *request* diterima oleh *controller*, *controller* memberikan *response* untuk menuju ke halaman *feedback*. *Response* hapus *feedback* dapat dilihat pada *list* 3.59

```
response.sendRedirect("feedback.jsp");
```

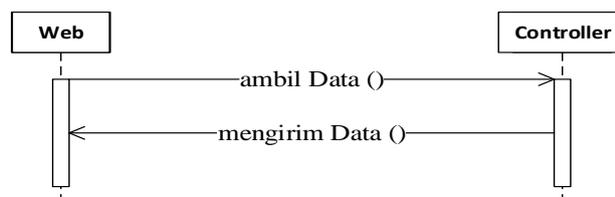
List 3.59 *Response* hapus *feedback*

12. Sequence diagram menerima data karyawan

Sequence diagram menerima data karyawan dirancang berdasarkan *activity* diagram menerima data karyawan. *Sequence* diagram menerima data karyawan dibagi menjadi 4 bagian, yaitu :

Sequence diagram *view* karyawan

Sequence diagram karyawan menggambarkan rancangan arsitektur untuk proses karyawan. *Sequence* diagram karyawan dibuat berdasarkan *activity* diagram karyawan. Rancangan *sequence* diagram karyawan dapat dilihat pada gambar dibawah ini



Gambar 3.68 *Sequence* diagram *view* karyawan

Berikut penjelasan *sequence* diagram karyawan pada gambar 3.68 :

- a. Web mengirimkan *request* data untuk mengirim data karyawan. *Request* data karyawan dapat dilihat pada *list* 3.60.

```
String datKaryawan = request.getParameter("datKaryawan");
```

List 3.60 *Request* karyawan

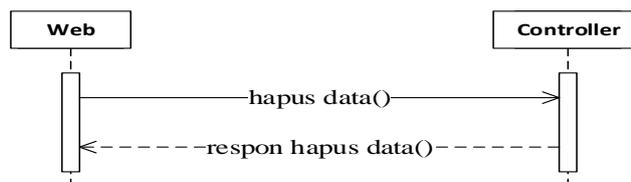
- b. Setelah *request* diterima oleh *controller*, *controller* mengirimkan *response* berisi data karyawan yang akan ditampilkan di web. *Response* data karyawan dapat dilihat pada *list 3.61*

```
response.setNik(Integer.parseInt(request.getParameter("nik")));
response.setDept(request.getParameter("dept"));
response.setEmail(request.getParameter("email"));
response.setImg_user(request.getParameter("img_user"));
response.setName(request.getParameter("name"));
response.setPhone(request.getParameter("phone"));
response.setPlant(request.getParameter("plant"));
response.setPwd(request.getParameter("pwd"));
response.setReg_date(request.getParameter("reg_date"));
response.setTag(request.getParameter("tag"));
```

List 3.61 *Response* karyawan

Sequence diagram hapus karyawan

Sequence diagram hapus karyawan menggambarkan rancangan arsitektur untuk proses hapus karyawan. *Sequence* diagram hapus karyawan dibuat berdasarkan *activity* diagram hapus karyawan. Rancangan *sequence* diagram hapus karyawan dapat dilihat pada gambar dibawah ini.



Gambar 3.69 *Sequence* diagram hapus karyawan

Berikut penjelasan *sequence* diagram hapus karyawan pada gambar 3.69 :

- a. Web mengirimkan *request* untuk menghapus data karyawan. *Request* data hapus karyawan dapat dilihat pada *list 3.62*

```
String proses = request.getParameter("hapus-user");
```

List 3.62 *Request* hapus karyawan

- b. Setelah *request* diterima oleh *controller*, *controller* memberikan *response* untuk menuju ke halaman karyawan. *Response* hapus keluhan absensi dapat dilihat pada *list 3.63*

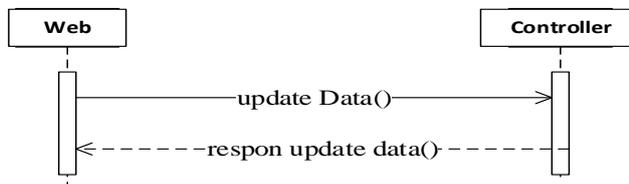
```
response.sendRedirect("User.jsp");
```

List 3.63 *Response* hapus karyawan

Sequence diagram update karyawan

Sequence diagram *update* karyawan menggambarkan rancangan arsitektur untuk proses *update* karyawan. *Sequence* diagram *update* karyawan dibuat

berdasarkan *activity* diagram *update* karyawan. Rancangan *sequence* diagram *update* karyawan dapat dilihat pada gambar dibawah ini.



Gambar 3.70 *Sequence* diagram *update* karyawan

Berikut penjelasan *sequence* diagram *update* karyawan pada gambar 3.70 :

- a. Web mengirimkan *request* untuk mengupdate data karyawan. *Request* data *update* karyawan dapat dilihat pada list 3.64

```
String proses = request.getParameter("update-user");
```

List 3.64 *Request* update karyawan

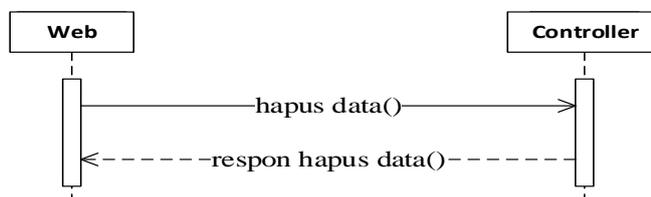
- b. Setelah *request* diterima oleh *controller*, *controller* memberikan *response* untuk mengupdate data karyawan dan menuju halaman *update* karyawan. *Response* update data karyawan dapat dilihat pada list 3.65.

```
response.sendRedirect("update_user.jsp?nik="
request.getParameter("nik"));
```

List 3.65 *Response* update karyawan

Sequence diagram tambah karyawan

Sequence diagram tambah karyawan menggambarkan rancangan arsitektur untuk proses tambah karyawan. *Sequence* diagram tambah karyawan dibuat berdasarkan *activity* diagram tambah karyawan. Rancangan *sequence* diagram tambah karyawan dapat dilihat pada gambar dibawah ini.



Gambar 3.71 *Sequence* diagram tambah karyawan

Berikut penjelasan *sequence* diagram tambah karyawan pada gambar 3.71 :

- a. Web mengirimkan *request* untuk menambah data karyawan. *Request* data tambah karyawan dapat dilihat pada list 3.66

```
String proses = request.getParameter("input-karyawan");
```

List 3.66 Request tambah karyawan

- b. Setelah *request* diterima oleh *controller*, *controller* memberikan *response* untuk menuju ke halaman karyawan. *Response* tambah karyawan dapat dilihat pada *list 3.67*

```
response.sendRedirect("insert_user.jsp");
```

List 3.67 Response tambah karyawan

3.3.6 ER Diagram

Berdasarkan kebutuhan pada *class* diagram maka dirancanglah *Entity Relationship* Diagram (ERD). ER diagram digunakan untuk mengatur dan mendokumentasikan data dalam pembangunan web admin.

ER Diagram yang digunakan dapat dilihat pada gambar 3.71.

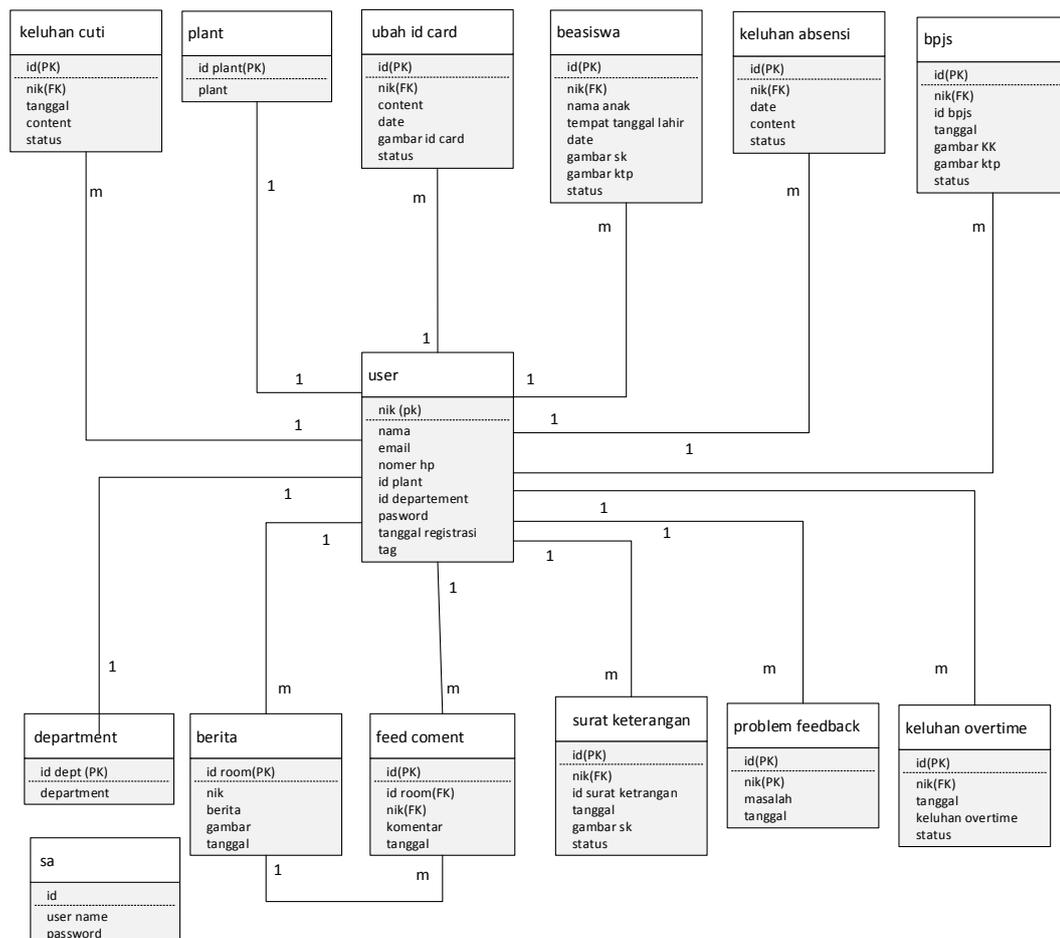
Pada web admin di PT Indocement Tunggal Prakasa Tbk. terdapat 13 entitas yaitu:

1. Entitas user memiliki atribut nik, nama, plant, departement, nomor telepon, email, password, foto, tanggal registrasi dan tag.
2. Entitas user memiliki relasi *one to one* dengan entitas plant, karena user hanya dapat menggunakan 1 plant.
3. Entitas user memiliki relasi *one to one* dengan entitas departemen, karena user hanya dapat menggunakan 1 departemen.
4. Entitas user memiliki relasi *one to many* dengan entitas beranda, karena user dapat melihat semua berita yang ditampilkan.
5. Entitas beranda memiliki relasi *one to many* dengan entitas komentar beranda, karena dalam 1 beranda dapat berisi banyak komentar dari user.
6. Entitas user memiliki relasi *one to many* dengan entitas feedback, user dapat mengirimkan feedback lebih dari 1 kali.
7. Entitas user memiliki relasi *one to many* dengan entitas keluhan absensi, karena user dapat mengirimkan keluhan absensi lebih dari 1 kali.
8. Entitas user memiliki relasi *one to many* dengan entitas keluhan cuti, karena user dapat mengirimkan keluhan cuti lebih dari 1 kali.
9. Entitas user memiliki relasi *one to many* dengan entitas keluhan overtime, karena user dapat mengirimkan keluhan overtime lebih dari 1 kali.

10. Entitas user memiliki relasi *one to many* dengan Entitas Pengajuan beasiswa anak, karena user dapat mengirimkan pengajuan beasiswa anak lebih dari 1 kali.
11. Entitas user memiliki relasi *one to many* dengan Entitas pendaftaran bpjs, karena user dapat mengirimkan pendaftaran bpjs lebih dari 1 kali.
12. Entitas user memiliki relasi *one to many* dengan entitas perubahan id card, karena user dapat mengirimkan perubahan id card lebih dari 1 kali.
13. Entitas user memiliki relasi *one to many* dengan pengajuan surat keputusan, karena user dapat mengirimkan pengajuan surat keputusan lebih dari 1 kali.
14. Pada entitas user terdapat atribut *tag*, fungsi *tag* disini adalah untuk membedakan antara karyawan dengan admin. Tag admin berfungsi pada saat proses login admin.

Untuk memperjelas penjelasan diatas maka dapat dilihat pada gambar ER

Diagram web admin pada gambar dibawah ini.



Gambar 3.71 Entity Relationship Diagram

3.3.7 Perancangan Sistem

Pada tahap ini dilakukan perancangan arsitektur sistem yang akan digunakan. Tahap perancangan sistem akan dibagi menjadi 2 bagian yaitu :

A. Perancangan Model Data

Berdasarkan rancangan *sequence* diagram pada sub bab sebelumnya, maka dirancanglah model data. Model data dirancang berdasarkan kebutuhan data pada *sequence* diagram. Rancangan model data dibagi menjadi beberapa bagian yaitu sebagai berikut :

Model data user

Model data *user* dirancang berdasarkan data yang dikirimkan pada *sequence* diagram karyawan dan *login* pada *list sequence login* dan karyawan. Model data *user* dapat dilihat pada *list 3.68*.

```
public class mdl_user {
    private String dept,email, img_user, name, phone, plant, pwd, reg_date, tag;
    private Integer nik;
    public mdl_user(String dept, String email, String user, String name, String phone, String plant, String pwd, String reg_date, String tag, Integer nik){
        super();
        this.nik = nik;
        this.dept = dept;
        this.emsil = emsil;
        this.img_user = img_user;
        this.name = name;
        this.phone = phone;
        this.plant = plant;
        this.pwd = pwd;
        this.reg_date = reg_date;
        this.tag = tag;
    }
}
```

List 3.68 Model data user

Model data berita indek

Model data berita dirancang berdasarkan data yang dikirimkan pada *sequence* diagram indek dan berita pada *list sequence indek dan berita*. Model data berita dapat dilihat pada *list 3.69*.

```
public class mdl_berita {
    private String conten,img_content, posdate;
    private Integer nik, room_id;
    public mdl_berita(String content, String img_usr, String postdate,Integer nik, Integer room_id){
        super();
        this.room_id = room-id;
        this.content = content;
        this.img_content = img_content;
        this.postdate = postdate;
        this.nik = nik;
    }
}
```

List 3.69 Model data berita

Model data keluhan absensi

Model data absensi dirancang berdasarkan data yang dikirimkan pada *sequence* diagram hapus, *update* dan keluhan absensi pada *list sequence* hapus, *update* dan keluhan absensi. Model data absensi dapat dilihat pada *list 3.70*.

```
public class mdl_absensi {
    private String content, date, status;
    private Integer nik, id;
    public mdl_absensi(String content, String date, String status, Integer
    nik, Integer id)
    super(); }
    this.id = id;
    this.content = content;
    this.date = date;
    this.status = status;
    this.nik = nik;
}
```

List 3.70 Model data absensi

Model data keluhan overtime

Model data overtime dirancang berdasarkan data yang dikirimkan pada *sequence* diagram hapus, *update* dan keluhan overtime pada *list sequence* hapus, *update* dan keluhan overtime. Model data overtime dapat dilihat pada *list 3.71*.

```
public class mdl_overtime {
    private String content, date, status;
    private Integer nik, id;
    public mdl_overtime(private String content, String content, String
    content, Integer nik, Integer id){
    super();
        this.id = id;
        this.content = content;
        this.date = date;
        this.status = status;
        this.nik = nik;
    }
}
```

List 3.71 Model data overtime

Model data keluhan cuti

Model Model data cuti dirancang berdasarkan data yang dikirimkan pada *sequence* diagram hapus, *update* dan keluhan cuti pada *list sequence* hapus, *update* dan keluhan cuti. data cuti dapat dilihat pada *list 3.72*.

```
public class mdl_cuti {
    private String content, date, status;
    private Integer nik, Integer id;
    public mdl_cuti (private String content, String content, String
    content, Integer nik, Integer id){
    super();
        this.id = id;
        this.content = content;
        this.date = date;
    }
}
```

```

        this.status = status;
        this.nik = nik;
    }

```

List 3.72 Model data cuti

Model data pendaftaran beasiswa

Model data beasiswa dirancang berdasarkan data yang dikirimkan pada *sequence* diagram hapus, *update* dan pendafrtan beasiswa pada *list sequence* hapus, *update* dan pendafrtan beasiswa. Model data beasiswa dapat dilihat pada *list 3.73*.

```

public class mdl_beasiswa {

    private String date, doc, nm_ana, status, ttl;
    private Integer nik, id;
    public mdl_beasiswa(
String date, String doc, String nm_ana, String status, String ttl, Integer
nik, Integer id){
super();
        this.id = id;
        this.date = date;
        this.doc = doc;
        return nm_anak;
        this.nm_anak = nm_anak;
        this.status = status;
        this.ttl = ttl;
        this.nik = nik;
    }
}

```

List 3.73 Model data beasiswa

Model data pendaftaran BPJS

Model data BPJS dirancang berdasarkan data yang dikirimkan pada *sequence* diagram hapus, *update* dan pendafrtan BPJS pada *list sequence* hapus, *update* dan pendafrtan BPJS. Model data BPJS dapat dilihat pada *list 3.74*.

```

public class mdl_bpjs {
    private String date, nokk, noktp, Status;
    private Integer nik, id;
    public mdl_bpjs(String date, String nokk, String noktp, String Status,
Integer nik, Integer id) {
super();
        db = new Koneksi();
        this.id = id;
        this.date = date;
        this.nokk = nokk;
        this.noktp = noktp;
        this.doc = doc;
        this.nik = nik;
        this.Status = Status;
    }
}

```

List 3.74 Model data BPJS

Model data pengajuan surat keterangan

Model data surat keterangan dirancang berdasarkan data yang dikirimkan pada *sequence* diagram hapus, *update* dan pengajuan surat keterangan pada *list sequence* hapus, *update* dan pengajuan surat keterangan. Model data surat keterangan dapat dilihat pada *list 3.75*.

```
public class mdl_suratKeterangan {
    private String date, jnssurat, doc, status;
    private Integer nik, id;
    public mdl_suratKeterangan(String date, String jnssurat, String doc,
String status, Integer nik, Integer id;) {
super();
    this.id = id;
    this.date = date;
    this.jnssurat = jnssurat;
    this.doc = doc;
    this.status = status;
    this.nik = nik;
}
}
```

List 3.75 Model data bsurat keterangan

Model data pengajuan id card

Model data *id card* dirancang berdasarkan data yang dikirimkan pada *sequence* diagram hapus, *update* dan pengajuan *id card* pada *list sequence* hapus, *update* dan pengajuan *id card*. Model data *id card* dapat dilihat pada *list 3.76*.

```
public class mdl_card {
    private String content, date, doc, status;
    private Integer nik, id;
    public mdl_card(String content, String date, String doc, String
status, Integer nik, Integer id;) {
super();
    this.id = id;
    this.content = content;
    this.date = date;
    this.doc = doc;
    this.status = status;
    this.nik = nik;
}
}
```

List 3.76 Model data *id card*

Model data komentar

Model data komentar dirancang berdasarkan data yang dikirimkan pada *sequence* diagram hapus, tambah dan komentar pada *list sequence* hapus, tambah dan komentar. Model data komentar dapat dilihat pada *list 3.77*.

```
public class mdl_komentar {
    private String cmtcontent, cmtposdate;
    private Integer nik, room_id, id;
    public mdl_komentar(String cmtcontent, String cmtposdate, Integer nik,
Integer room_id, Integer id ) {
}
```

```

super();
    this.id = id;
    this.cmtcontent = cmtcontent;
    this.cmtposdate = cmtposdate;
    this.nik = nik;
    this.room_id = room_id;
}

```

List 3.77 Model data komentar

Model data *feedback*

Model data *feedback* dirancang berdasarkan data yang dikirimkan pada *sequence* diagram hapus *feedback* dan *feedback* pada *list sequence* hapus *feedback* dan *feedback*. Model data *feedback* dapat dilihat pada *list* 3.78.

```

public class mdl_feedbak {
    private String content, posdate;
    private Integer nik, id;
    public mdl_feedbak(String content, String postdate, Integer nik,
Integer id;) {
super();
    this.id = id;
    this.content = content;
    this.posdate = posdate;
    this.nik = nik;
}
}

```

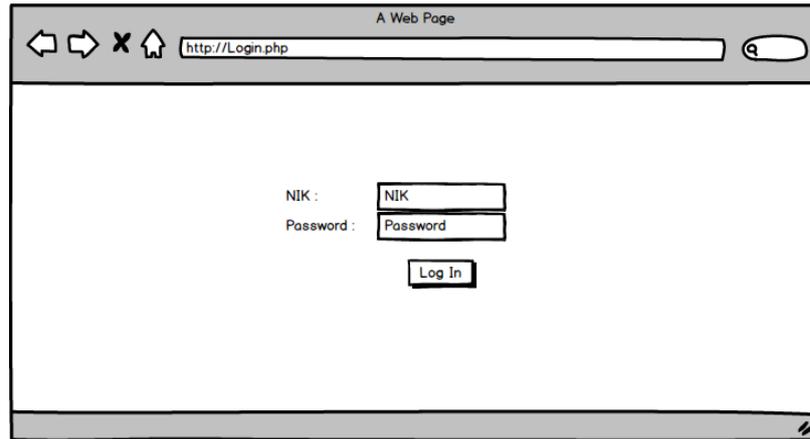
List 3.78 Model data *feedback*

B. Perancangan Antarmuka

Interface atau antarmuka merupakan tampilan dari aplikasi yang menjadi sarana komunikasi antara pengguna dengan sistem, serta memberikan kemudahan kepada pengguna dalam melakukan aktivitas didalam penggunaan *web*.

Rancangan Tampilan *Login*

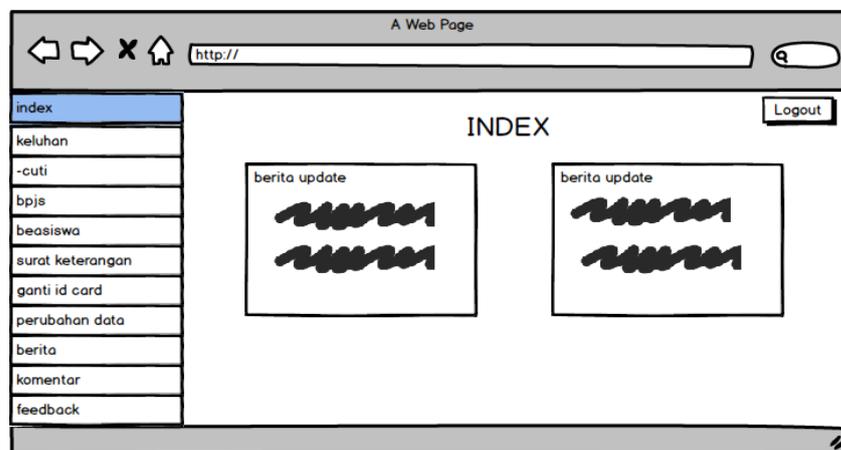
Rancangan antarmuka ini merupakan halaman validasi *nik* dan *password* yang berfungsi untuk melakukan *login* kedalam aplikasi. Gambaran rancangan antarmuka halaman *login* dapat dilihat pada Gambar 3.72.



Gambar 3.72 Tampilan *login*

Tampilan Halaman Indek Admin

Rancangan antarmuka ini merupakan halaman Indek yang akan tampil setelah melakukan *login*. Pada halaman ini terdapat fitur – fitur yang dapat digunakan oleh admin dalam melakukan pengelolaan data. Terdapat beberapa fitur diantaranya adalah fitur keluhan absensi, keluhan cuti, keluhan overtime, BPJS, beasiswa, perubahan *id card*, surat keterangan, data karyawan dan lain-lain. Gambaran rancangan antarmuka halaman ini dapat dilihat pada gambar 3.73.

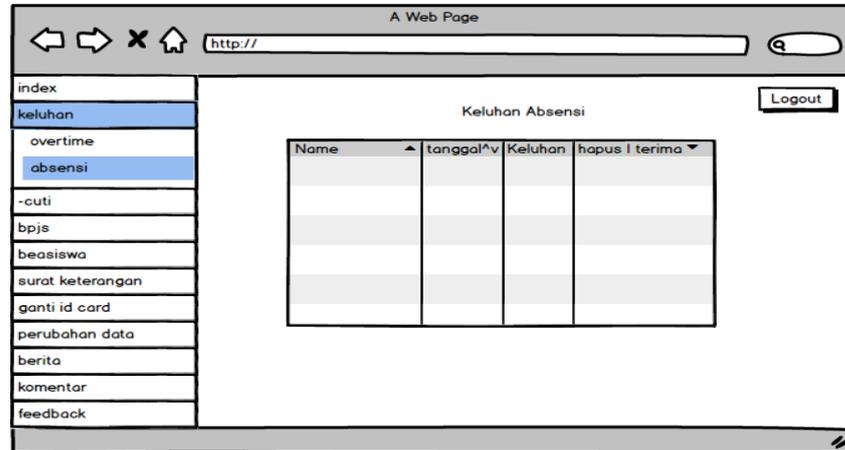


Gambar 3.73 Tampilan halaman indek admin

Tampilan Halaman Keluhan Absensi

Rancangan antarmuka ini merupakan halaman tampilan dari halaman keluhan absensi yang berisi data keluhan absensi yang menggunakan aplikasi. Pada halaman ini terdapat beberapa fitur yang admin bisa lakukan seperti melihat data, menerima, mencari, meng*update* dan juga ada tombol logout jika dirasa admin sudah beres mengerjakan aktifitas di

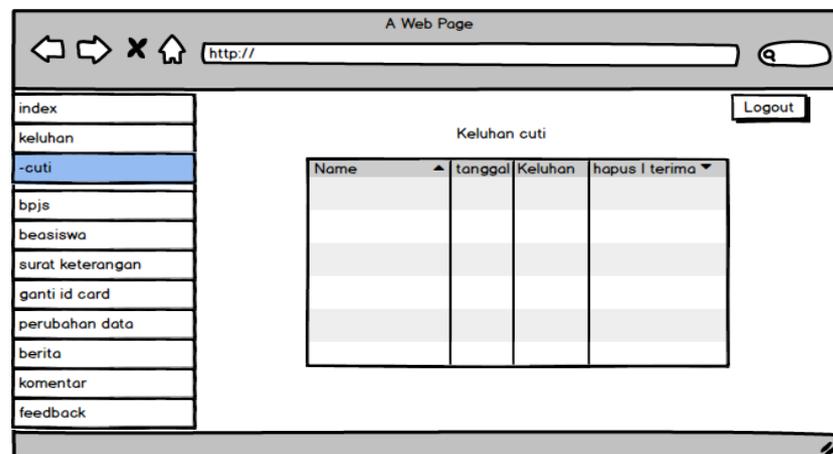
halaman keluhan absensi. Pada fitur hapus disini admin bisa menghapus data yang sekiranya sudah tidak di perlukan lagi. fitur *update* berguna untuk mengganti status dari *waiting* menjadi *accpet*. Gambaran rancangan antarmuka halaman ini dapat dilihat pada gambar 3.74.



Gambar 3.74 Tampilan halaman keluhan absensi

Tampilan Halaman Keluhan Cuti

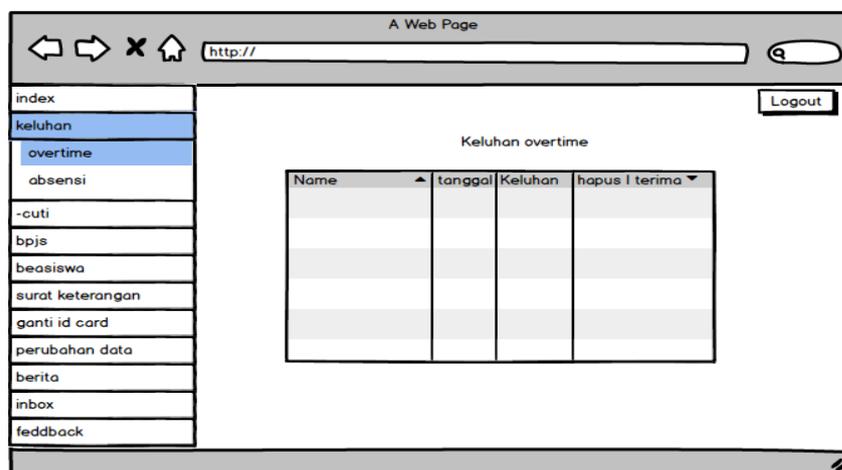
Rancangan antarmuka ini merupakan halaman tampilan dari halaman keluhan cuti yang berisi data keluhan cuti karyawan yang menggunakan aplikasi. Pada halaman ini terdapat beberapa yang admin bisa lakukan seperti melihat data, menerima, mencari, meng*update* dan juga ada tombol logout jika dirasa admin sudah beres mengerjakan aktifitas di halaman keluhan cuti. Pada fitur hapus disini admin bisa menghapus data yang sekiranya sudah tidak di perlukan lagi. fitur *update* berguna untuk mengganti status dari *waiting* menjadi *accpet*. Gambaran rancangan antarmuka halaman ini dapat dilihat pada gambar 3.75



Gambar 3.75 Tampilan halaman keluhan cuti

Tampilan Halaman Keluhan Overtime

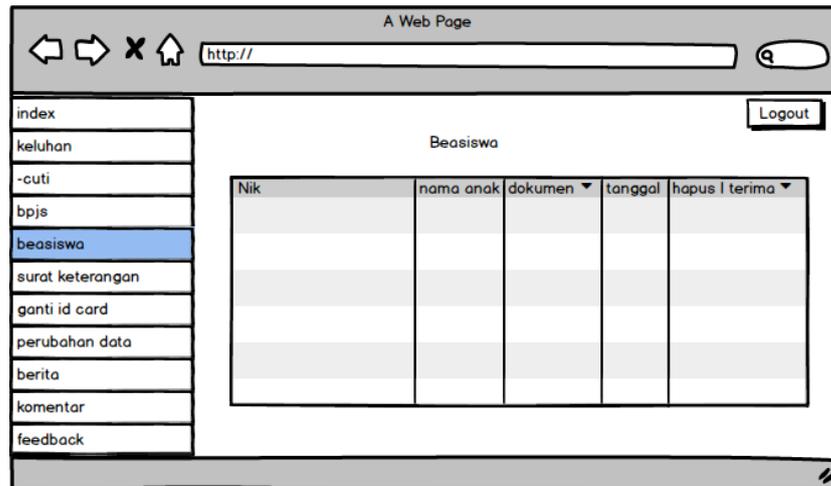
Rancangan antarmuka ini merupakan halaman tampilan dari halaman keluhan overtime yang berisi data keluhan overtime karyawan yang menggunakan aplikasi. Pada halaman ini terdapat beberapa yang admin bisa lakukan seperti melihat data, menerima, mencari, mengupdate dan juga ada tombol logout jika dirasa admin sudah beres mengerjakan aktifitas di halaman keluhan overtime. Pada fitur hapus disini admin bisa menghapus data yang sekiranya sudah tidak di perlukan lagi. fitur *update* berguna untuk mengganti status dari *waiting* menjadi *accpet*. Gambaran rancangan antarmuka halaman ini dapat dilihat pada gambar 3.76.



Gambar 3.76 Tampilan halaman keluhan overtime.

Tampilan Halaman Pengajuan Beasiswa

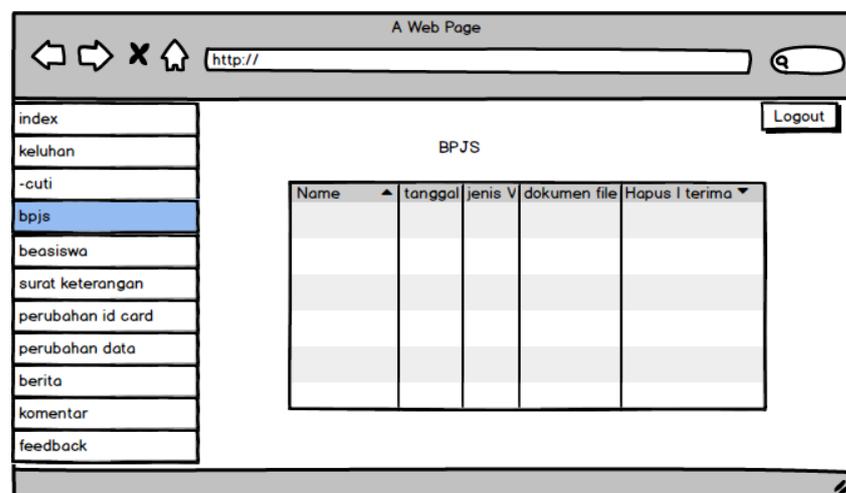
Rancangan antarmuka ini merupakan halaman tampilan dari halaman pengajuan beasiswa yang berisi data pengajuan beasiswa karyawan yang menggunakan aplikasi. Pada halaman ini terdapat beberapa yang admin bisa lakukan seperti melihat data, menerima, mencari, mengupdate dan juga ada tombol logout jika dirasa admin sudah cukup mengerjakan aktifitas di halaman pengajuan beasiswa. Pada fitur hapus disini admin bisa menghapus data yang sekiranya sudah tidak di perlukan lagi yang berada di table tersebut. fitur *update* berguna untuk mengganti status dari *waiting* menjadi *accpet*. Gambaran rancangan antarmuka halaman ini dapat dilihat pada gambar 3.77



Gambar 3.77 Tampilan halaman pengajuan beasiswa

Tampilan Halaman Pengajuan BPJS

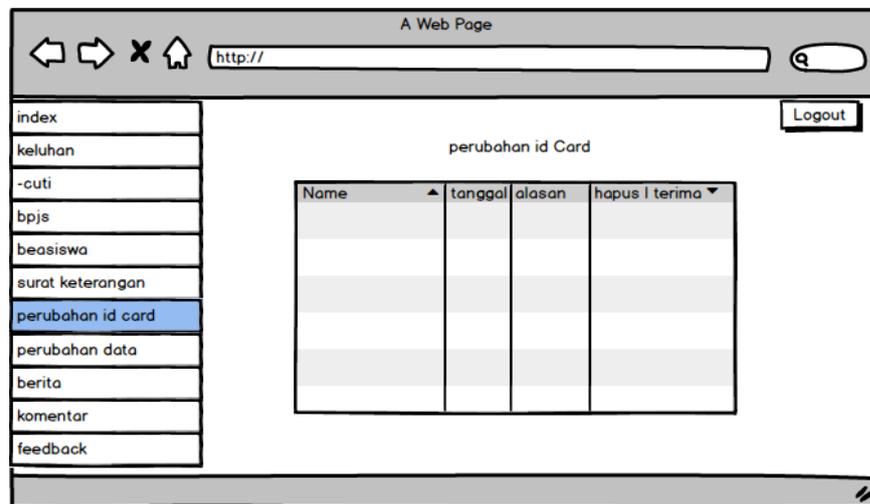
Rancangan antarmuka ini merupakan halaman tampilan dari halaman Pengajuan BPJS yang berisi data Pengajuan BPJS karyawan yang menggunakan aplikasi. Pada halaman ini terdapat beberapa yang admin bisa lakukan seperti melihat data, menerima, mencari, *update* dan juga ada tombol logout jika dirasa admin sudah cukup mengerjakan aktifitas di halaman Pengajuan BPJS. Pada fitur hapus disini admin bisa menghapus data yang sekiranya sudah tidak di perlukan lagi yang berada di table tersebut. fitur *update* berguna untuk mengganti status dari *waiting* menjadi *accpet*. Gambaran rancangan antarmuka halaman ini dapat dilihat pada gambar 3.78.



Gambar 3.78 Tampilan halaman pengajuan BPJS

Tampilan Halaman Perubahan Id Card

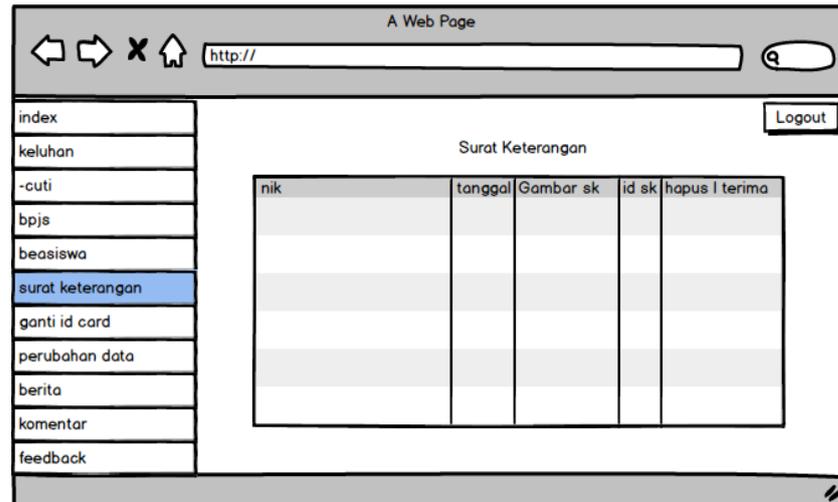
Rancangan antarmuka ini merupakan halaman tampilan dari halaman perubahan id card yang berisi data perubahan id card karyawan yang menggunakan aplikasi. Pada halaman ini terdapat beberapa yang admin bisa lakukan seperti melihat data, menerima, mencari, meng*update* dan juga ada tombol logout jika dirasa admin sudah cukup mengerjakan aktifitas di halaman perubahan id card. Pada fitur hapus disini admin bisa menghapus data yang sekiranya sudah tidak diperlukan lagi yang berada di table tersebut. fitur *update* berguna untuk mengganti status dari *waiting* menjadi *accpet*. Gambaran rancangan antarmuka halaman ini dapat dilihat pada gambar 3.79.



Gambar 3.79 Tampilan halaman perubahan id card

Tampilan Halaman Pengajuan Surat Keterangan (SK)

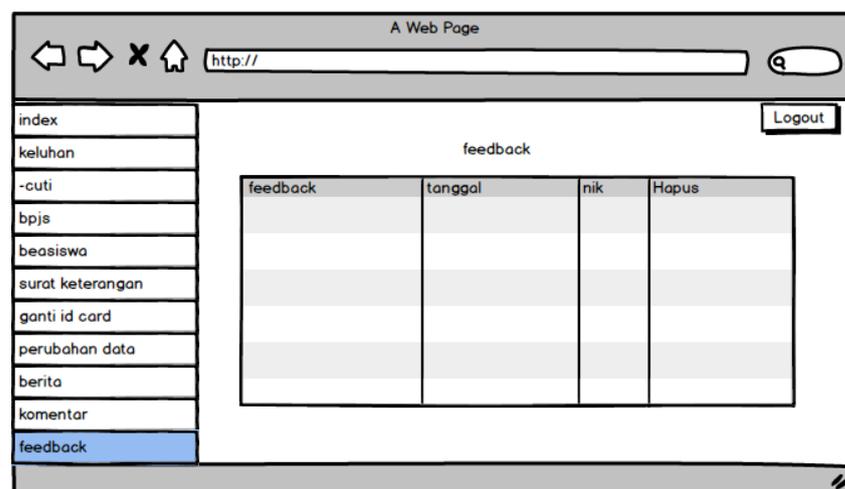
Rancangan antarmuka ini merupakan halaman tampilan dari halaman Pengajuan SK yang berisi data Pengajuan SK karyawan yang menggunakan aplikasi. Pada halaman ini terdapat beberapa yang admin bisa lakukan seperti melihat data, menerima, mencari, meng*update* dan juga ada tombol logout jika dirasa admin sudah cukup mengerjakan aktifitas di halaman Pengajuan SK. Pada fitur hapus disini admin bisa menghapus data yang sekiranya sudah tidak di perlukan lagi yang berada di table tersebut. fitur *update* berguna untuk mengganti status dari *waiting* menjadi *accpet*. Gambaran rancangan antarmuka halaman ini dapat dilihat pada gambar 3.80



Gambar 3.80 Tampilan halaman pengajuan surat keterangan

Tampilan Halaman Feedback

Rancangan antarmuka ini merupakan halaman tampilan dari halaman feedback yang berisi data feedback para karyawan yang menggunakan aplikasi. Pada halaman ini terdapat beberapa yang admin bisa lakukan seperti melihat data, menerima, mencari dan juga ada tombol logout jika dirasa admin sudah cukup mengerjakan aktifitas di halaman feedback. Pada fitur hapus disini admin bisa menghapus data yang sekiranya sudah tidak di perlukan lagi yang berada di table tersebut. Gambaran rancangan antarmuka halaman ini dapat dilihat pada gambar 3.81

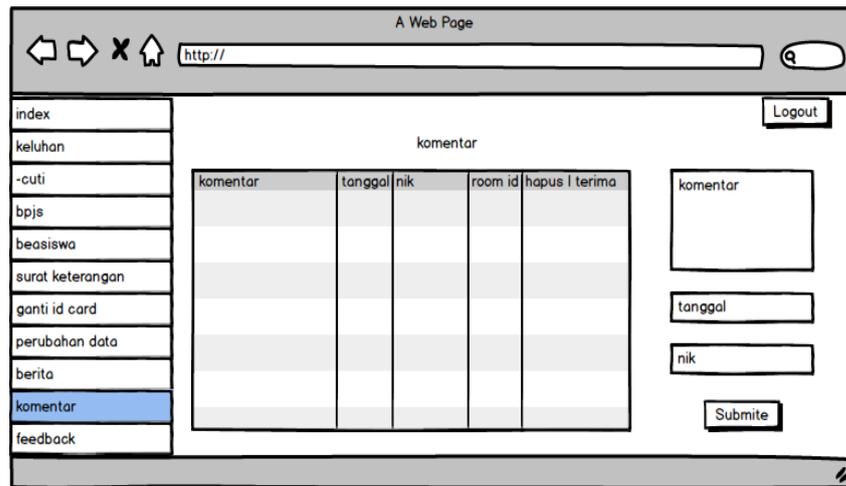


Gambar 3.81 Tampilan halaman feedback

Tampilan Halaman Komentar

Rancangan antarmuka ini merupakan halaman tampilan dari halaman komentar yang berisi data komentar para karyawan yang

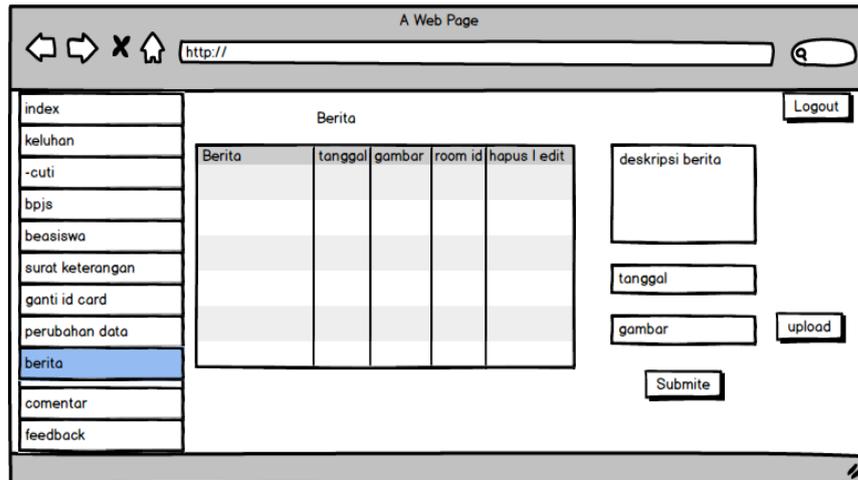
menggunakan aplikasi. Pada halaman ini terdapat beberapa yang admin bisa lakukan seperti menambah data, melihat data, menerima, mencari, menghapus dan juga ada tombol logout jika dirasa admin sudah cukup mengerjakan aktifitas di halaman feedback. Pada fitur hapus disini admin bisa menghapus komentar yang sekiranya sudah tidak di perlukan lagi yang berada di table tersebut. Pada fitur membuat komentar admin akan membuat komentar guna membalas komentar yang berada di *newsfeed*. Gambaran rancangan antarmuka halaman ini dapat dilihat pada gambar 3.82.



Gambar 3.82 Tampilan halaman komentar

Tampilan Halaman Berita

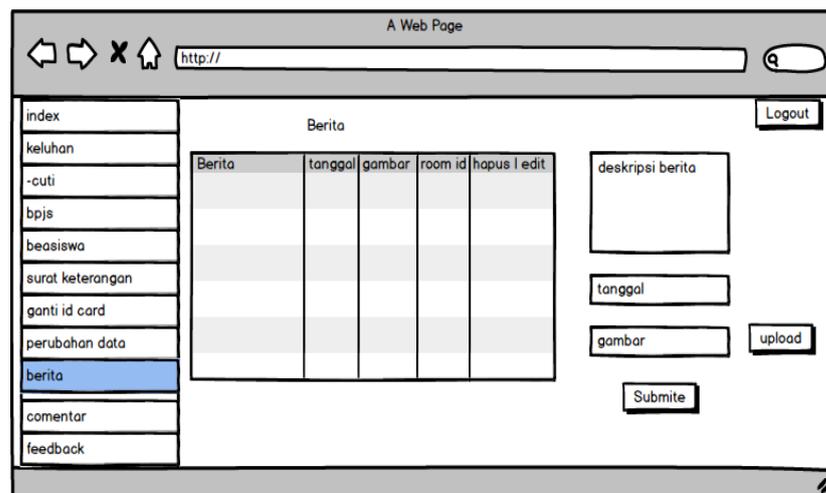
Rancangan antarmuka ini merupakan halaman tampilan dari halaman berita yang berisi data berita yang admin buat guna memberi informasi kepada karyawan di aplikasi. Pada halaman ini terdapat beberapa yang admin bisa lakukan seperti melihat data, membuat, menerima, mencari, menghapus dan juga ada tombol logout jika dirasa admin sudah cukup mengerjakan aktifitas di halaman *newsfeed*. Pada fitur hapus disini admin bisa menghapus berita yang sekiranya sudah tidak di perlukan lagi yang berada di table tersebut. Pada fitur membuat berita admin akan membuat berita guna menyebarkan informasi yang berada di *newsfeed*. Gambaran rancangan antarmuka halaman ini dapat dilihat pada gambar 3.83.



Gambar 3.83 Tampilan halaman Berita

Tampilan Halaman Karyawan

Rancangan antarmuka ini merupakan halaman tampilan dari halaman karyawan yang berisi data karyawan karyawan yang menggunakan aplikasi. Pada halaman ini terdapat beberapa yang admin bisa lakukan seperti melihat data, membuat, menghapus, menerima, mencari, mengupdate dan juga ada tombol logout jika dirasa admin sudah cukup mengerjakan aktifitas di halaman pengajuan beasiswa. Pada fitur hapus disini admin bisa menghapus data yang sekiranya sudah tidak di perlukan lagi yang berada di table tersebut. Adapun fitur *update* digunakan untuk mengupdate data karyawan yang sekiranya perlu diupdate. Gambaran rancangan antarmuka halaman ini dapat dilihat pada gambar 3.84.



Gambar 3.84 Tampilan halaman karyawan