

BAB 3 METODOLOGI PENELITIAN

3.1 Studi Literatur

Pada tahapan ini dikumpulkanlah berbagai informasi mengenai sistem pakar, metode Dempster-Shafer pada sistem pakar, dan mengukur tingkat kepuasan pengguna menggunakan *End User Computing Satisfaction* (EUCS). Informasi diperoleh melalui buku, jurnal dan artikel.

3.2 Pengumpulan Data

Pada tahapan ini pengumpulan data seperti penyakit pada hewan ternak, gejala yang dialami oleh hewan ternak, penjelasan dari suatu penyakit, dan cara pengobatan atau pencegahan bagi hewan ternak yang terdampak. Informasi tersebut didapat dari buku.

Setelah dikumpulkannya berbagai penyakit, gejala, penjelasan dan solusi atau pencegahan maka akan dilakukan sesi wawancara dengan dokter hewan, yang bertujuan untuk memverifikasi informasi solusi, gejala, penjelasan dan solusi atau pencegahan pada hewan ternak.

3.3 Perancangan Aplikasi

Pada tahapan ini dilakukan perancangan dari kebutuhan yang dimiliki aplikasi yang akan dibangun, seperti menggunakan *Unified Modeling Language* (UML), *Sitemap*, dan *Design Interface*.

3.4 Implementasi

Pada tahapan ini dibuatlah sebuah website sesuai dengan rancangan yang telah dilakukan, website dibangun menggunakan bahasa pemrograman PHP, dengan *framework* Laravel serta menggunakan *database* MySQL.

Pada tahapan ini juga diimplementasikan metode Dempster-Shafer untuk mendiagnosa penyakit pada hewan ternak dengan data yang telah disiapkan.

3.5 Pengujian

Pada tahapan ini dilakukannya pengujian terhadap sistem pakar yang dibuat, diuji oleh pakar, untuk memastikan hasil yang diuji sesuai dengan hasil yang diharapkan.

3.6 Evaluasi

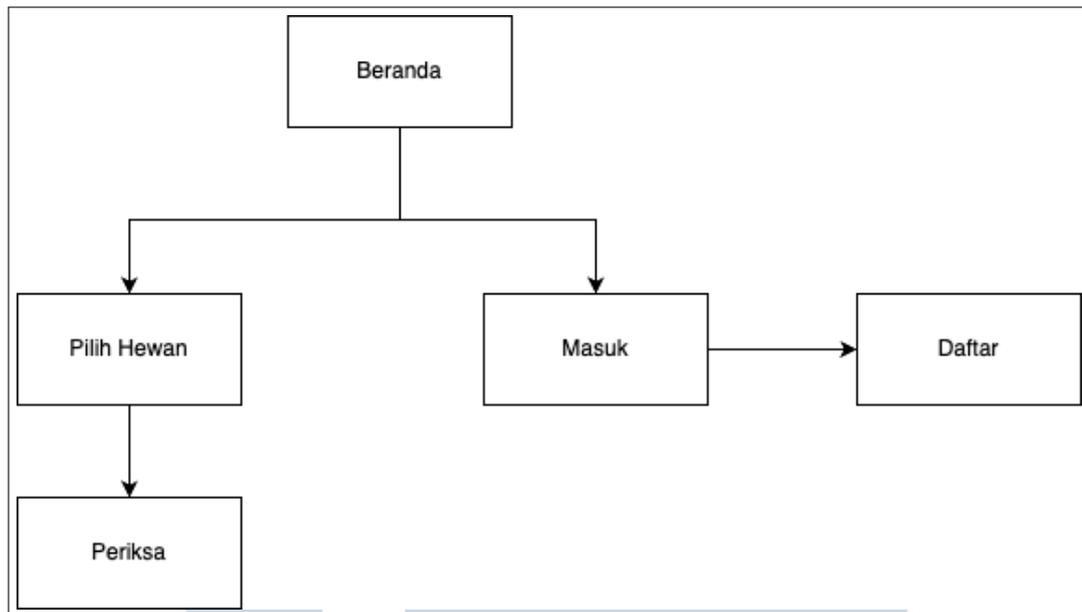
Pada tahapan ini dilakukannya evaluasi terhadap kesesuaian hasil diagnosis dari sistem dan diagnosis yang dilakukan oleh pakar, untuk mendapatkan tingkat akurasi pada sistem pakar yang dibuat, serta dibuat juga evaluasi tingkat kepuasan pengguna dengan metode EUCS.

3.7 Sitemap

Sitemap ini dibuat bertujuan untuk memperlihatkan halaman-halaman apa saja yang terdapat pada website, pada penelitian ini. Ada tiga *sitemap* berdasarkan kategori *user*, yaitu, *sitemap user* belum masuk, *sitemap user* sudah masuk, dan *sitemap user* admin.

3.7.1 Sitemap User Belum Masuk

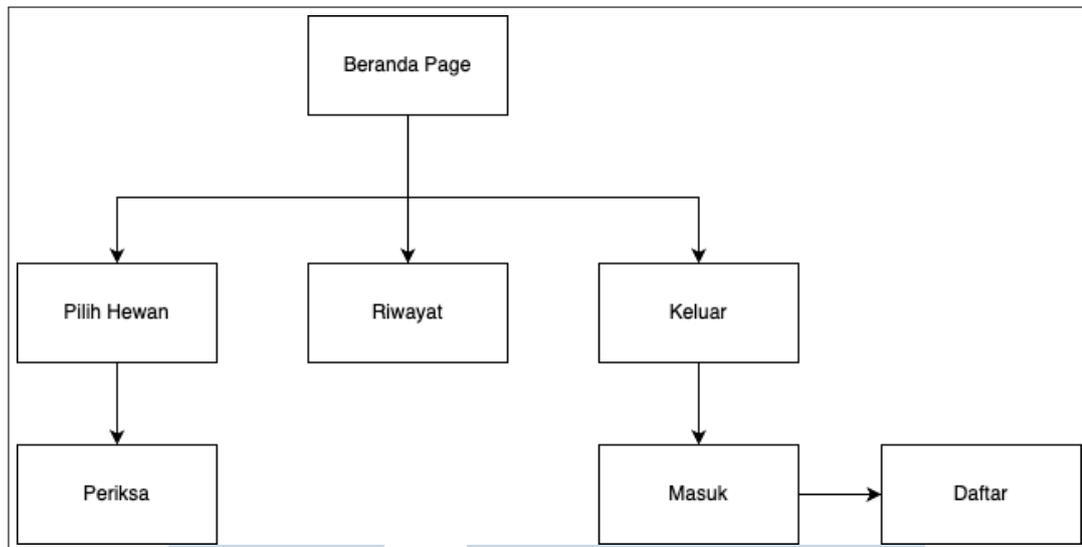
Pada Gambar 3.1 terlihat bahwa hanya terdapat lima halaman yang bisa diakses untuk user yang belum login. Yaitu halaman beranda, halaman pilih hewan, halaman periksa, halaman masuk, dan halaman daftar. Halaman beranda adalah halaman pertama yang diakses. Pada halaman beranda pengguna bisa berpindah ke halaman pilih hewan atau halaman masuk. Pada halaman pilih hewan pengguna diharuskan untuk memilih hewan ternak apa yang akan didiagnosis penyakitnya. Pada penelitian ini hanya terdapat dua hewan ternak saja, yaitu, sapi dan kambing. Setelah memilih hewan yang akan didiagnosis, maka pengguna akan masuk ke halaman periksa. Pada halaman periksa pengguna diharuskan untuk memilih minimal satu gejala yang dialami oleh hewan ternaknya, seperti penyakit bulu rontok pada sapi atau penyakit keguguran pada kambing. Setelah memilih gejala makan akan ditampilkan hasil diagnosisnya. Halaman lainnya adalah halaman masuk, pada halaman ini pengguna diharuskan untuk mempunyai sebuah akun terlebih dahulu, jika tidak mempunyai akun, maka akan diarahkan ke halaman daftar.



Gambar 3.1. Sitemap User Belum Masuk

3.7.2 Sitemap User Sudah Masuk

Pada Gambar 3.2 terlihat bahwa terdapat tujuh halaman yang dapat diakses untuk user yang sudah masuk. Yaitu halaman beranda, halaman pilih hewan, halaman periksa, halaman riwayat, halaman keluar, halaman masuk, dan halaman daftar. Halaman beranda adalah halaman yang diakses setelah pengguna masuk. Pada halaman beranda pengguna bisa berpindah ke halaman pilih hewan, halaman riwayat, dan halaman masuk. Pada halaman pilih hewan pengguna diharuskan untuk memilih hewan ternak apa yang akan didiagnosis penyakitnya. Pada penelitian ini hanya terdapat dua hewan ternak saja, yaitu sapi dan kambing. Setelah memilih hewan yang akan didiagnosis, maka pengguna akan masuk ke halaman periksa. Pada halaman periksa pengguna diharuskan untuk memilih minimal satu gejala yang dialami oleh hewan ternaknya, seperti penyakit bulu rontok pada sapi atau penyakit keguguran pada kambing. Setelah memilih gejala maka akan ditampilkan hasil diagnosisnya. Sedangkan pada halaman riwayat dapat menampilkan riwayat dari diagnosis yang telah dilakukan sebelumnya. Halaman lainnya adalah halaman keluar, pada halaman ini pengguna akan dipindahkan ke halaman masuk, serta menjadi kategori user belum masuk.

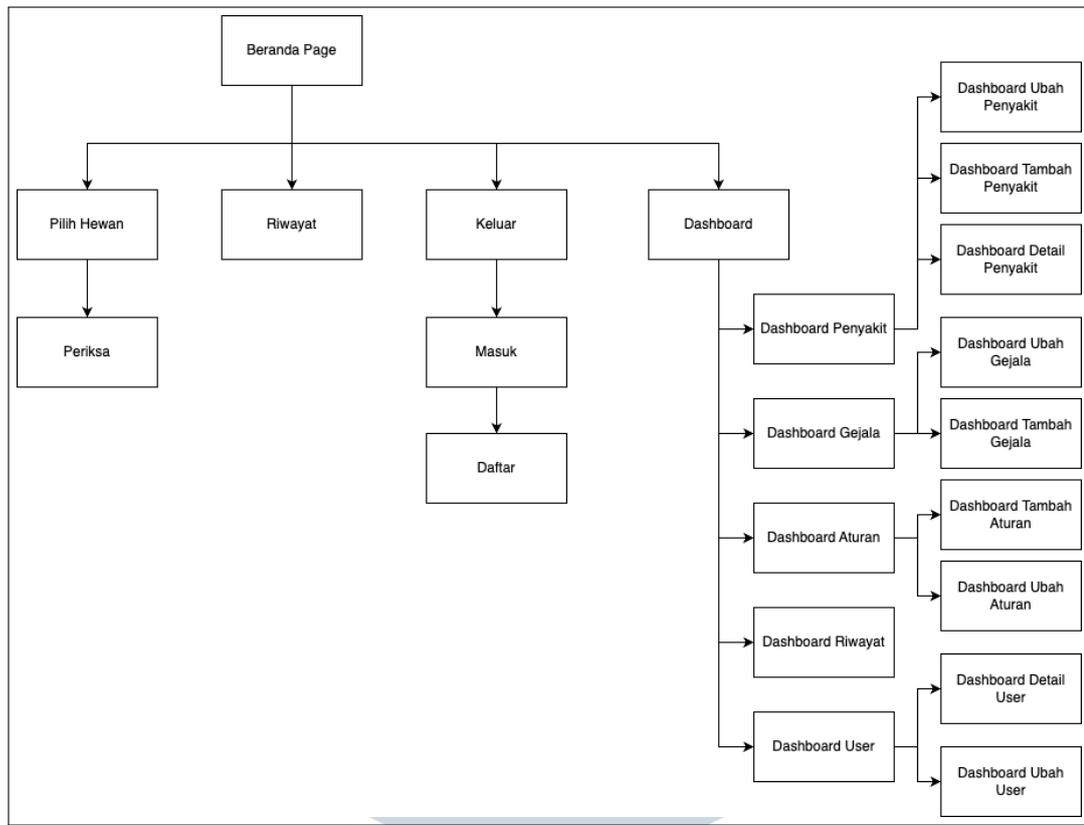


Gambar 3.2. Sitemap User Sudah Masuk

3.7.3 Sitemap User Admin

Pada Gambar 3.3 terlihat bahwa ada lebih banyak halaman dibandingkan dengan kategori user sudah masuk. Yaitu halaman *dashboard*, halaman *dashboard* penyakit, halaman *dashboard* ubah penyakit, halaman *dashboard* tambah penyakit, halaman *dashboard* detail penyakit, halaman *dashboard* gejala, halaman *dashboard* ubah gejala, halaman *dashboard* tambah gejala, halaman *dashboard* aturan, halaman *dashboard* tambah aturan, halaman *dashboard* ubah aturan, halaman *dashboard* hapus aturan, halaman *dashboard* riwayat, halaman *dashboard* user, halaman *dashboard* detail user, halaman *dashboard* ubah user, dan halaman *dashboard* hapus user. Pada kategori user admin dikhususkan untuk akun pakar dan akun pengembang. Pada halaman *dashboard* tersedia beberapa fitur yang dibuat untuk memudahkan memasukkan data ke dalam aplikasi sistem pakar.

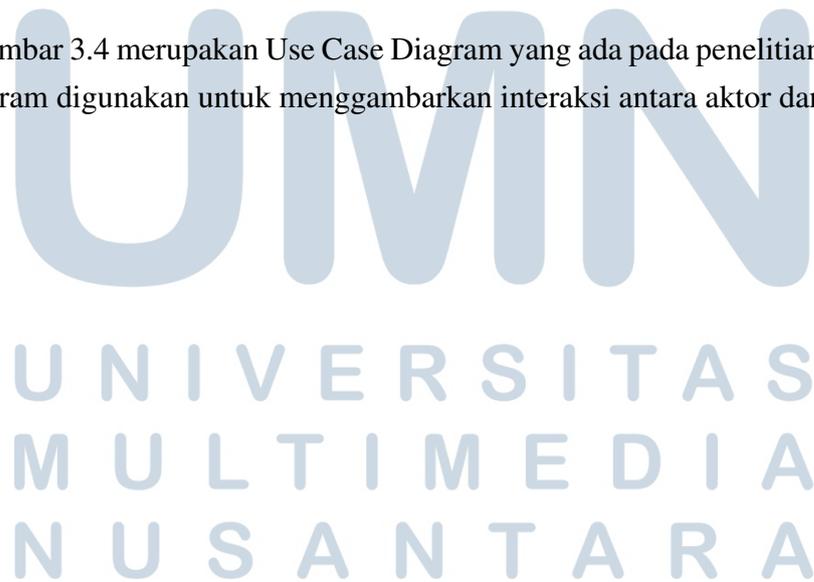
UNIVERSITAS
MULTIMEDIA
NUSANTARA

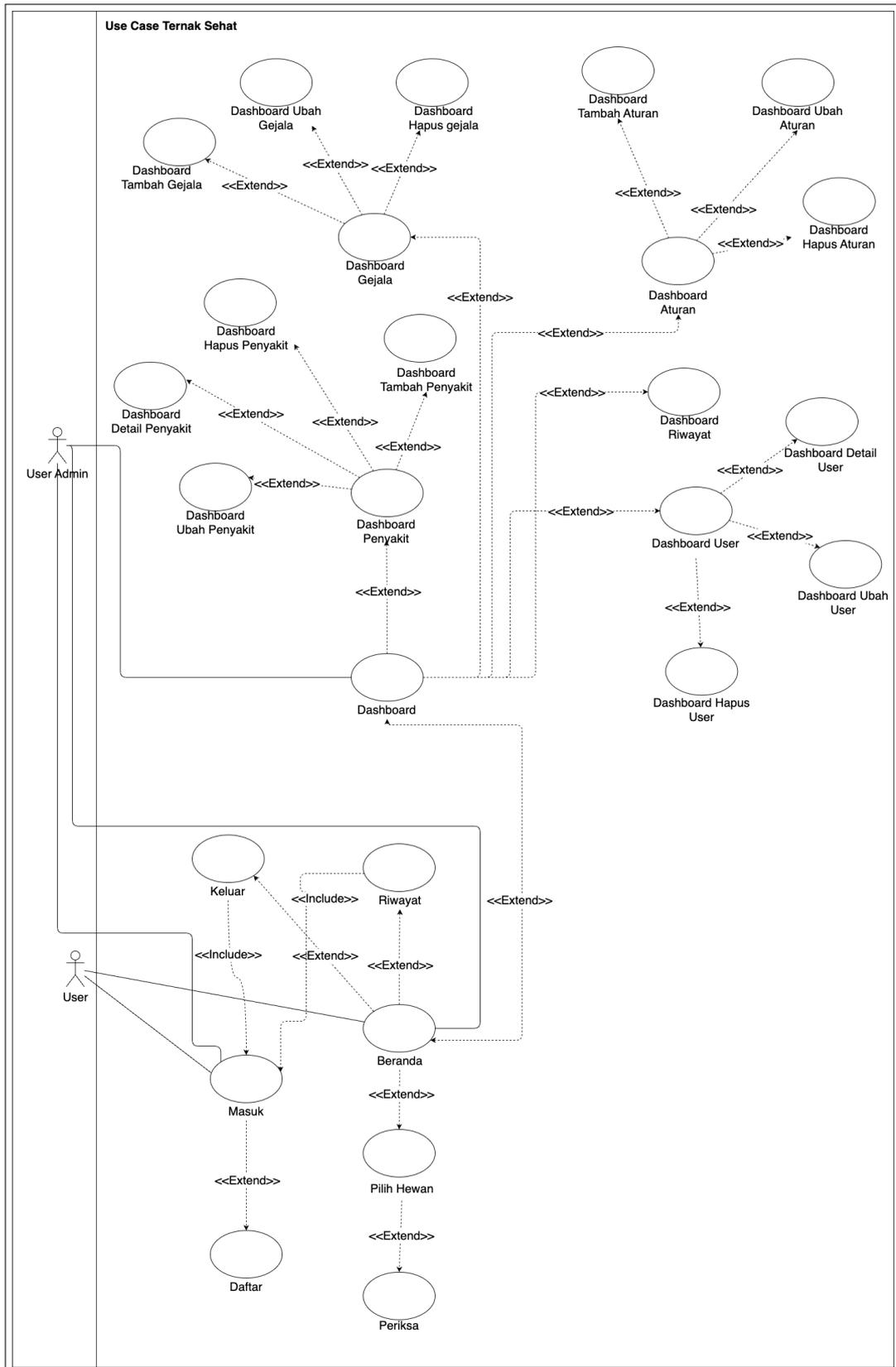


Gambar 3.3. Sitemap User Admin

3.8 Use Case Diagram

Gambar 3.4 merupakan Use Case Diagram yang ada pada penelitian ini. Use Case Diagram digunakan untuk menggambarkan interaksi antara aktor dan sistem.





Gambar 3.4. Use Case Diagram

Pada Use Case Diagram terdapat dua aktor, yaitu user admin, dan user. User berperan sebagai kategori user belum masuk dan user sudah masuk. Aktor user dapat mengakses halaman beranda atau ke halaman masuk. Jika user mengakses halaman beranda terlebih dahulu, maka user dapat dikategorikan sebagai user belum masuk dan user tidak dapat mengakses halaman riwayat, namun user tetap bisa melakukan diagnosis penyakit pada halaman pilih hewan dan halaman periksa. Sedangkan user yang mengakses halaman masuk lebih dahulu dan memasukkan akun dengan *role* user yang telah dibuat pada halaman daftar, maka akan dikategorikan user sudah masuk dan user dapat mengakses halaman riwayat. Jika user memasukkan akun yang memiliki *role* admin, maka user dapat mengakses halaman riwayat dan halaman *dashboard*.

Pada aktor user admin, user admin bisa mengakses semua halaman yang aktor user punya. Jika user sudah memasukkan akun pada halaman masuk, user dapat mengakses halaman khusus, yaitu:

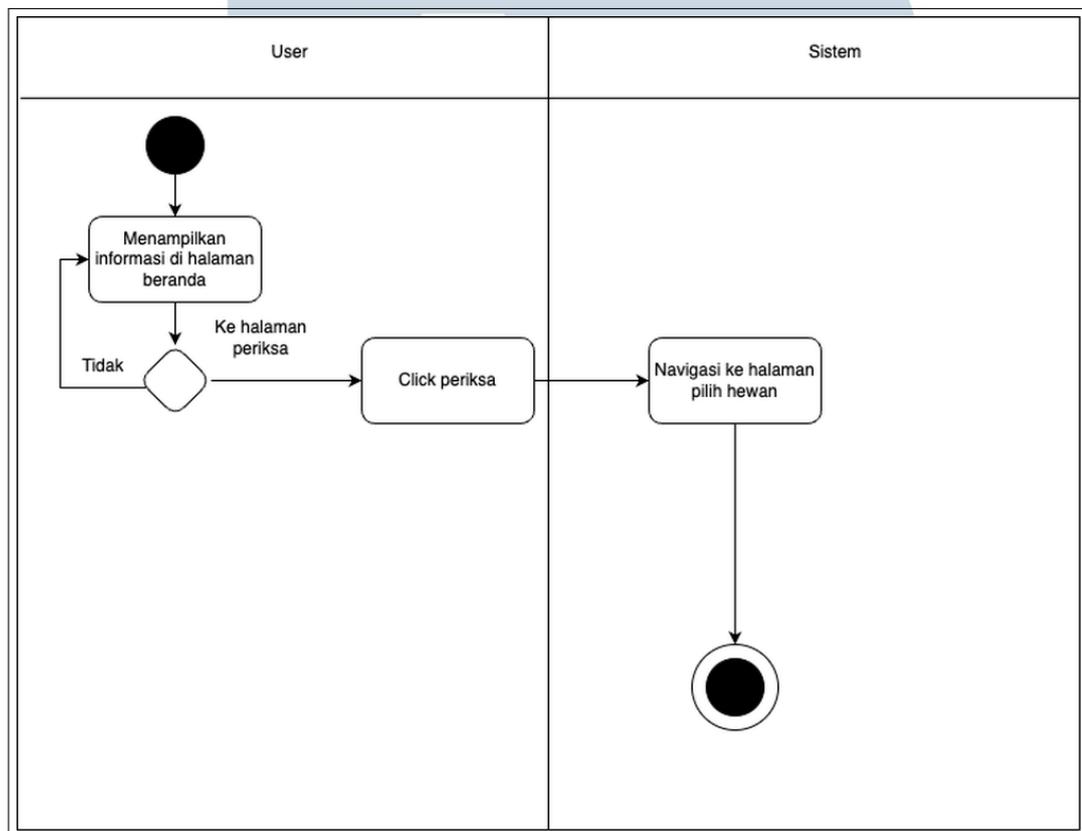
1. Dashboard, bertujuan untuk melihat banyaknya penyakit, gejala, riwayat, dan pengguna yang terdaftar pada website.
2. Dashboard Penyakit, bertujuan untuk menambah penyakit, mengubah penyakit, melihat detail penyakit, dan menghapus penyakit.
3. Dashboard Gejala, bertujuan untuk menambah gejala, mengubah gejala, dan menghapus gejala.
4. Dashboard Aturan, bertujuan untuk menambah aturan, mengubah aturan, dan menghapus aturan.
5. Dashboard riwayat.
6. Dashboard User, bertujuan untuk melihat detail user, mengubah user, dan menghapus user.

3.9 Activity Diagram

Pada bagian ini dijabarkan tentang Activity Diagram yang ada pada penelitian ini. Activity Diagram digunakan untuk menggambarkan alur kerja sistem pada setiap halaman.

3.9.1 Beranda

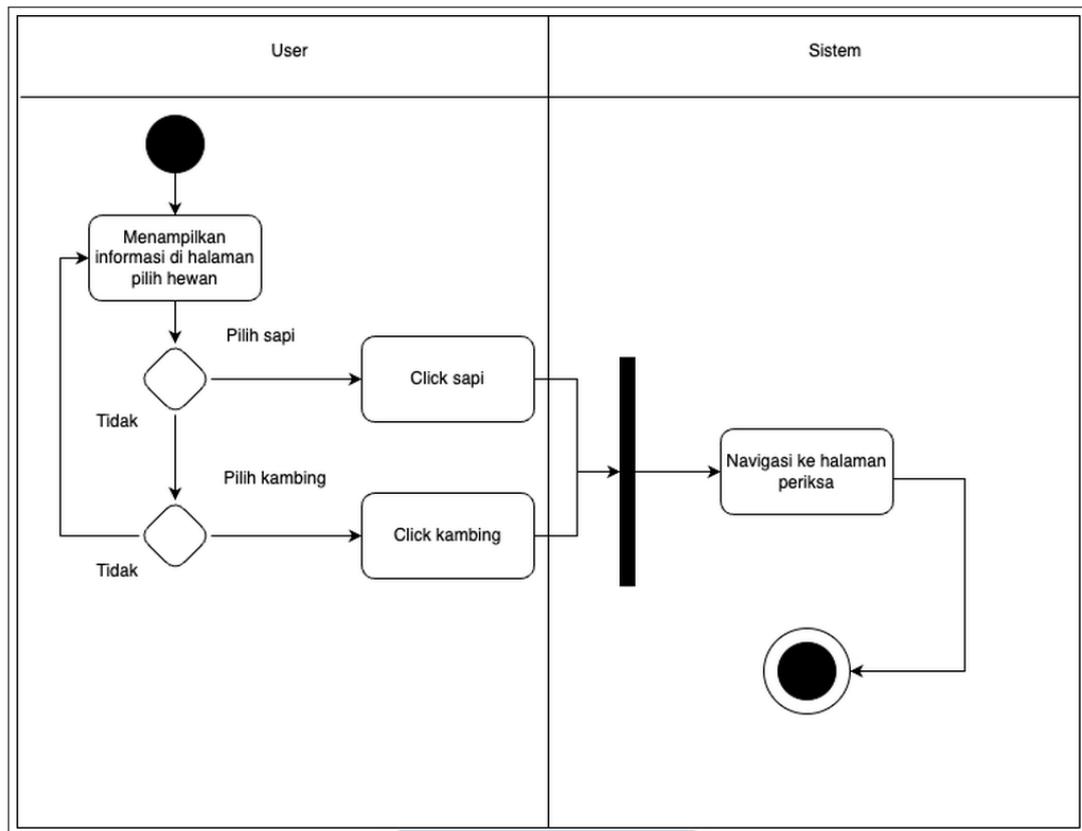
Pada Gambar 3.5 merupakan alur kerja dari halaman beranda. Saat pengguna mengakses halaman beranda, maka akan ditampilkan sebuah informasi mengenai sistem pakar hewan ternak. Pengguna dapat melakukan diagnosis pada hewan ternaknya dengan menekan tombol periksa dan akan diarahkan ke halaman pilih hewan.



Gambar 3.5. Activity Diagram Beranda

3.9.2 Pilih Hewan

Pada Gambar 3.6 merupakan alur kerja dari halaman beranda. Pada halaman ini pengguna diharuskan untuk memilih hewan ternak apa yang ingin didiagnosis. Setelah memilih hewan ternak yang akan didiagnosis, maka akan diarahkan ke halaman periksa.

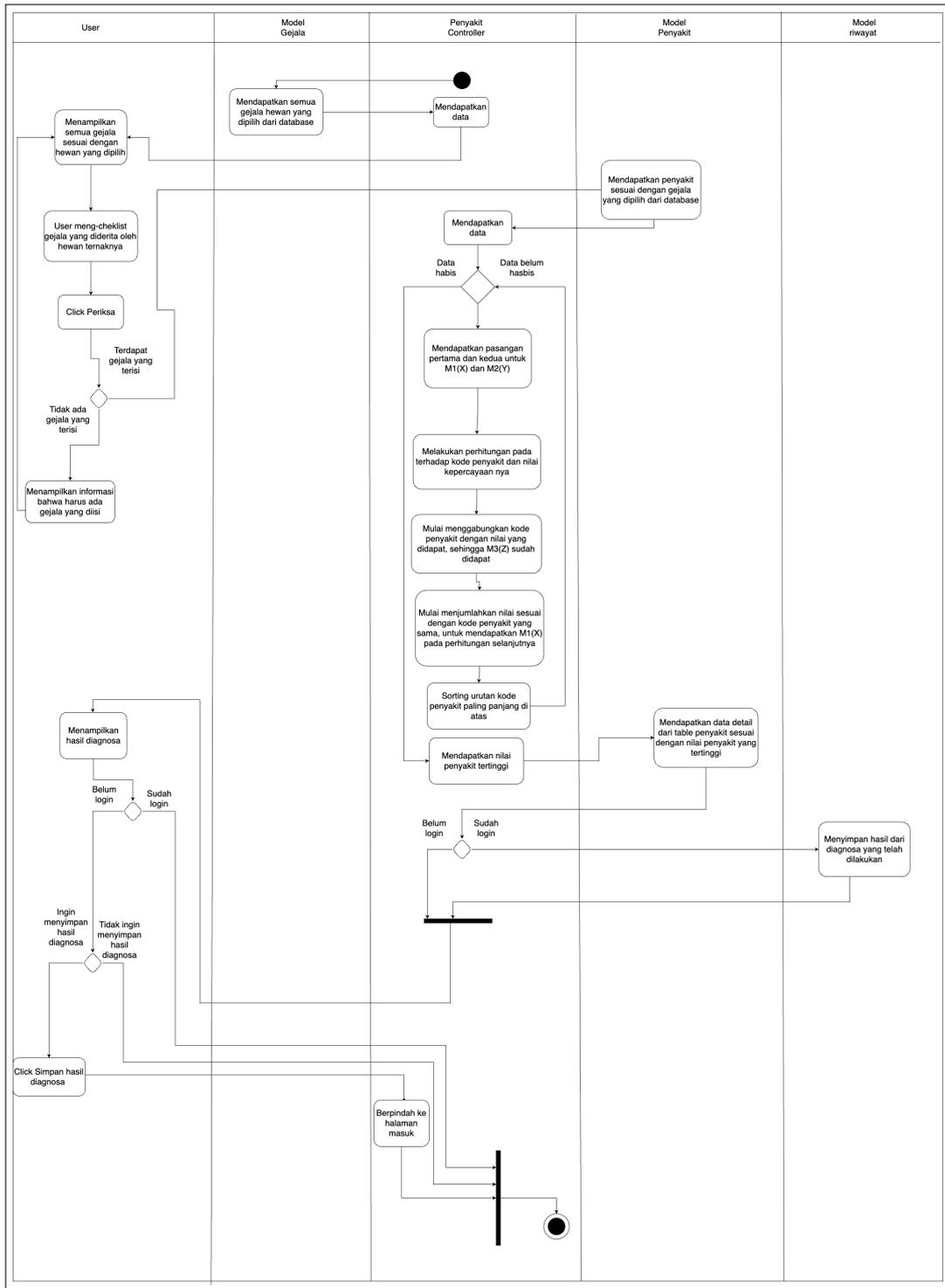


Gambar 3.6. Activity Diagram Pilih Hewan

3.9.3 Periksa

Pada Gambar 3.7 merupakan alur kerja dari halaman periksa. Pada halaman ini pengguna ditampilkan seluruh gejala dari salah satu hewan ternak yang dipilih pada halaman pilih hewan dan diharuskan untuk memilih minimal satu gejala. Setelah memilih gejala, maka akan dilakukan perhitungan sistem pakar dengan metode Dempster-Shafer. Setelah perhitungan selesai akan ada pengecekan apakah pengguna sudah masuk atau belum. Jika pengguna sudah masuk, maka hasil diagnosis akan disimpan pada halaman riwayat. Jika pengguna belum masuk, maka hasil diagnosis tidak disimpan dan pada halaman ini dimunculkan tombol simpan hasil diagnosis, yang mana jika ditekan akan diarahkan ke halaman masuk.

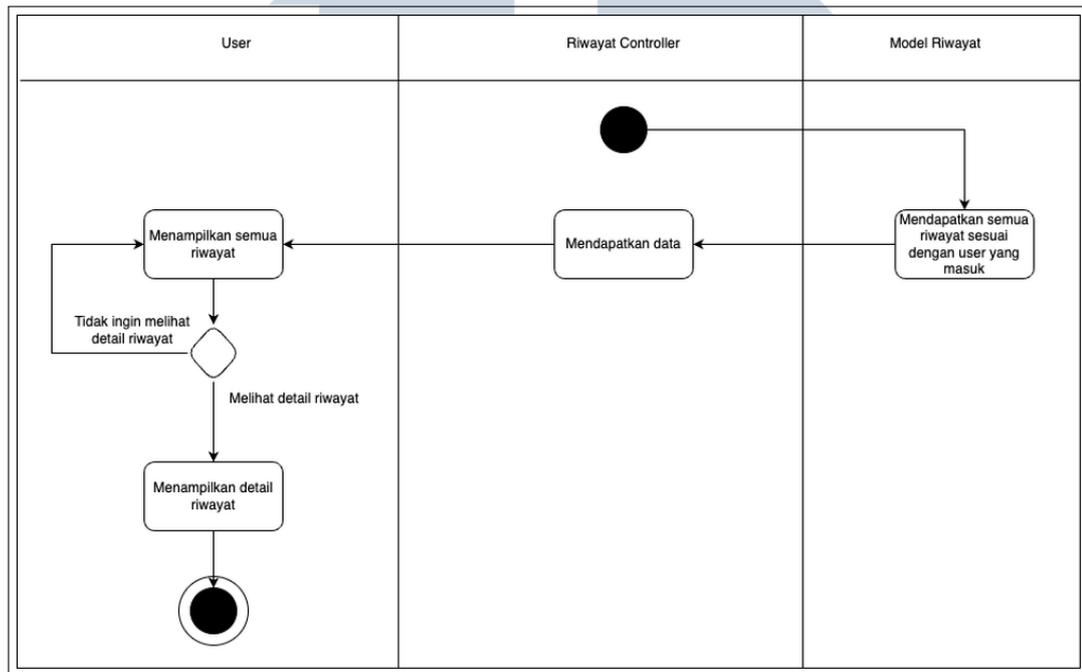
Pada halaman ini setelah melakukan pengecekan pengguna sudah masuk atau belum, akan menampilkan hasil diagnosis penyakitnya. Ditampilkannya juga nama penyakit, penjelasan penyakit, solusi atau pencegahan yang dapat dilakukan.



N U S A N T A R A
Gambar 3.7. Activity Diagram Periksa

3.9.4 Riwayat

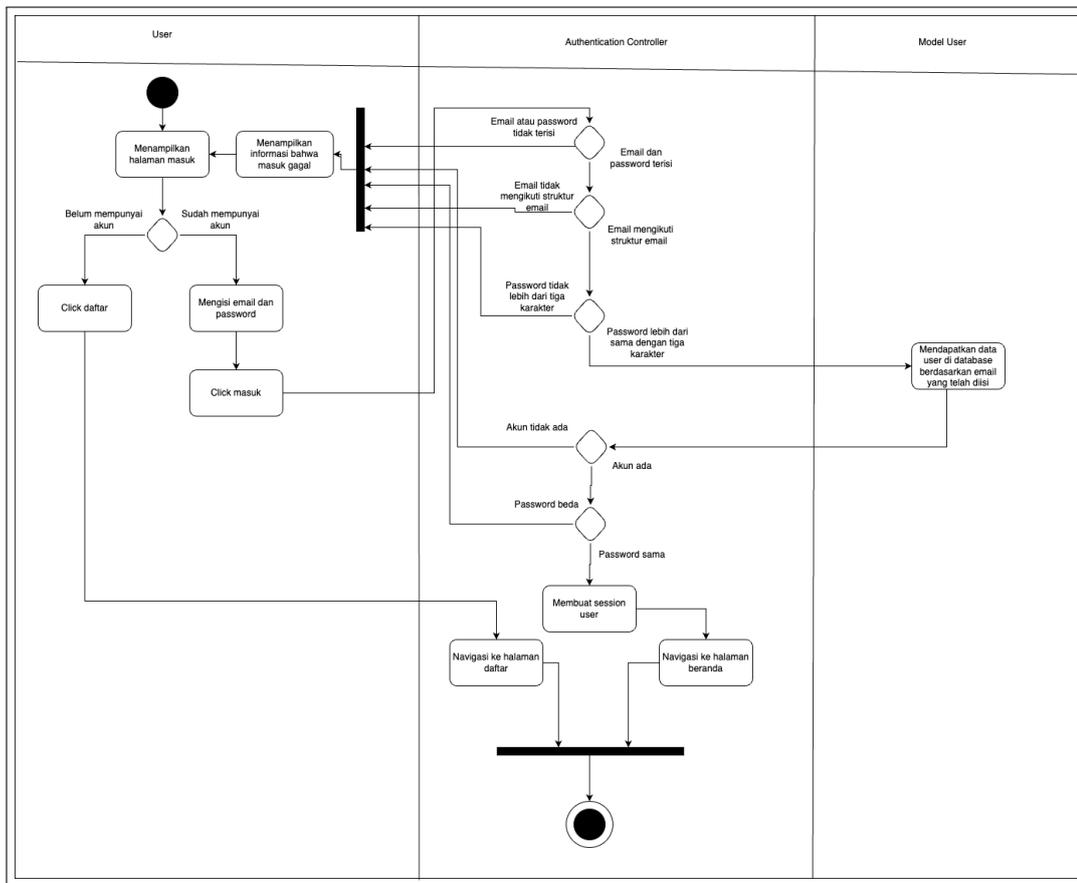
Pada Gambar 3.8 merupakan alur kerja dari halaman riwayat. Pada halaman ini pengguna yang sudah masuk dapat melihat hasil diagnosis terdahulu.



Gambar 3.8. Activity Diagram Riwayat

3.9.5 Masuk

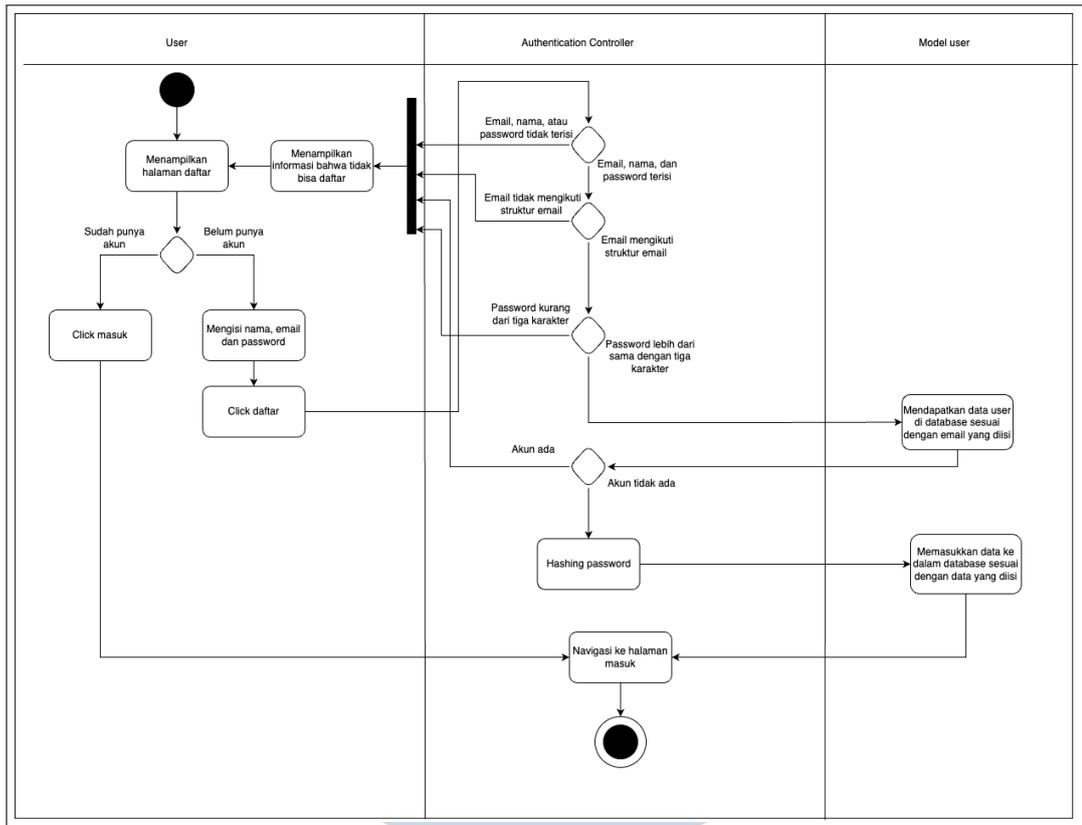
Pada Gambar 3.9 merupakan alur kerja dari halaman masuk. Pada halaman ini pengguna diharuskan untuk mengisi email dan password. Jika pengguna tidak mempunyai akun dapat menekan tombol daftar untuk mendaftar di halaman daftar. Setelah pengguna mengisi email dan password maka akan ada beberapa pengecekan, yaitu, pengecekan apakah email dan password sudah terisi, pengecekan apakah email mengikuti struktur email, pengecekan apakah email yang terdaftar ada pada database, dan pengecekan apakah password yang diisi dan password yang terdaftar sama. Setelah dilakukan semua pengecekan, jika ada yang tidak sesuai, maka akan ada informasi bahwa ada yang salah. Jika dari pengecekan sudah sesuai semua, maka akan dibuatkan session user, lalu berpindah ke halaman beranda.



Gambar 3.9. Activity Diagram Masuk

3.9.6 Daftar

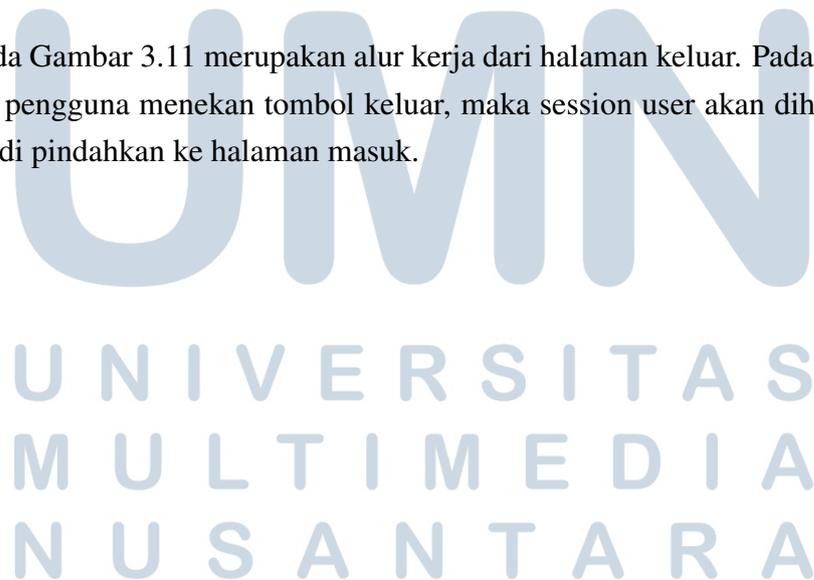
Pada Gambar 3.10 terlihat bahwa mengunjungi halaman, pengguna disuguhkan form yang dapat diisi jika ingin daftar dan belum mempunyai akun, pengguna dapat menuliskan nama, email dan password yang ingin didaftarkan, setelah menekan tombol daftar, maka akan ada beberapa validasi seperti, nama, email atau password sudah terisi apa belum, email sudah sesuai dengan struktur email atau belum dan akun yang didaftarkan sudah ada atau belum, jika dirasa sudah sesuai semua, maka akan password akan di *hashing* lalu data akan dimasukkan ke database, selanjutnya pengguna langsung berpindah ke halaman masuk. Jika pengguna sudah memiliki akun, maka dapat langsung masuk dengan menekan tombol masuk dan akan langsung menampilkan halaman masuk.

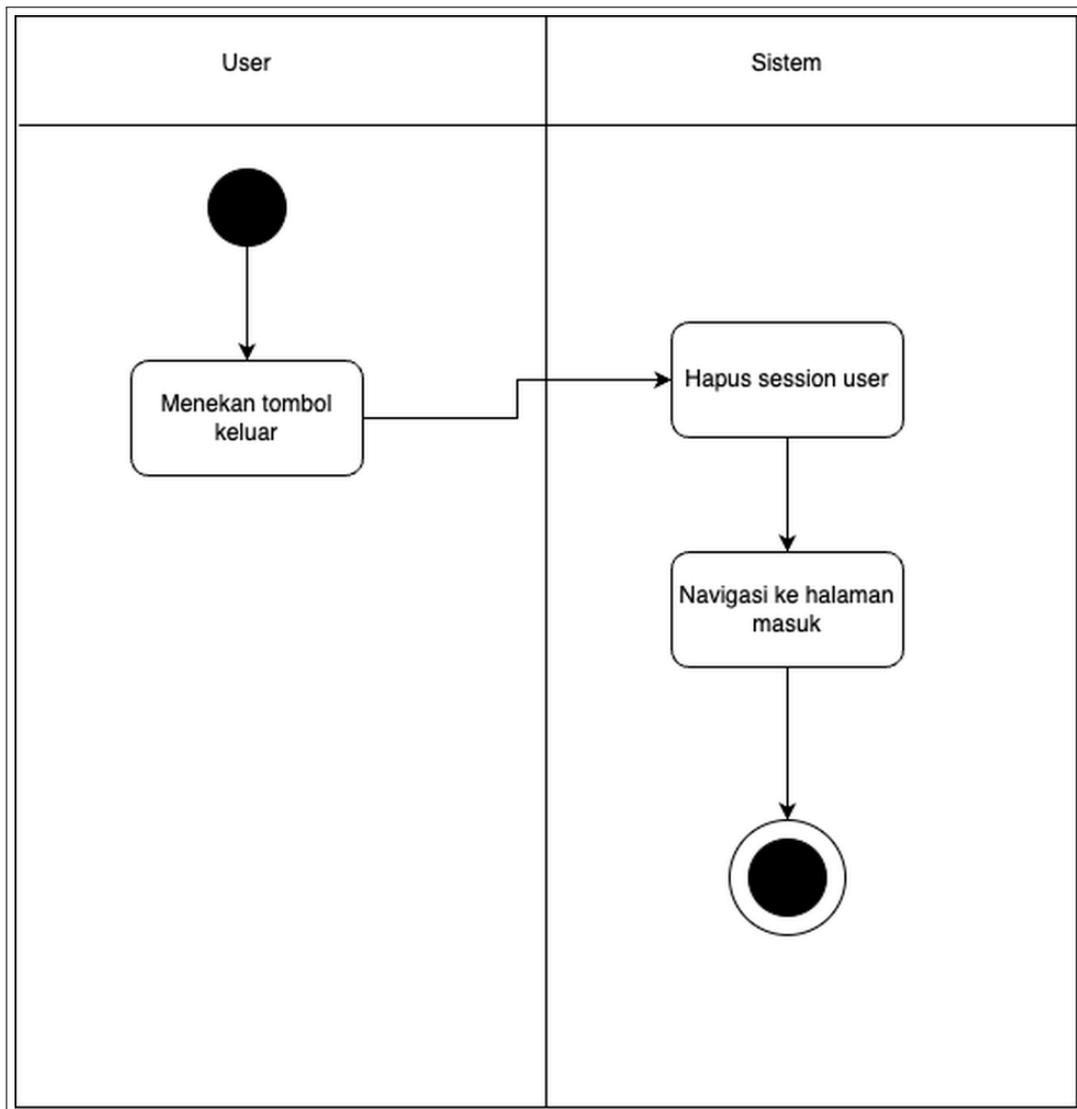


Gambar 3.10. Activity Diagram Daftar

3.9.7 Keluar

Pada Gambar 3.11 merupakan alur kerja dari halaman keluar. Pada halaman ini, ketika pengguna menekan tombol keluar, maka session user akan dihapus dan pengguna di pindahkan ke halaman masuk.

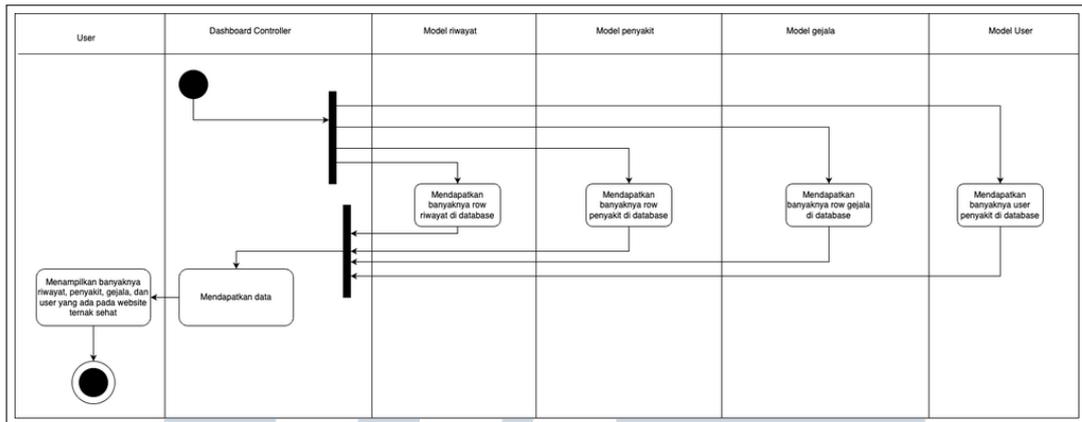




Gambar 3.11. Activity Diagram Keluar

3.9.8 Dashboard

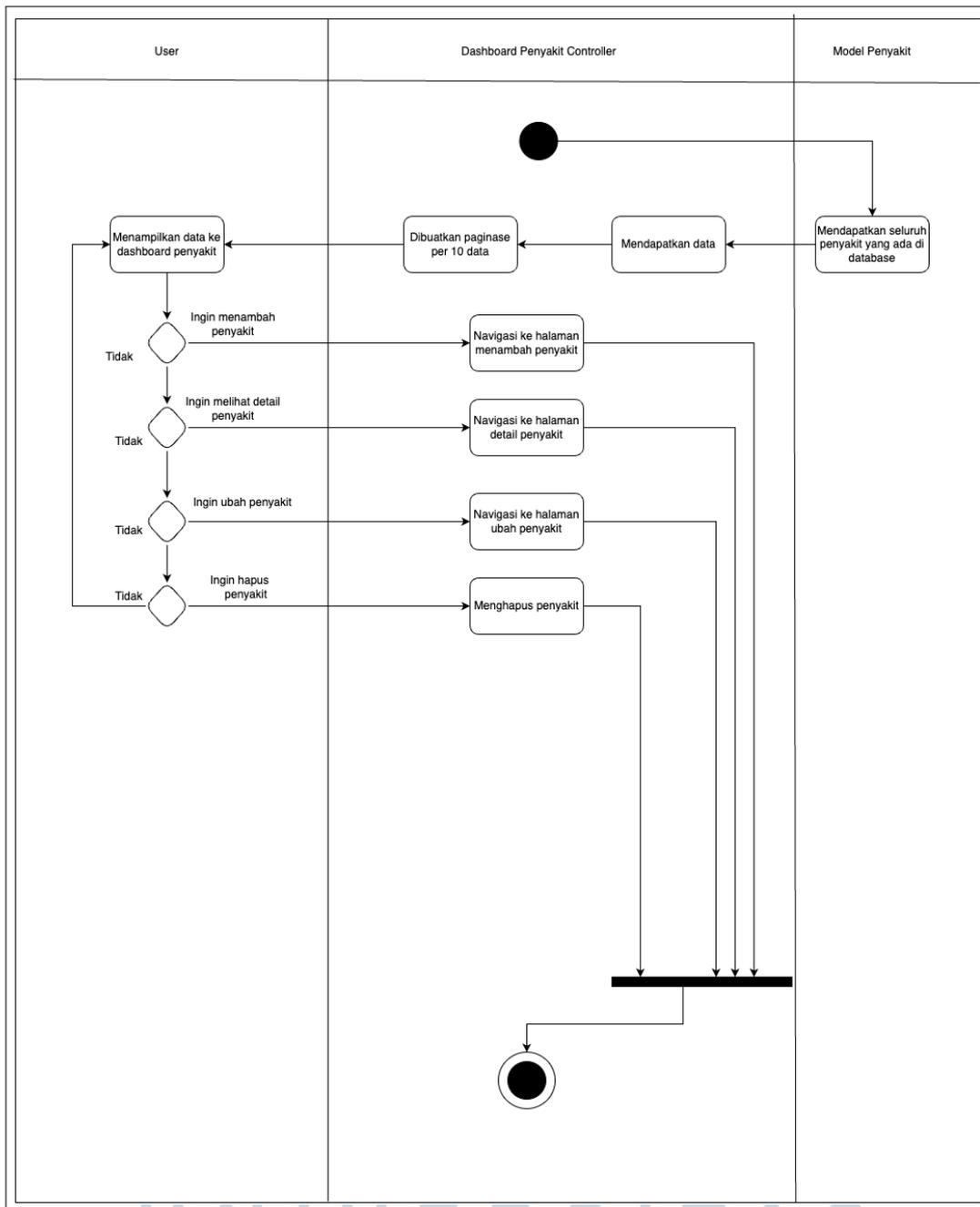
Pada Gambar 3.12 merupakan alur kerja halaman *dashbaord*. Pada halaman ini pengguna diperlihatkan banyaknya data riwayat, data penyakit, data gejala dan data user yang dimiliki oleh aplikasi.



Gambar 3.12. Activity Diagram Dashboard

3.9.9 Dashboard Penyakit

Pada Gambar 3.13 merupakan alur kerja dari halaman *dashboard* penyakit. Pada halaman ini pengguna ditampilkan seluruh penyakit dari hewan sapi dan kambing. Pengguna bisa menambah penyakit pada sapi atau kambing dengan menekan tombol tambah penyakit, lalu pengguna akan dipindahkan ke halaman dashboard tambah penyakit. Pengguna dapat mengubah isi penyakit dari penyakit yang sudah ada dengan menekan tombol ubah, lalu pengguna akan dipindahkan ke halaman dashboard ubah penyakit. Pengguna dapat melihat detail dari penyakit dengan cara menekan tombol detail, lalu pengguna akan dipindahkan ke halaman dashboard detail penyakit. Pengguna juga dapat untuk menghapus penyakit.

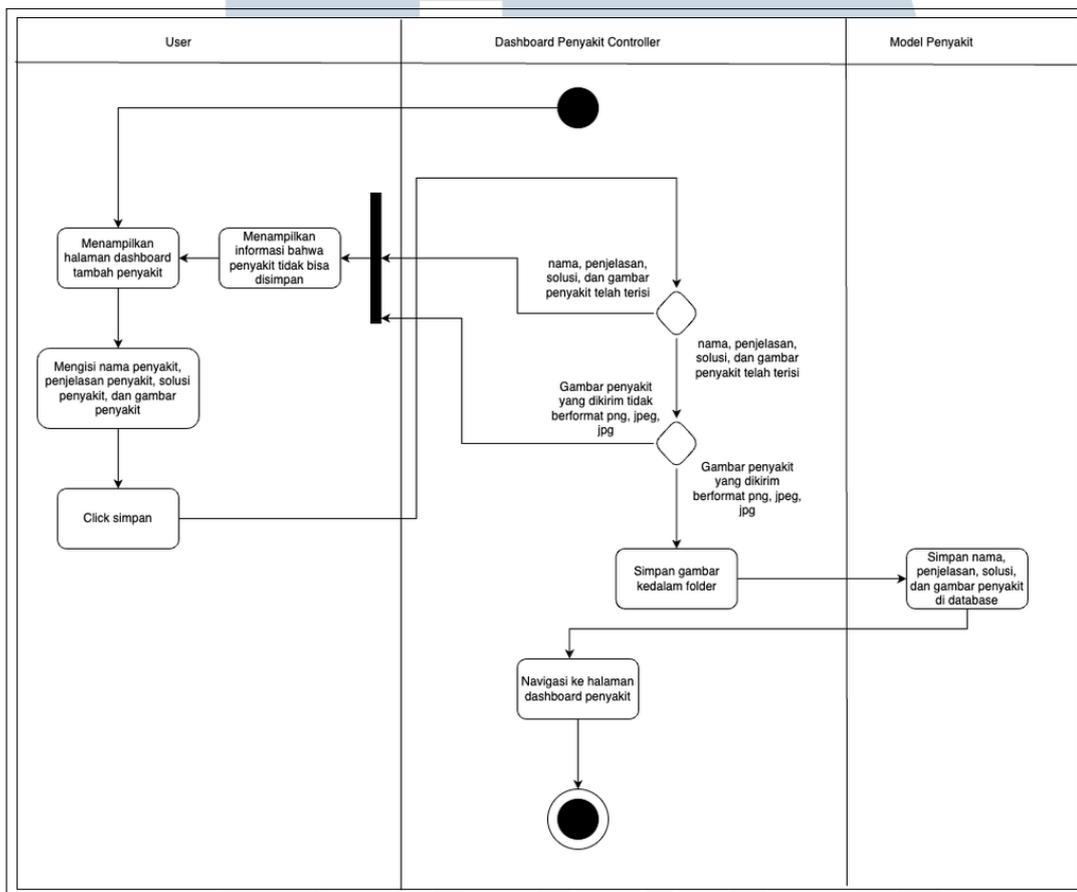


Gambar 3.13. Activity Diagram Dashboard Penyakit

3.9.10 Dashboard Tambah penyakit

Pada Gambar 3.14 merupakan alur kerja dari halaman *dashboard* tambah penyakit. Pada halaman ini pengguna bisa menambahkan penyakit dengan mengisi nama penyakit, penjelasan penyakit, solusi penyakit, dan gambar penyakit. Setelah semua terisi, maka akan ada beberapa pengecekan, yaitu, pengecekan apakah nama

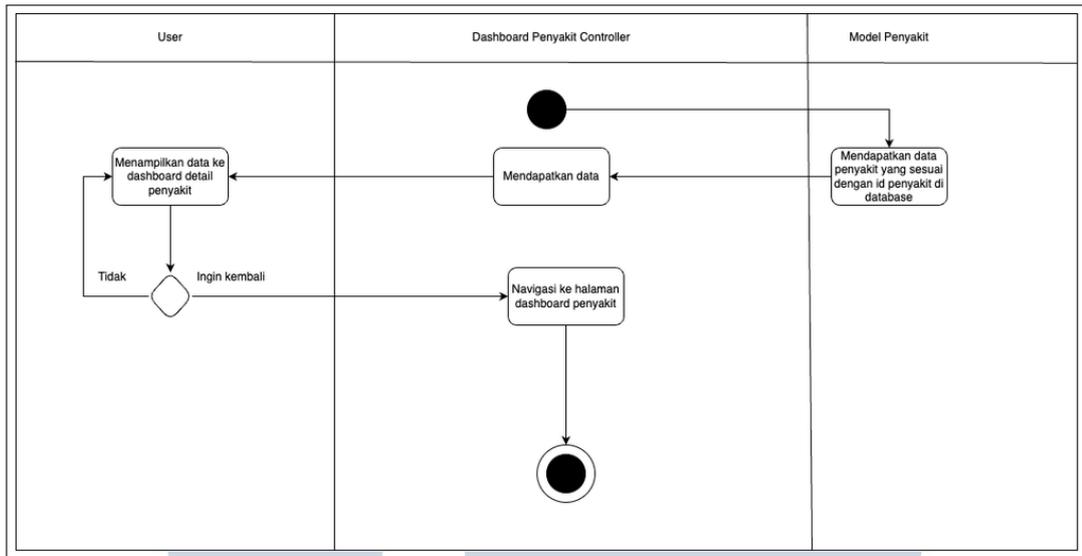
penyakit, penjelasan penyakit, solusi penyakit, dan gambar penyakit sudah terisi dan pengecekan apakah gambar penyakit berformat png, jpeg, atau jpg. Jika pengecekan ada yang tidak sesuai, maka akan menampilkan informasi bahwa penyakit tidak bisa disimpan. Setelah dilakukannya pengecekan, maka nama penyakit, penjelasan penyakit, solusi penyakit, dan gambar penyakit akan disimpan pada database. Setelah data disimpan pada database, pengguna dipindahkan ke halaman *dashboard* penyakit.



Gambar 3.14. Activity Diagram Dashboard Tambah Penyakit

3.9.11 Dashboard Detail Penyakit

Pada Gambar 3.15 merupakan alur kerja dari halaman *dashboard* detail penyakit. Pada halaman ini menampilkan data detail penyakit, sesuai dengan penyakit yang dipilih.

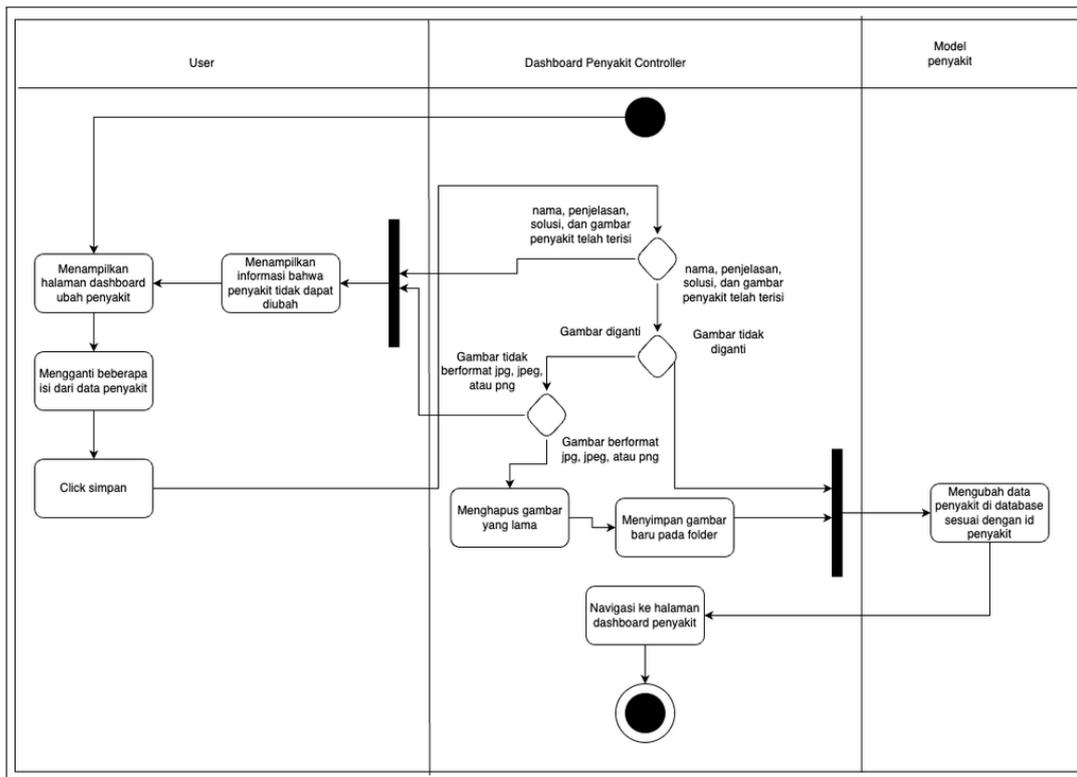


Gambar 3.15. Activity Diagram Dashboard Detail Penyakit

3.9.12 Dashboard Ubah Penyakit

Pada Gambar 3.16 merupakan alur kerja dari halaman *dashboard* ubah penyakit. Pada halaman ini pengguna dapat mengubah detail dari penyakit. Pengguna dapat merubah nama penyakit, penjelasan penyakit, gejala penyakit, dan gambar penyakit. Setelah mengisi data baru, maka akan ada pengecekan, yaitu, pengecekan apakah nama penyakit, penjelasan penyakit, gejala penyakit, atau gambar penyakit sudah terisi dan pengecekan apakah gambar berformat jpg, jpeg, png. Jika pengecekan ada yang tidak sesuai, maka akan menampilkan informasi bahwa penyakit tidak dapat diubah. Setelah melewati pengecekan, maka data baru akan mengganti data lama di database. Setelah menyimpan data baru di database pengguna dipindahkan ke halaman *dashboard* penyakit.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

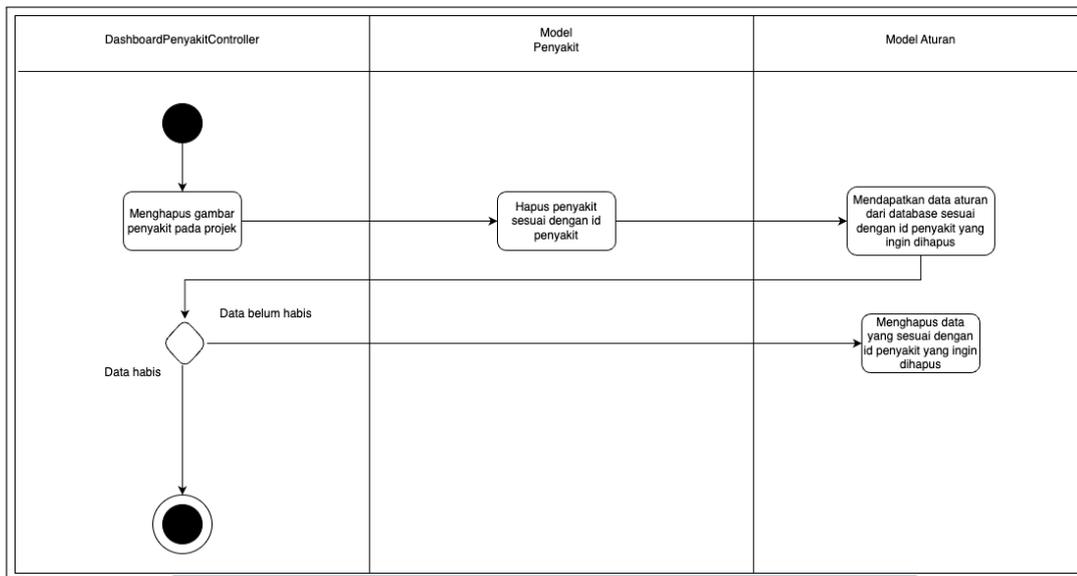


Gambar 3.16. Activity Diagram Dashboard Ubah Penyakit

3.9.13 Dashboard Hapus Penyakit

Pada Gambar 3.17 merupakan alur kerja dari fungsi *dashboard* hapus penyakit. Pada fungsi ini yang dilakukan pertama kali adalah penghapusan gambar penyakit pada projek. Selanjutnya menghapus penyakit yang ada pada database. Setelah menghapus penyakit yang ada pada database, maka didapatkanlah data aturan yang memakai penyakit yang ingin dihapus. Setelah mendapatkan data aturan, maka dihapus data tersebut satu persatu, hingga data habis.

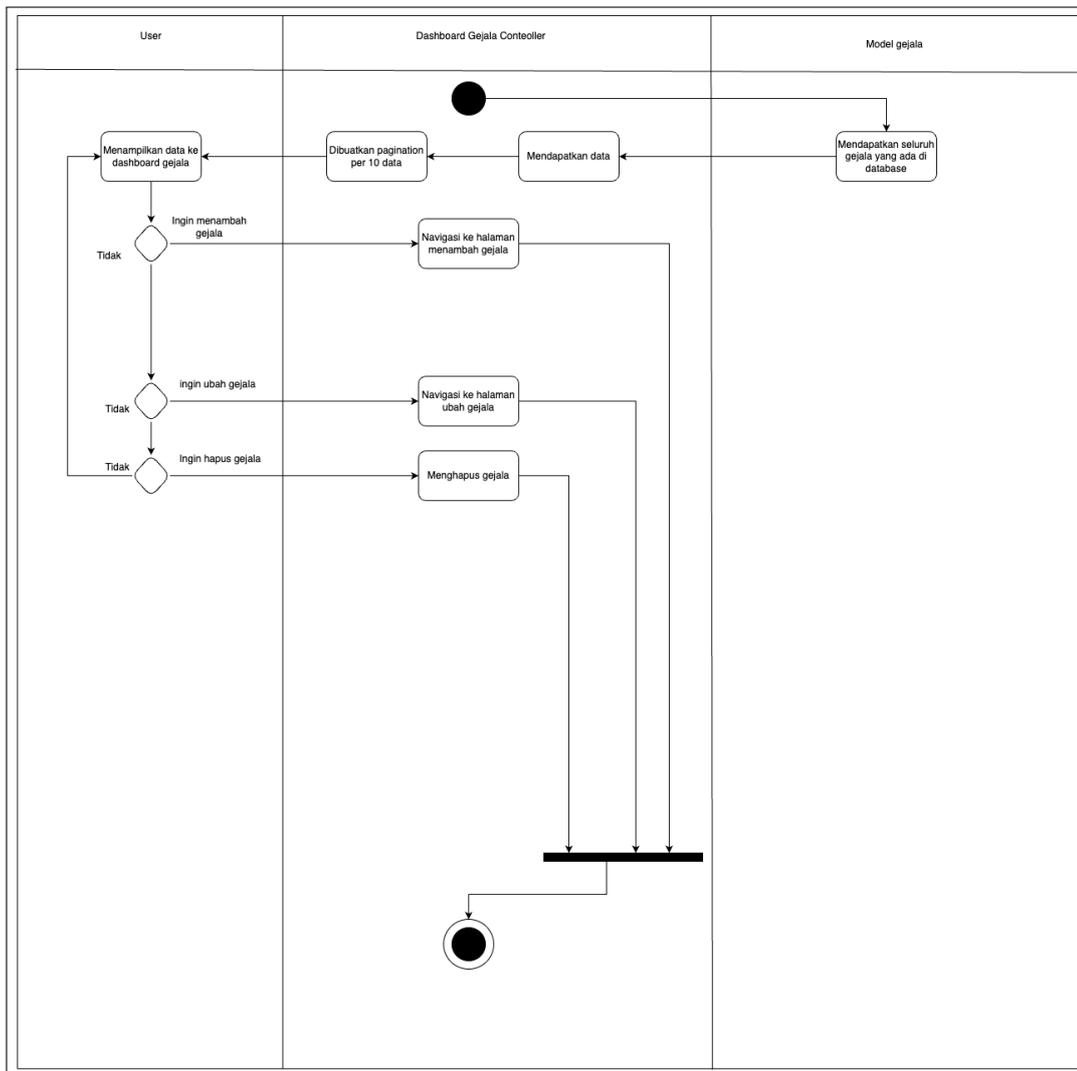
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.17. Activity Diagram Dashboard Hapus Penyakit

3.9.14 Dashboard Gejala

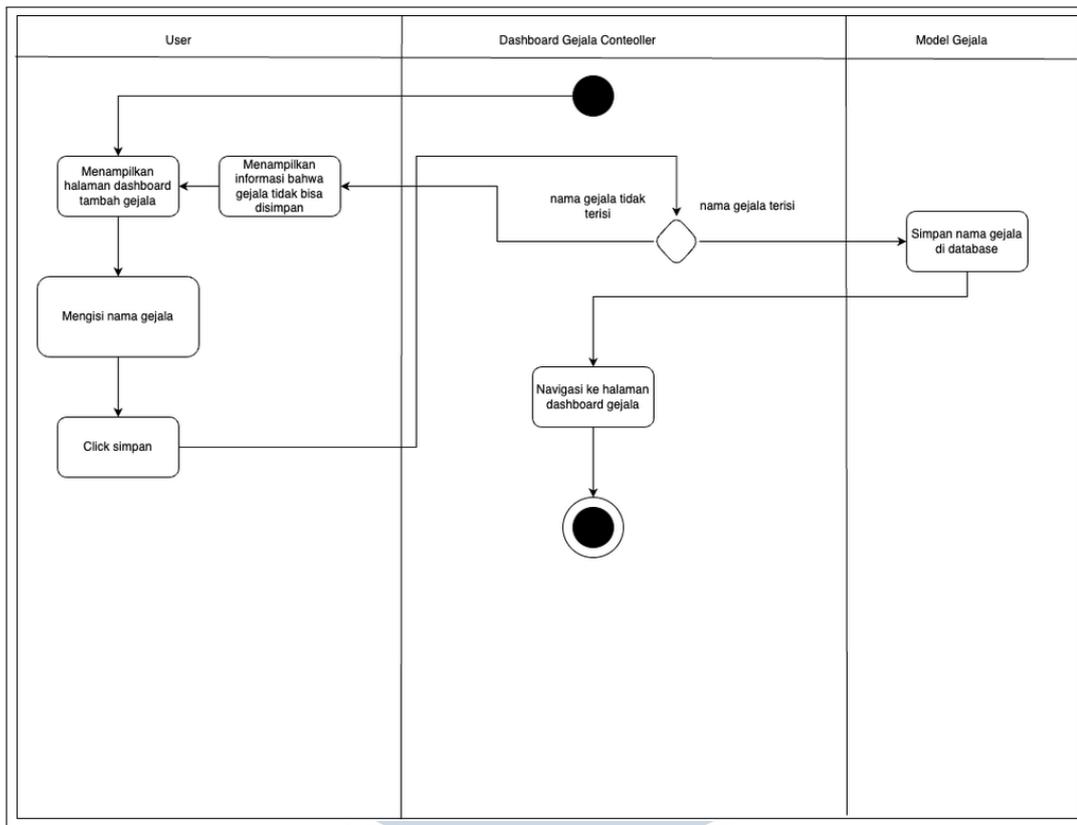
Pada Gambar 3.18 merupakan alur kerja dari halaman *dashboard* gejala. Pada halaman ini pengguna ditampilkan seluruh gejala dari hewan sapi dan kambing. Pengguna bisa menambah gejala pada sapi atau kambing dengan menekan tombol tambah gejala, lalu pengguna akan dipindahkan ke halaman dashboard tambah gejala. Pengguna dapat mengubah isi gejala dari gejala yang sudah ada dengan menekan tombol ubah, lalu pengguna akan dipindahkan ke halaman dashboard ubah gejala. Pengguna juga bisa untuk menghapus gejala.



Gambar 3.18. Activity Diagram Dashboard Gejala

3.9.15 Dashboard Tambah Gejala

Pada Gambar 3.19 merupakan alur kerja dari halaman *dashboard* tambah gejala. Pada halaman ini pengguna bisa menambahkan gejala dengan mengisi nama gejala. Setelah nama gejala terisi, maka akan ada pengecekan, yaitu, pengecekan apakah nama gejala sudah terisi. Jika pengecekan ada yang tidak sesuai, maka akan menampilkan informasi bahwa gejala tidak bisa disimpan. Setelah dilakukannya pengecekan, maka nama gejala akan disimpan pada database. Setelah data disimpan pada database, pengguna dipindahkan ke halaman *dashboard* gejala.

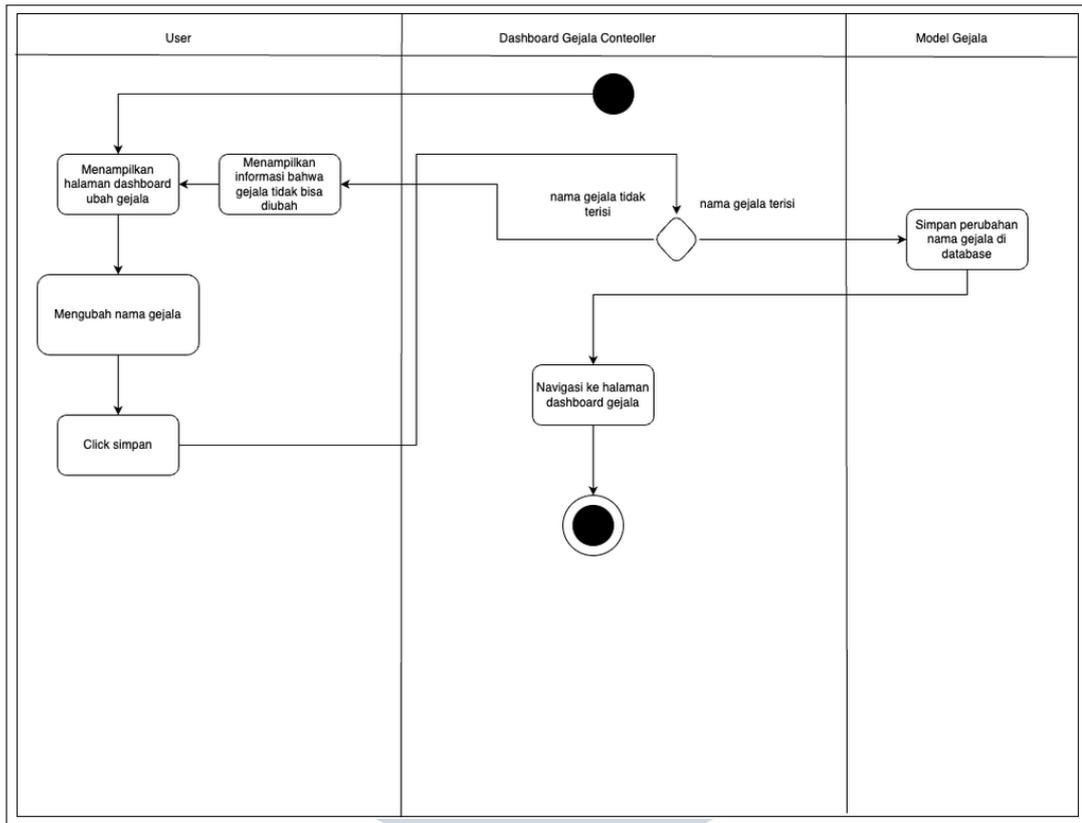


Gambar 3.19. Activity Diagram Dashboard Tambah Gejala

3.9.16 Dashboard Ubah Gejala

Pada Gambar 3.20 merupakan alur kerja dari halaman *dashboard* ubah gejala. Pada halaman ini pengguna dapat mengubah detail dari gejala. Pengguna dapat merubah nama gejala. Setelah mengisi data baru, maka akan ada pengecekan, yaitu, pengecekan apakah nama gejala sudah terisi . Jika pengecekan ada yang tidak sesuai, maka akan menampilkan informasi bahwa gejala tidak dapat diubah. Setelah melewati pengecekan, maka data baru akan mengganti data lama di database. Setelah menyimpan data baru di database pengguna dipindahkan ke halaman *dashboard* gejala.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

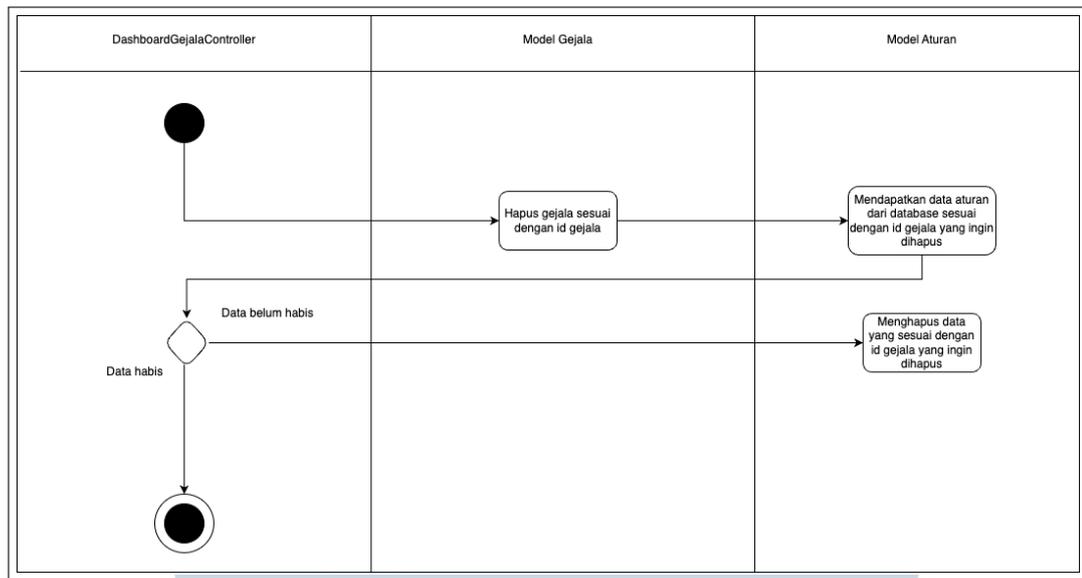


Gambar 3.20. Activity Diagram Dashboard Ubah Gejala

3.9.17 Dashboard Hapus Gejala

Pada Gambar 3.21 merupakan alur kerja dari fungsi *dashboard* hapus gejala. Pada fungsi ini yang dilakukan pertama kali adalah penghapusan gejala yang ada pada database. Setelah menghapus gejala yang ada pada database, maka didapatkanlah data aturan yang memakai gejala yang ingin dihapus. Setelah mendapatkan data aturan, maka dihapus data tersebut satu persatu, hingga data habis.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

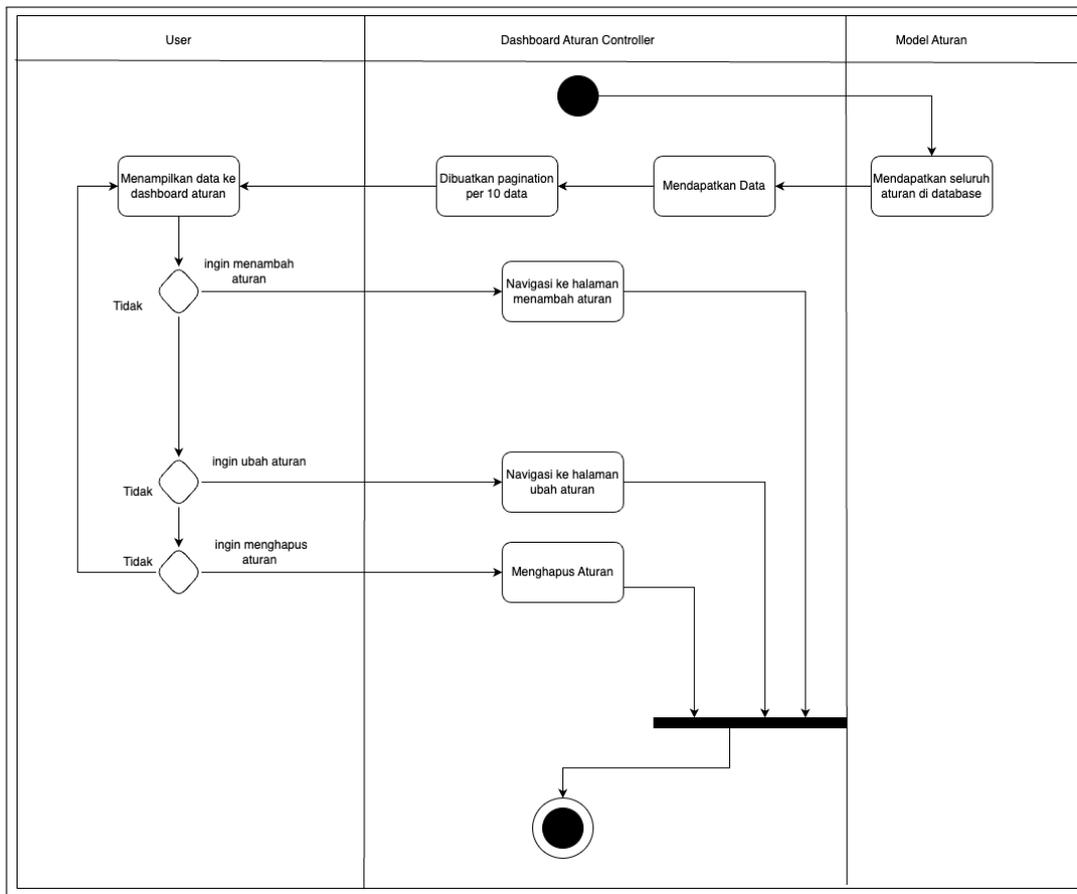


Gambar 3.21. Activity Diagram Dashboard Hapus Gejala

3.9.18 Dashboard Aturan

Pada Gambar 3.22 merupakan alur kerja dari halaman *dashboard* aturan. Pada halaman ini pengguna ditampilkan seluruh aturan dari hewan sapi dan kambing. Pengguna bisa menambah aturan pada sapi atau kambing dengan menekan tombol tambah aturan, lalu pengguna akan dipindahkan ke halaman dashboard tambah aturan. Pengguna dapat mengubah isi aturan dari aturan yang sudah ada dengan menekan tombol ubah, lalu pengguna akan dipindahkan ke halaman dashboard ubah aturan. Pengguna juga bisa untuk menghapus aturan dengan cara menekan tombol hapus.

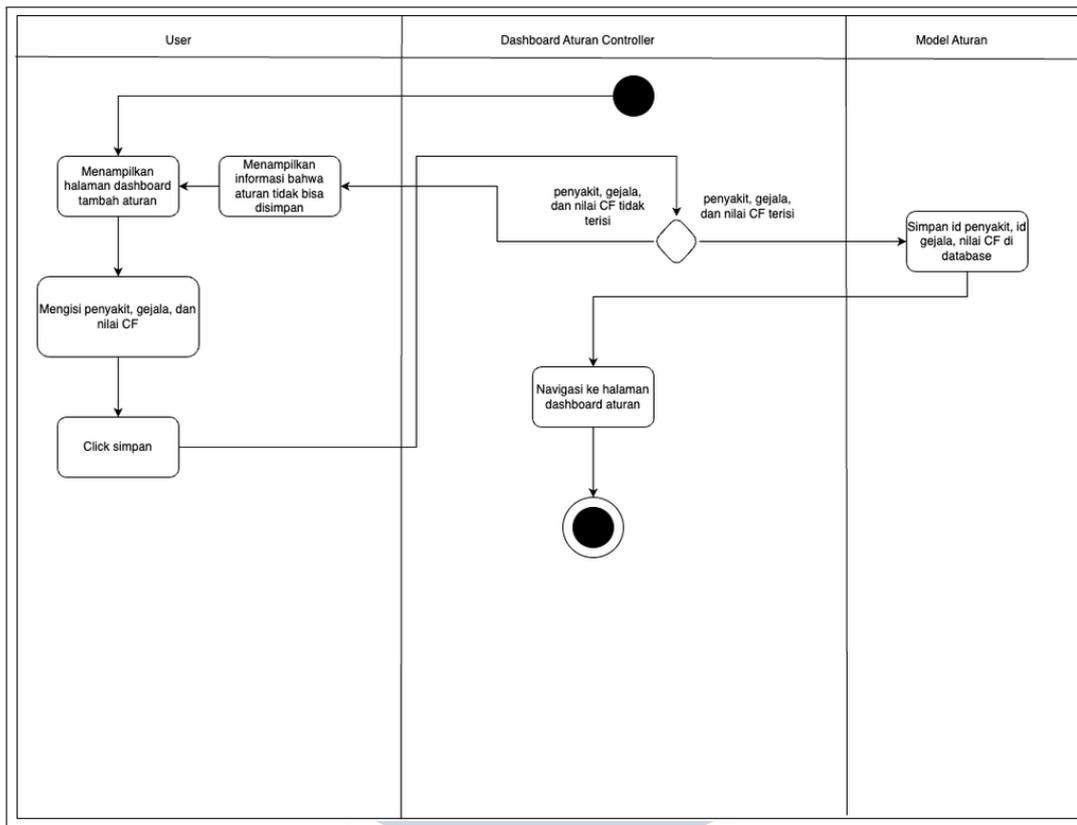
U M V I N
 U N I V E R S I T A S
 M U L T I M E D I A
 N U S A N T A R A



Gambar 3.22. Activity Diagram Dashboard Aturan

3.9.19 Dashboard Tambah Aturan

Pada Gambar 3.23 merupakan alur kerja dari halaman *dashboard* tambah aturan. Pada halaman ini pengguna bisa menambahkan aturan dengan mengisi penyakit, gejala, dan nilai kepercayaan. Setelah penyakit, gejala, dan nilai kepercayaan terisi, maka akan ada pengecekan, yaitu, pengecekan apakah penyakit, gejala, dan nilai kepercayaan sudah terisi. Jika pengecekan ada yang tidak sesuai, maka akan menampilkan informasi bahwa aturan tidak bisa disimpan. Setelah dilakukannya pengecekan, maka penyakit, gejala, dan nilai kepercayaan akan disimpan pada database. Setelah data disimpan pada database, pengguna dipindahkan ke halaman *dashboard* aturan.

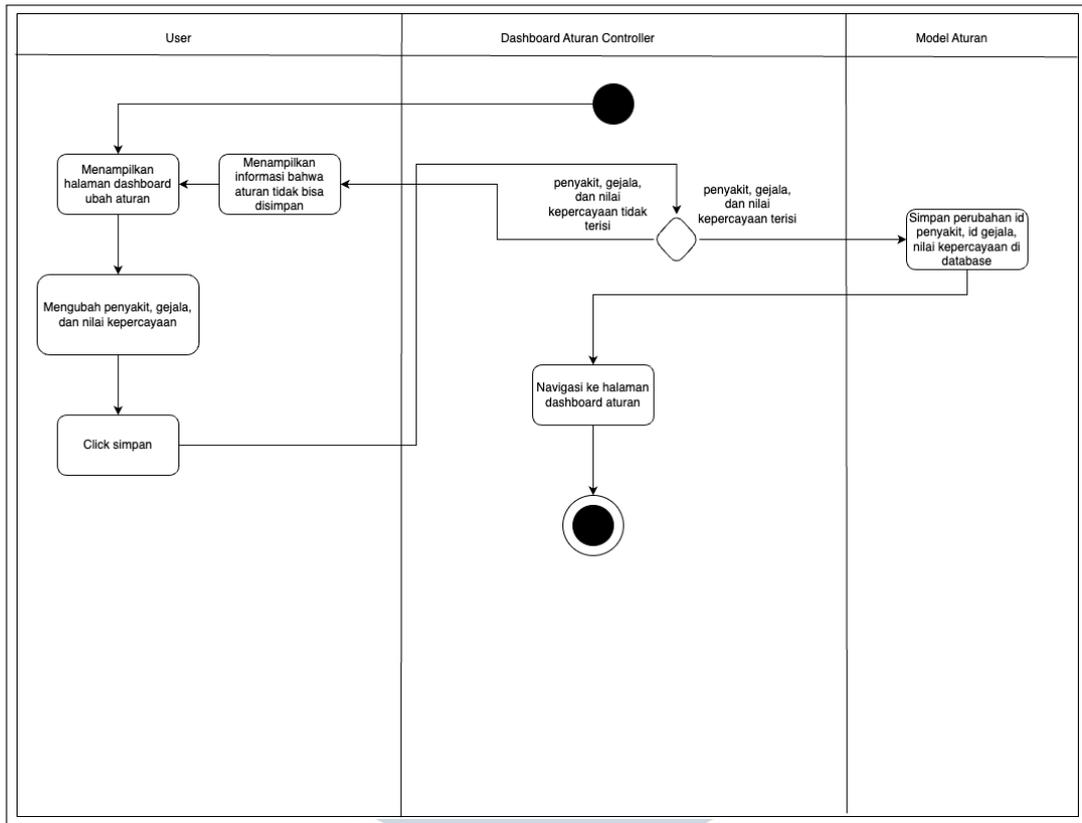


Gambar 3.23. Activity Diagram Dashboard Tambah Aturan

3.9.20 Dashboard Ubah Aturan

Pada Gambar 3.24 merupakan alur kerja dari halaman *dashboard* ubah aturan. Pada halaman ini pengguna dapat mengubah detail dari aturan. Pengguna dapat merubah penyakit, gejala, dan nilai kepercayaan. Setelah mengisi data baru, maka akan ada pengecekan, yaitu, pengecekan apakah penyakit, gejala, dan nilai kepercayaan sudah terisi . Jika pengecekan ada yang tidak sesuai, maka akan menampilkan informasi bahwa aturan tidak dapat diubah. Setelah melewati pengecekan, maka data baru akan mengganti data lama di database. Setelah menyimpan data baru di database pengguna dipindahkan ke halaman *dashboard* aturan.

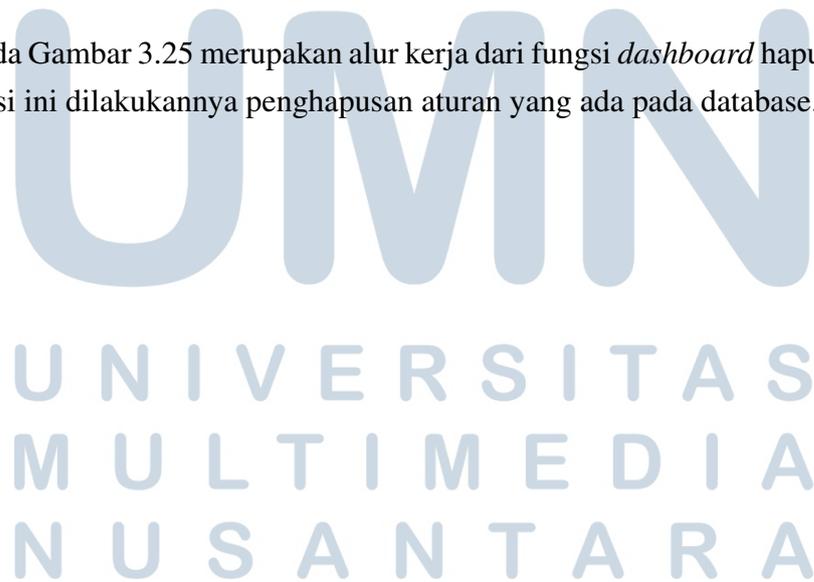
MULTIMEDIA
NUSANTARA

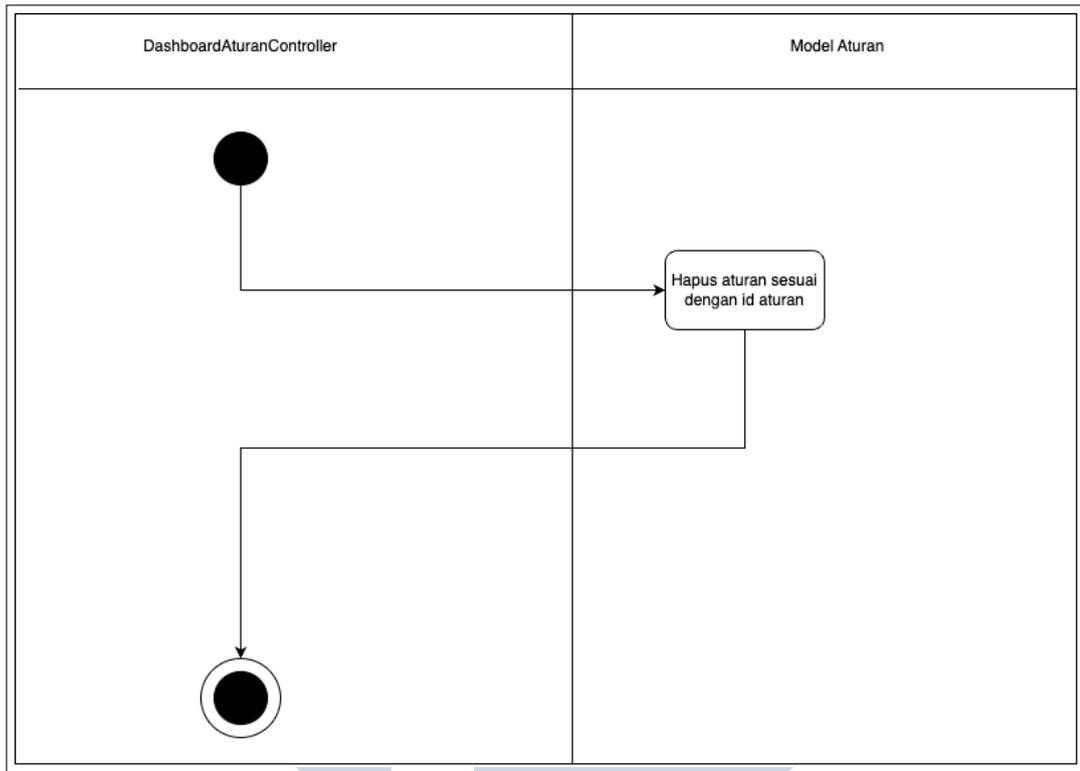


Gambar 3.24. Activity Diagram Dashboard Ubah Aturan

3.9.21 Dashboard Hapus Aturan

Pada Gambar 3.25 merupakan alur kerja dari fungsi *dashboard* hapus aturan. Pada fungsi ini dilakukannya penghapusan aturan yang ada pada database.



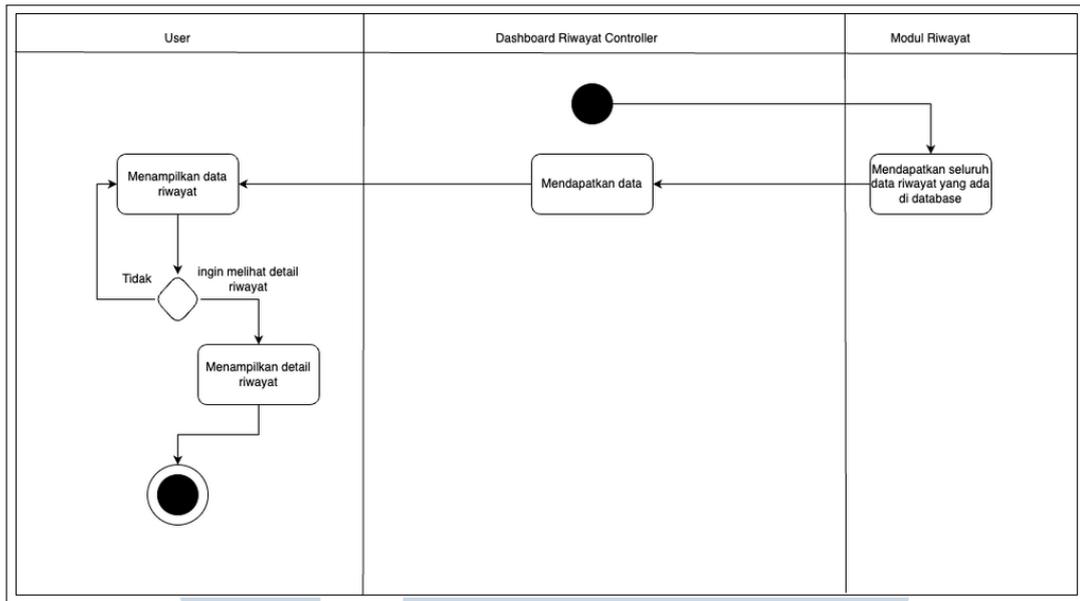


Gambar 3.25. Activity Diagram Dashboard Hapus Aturan

3.9.22 Dashboard Riwayat

Pada Gambar 3.26 merupakan alur kerja dari halaman *dashboard* riwayat. Pada halaman ini pengguna dapat melihat riwayat dari keseluruhan diagnosis yang pernah dilakukan oleh kategori user sudah masuk dan kategori user admin.

UIN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

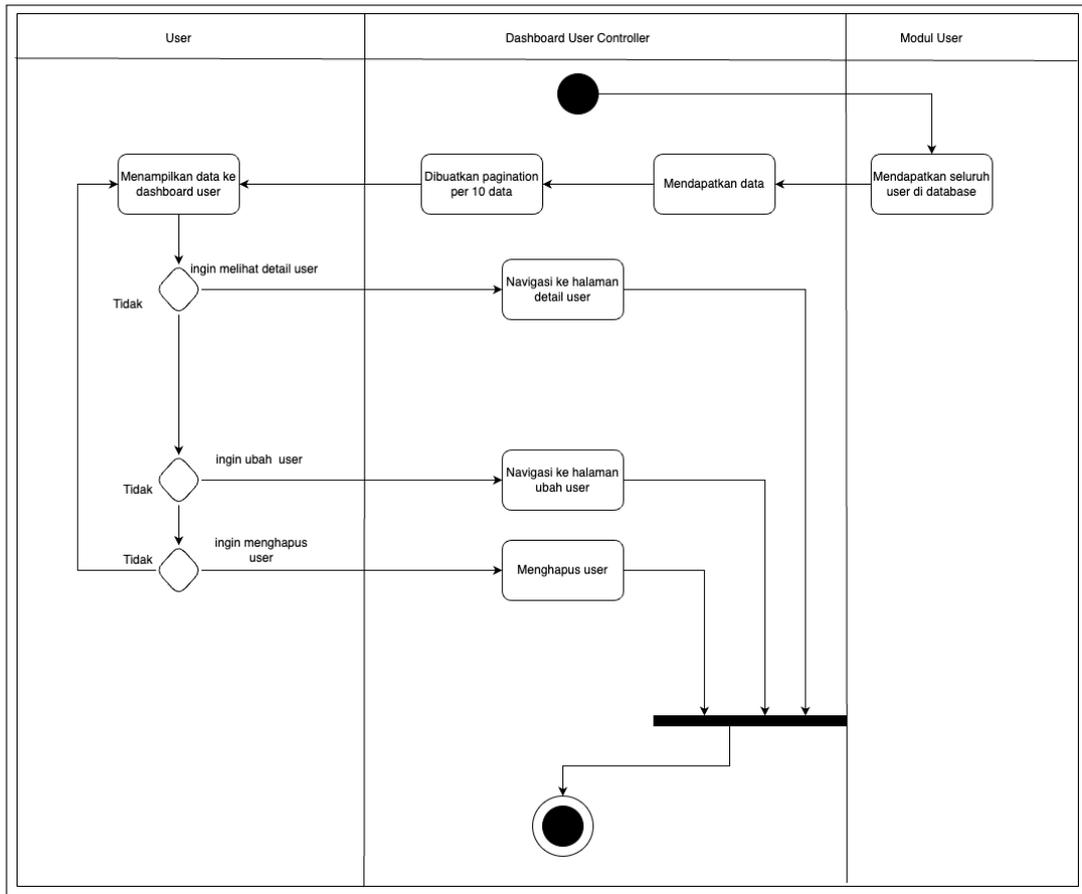


Gambar 3.26. Activity Diagram Dashboard Riwayat

3.9.23 Dashboard User

Pada Gambar 3.27 merupakan alur kerja dari halaman *dashboard* user. Pada halaman ini pengguna ditampilkan seluruh user yang pernah membuat akun. Pengguna bisa mengubah detail user dari user yang sudah ada dengan menekan tombol ubah, lalu pengguna akan dipindahkan ke halaman dashboard ubah user. Pengguna dapat melihat detail dari user dengan cara menekan tombol detail, lalu pengguna akan dipindahkan ke halaman dashboard detail user. Pengguna juga bisa untuk menghapus user dengan cara menekan tombol hapus.

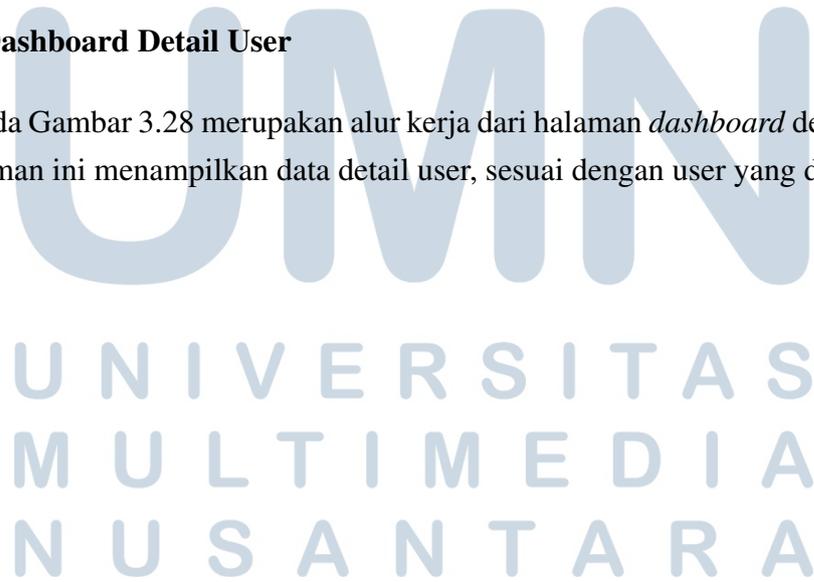
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

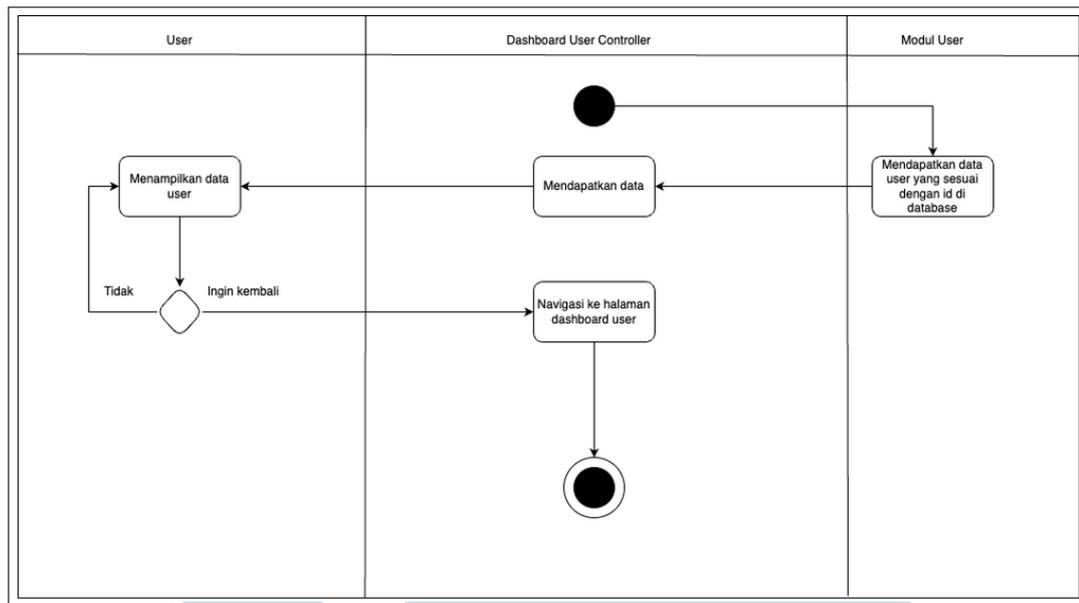


Gambar 3.27. Activity Diagram Dashboard User

3.9.24 Dashboard Detail User

Pada Gambar 3.28 merupakan alur kerja dari halaman *dashboard* detail user. Pada halaman ini menampilkan data detail user, sesuai dengan user yang dipilih.



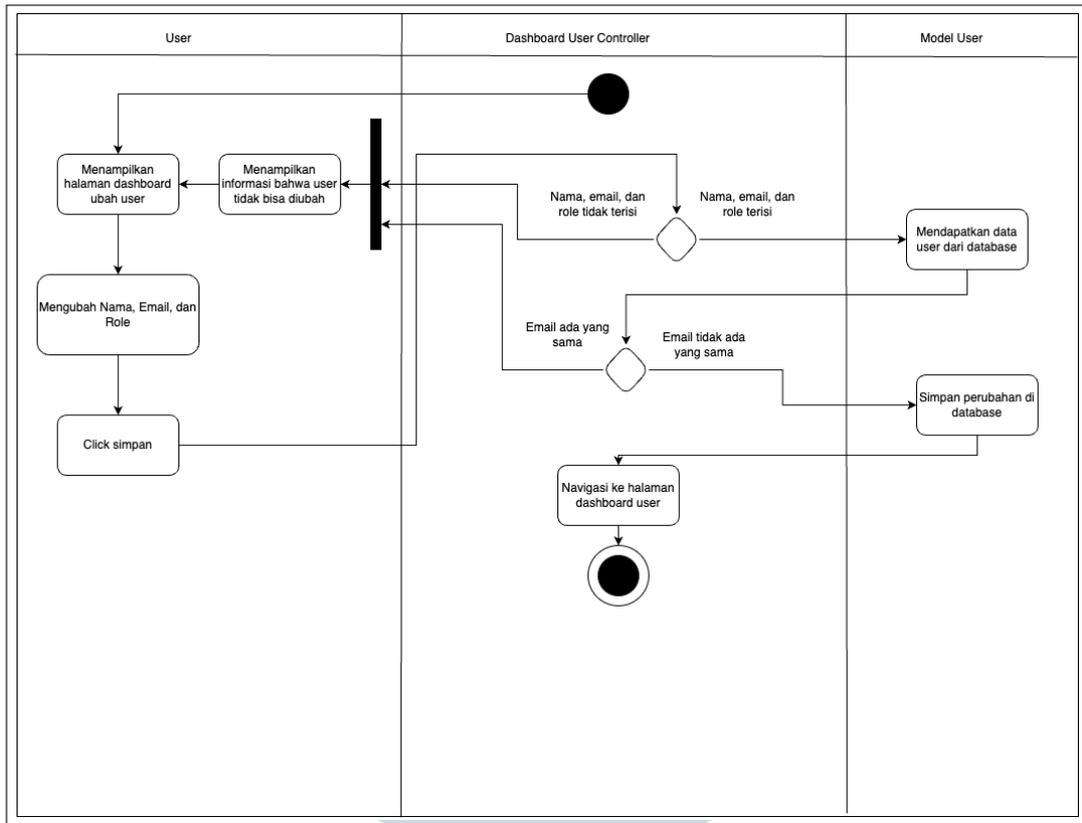


Gambar 3.28. Activity Diagram Dashboard Detail User

3.9.25 Dashboard Ubah User

Pada Gambar 3.29 merupakan alur kerja dari halaman *dashboard* ubah user. Pada halaman ini pengguna dapat mengubah detail dari user. Pengguna dapat merubah nama, email, dan role. Setelah mengisi data baru, maka akan ada pengecekan, yaitu, pengecekan apakah nama, email, dan role sudah terisi dan pengecekan apakah email yang akan diganti sudah digunakan. Jika pengecekan ada yang tidak sesuai, maka akan menampilkan informasi bahwa user tidak dapat diubah. Setelah melewati pengecekan, maka data baru akan mengganti data lama di database. Setelah menyimpan data baru di database pengguna dipindahkan ke halaman *dashboard* user.

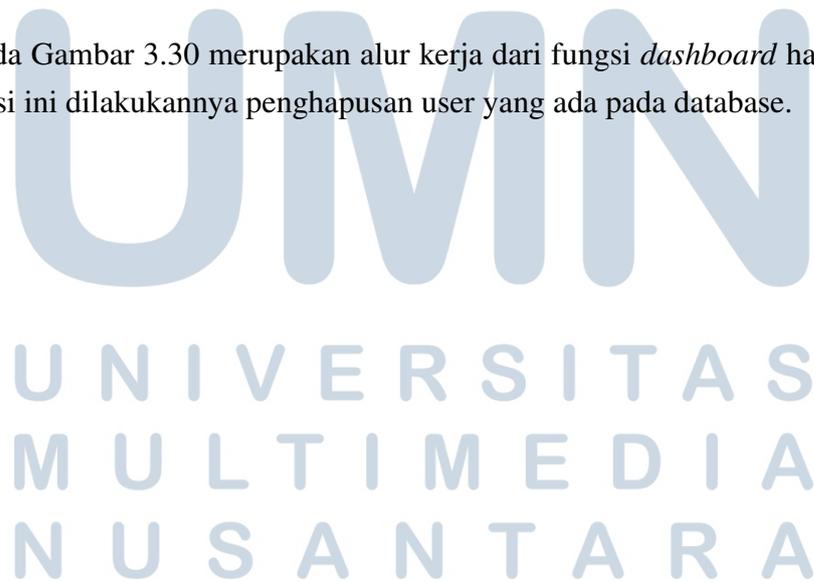
UNIVERSITAS
MULTIMEDIA
NUSANTARA

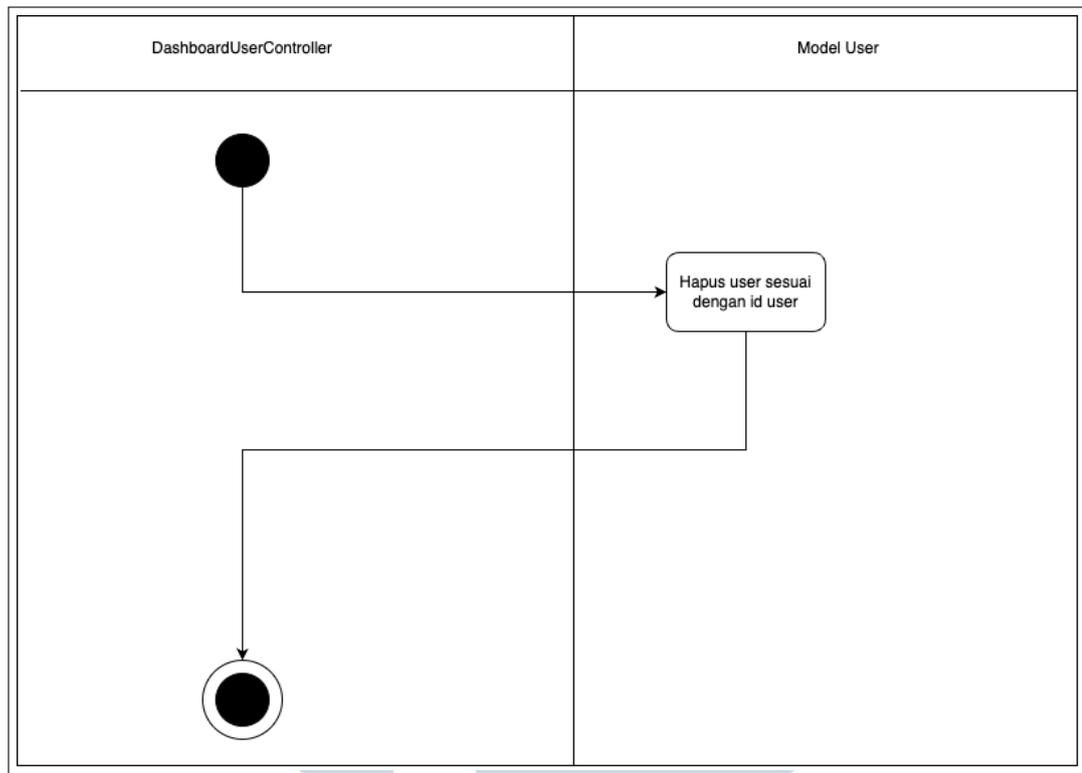


Gambar 3.29. Activity Diagram Dashboard Ubah User

3.9.26 Dashboard Hapus User

Pada Gambar 3.30 merupakan alur kerja dari fungsi *dashboard* hapus user. Pada fungsi ini dilakukannya penghapusan user yang ada pada database.





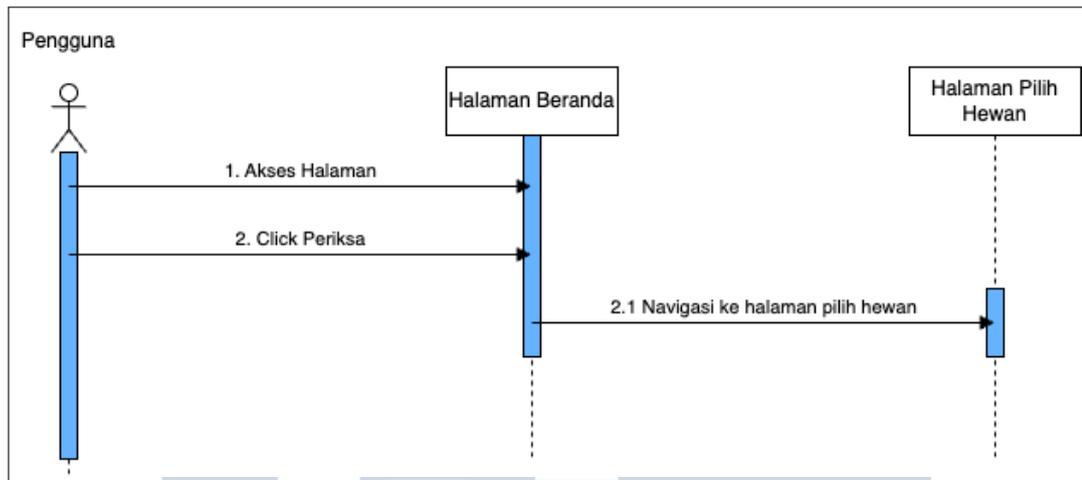
Gambar 3.30. Activity Diagram Dashboard Hapus User

3.10 Sequence Diagram

Pada bagian ini dijabarkan tentang Sequence Diagram yang ada pada penelitian ini. Sequence Diagram digunakan untuk menggambarkan interaksi antara berbagai objek.

3.10.1 Beranda

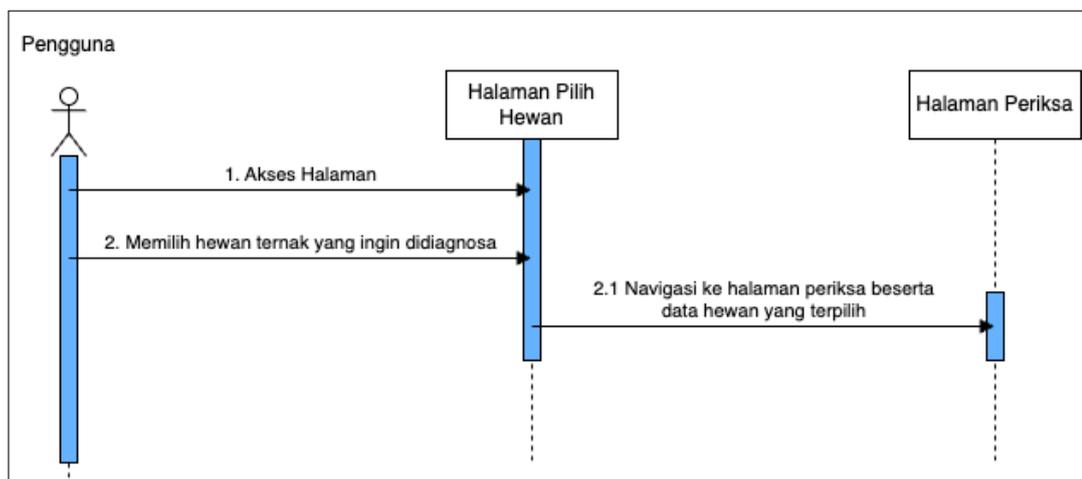
Pada Gambar 3.31 merupakan gambaran interaksi antara objek dari halaman beranda. Pada halaman ini ketika pengguna mengakses pertama kali aplikasi, akan menampilkan halaman beranda. Dari halaman beranda pengguna dapat melakukan diagnosis hewan ternak dengan menekan tombol periksa. Setelah menekan tombol periksa, pengguna akan diarahkan ke halaman pilih hewan.



Gambar 3.31. Sequence Diagram Beranda

3.10.2 Pilih Hewan

Pada Gambar 3.32 merupakan gambaran interaksi antara objek dari halaman pilih hewan. Pada halaman ini pengguna diharuskan untuk memilih hewan ternak yang ingin didiagnosis terlebih dahulu. Setelah memilih hewan ternak yang ingin didiagnosa, pengguna akan diarahkan ke halaman periksa.



Gambar 3.32. Sequence Diagram Pilih Hewan

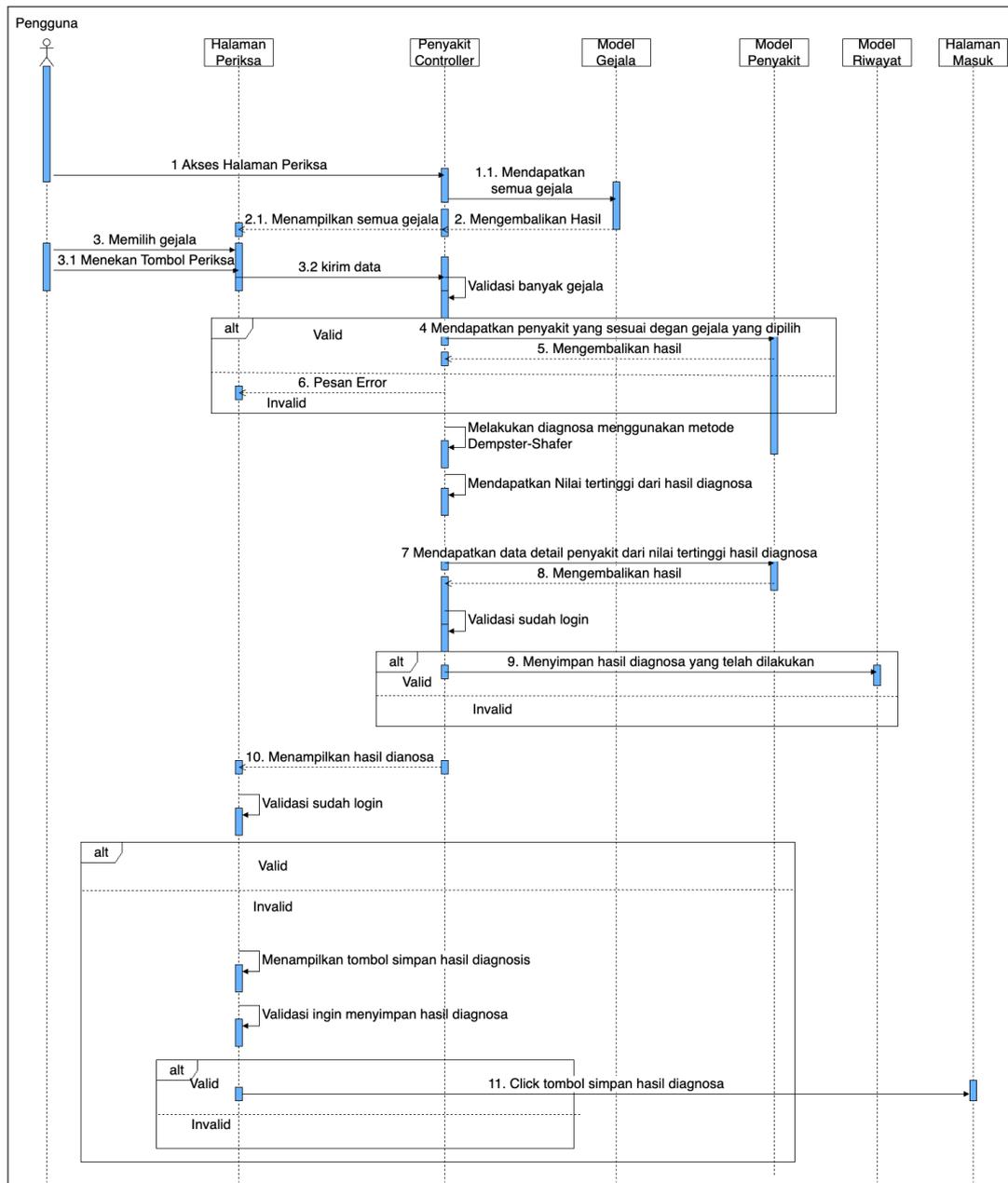
3.10.3 Periksa

Pada Gambar 3.33 merupakan gambaran interaksi antara objek dari halaman periksa. Saat pengguna mengakses halaman periksa, hal pertama yang dilakukan adalah mendapatkan data gejala sesuai dengan hewan ternak yang telah dipilih pada halaman pilih hewan. Pemanggilan data gejala tersebut dilakukan di *class*

PenyakitController dan ditampilkan pada halaman periksa. Setelah ditampilkannya gejala, pengguna diharuskan untuk memilih gejala sesuai dengan gejala yang dialami. Setelah memilih gejala, maka akan ada pengecekan apakah ada gejala yang dikirim pada *class* PenyakitController. Jika tidak ada gejala yang dikirim, maka *class* PenyakitController akan mengirimkan pesan *error* ke halaman periksa.

Setelah dilakukannya pengecekan, maka *class* PenyakitController akan mengambil data penyakit dari *database* berupa penyakit yang sesuai dengan gejala yang dipilih. Setelah mendapatkan penyakit yang sesuai, maka dilakukan perhitungan Dempster-Shafer dan didapatkan nilai presentase tertinggi penyakit. Setelah mendapatkan nilai presentase tertinggi penyakit, maka *class* PenyakitController akan mengambil data pada *database* berupa detail penyakit dari penyakit dengan nilai presentase tertinggi. Setelah mendapatkan detail penyakit, maka dilakukan pengecekan apakah pengguna sudah masuk atau belum. Jika pengguna sudah masuk, hasil diagnosis tersebut akan disimpan pada *database*. Setelah dilakukannya pengecekan, maka ditampilkannya hasil diagnosis ke halaman periksa. Pada halaman periksa dilakukan pengecekan apakah pengguna sudah masuk atau belum. Jika pengguna belum masuk maka akan menampilkan tombol simpan hasil diagnosis. Jika pengguna menekan tombol simpan hasil diagnosis, maka pengguna akan diarahkan ke halaman masuk.



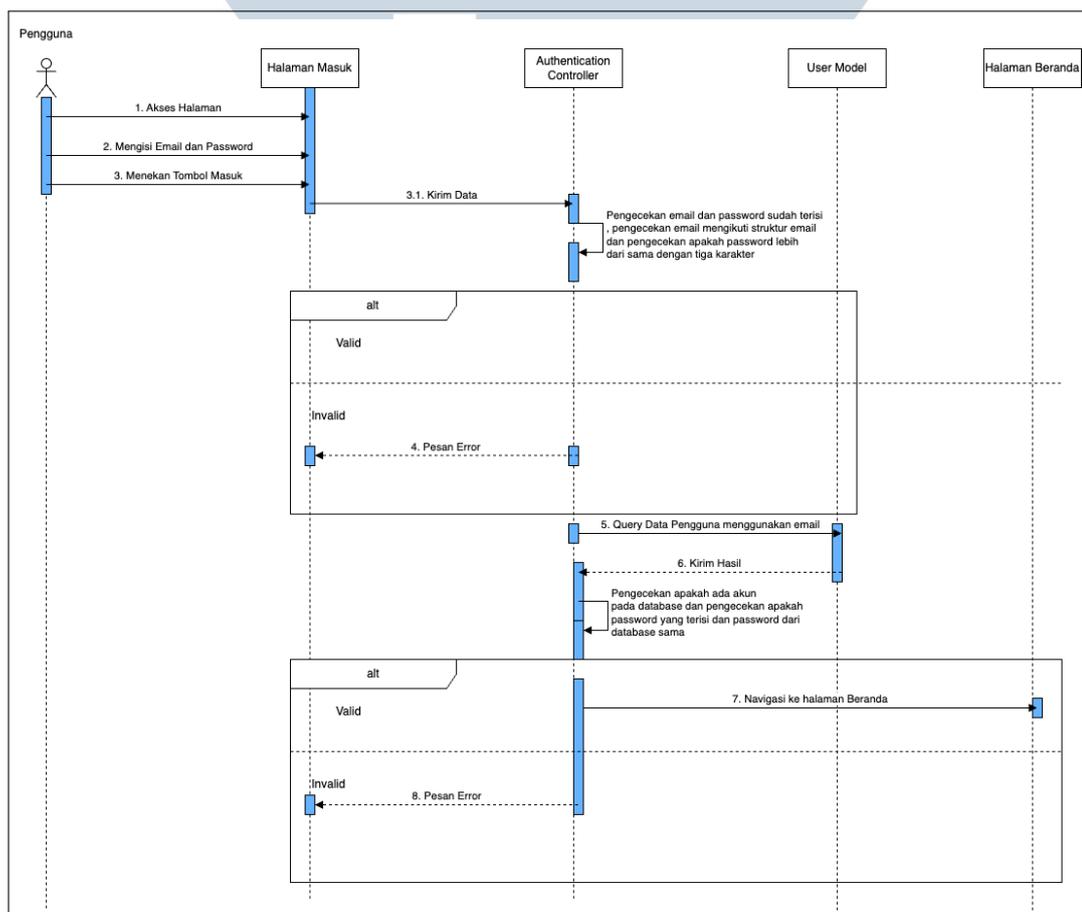


Gambar 3.33. Sequence Diagram Periksa

3.10.4 Masuk

Pada Gambar 3.34 merupakan gambaran interaksi antara objek dari halaman masuk. Ketika pengguna mengakses halaman masuk, pengguna dapat memasukkan *email* dan *password* dari akun yang telah dibuat. Jika pengguna tidak memiliki akun, pengguna dapat membuat akun terlebih dahulu, dengan menekan tombol daftar dan pengguna akan dipindahkan ke halaman daftar. Jika telah mengisi email

dan password serta menekan tombol masuk, halaman masuk akan mengirim data tersebut ke *class* AuthenticationController. Pada *class* AuthenticationController setelah menerima data dari halaman masuk, akan ada beberapa pengecekan, yaitu, pengecekan apakah email dan password sudah terisi, pengecekan apakah email mengikuti struktur email, dan pengecekan apakah password yang diisi kurang dari tiga karakter. Jika ada pengecekan tidak sesuai, maka akan menampilkan *error* dihalaman masuk. Setelah dilakukannya pengecekan, lalu *class* AuthenticationController akan mengambil data dari akun pengguna *database*. Setelah mendapatkan akun pengguna dari *database*, dilakukanlah beberapa pengecekan, yaitu, pengecekan apakah akun tersebut ada pada database dan pengecekan apakah password yang diisi dan ditabase sama. Jika dari pengecekan ada yang tidak sesuai, maka pada halaman masuk akan menampilkan pesan *error*. Jika semua pengecekan sudah sesuai, pengguna akan dipindahkan ke halaman beranda.

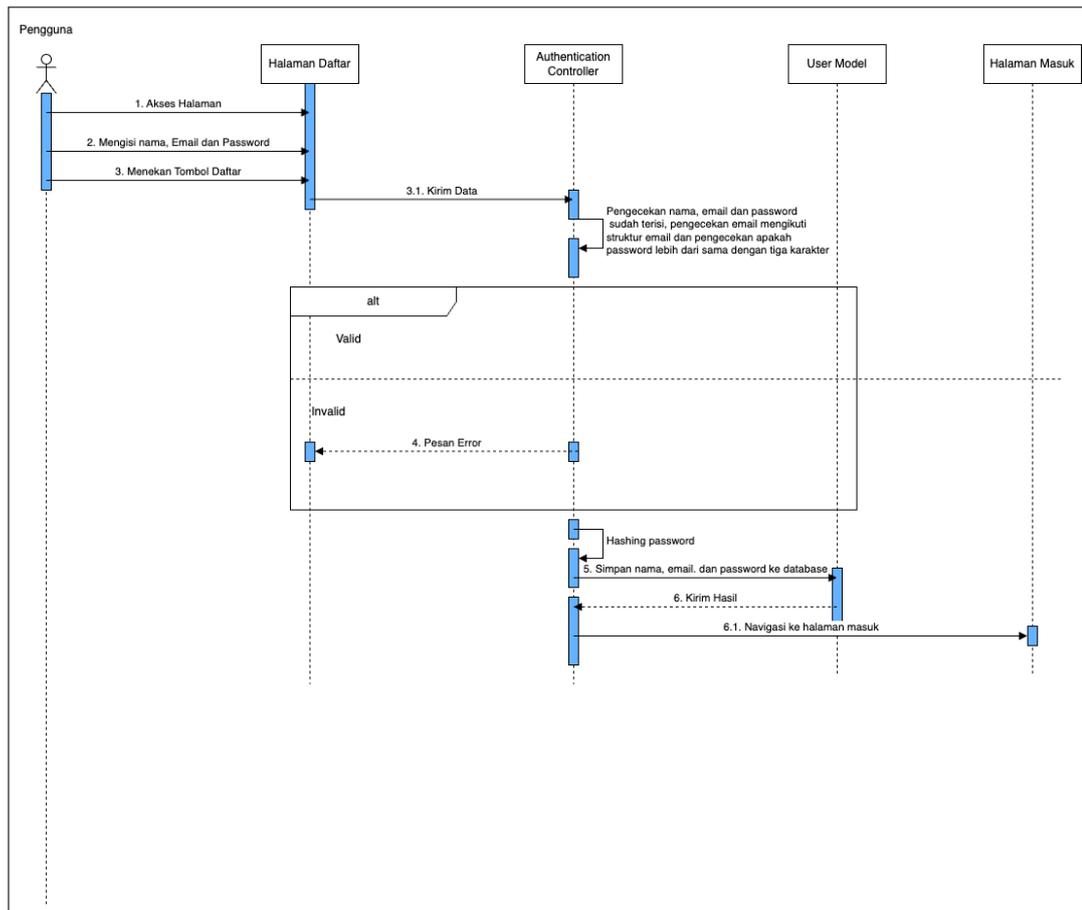


Gambar 3.34. Sequence Diagram Masuk

3.10.5 Daftar

Pada Gambar 3.35 merupakan gambaran interaksi antara objek dari halaman daftar. Ketika pengguna mengakses halaman daftar, pengguna dapat memasukkan nama, *email*, dan *password*. Jika pengguna sudah memiliki akun, pengguna dapat langsung masuk, dengan menekan tombol masuk dan pengguna akan dipindahkan ke halaman masuk. Jika telah mengisi nama, email, dan password serta menekan tombol daftar, halaman daftar akan mengirim data tersebut ke *class* *AuthenticationController*. Pada *class* *AuthenticationController* setelah menerima data dari halaman daftar, akan ada beberapa pengecekan, yaitu, pengecekan apakah nama, email, dan password sudah terisi, pengecekan apakah email mengikuti struktur email, dan pengecekan apakah password yang diisi kurang dari tiga karakter. Jika ada pengecekan tidak sesuai, maka akan menampilkan pesan *error* di halaman daftar. Setelah dilakukannya pengecekan, lalu *class* *AuthenticationController* akan mencoba mengambil data berdasarkan email yang telah diisi. Setelah dilakukan pengambilan data berdasarkan email dari *database*, dilakukanlah pengecekan apakah email yang telah diisi sudah digunakan oleh pengguna lain. Jika dari pengecekan ada yang tidak sesuai, maka pada halaman daftar akan menampilkan pesan *error*. Jika semua pengecekan sudah sesuai, pengguna akan dipindahkan ke halaman masuk.



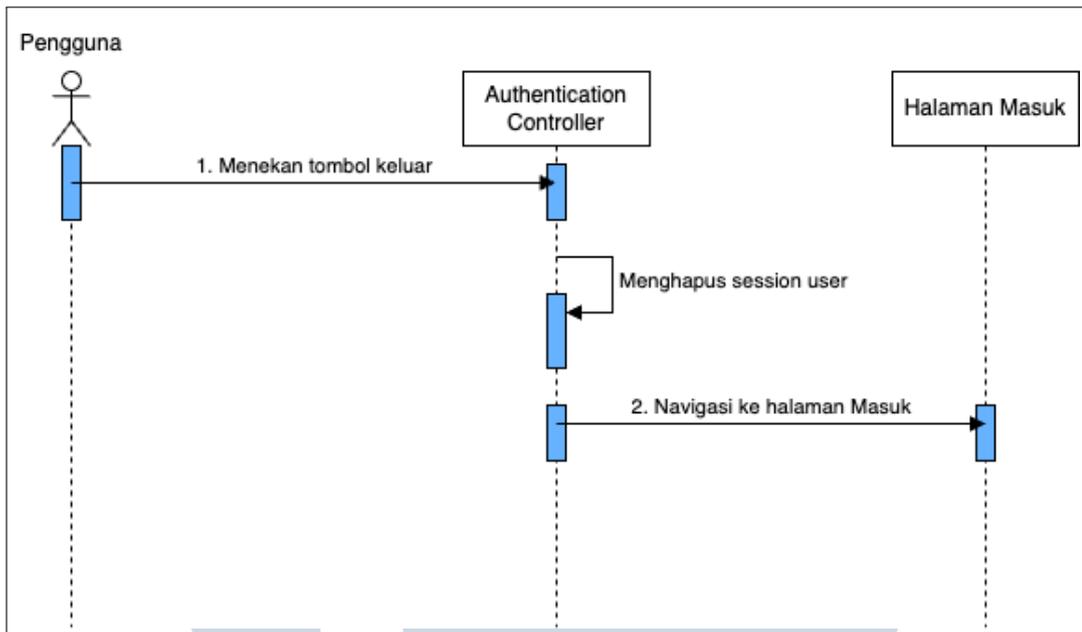


Gambar 3.35. Sequence Diagram Daftar

3.10.6 Keluar

Pada Gambar 3.36 merupakan gambaran interaksi antara objek dari halaman keluar. Ketika pengguna menekan tombol keluar, maka pada *class* AuthenticationController akan menghapus *session user* dan memindahkan pengguna ke halaman beranda.

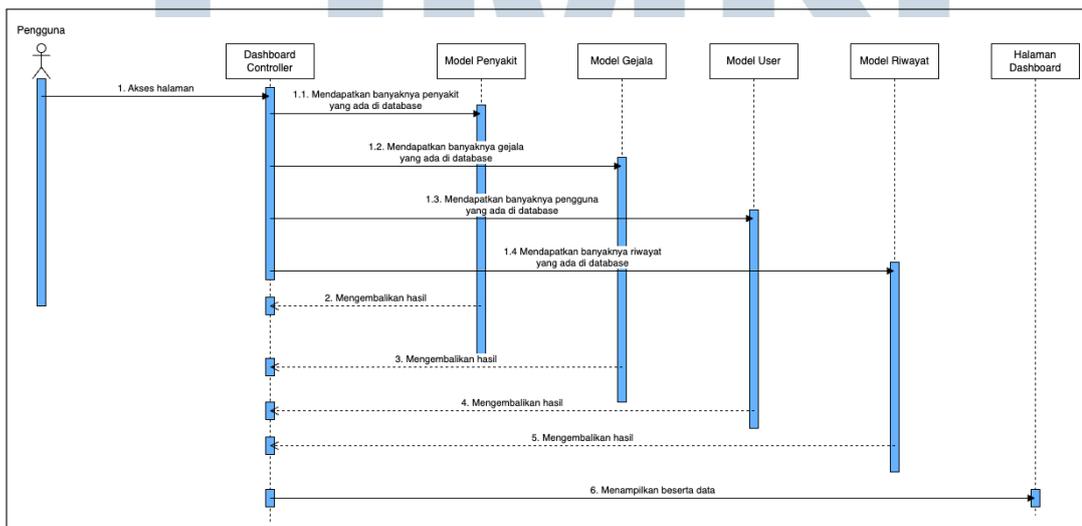
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.36. Sequence Diagram Keluar

3.10.7 Dashboard

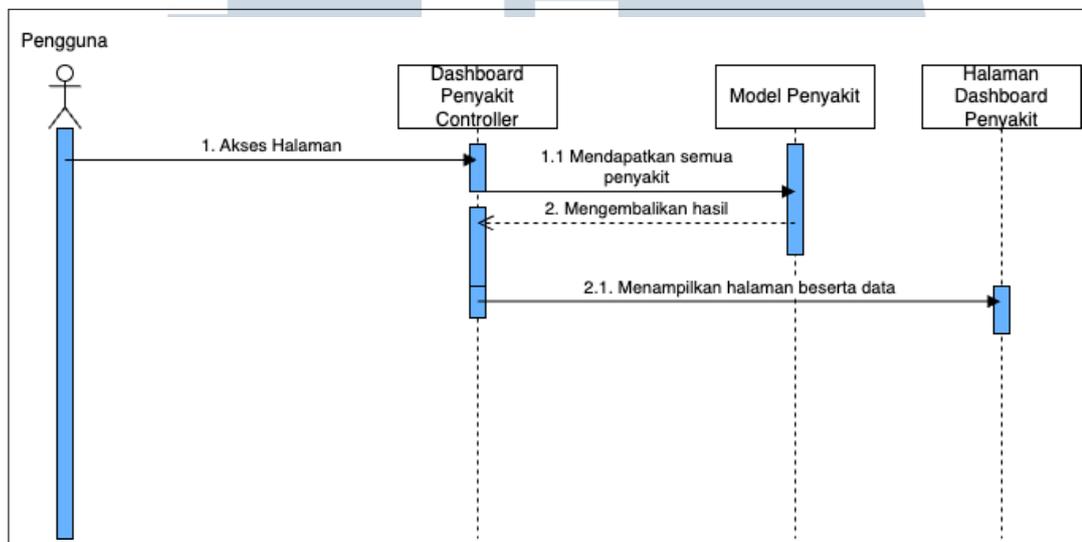
Pada Gambar 3.37 merupakan gambaran interaksi antara objek dari halaman *dashboard*. Ketika pengguna mengakses halaman *dashboard*, *class* *DashboardController* akan melakukan menghitung banyaknya penyakit, banyaknya gejala, banyaknya pengguna, dan banyaknya riwayat yang ada pada *database*. Setelah mendapatkan data dari *database*, maka data tersebut akan ditampilkan pada halaman *dashboard*.



Gambar 3.37. Sequence Diagram Dashboard

3.10.8 Dashboard Penyakit

Pada Gambar 3.38 merupakan gambaran interaksi antara objek dari halaman *dashboard* penyakit. Ketika pengguna mengakses halaman *dashboard* penyakit, *class* `DashboardPenyakitController` akan mengambil data seluruh penyakit dari database. Setelah mendapatkan data dari database, *class* `DashboardPenyakitController` akan menampilkan data tersebut ke halaman *dashboard* penyakit.

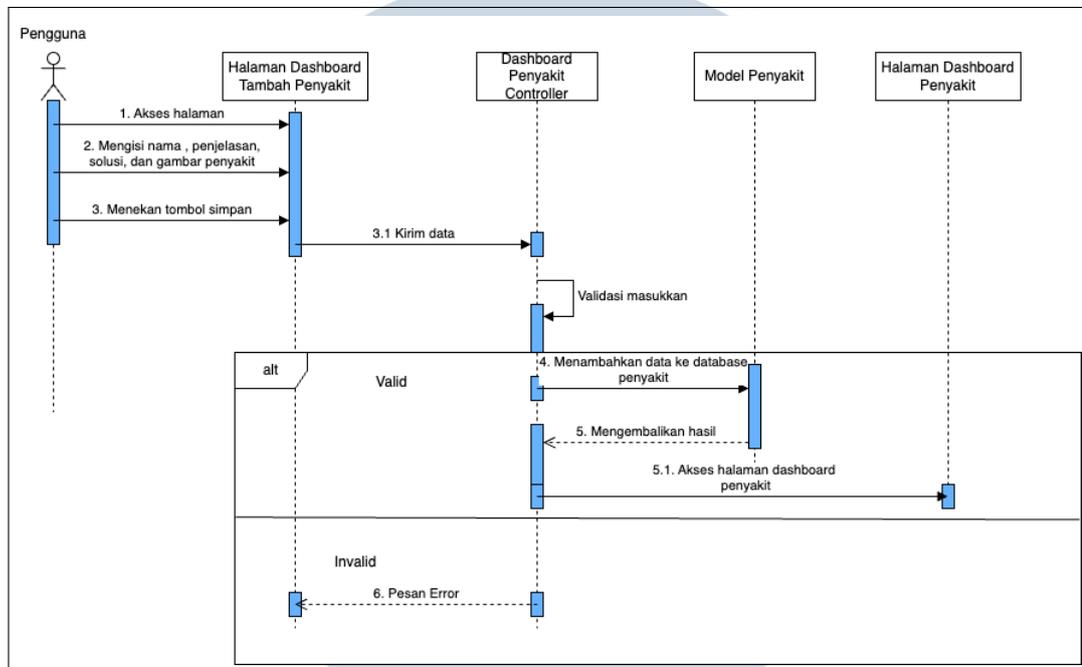


Gambar 3.38. Sequence Diagram Dashboard Penyakit

3.10.9 Dashboard Tambah Penyakit

Pada Gambar 3.39 merupakan gambaran interaksi antara objek dari halaman *dashboard* tambah penyakit. Ketika pengguna mengakses halaman *dashboard* tambah penyakit, pengguna dapat mengisi nama penyakit, penjelasan penyakit, solusi penyakit, dan gambar penyakit. Setelah mengisi nama penyakit, penjelasan penyakit, solusi penyakit, dan gambar penyakit, data tersebut akan dikirim ke *class* `DashboardPenyakitController`. Pada *class* `DashboardPenyakitController` setelah menerima data dari halaman *dashboard* tambah penyakit akan melakukan pengecekan apakah nama penyakit, penjelasan penyakit, solusi penyakit, dan gambar penyakit sudah terisi dan apakah gambar penyakit berformat png, jpeg, dan jpg. Jika pengecekan tidak sesuai, maka *class* `DashboardPenyakitController` akan mengirimkan pesan *error* ke halaman *dashboard* tambah penyakit. Setelah melakukan pengecekan, *class* `DashboardPenyakitController` akan menyimpan data

tersebut ke dalam database. Setelah menambahkan data ke dalam database berhasil, pengguna akan diarahkan ke halaman *dashboard* penyakit

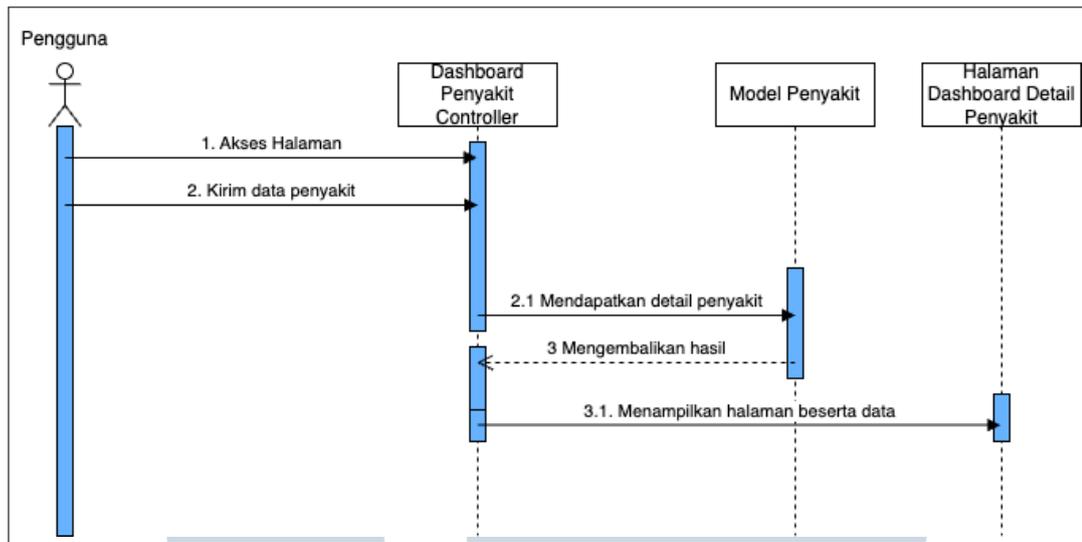


Gambar 3.39. Sequence Diagram Dashboard Tambah Penyakit

3.10.10 Dashboard Detail Penyakit

Pada Gambar 3.40 merupakan gambaran interaksi antara objek dari halaman *dashboard* detail penyakit. Ketika pengguna mengakses halaman *dashboard* detail penyakit beserta mengirim data penyakit yang ingin dilihat detailnya, maka *class* DashboardPenyakitController akan mengambil dari *database* detail dari penyakit sesuai dengan data yang dikirim . Setelah mendapatkan data detail penyakit, *class* DashboardPenyakitController akan menampilkan data tersebut pada halaman *dashboard* detail penyakit.

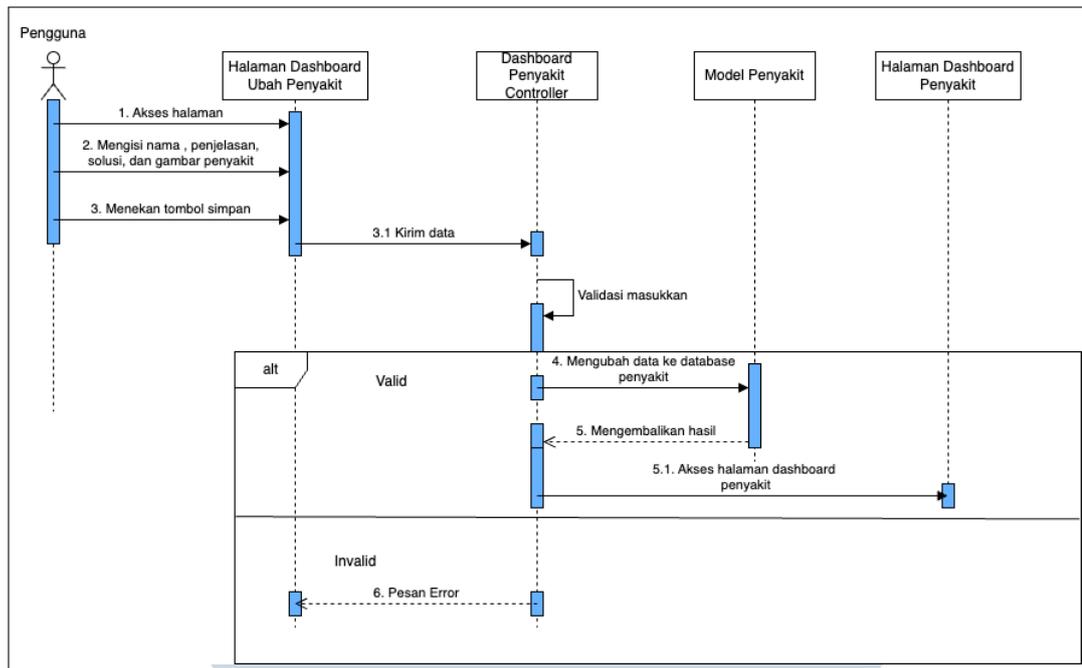
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.40. Sequence Diagram Dashboard Detail Penyakit

3.10.11 Dashboard Ubah Penyakit

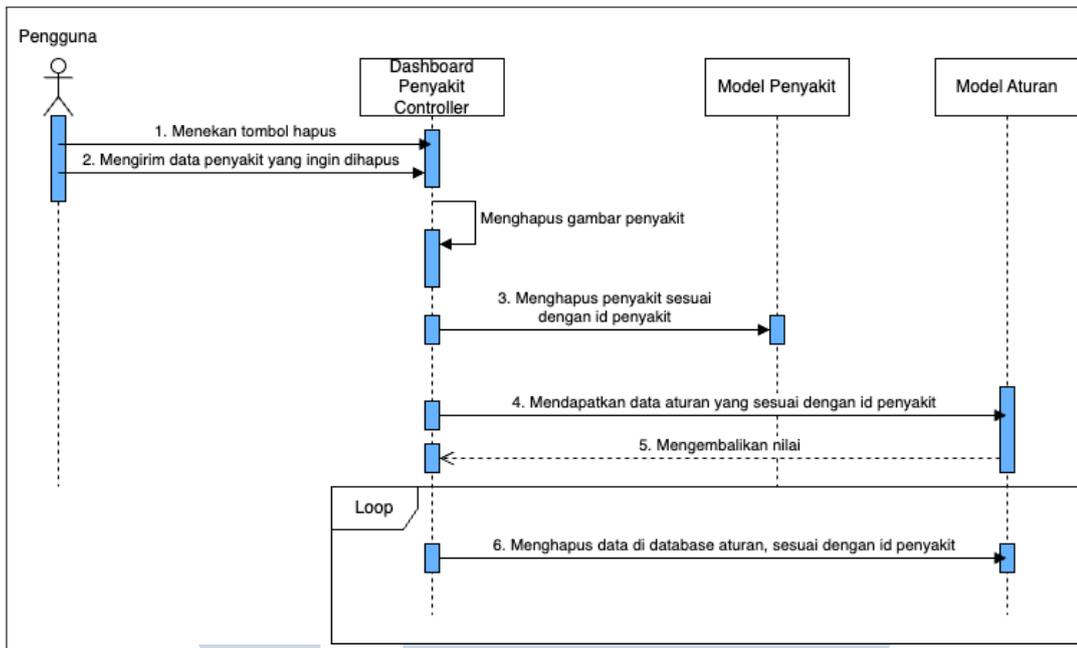
Pada Gambar 3.41 merupakan gambaran interaksi antara objek dari halaman *dashboard* ubah penyakit. Ketika pengguna mengakses halaman *dashboard* ubah penyakit, pengguna dapat mengubah nama penyakit, penjelasan penyakit, solusi penyakit, dan gambar penyakit. Setelah mengisi nama penyakit, penjelasan penyakit, solusi penyakit, atau gambar penyakit, data tersebut akan dikirim ke *class* *DashboardPenyakitController*. Pada *class* *DashboardPenyakitController* setelah menerima data dari halaman *dashboard* ubah penyakit akan melakukan pengecekan apakah nama penyakit, penjelasan penyakit, solusi penyakit, dan gambar penyakit sudah terisi dan apakah gambar penyakit berformat png, jpeg, dan jpg. Jika pengecekan tidak sesuai, maka *class* *DashboardPenyakitController* akan mengirimkan pesan *error* ke halaman *dashboard* ubah penyakit. Setelah melakukan pengecekan, *class* *DashboardPenyakitController* akan mengubah data lama yang ada pada *database* menjadi data baru. Setelah mengubah data ke dalam *database* berhasil, pengguna akan diarahkan ke halaman *dashboard* penyakit.



Gambar 3.41. Sequence Diagram Dashboard Ubah Penyakit

3.10.12 Dashboard Hapus Penyakit

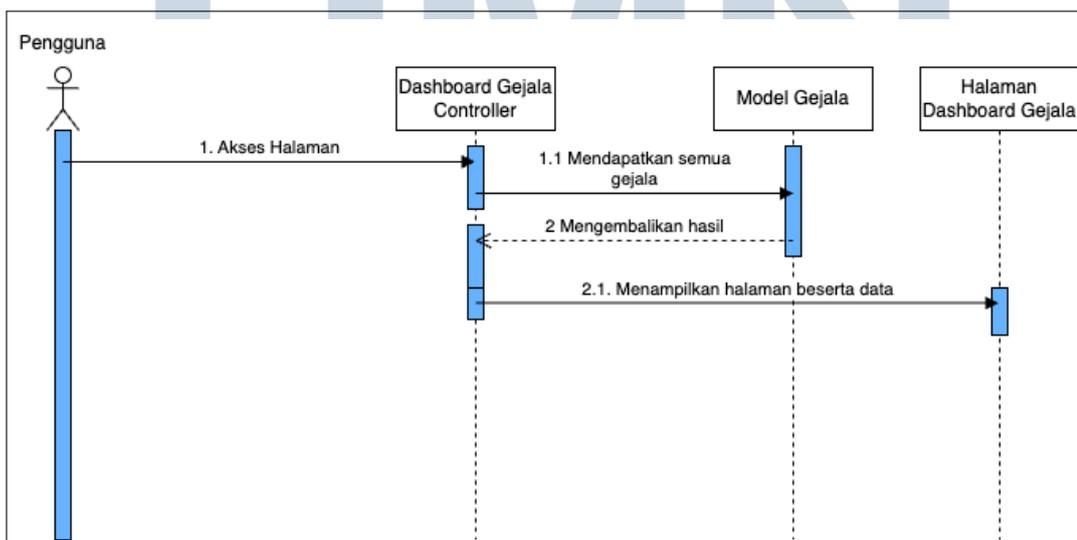
Pada Gambar 3.42 merupakan gambaran interaksi antara objek dari fungsi *dashboard* hapus penyakit. ketika pengguna menekan tombol hapus, maka dari halaman dashboard penyakit akan mengirim data penyakit yang ingin dihapus ke *class* DashboardPenyakitController. Pada *class* DashboardPenyakitController setelah menerima data dari halaman *dashboard* penyakit, dilakukanlah penghapusan gambar penyakit yang disimpan pada *project*. Setelah menghapus gambar penyakit, dilanjutkan dengan menghapus penyakit sesuai dengan data yang diterima dari halaman *dashboard* penyakit pada *database*. Setelah menghapus data penyakit yang ada pada *database*, dilanjutkanlah dengan mendapatkan data aturan dari *database* yang aturannya memakai penyakit yang ingin dihapus. Setelah didapat aturan yang penyakitnya sama dengan penyakit yang ingin dihapus, maka dihapuslah aturan tersebut pada *database*.



Gambar 3.42. Sequence Diagram Dashboard Hapus Penyakit

3.10.13 Dashboard Gejala

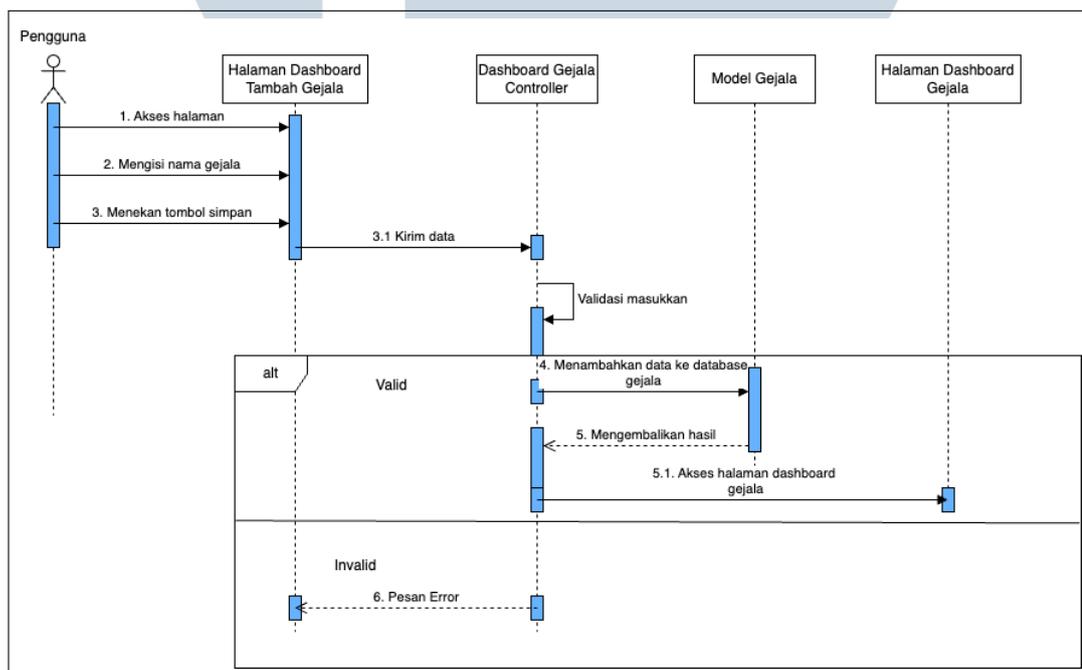
Pada Gambar 3.43 merupakan gambaran interaksi antara objek dari halaman *dashboard* gejala. Ketika pengguna mengakses halaman *dashboard* gejala, *class* DashboardGejalaController akan mengambil data seluruh gejala dari database. Setelah mendapatkan data dari database, *class* DashboardGejalaController akan menampilkan data tersebut ke halaman *dashboard* gejala.



Gambar 3.43. Sequence Diagram Dashboard Gejala

3.10.14 Dashboard Tambah Gejala

Pada Gambar 3.44 merupakan gambaran interaksi antara objek dari halaman *dashboard* tambah gejala. Ketika pengguna mengakses halaman *dashboard* tambah gejala, pengguna dapat mengisi nama gejala. Setelah mengisi nama gejala, data tersebut akan dikirim ke *class* DashboardGejalaController. Pada *class* DashboardGejalaController setelah menerima data dari halaman *dashboard* tambah gejala akan melakukan pengecekan apakah nama gejala sudah terisi. Jika pengecekan tidak sesuai, maka *class* DashboardGejalaController akan mengirimkan pesan *error* ke halaman *dashboard* tambah gejala. Setelah melakukan pengecekan, *class* DashboardGejalaController akan menyimpan data tersebut ke dalam *database*. Setelah menambahkan data ke dalam *database* berhasil, pengguna akan diarahkan ke halaman *dashboard* gejala.

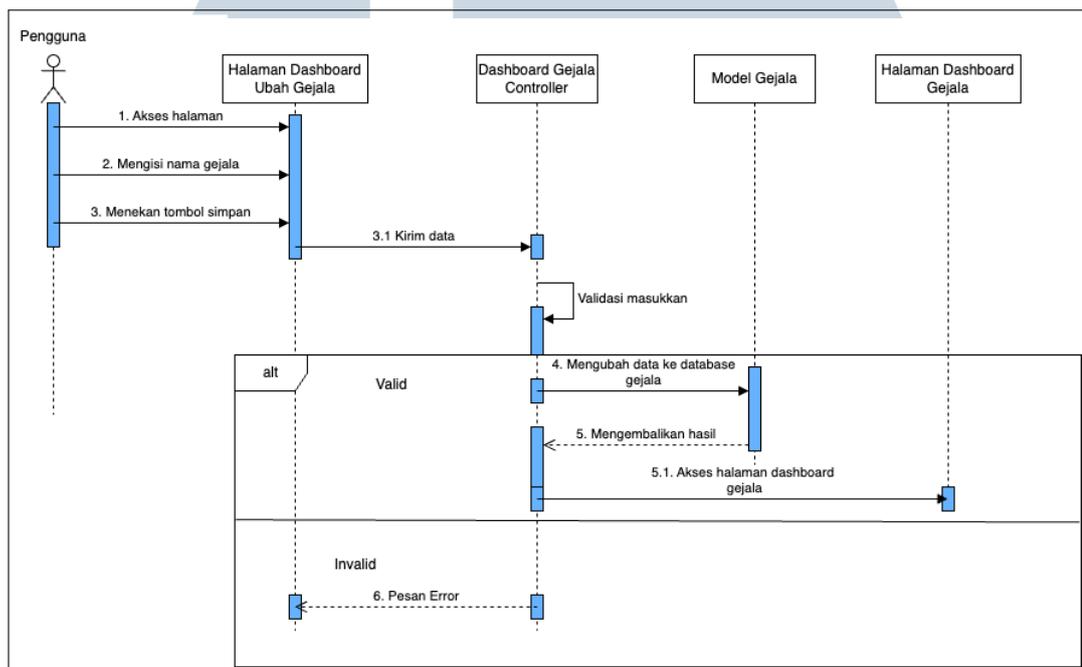


Gambar 3.44. Sequence Diagram Dashboard Tambah Gejala

3.10.15 Dashboard Ubah Gejala

Pada Gambar 3.45 merupakan gambaran interaksi antara objek dari halaman *dashboard* ubah gejala. Ketika pengguna mengakses halaman *dashboard* ubah gejala, pengguna dapat mengubah nama gejala. Setelah mengisi nama gejala, data tersebut akan dikirim ke *class* DashboardGejalaController. Pada *class* DashboardGejalaController setelah menerima data dari halaman *dashboard*

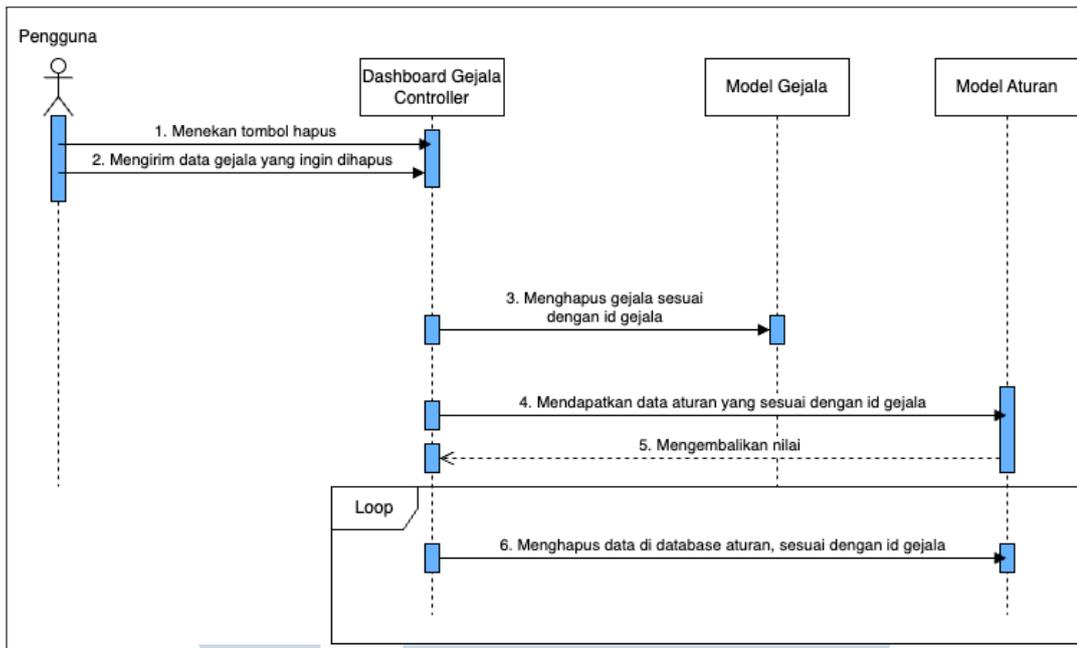
ubah gejala akan melakukan pengecekan apakah nama gejala sudah terisi. Jika pengecekan tidak sesuai, maka *class* DashboardGejalaController akan mengirimkan pesan *error* ke halaman *dashboard* ubah gejala. Setelah melakukan pengecekan, *class* DashboardGejalaController akan mengubah data lama yang ada pada *database* menjadi data baru. Setelah mengubah data ke dalam *database* berhasil, pengguna akan diarahkan ke halaman *dashboard* gejala.



Gambar 3.45. Sequence Diagram Dashboard Ubah Gejala

3.10.16 Dashboard Hapus Gejala

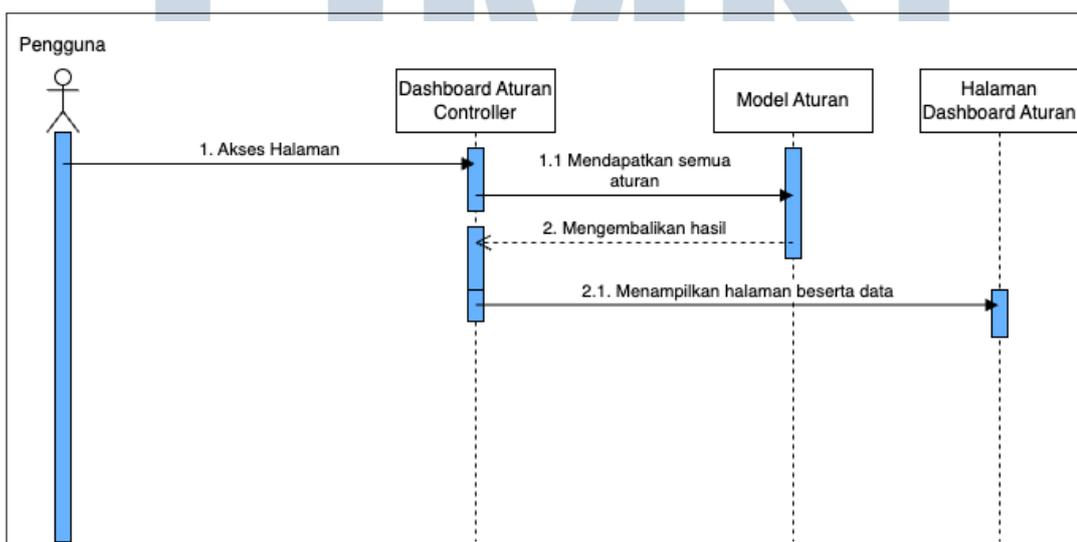
Pada Gambar 3.46 merupakan gambaran interaksi antara objek dari fungsi *dashboard* hapus gejala. ketika pengguna menekan tombol hapus, maka dari halaman *dashboard* gejala akan mengirim data gejala yang ingin dihapus ke *class* DashboardGejalaController. Pada *class* DashboardGejalaController setelah menerima data dari halaman *dashboard* gejala, dilakukanlah penghapusan gejala sesuai dengan data yang diterima dari halaman *dashboard* gejala pada *database*. Setelah menghapus data gejala yang ada pada *database*, dilanjutkan dengan mendapatkan data aturan dari *database* yang aturannya memakai gejala yang ingin dihapus. Setelah didapat aturan yang gejalanya sama dengan gejala yang ingin dihapus, maka dihapuslah aturan tersebut pada *database*.



Gambar 3.46. Sequence Diagram Dashboard Hapus Gejala

3.10.17 Dashboard Aturan

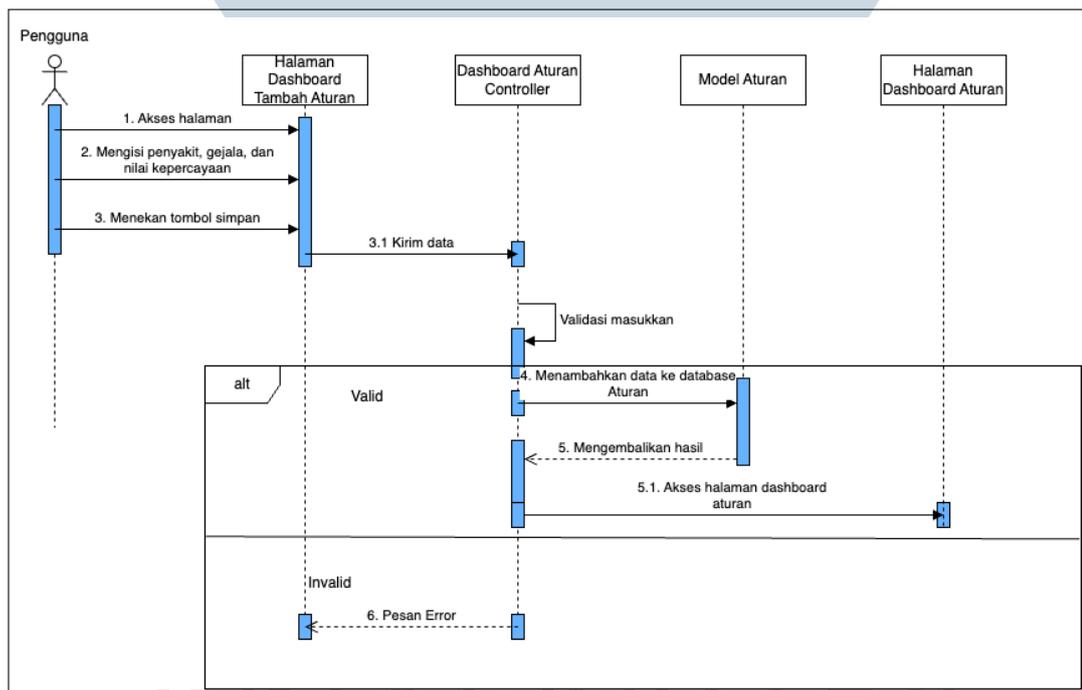
Pada Gambar 3.47 merupakan gambaran interaksi antara objek dari halaman *dashboard* aturan. Ketika pengguna mengakses halaman *dashboard* aturan, *class* DashboardAturanController akan mengambil data seluruh gejala dari database. Setelah mendapatkan data dari database, *class* DashboardAturanController akan menampilkan data tersebut ke halaman *dashboard* aturan.



Gambar 3.47. Sequence Diagram Dashboard Aturan

3.10.18 Dashboard Tambah Aturan

Pada Gambar 3.48 merupakan gambaran interaksi antara objek dari halaman *dashboard* tambah aturan. Ketika pengguna mengakses halaman *dashboard* tambah aturan, pengguna dapat mengisi penyakit, gejala, dan nilai kepercayaan. Setelah mengisi penyakit, gejala, dan nilai kepercayaan, data tersebut akan dikirim ke *class* *DashboardAturanController*. Pada *class* *DashboardAturanController* setelah menerima data dari halaman *dashboard* tambah aturan akan melakukan pengecekan apakah penyakit, gejala, dan nilai kepercayaan sudah terisi. Jika pengecekan tidak sesuai, maka *class* *DashboardAturanController* akan mengirimkan pesan *error* ke halaman *dashboard* tambah aturan. Setelah melakukan pengecekan, *class* *DashboardAturanController* akan menyimpan data tersebut ke dalam *database*. Setelah menambahkan data ke dalam *database* berhasil, pengguna akan diarahkan ke halaman *dashboard* aturan.

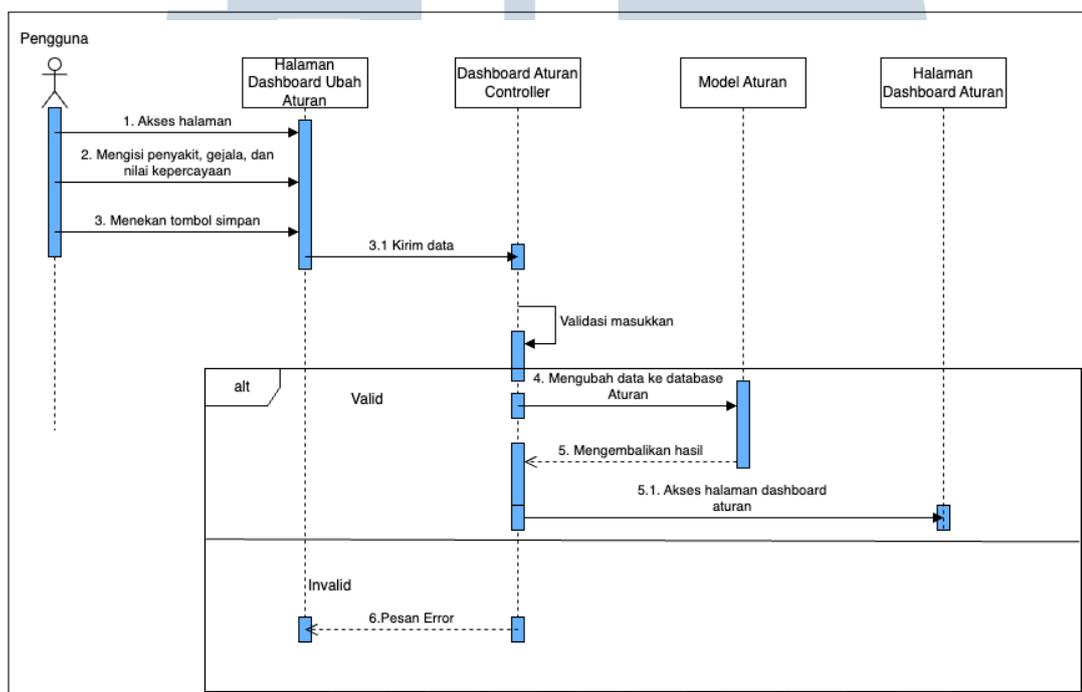


Gambar 3.48. Sequence Diagram Dashboard Tambah Aturan

3.10.19 Dashboard Ubah Aturan

Pada Gambar 3.49 merupakan gambaran interaksi antara objek dari halaman *dashboard* ubah aturan. Ketika pengguna mengakses halaman *dashboard* ubah aturan, pengguna dapat mengubah penyakit, gejala, dan nilai kepercayaan. Setelah mengisi penyakit, gejala, dan nilai kepercayaan, data tersebut akan dikirim ke

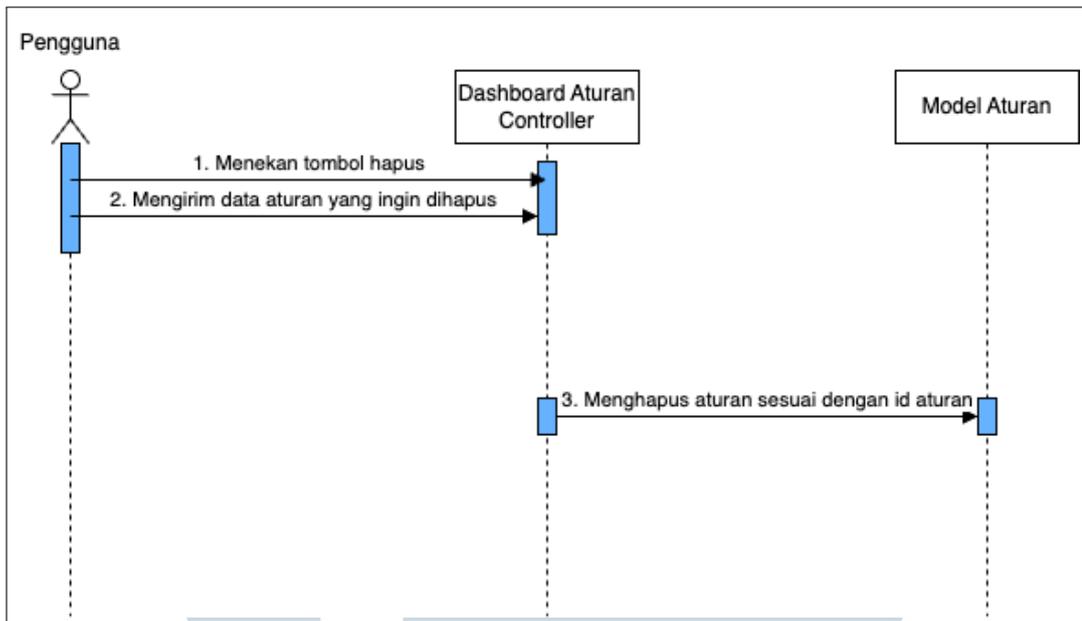
class DashboardAturanController. Pada *class* DashboardAturanController setelah menerima data dari halaman *dashboard* ubah aturan akan melakukan pengecekan apakah penyakit, gejala, dan nilai kepercayaan sudah terisi. Jika pengecekan tidak sesuai, maka *class* DashboardAturanController akan mengirimkan pesan *error* ke halaman *dashboard* ubah aturan. Setelah melakukan pengecekan, *class* DashboardAturanController akan mengubah data lama yang ada pada *database* menjadi data baru. Setelah mengubah data ke dalam *database* berhasil, pengguna akan diarahkan ke halaman *dashboard* aturan.



Gambar 3.49. Sequence Diagram Dashboard Ubah Aturan

3.10.20 Dashboard Hapus Aturan

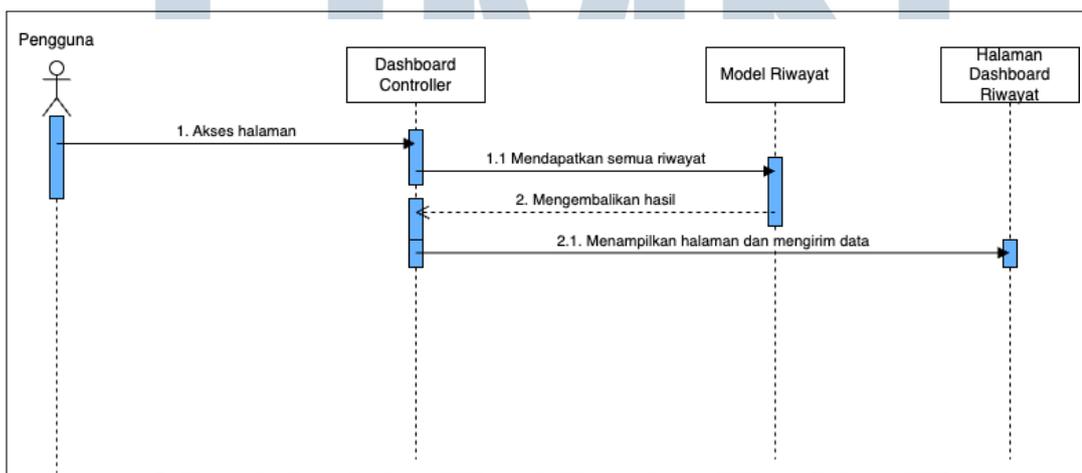
Pada Gambar 3.50 merupakan gambaran interaksi antara objek dari fungsi *dashboard* hapus aturan. ketika pengguna menekan tombol hapus, maka dari halaman *dashboard* aturan akan mengirim data aturan yang ingin dihapus ke *class* DashboardAturanController. Pada *class* DashboardAturanController setelah menerima data dari halaman *dashboard* aturan, dilakukanlah penghapusan aturan sesuai dengan data yang diterima dari halaman *dashboard* aturan pada *database*.



Gambar 3.50. Sequence Diagram Dashboard Hapus Aturan

3.10.21 Dashboard Riwayat

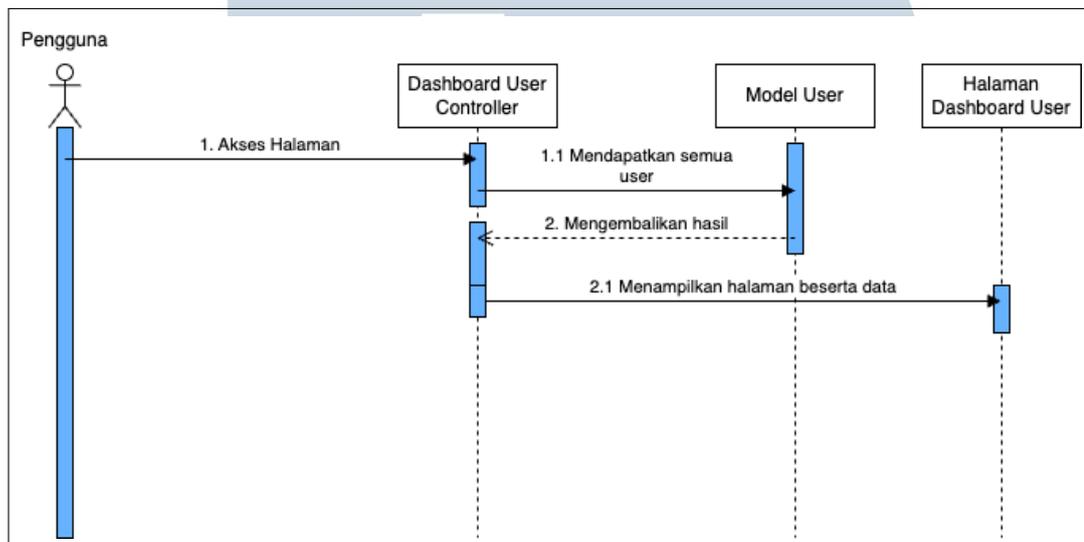
Pada Gambar 3.51 merupakan gambaran interaksi antara objek dari halaman *dashboard* riwayat. Ketika pengguna mengakses halaman *dashboard* riwayat, *class* DashboardController akan mengambil data seluruh riwayat dari database. Setelah mendapatkan data dari database, *class* DashboardController akan menampilkan data tersebut ke halaman *dashboard* riwayat.



Gambar 3.51. Sequence Diagram Dashboard Riwayat

3.10.22 Dashboard User

Pada Gambar 3.52 merupakan gambaran interaksi antara objek dari halaman *dashboard* user. Ketika pengguna mengakses halaman *dashboard* user, *class* *DashboardUserController* akan mengambil data seluruh user dari database. Setelah mendapatkan data dari database, *class* *DashboardUserController* akan menampilkan data tersebut ke halaman *dashboard* user.

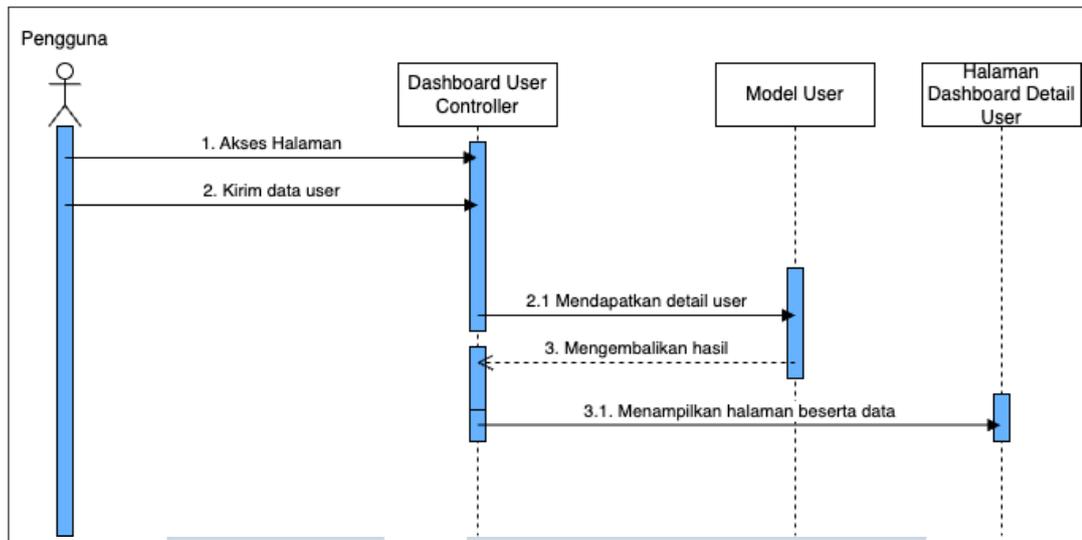


Gambar 3.52. Sequence Diagram Dashboard User

3.10.23 Dashboard Detail User

Pada Gambar 3.53 merupakan gambaran interaksi antara objek dari halaman *dashboard* detail user. Ketika pengguna mengakses halaman *dashboard* detail user beserta mengirim data user yang ingin dilihat detailnya, maka *class* *DashboardUserController* akan mengambil dari *database* detail dari user sesuai dengan data yang dikirim. Setelah mendapatkan data detail user, *class* *DashboardUserController* akan menampilkan data tersebut pada halaman *dashboard* detail user.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

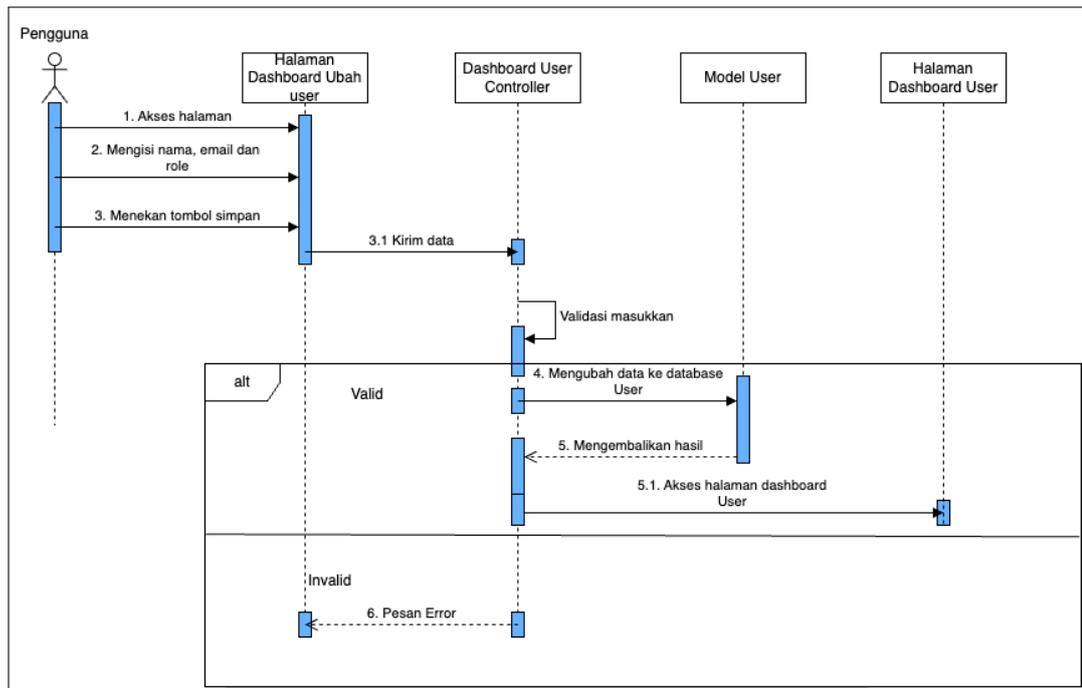


Gambar 3.53. Sequence Diagram Dashboard Detail User

3.10.24 Dashboard Ubah User

Pada Gambar 3.54 merupakan gambaran interaksi antara objek dari halaman *dashboard* ubah user. Ketika pengguna mengakses halaman *dashboard* ubah user, pengguna dapat mengubah nama, email, dan role. Setelah mengisi nama, email, dan role, data tersebut akan dikirim ke *class* `DashboardUserController`. Pada *class* `DashboardUserController` setelah menerima data dari halaman *dashboard* ubah user akan melakukan pengecekan apakah nama, email, dan role sudah terisi. Jika pengecekan tidak sesuai, maka *class* `DashboardUserController` akan mengirimkan pesan *error* ke halaman *dashboard* ubah user. Setelah melakukan pengecekan, *class* `DashboardUserController` akan mengubah data lama yang ada pada *database* menjadi data baru. Setelah mengubah data ke dalam *database* berhasil, pengguna akan diarahkan ke halaman *dashboard* user.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

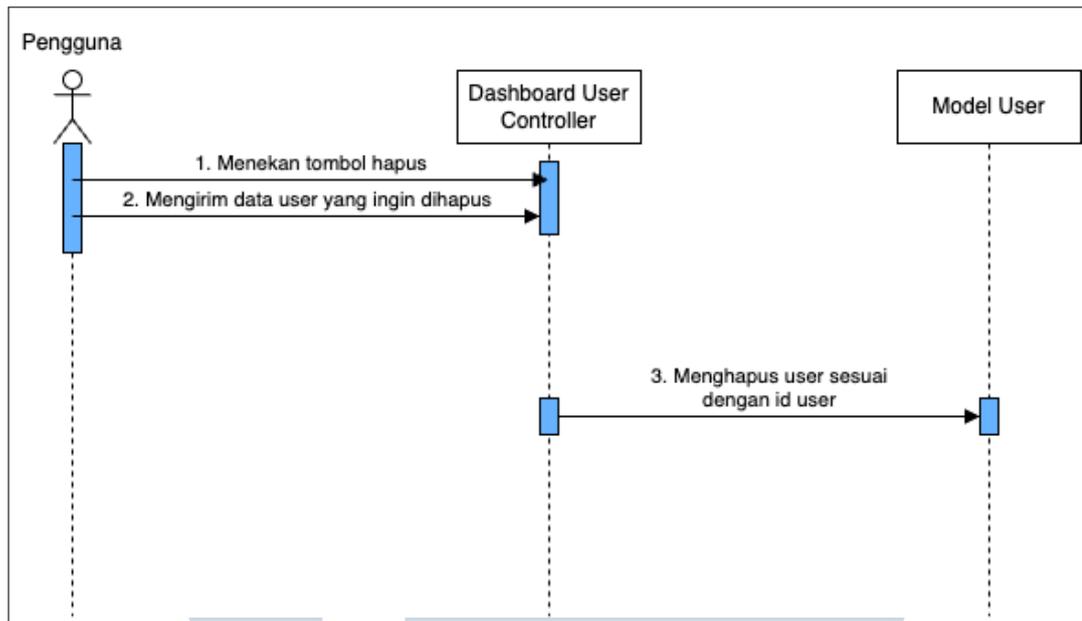


Gambar 3.54. Sequence Diagram Dashboard Ubah User

3.10.25 Dashboard Hapus User

Pada Gambar 3.55 merupakan gambaran interaksi antara objek dari fungsi *dashboard* hapus user. ketika pengguna menekan tombol hapus, maka dari halaman dashboard user akan mengirim data user yang ingin dihapus ke *class* *DashboardUserController*. Pada *class* *DashboardUserController* setelah menerima data dari halaman *dashboard* user, dilakukanlah penghapusan user sesuai dengan data yang diterima dari halaman *dashboard* user pada *database*.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.55. Sequence Diagram Dashboard Hapus User

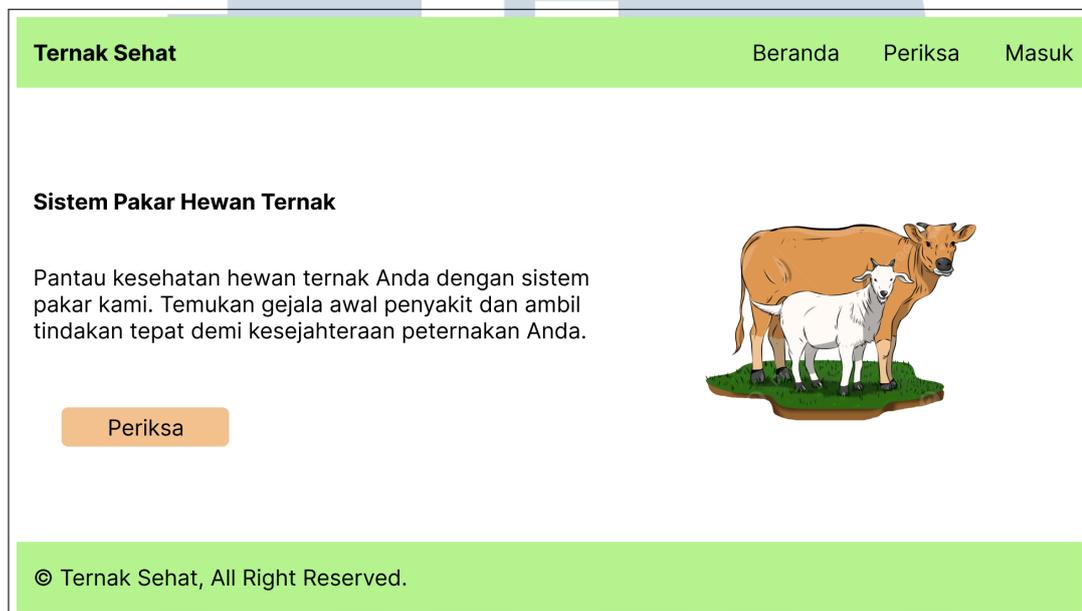
3.11 Class Diagram

Pada Gambar 3.56 merupakan Class Diagram yang ada pada penelitian ini. Class Diagram digunakan untuk menggambarkan hubungan antar *class*. Pada *class* *UserSession* menjadi tumpu atas terhubungnya *class-class* lainnya. Pada *class* *UserSession* memiliki beberapa atribut, yaitu, *id*, *name*, *email*, *password* yang telah dilakukan *hashing*, *is_admin*, *created_at*, dan *updated_at*. Pada *class* *UserSession* juga mempunyai relasi one to many, kepada *class* *PenyakitController*, *class* *DashboardGejalaController*, *class* *DashboardAturanController*, *class* *DashboardPenyakitController*, *class* *RiwayatController*, *class* *DashboardController*, *class* *AuthenticationController*, dan *class* *DashboardUserController*.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

A Beranda

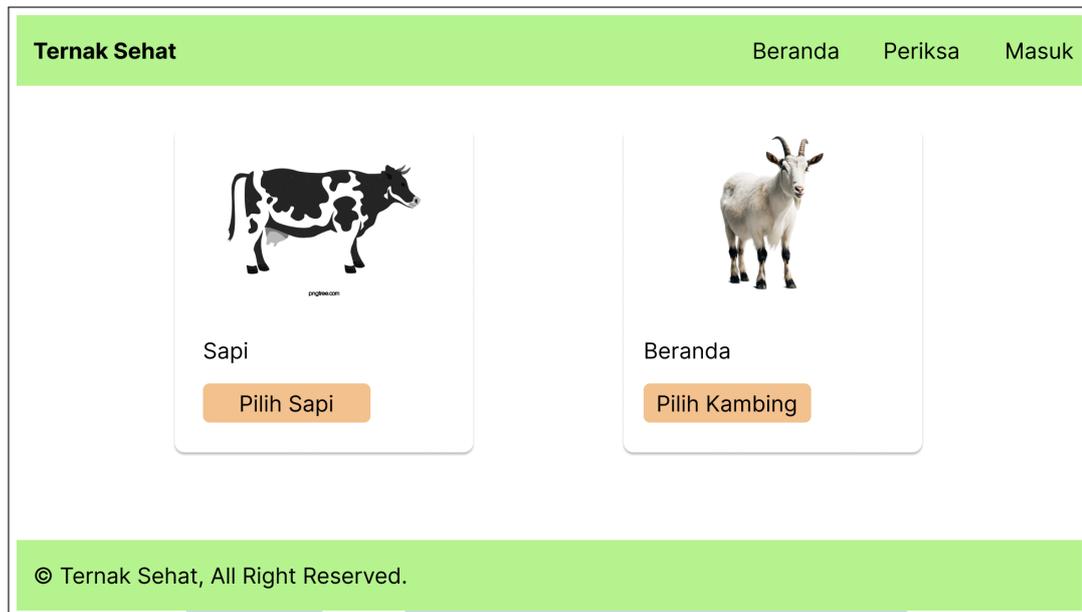
Pada Gambar 3.57 merupakan *design interface* dari halaman beranda. Pada halaman ini terdapat *header* berisikan nama website yang terletak pada bagian kiri dan beberapa fitur pada bagian kanan. pada bagian tengah website menampilkan informasi mengenai manfaat dari penelitian ini. Dari halaman beranda, pengguna dapat melakukan diagnosis pada hewan ternak dengan menekan tombol periksa yang berada di bagian tengah dan tombol periksa pada *header* bagian kanan.



Gambar 3.57. Design Interface User Belum Masuk Beranda

B Pilih Hewan

Pada Gambar 3.58 merupakan *design interface* dari halaman pilih hewan. Pada halaman ini bagian *header* berisikan nama website yang terletak pada bagian kiri dan beberapa fitur pada bagian kanan. pada bagian tengah website menampilkan dua hewan ternak yang pengguna dapat pilih untuk didiagnosis, dengan menekan tombol pilih sapi atau pilih kambing.



Gambar 3.58. Design Interface User Belum Masuk Pilih Hewan

C Periksa

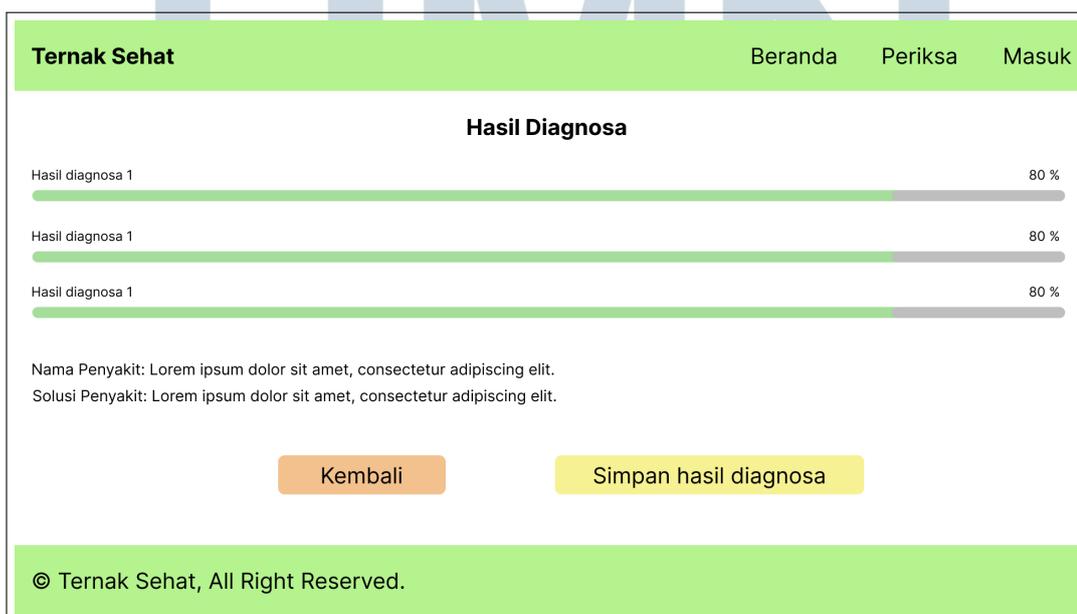
Pada Gambar 3.59 merupakan *design interface* dari halaman periksa ketika memilih gejala. Pada halaman ini bagian *header* berisikan nama website yang terletak pada bagian kiri dan beberapa fitur pada bagian kanan. pada bagian tengah website menampilkan gejala yang pengguna dapat pilih sesuai dengan gejala yang hewan ternaknya alami. Setelah pengguna memilih gejala, pengguna dapat menekan tombol periksa untuk dapat melakukan perhitungan.

UIN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.59. Design Interface User Belum Masuk Pilih Gejala

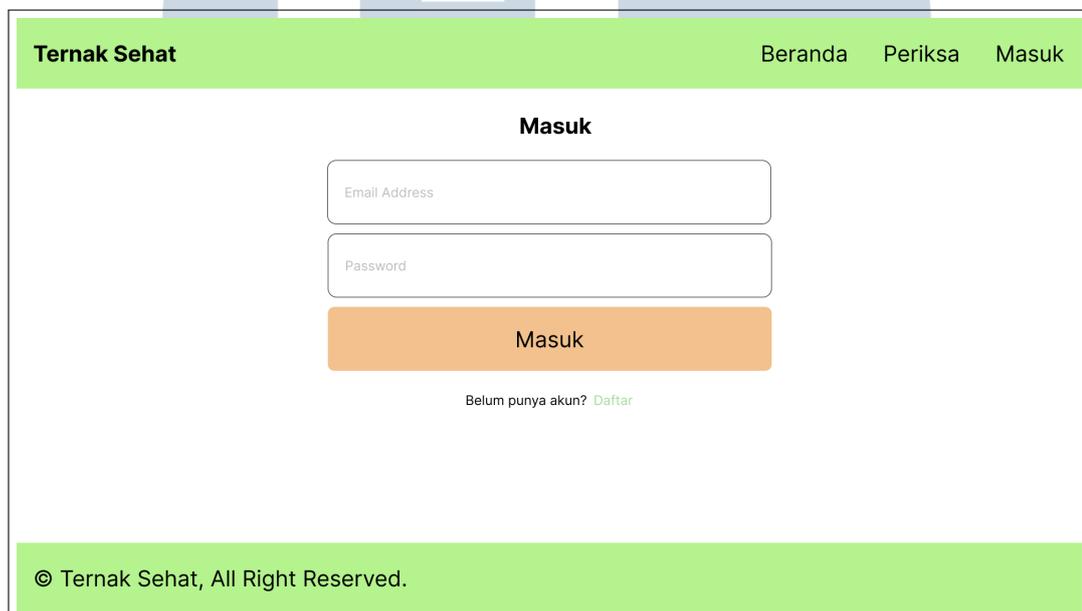
Pada Gambar 3.60 merupakan *design interface* dari halaman periksa ketika menampilkan hasil diagnosis. Pada halaman ini menampilkan hasil diagnosis berupa nilai presentase penyakit dan detail dari penyakit. Pengguna dapat melakukan diagnosis ulang dengan menekan tombol kembali dan tombol periksa pada *header* bagian kanan. Pengguna dapat menyimpan hasil diagnosis dengan cara masuk terlebih dahulu, dengan cara menekan tombol simpan hasil diagnosis atau menekan tombol masuk pada *header* bagian kanan.



Gambar 3.60. Design Interface User Belum Masuk Hasil Diagnosa

D Masuk

Pada Gambar 3.61 merupakan *design interface* dari halaman masuk. Pada halaman ini bagian *header* berisikan nama website yang terletak pada bagian kiri dan beberapa fitur pada bagian kanan. pada bagian tengah website menampilkan beberapa kolom untuk *email* dan kolom untuk *password*. Pada halaman ini pengguna dapat masuk dengan mengisi kolom *email* dan kolom *password*, serta menekan tombol masuk. Di bawah tombol masuk terdapat tombol untuk daftar jika dirasa pengguna belum mempunyai akun.



The image shows a web interface for 'Ternak Sehat'. At the top, there is a green header bar with the text 'Ternak Sehat' on the left and 'Beranda Periksa Masuk' on the right. Below the header, the word 'Masuk' is centered. There are two input fields: 'Email Address' and 'Password'. Below these fields is an orange button labeled 'Masuk'. Underneath the button, there is a link that says 'Belum punya akun? [Daftar](#)'. At the bottom of the page, there is a green footer bar with the text '© Ternak Sehat, All Right Reserved.'

Gambar 3.61. Design Interface User Belum Masuk Halaman Masuk

E Daftar

Pada Gambar 3.62 merupakan *design interface* dari halaman daftar. Pada halaman ini bagian *header* berisikan nama website yang terletak pada bagian kiri dan beberapa fitur pada bagian kanan. pada bagian tengah website menampilkan beberapa kolom untuk nama, kolom untuk *email* dan kolom untuk *password*. Pada halaman ini pengguna dapat daftar dengan mengisi kolom nama, kolom *email* dan kolom *password*, serta menekan tombol daftar. Di bawah tombol daftar terdapat tombol untuk masuk jika dirasa pengguna sudah mempunyai akun.

Gambar 3.62. Design Interface User Belum Masuk Halaman Daftar

3.12.2 User Sudah Masuk

Pada bagian ini merupakan *design interface* dari tampilan website, dengan kategori user sudah masuk.

A Riwayat

Pada Gambar 3.63 merupakan *design interface* dari halaman riwayat. Pada halaman ini bagian *header* berisikan nama website yang terletak pada bagian kiri dan beberapa fitur pada bagian kanan. pada bagian tengah website menampilkan riwayat dari hasil diagnosis yang telah dilakukan sebelumnya.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.63. Design Interface User Sudah Masuk Riwayat

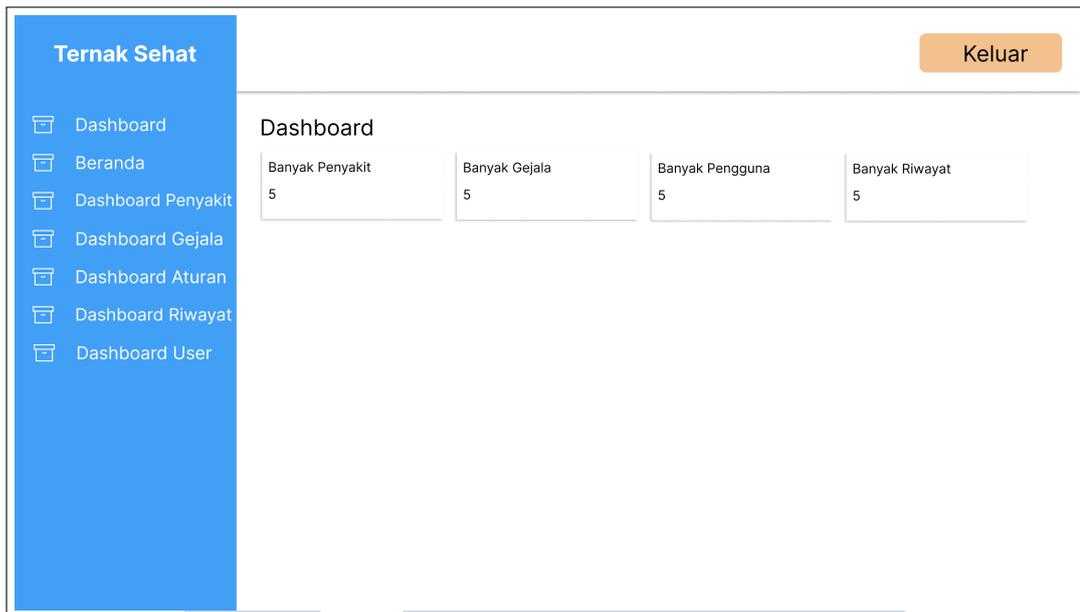
3.12.3 User Admin

Pada bagian ini merupakan *design interface* dari tampilan website, dengan kategori user admin.

A Dashboard

Pada Gambar 3.64 merupakan *design interface* dari halaman *dashboard*. Pada halaman ini bagian sidebar menampilkan nama website dan menampilkan beberapa fitur. Pada halaman ini menampilkan bayaknya penyakit, banyaknya gejala, banyaknya penggunda, dan banyaknya riwayat yang ada pada website.

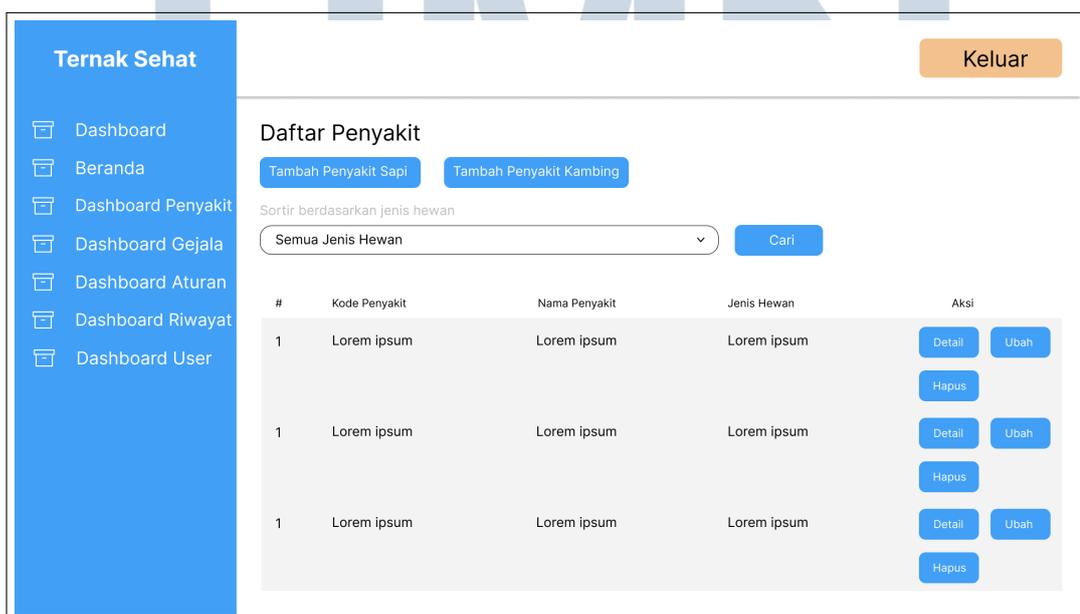
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.64. Design Interface User Admin Dashboard

B Dashboard Penyakit

Pada Gambar 3.65 merupakan *design interface* dari halaman *dashboard* penyakit. Pada halaman ini bagian sidebar menampilkan nama website dan menampilkan beberapa fitur. Pada halaman ini menampilkan semua penyakit yang ada, serta tombol-tombol untuk tambah penyakit, melihat detail penyakit, ubah penyakit, dan hapus penyakit.



Gambar 3.65. Design Interface User Admin Dashboard Penyakit

C Dashboard Tambah Penyakit

Pada Gambar 3.66 merupakan *design interface* dari halaman *dashboard* tambah penyakit. Pada halaman ini bagian sidebar menampilkan nama website dan menampilkan beberapa fitur. Pada halaman ini menampilkan beberapa kolom yang dapat diisi, seperti, kolom nama penyakit, kolom penjelasan penyakit, kolom solusi penyakit, dan kolom gambar penyakit.

The screenshot shows the 'Tambah Penyakit' form in the 'Ternak Sehat' admin dashboard. The form is titled 'Tambah Penyakit' and is located in the main content area. On the left, there is a blue sidebar with the 'Ternak Sehat' logo and a list of navigation items: Dashboard, Beranda, Dashboard Penyakit, Dashboard Gejala, Dashboard Aturan, Dashboard Riwayat, and Dashboard User. In the top right corner of the main area, there is an orange 'Keluar' button. The form itself contains the following fields and values:

Kode Penyakit	PS001
Nama Penyakit	<input type="text"/>
Penjelasan Penyakit	<input type="text"/>
Solusi Penyakit	<input type="text"/>
Gambar Penyakit	<input type="text"/>
Jenis Hewan	Sapi

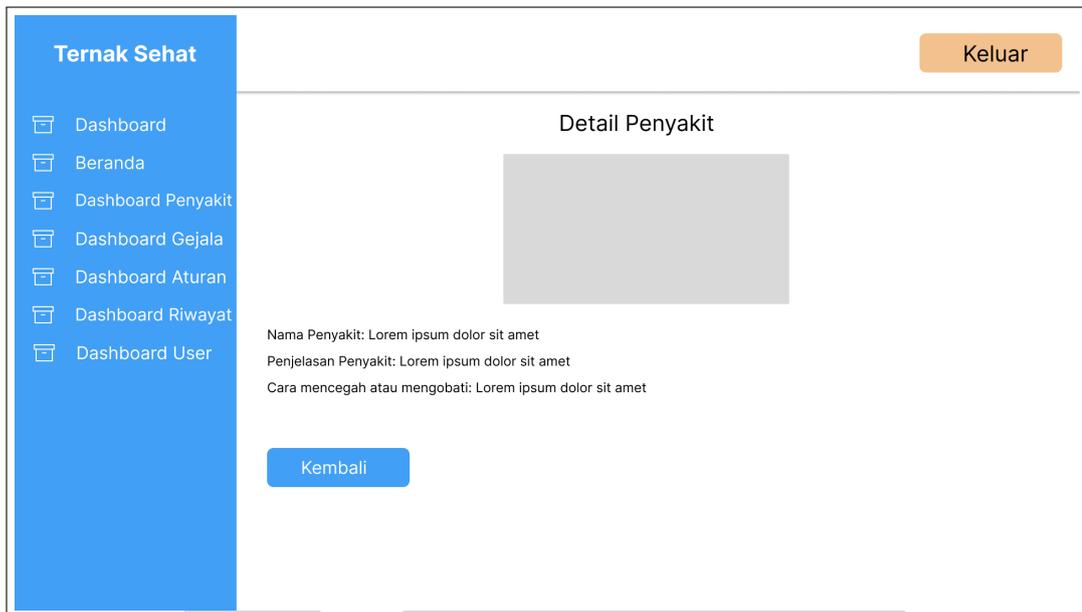
At the bottom center of the form, there is a blue 'Simpan' button.

Gambar 3.66. Design Interface User Admin Dashboard Tambah Penyakit

D Dashboard Detail Penyakit

Pada Gambar 3.67 merupakan *design interface* dari halaman *dashboard* detail penyakit. Pada halaman ini bagian sidebar menampilkan nama website dan menampilkan beberapa fitur. Pada halaman ini menampilkan detail dari suatu penyakit.

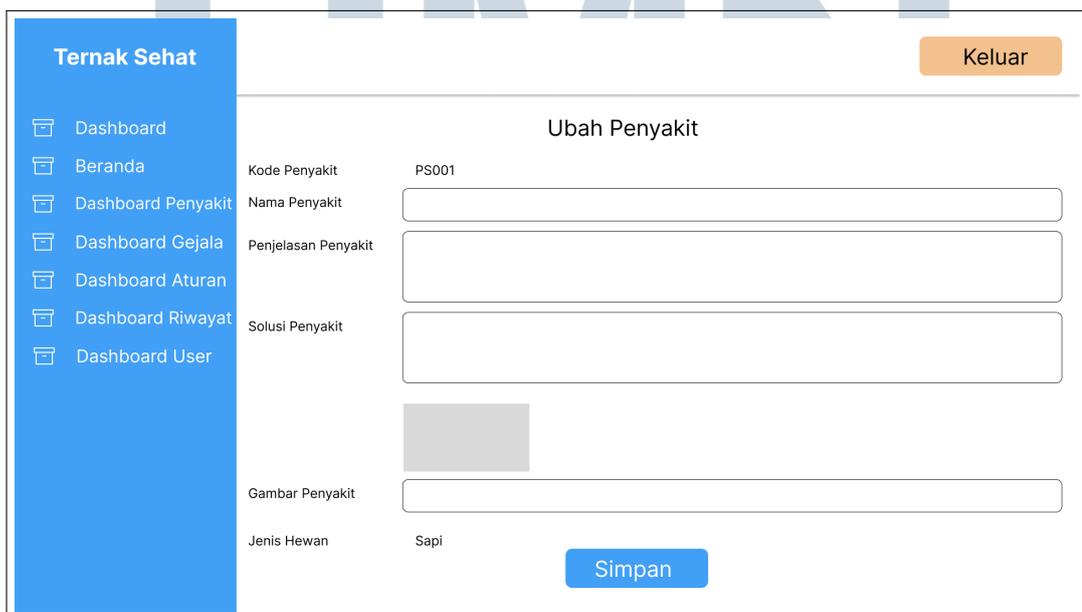
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.67. Design Interface User Admin Dashboard Detail Penyakit

E Dashboard Ubah Penyakit

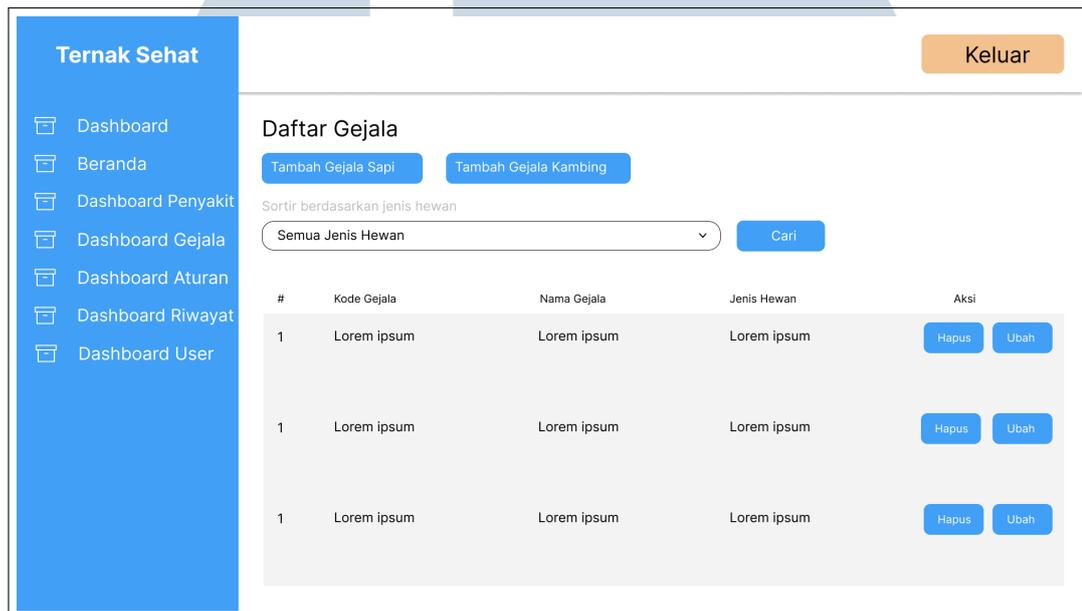
Pada Gambar 3.68 merupakan *design interface* dari halaman *dashboard* ubah penyakit. Pada halaman ini bagian sidebar menampilkan nama website dan menampilkan beberapa fitur. Pada halaman ini menampilkan beberapa kolom yang dapat diganti isinya, seperti, kolom nama penyakit, kolom penjelasan penyakit, kolom solusi penyakit, dan kolom gambar penyakit.



Gambar 3.68. Design Interface User Admin Dashboard Ubah Penyakit

F Dashboard Gejala

Pada Gambar 3.69 merupakan *design interface* dari halaman *dashboard* gejala. Pada halaman ini bagian sidebar menampilkan nama website dan menampilkan beberapa fitur. Pada halaman ini menampilkan semua gejala yang ada, serta tombol-tombol untuk tambah gejala, ubah gejala, dan hapus gejala.



Gambar 3.69. Design Interface User Admin Dashboard Gejala

G Dashboard Tambah Gejala

Pada Gambar 3.70 merupakan *design interface* dari halaman *dashboard* tambah gejala. Pada halaman ini bagian sidebar menampilkan nama website dan menampilkan beberapa fitur. Pada halaman ini menampilkan kolom nama gejala yang dapat diisi.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Gambar 3.70. Design Interface User Admin Dashboard Tambah Gejala

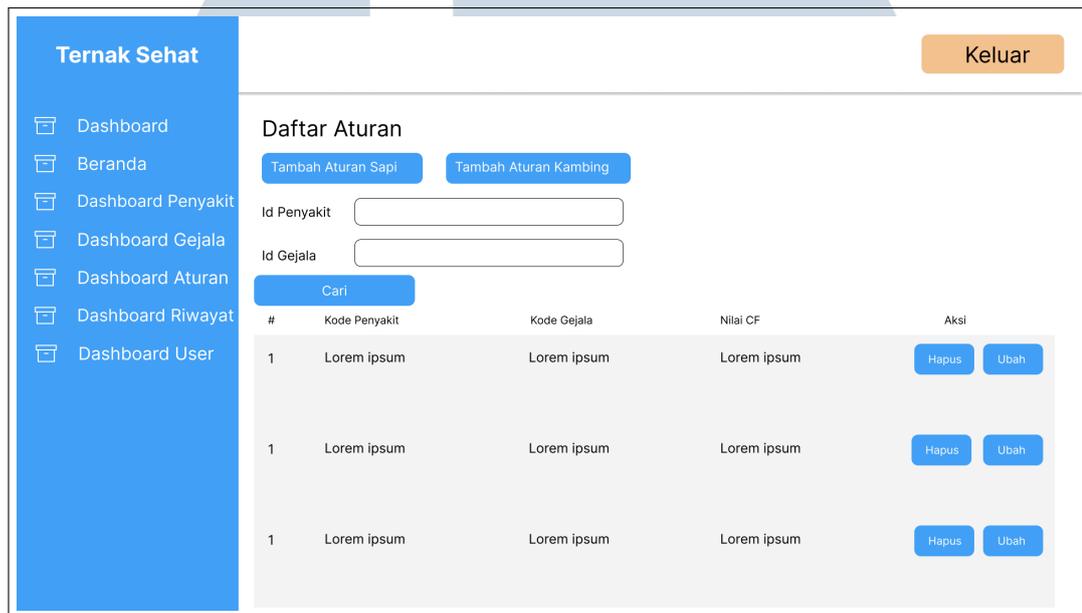
H Dashboard Ubah Gejala

Pada Gambar 3.71 merupakan *design interface* dari halaman *dashboard* ubah gejala. Pada halaman ini bagian sidebar menampilkan nama website dan menampilkan beberapa fitur. Pada halaman ini menampilkan kolom nama gejala yang dapat diubah.

Gambar 3.71. Design Interface User Admin Dashboard Ubah Gejala

I Dashboard Aturan

Pada Gambar 3.72 merupakan *design interface* dari halaman *dashboard* aturan. Pada halaman ini bagian sidebar menampilkan nama website dan menampilkan beberapa fitur. Pada halaman ini menampilkan semua aturan yang ada, serta tombol-tombol untuk tambah aturan, ubah aturan, dan hapus aturan.



Gambar 3.72. Design Interface User Admin Dashboard Aturan

J Dashboard Tambah Aturan

Pada Gambar 3.73 merupakan *design interface* dari halaman *dashboard* tambah aturan. Pada halaman ini bagian sidebar menampilkan nama website dan menampilkan beberapa fitur. Pada halaman ini menampilkan beberapa kolom yang dapat diisi, seperti, kolom penyakit, kolom gejala, dan kolom nilai kepercayaan.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Gambar 3.73. Design Interface User Admin Dashboard Tambah Aturan

K Dashboard Ubah Aturan

Pada Gambar 3.74 merupakan *design interface* dari halaman *dashboard* ubah aturan. Pada halaman ini bagian sidebar menampilkan nama website dan menampilkan beberapa fitur. Pada halaman ini menampilkan beberapa kolom yang dapat diubah, seperti, kolom penyakit, kolom gejala, dan kolom nilai kepercayaan.

Gambar 3.74. Design Interface User Admin Dashboard Ubah Aturan

L Dashboard Riwayat

Pada Gambar 3.75 merupakan *design interface* dari halaman *dashboard* riwayat. Pada halaman ini bagian sidebar menampilkan nama website dan menampilkan beberapa fitur. Pada halaman ini menampilkan seluruh riwayat hasil diagnosis yang pernah dilakukan.

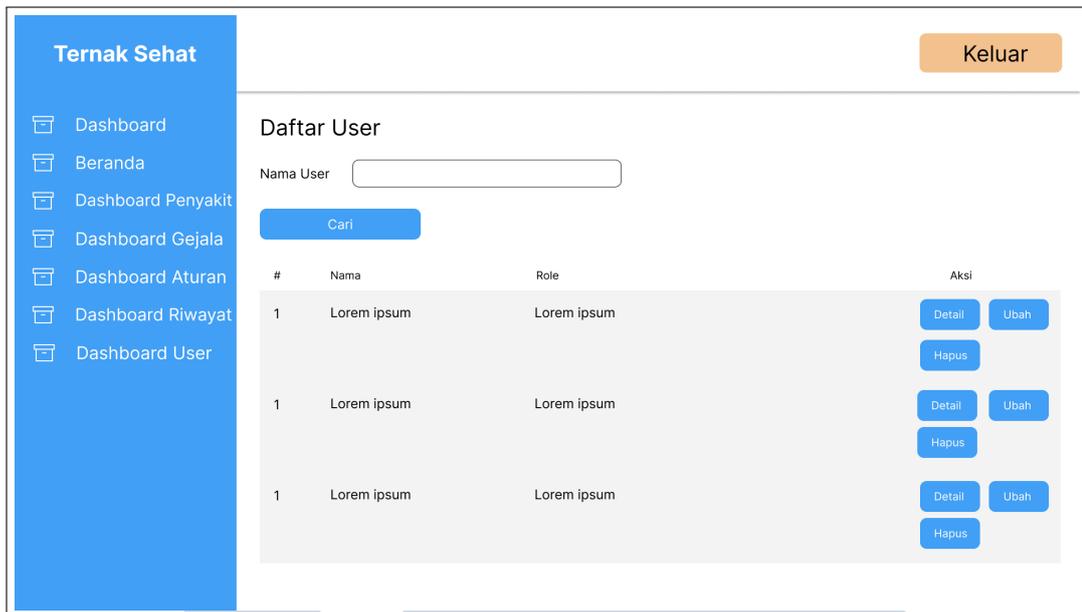


Gambar 3.75. Design Interface User Admin Dashboard Riwayat

M Dashboard User

Pada Gambar 3.76 merupakan *design interface* dari halaman *dashboard* user. Pada halaman ini bagian sidebar menampilkan nama website dan menampilkan beberapa fitur. Pada halaman ini menampilkan semua user yang ada, serta tombol-tombol untuk melihat detail user, ubah user, dan hapus user.

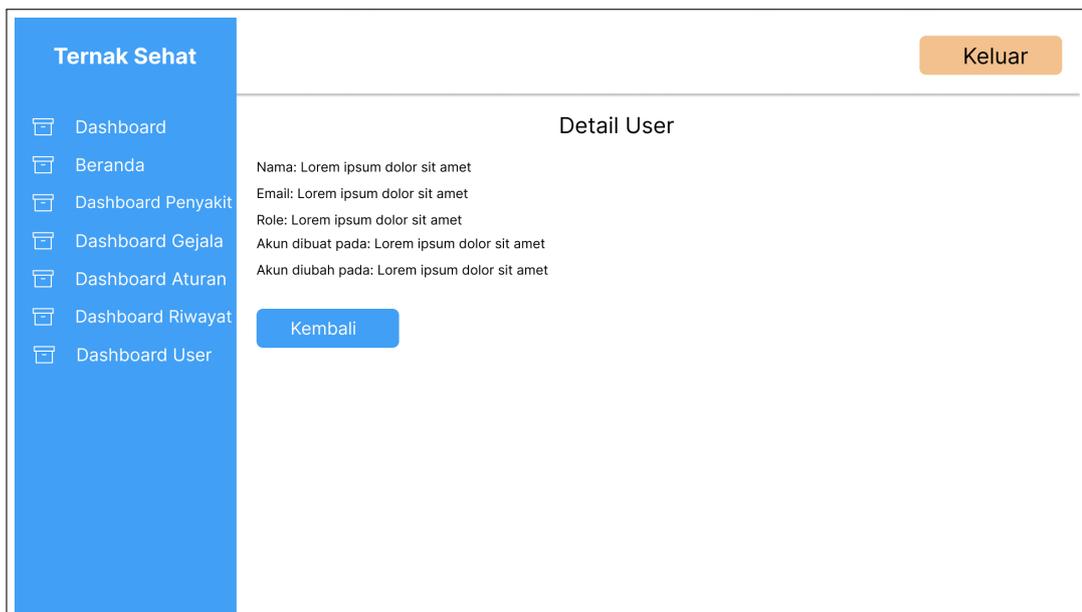
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.76. Design Interface User Admin Dashboard User

N Dashboard Detail User

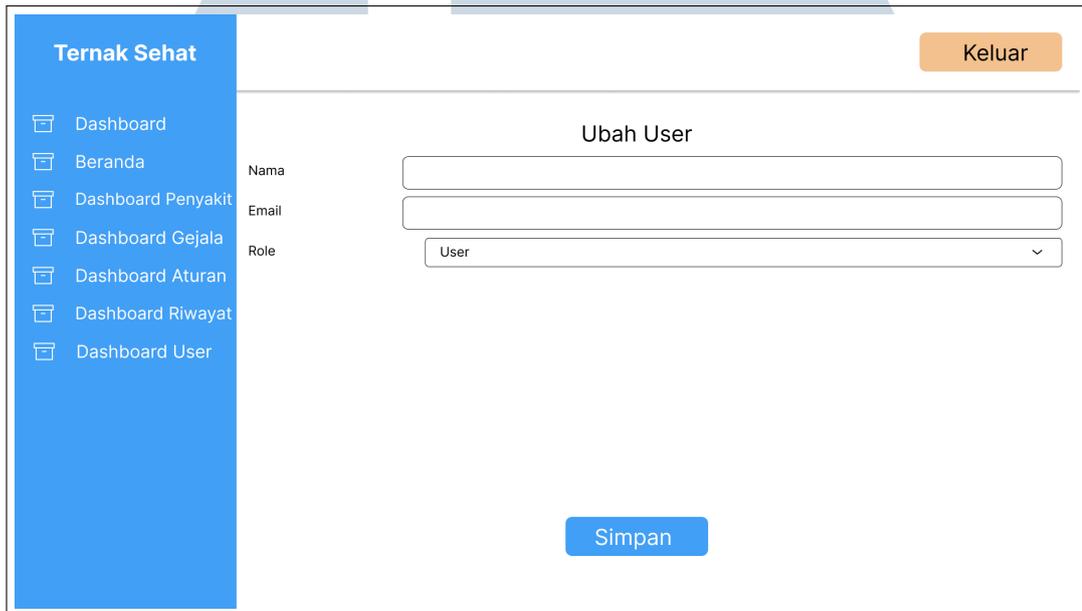
Pada Gambar 3.77 merupakan *design interface* dari halaman *dashboard* detail user. Pada halaman ini bagian sidebar menampilkan nama website dan menampilkan beberapa fitur. Pada halaman ini menampilkan detail dari suatu user.



Gambar 3.77. Design Interface User Admin Dashboard Detail User

O Dashboard Ubah User

Pada Gambar 3.78 merupakan *design interface* dari halaman *dashboard* ubah user. Pada halaman ini bagian sidebar menampilkan nama website dan menampilkan beberapa fitur. Pada halaman ini menampilkan beberapa kolom yang dapat diubah, seperti, kolom nama, kolom *email*, dan kolom *role*.



The image shows a web interface for changing a user. On the left is a blue sidebar with the text "Ternak Sehat" and a list of menu items: Dashboard, Beranda, Dashboard Penyakit, Dashboard Gejala, Dashboard Aturan, Dashboard Riwayat, and Dashboard User. In the top right corner of the main content area is an orange "Keluar" button. The main content area is titled "Ubah User" and contains three form fields: "Nama" (Name) with an empty text input, "Email" with an empty text input, and "Role" with a dropdown menu showing "User". At the bottom center of the form is a blue "Simpan" button.

Gambar 3.78. Design Interface User Admin Dashboard Ubah User

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA