

BAB IV HASIL DAN ANALISIS PENELITIAN

4.1 Analisa Perancangan Sistem

Analisa perancangan sistem yang didapatkan berdasarkan studi literatur yang telah dilakukan dan didukung dari hasil analisis sistem yang berjalan meliputi:

1. Analisis permasalahan sistem berjalan,
2. Analisis pemecahan masalah,
3. Analisis usulan sistem, dan
4. Analisis kebutuhan sistem

Penulis menyimpulkan dan memberikan sebuah solusi yaitu perancangan aplikasi berbasis web yang dapat melakukan pemindaian kerentanan keamanan *website* yang dimiliki oleh pemilik/orang yang bertanggung jawab pada pengembangan suatu aplikasi *website* dan dapat memberikan sebuah laporan hasil audit terhadap kerentanan *website* tersebut berupa Excel.

Dan dari sekian alat pemindaian yang telah dievaluasi oleh WAVSEP (*Web Application Vulnerability Scanner Evaluation Project*) serta diketahui secara resmi pada halaman website OWASP (*Open Web Application Security Project*) sebagai salah satu didalam daftar *Vulnerability Scanning Tools*, penulis memilih *tools* bernama Nuclei dikarenakan proyeknya memiliki kelebihan seperti kontribusi komunitas yang banyak dikarenakan bersifat *Open Source* dan komabilitasnya sangat bagus.

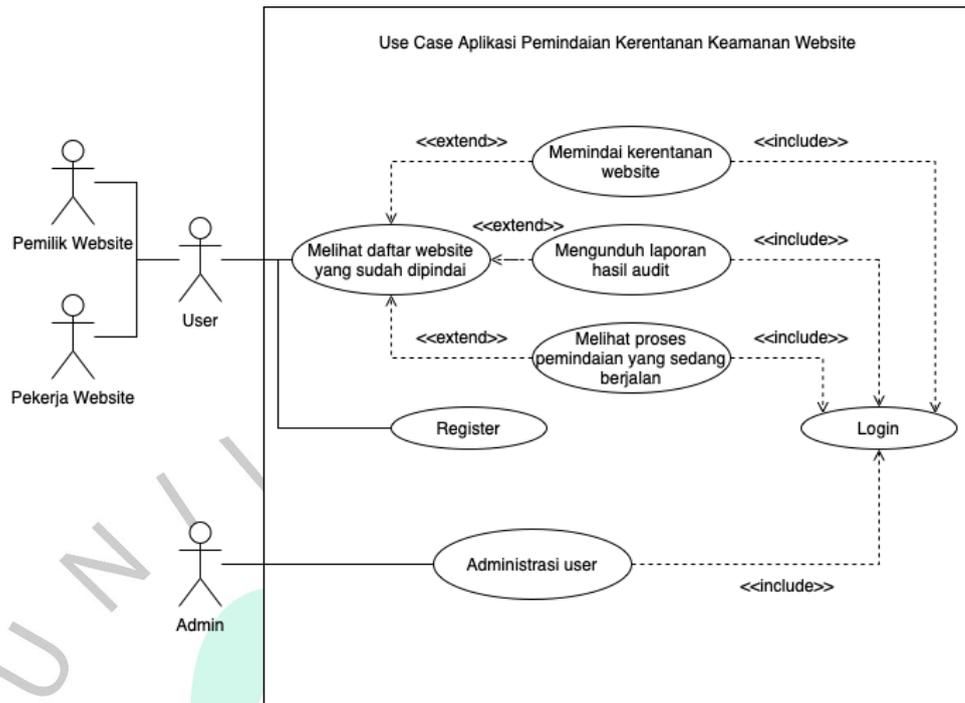
Oleh karena kelebihan yang Nuclei miliki, Nuclei dapat ditanamkan didalam *service Backend* Python yang menjadi *engine* utama sebagai pemindaian dalam perancangan aplikasi web penyedia pemindaian kerentanan keamanan *website* ini.

4.2 Perancangan Diagram Sistem Usulan

Dalam merancang sistem usulan berdasarkan dari hasil analisis penelitian perancangan sistem yang telah dilakukan, akan dijabarkan secara satu-persatu dengan menggunakan diagram UML (*Unified Modeling Language*) antara lain dengan *Use Case Diagram*, Spesifikasi *Use Case Diagram*, *Class Diagram*, *Activity Diagram*, dan *Sequence Diagram*.

4.2.1 Use Case Diagram

Use Case Diagram adalah sebuah jenis diagram yang menjelaskan mengenai apa yang dapat dilakukan oleh perancangan aplikasi tersebut. Diagram ini lebih melihat apa yang bisa dilakukan oleh aplikasi tersebut, dan siapa saja yang bisa menggunakan aplikasi tersebut. *Use Case* tersebut dapat dilihat pada **Gambar 4.1** sebagai berikut.



Gambar 4.1 Use Case Diagram Aplikasi Pemindaian Kerentanan Keamanan Website

Pada diagram diatas, pemilik website dan pekerja *website* menjadi user yang dapat dilihat tidak ada perbedaan role antara peran ini. Semua bisa menjadi user dan menggunakan semua fitur yang ada.

Dalam perancangan ini, untuk user yang sudah melakukan *login*, mempunyai 4 akses antara lain:

1. Melihat daftar *website* yang sudah di pindai.
2. Melakukan pemindaian baru terhadap kerentanan keamanan *website* dari url yang dimasukkan.
3. Melihat proses pemindaian yang sedang berjalan.
4. Mengunduh laporan hasil audit dari pemindaian kerentanan keamanan *website* yang telah selesai dipindai.

Dan user dapat melakukan registrasi secara mandiri tanpa didaftarkan oleh seorang administrator. Admin dapat melakukan manajemen akun user antara lain:

1. Menambahkan user baru.
2. Mengubah data user yang sudah ada.
3. Dan dapat menghapus user.

4.2.2 Spesifikasi Use Case Diagram

Spesifikasi *use case diagram* adalah sebuah penjelasan secara detail dari semua kegiatan yang ada di *use case diagram* tersebut. Spesifikasi tersebut digunakan agar peneliti dapat memahami bagaimana kegiatan tersebut.

Pada **Gambar 4.1**, *use case* yang dibuat oleh peneliti ada sepuluh kegiatan yang terjadi dalam aplikasi tersebut. Kegiatan tersebut akan dijelaskan secara detail tentang bagaimana kegiatan tersebut bekerja.

a. *Login*

Proses *login* tersebut dilakukan untuk masuk kedalam dashboard aplikasi tersebut. Proses tersebut dimulai dari user memasukkan *username* dan *password*.

Spesifikasi dari kegiatan *login* dapat dilihat pada **Tabel 4.1** dibawah.

Tabel 4.1 Spesifikasi *use case login*

Use Case Name	Login	
Actors	User	
Triggers	User menekan tombol <i>login</i>	
Precondition	User harus memiliki akun	
Post Condition	User berhasil masuk ke dalam dashboard aplikasi pemindaian	
Success Scenario	Actor	System
	1. User membuka halaman utama aplikasi web pemindaian.	
	2. User mengakses halaman <i>login</i> .	3. Sistem menampilkan form <i>login</i> .
	4. User mengisi form <i>login</i> .	5. Sistem memvalidasi input yang dikirim user
		6. Jika data yang dimasukkan user tidak sesuai, maka menampilkan pesan <i>username/password salah</i> .
		7. Jika sesuai, maka akan masuk ke <i>dashboard</i> .
		8. Selesai.

b. Registrasi

Proses registrasi diperlukan untuk membuat akun agar dapat menyimpan data user terkait data pemindaian. Proses tersebut dimlau dari user memasukkan *username*, *email*, dan *password*.

Spesifikasi dari kegiatan Registrasi dapat dilihat pada **Tabel 4.2** dibawah ini.

Tabel 4.2 Spesifikasi *use case* Registrasi

Use Case Name	Registrasi
Actors	User
Triggers	User menekan tombol register
Precondition	Tidak ada

Post Condition	User berhasil terdaftar dan masuk ke dalam aplikasi pemindaian	
Success Scenario	Actor	System
	1. User membuka halaman utama aplikasi web pemindaian.	
	2. User mengakses halaman registrasi.	3. Sistem menampilkan form registrasi.
	4. User mengisi form registrasi.	5. Sistem memvalidasi input yang dikirim user.
		6. Jika data sudah terdaftar, maka menampilkan akun sudah terdaftar.
		7. Jika sesuai, maka akan masuk ke dashboard.
		8. Selesai.

c. Memindai kerentanan website

Dalam proses ini, user dapat memindai kerentanan keamanan website dengan memasukkan web url yang ingin dipindai.

Spesifikasi dari kegiatan memindai kerentanan website dapat dilihat pada **Tabel 4.3** dibawah ini.

Tabel 4.3 Spesifikasi *use case* Memindai kerentanan *website*

Use Case Name	Memindai kerentanan website	
Actors	User	
Triggers	User ingin melakukan pemindaian kerentanan keamanan website	
Precondition	User harus berhasil melakukan <i>login</i>	
Post Condition	User berhasil masuk ke dalam aplikasi pemindaian	
Success Scenario	Actor	System
	1. User mengakses dashboard aplikasi.	2. Sistem menampilkan form url website yang ingin dipindai.
	3. User mengisi form url website yang ingin dipindai.	4. Sistem akan mulai memindai url website yang user input.
	6. User mendapatkan informasi atas proses pemindaian website yang dipindai.	5. Sistem menampilkan pembaruan proses pemindai.
	7. Selesai.	

d. Melihat proses pemindaian yang sedang berjalan

Setelah user melakukan proses pemindaian kerentanan website, proses ini menjadi proses kelanjutannya yaitu proses melihat *progress* pemindaian yang sedang berjalan.

Spesifikasi dari kegiatan melihat proses pemindaian yang sedang berjalan dapat dilihat pada **Tabel 4.4** dibawah ini.

Tabel 4.4 Spesifikasi *use case* Melihat proses pemindaian yang sedang berjalan

Use Case Name	Melihat proses pemindaian yang sedang berjalan	
Actors	User	
Triggers	User telah melakukan pemindaian	
Precondition	User harus berhasil melakukan pemindaian	
Post Condition	User berhasil melihat status proses pemindaian	
Success Scenario	Actor	System
	1. User baru saja melakukan pemindaian baru.	2. Sistem menampilkan status pembaruan proses pemindai.
	3. User mendapatkan informasi atas proses pemindaian website yang dipindai.	
	4. Selesai.	

e. Mengunduh laporan hasil audit

Proses ini terjadi ketika sebuah proses pemindaian telah sukses selesai. User dapat mengunduh laporan hasil audit ini dengan bentuk keluaran file excel di halaman dashboard aplikasi.

Spesifikasi dari kegiatan mengunduh laporan hasil audit dapat dilihat pada **Tabel 4.5** dibawah ini.

Tabel 4.5 Spesifikasi *use case* Mengunduh laporan hasil audit

Use Case Name	Mengunduh laporan hasil audit	
Actors	User	
Triggers	User menekan tombol download laporan	
Precondition	User harus memiliki website yang telah dipindai	
Post Condition	User berhasil berhasil mengunduh laporan hasil audit berupa excel	
Success Scenario	Actor	System
	1. User mengakses dashboard aplikasi.	2. Sistem menampilkan daftar website yang telah dipindai.
	3. User melihat daftar website yang telah dipindai.	

4. User menekan tombol download dari salah satu daftar website yang telah dipindai.	5. Sistem akan menyediakan permintaan laporan hasil audit.
6. User berhasil menyimpan laporan ke dalam sistem lokal.	
7. Selesai.	

f. Melihat daftar website yang telah dipindai

Proses ini terjadi ketika user mengakses halaman dashboard aplikasi dan tampilan yang dimuat ialah daftar website yang pernah dipindai sebelumnya.

Spesifikasi dari kegiatan melihat daftar website yang telah dipindai dapat dilihat pada **Tabel 4.6** dibawah ini.

Tabel 4.6 Spesifikasi *use case* Melihat daftar *website* yang telah dipindai

Use Case Name	Melihat daftar website yang telah dipindai	
Actors	User	
Triggers	User masuk ke dashboard aplikasi	
Precondition	User harus berhasil melakukan <i>login</i>	
Post Condition	User dapat melihat daftar website yang telah dipindai	
Success Scenario	Actor	System
	1. User mengakses dashboard aplikasi.	2. Sistem menampilkan daftar website yang telah dipindai.
	3. User melihat daftar website yang telah dipindai.	
	4. Selesai.	

g. Melihat user

Dalam proses ini, seorang Admin dapat melihat daftar user yang telah terdaftar dalam aplikasi ini.

Spesifikasi dari kegiatan melihat user dapat dilihat pada **Tabel 4.7** dibawah ini.

Tabel 4.7 Spesifikasi *use case* Melihat *user*

Use Case Name	Melihat user	
Actors	Admin	
Triggers	Admin ingin melihat daftar user	
Precondition	Admin harus berhasil melakukan <i>login</i>	
Post Condition	Admin dapat melihat daftar user yang terdaftar	
Success Scenario	Actor	System

1. Admin mengakses daftar user yang terdaftar.	2. Sistem menampilkan daftar user yang telah terdaftar.
3. Admin melihat daftar user yang terdaftar.	
4. Selesai.	

h. Menambahkan user

Dalam proses ini, seorang Admin dapat menambahkan user ke dalam aplikasi secara manual.

Spesifikasi dari kegiatan menambahkan user dapat dilihat pada **Tabel 4.8** dibawah ini.

Tabel 4.8 Spesifikasi *use case* Menambahkan user

Use Case Name	Menambahkan user	
Actors	Admin	
Triggers	Admin ingin mendaftarkan user	
Precondition	Admin harus berhasil melakukan <i>login</i>	
Post Condition	Admin berhasil mendaftarkan user	
Success Scenario	Actor	System
	1. Admin mengakses menambahkan user.	2. Sistem menampilkan form menambahkan user.
	3. Admin mengisi form menambahkan user.	4. Sistem menerima input form menambahkan user.
		5. Jika data belum sesuai, maka menampilkan error.
		6. Jika semua data telah sesuai, maka menampilkan berhasil menambahkan user.
		7. Selesai.

i. Mengubah user

Dalam proses ini, seorang Admin dapat mengubah/melakukan perubahan atribut dari suatu user yang terdaftar.

Spesifikasi dari kegiatan mengubah user dapat dilihat pada **Tabel 4.9** dibawah ini.

Tabel 4.9 Spesifikasi *use case* Mengubah user

Use Case Name	Mengubah user
Actors	Admin
Triggers	Admin ingin mengubah data user

Precondition	Admin harus berhasil melakukan <i>login</i>	
Post Condition	Admin berhasil mengubah data user	
Success Scenario	Actor	System
	1. Admin mengakses daftar user yang terdaftar.	2. Sistem menampilkan daftar user yang telah terdaftar.
	3. Admin memilih user yang ingin diubah.	4. Sistem menampilkan sebuah form ubah user.
	5. Admin mengisi form ubah user.	6. Sistem menerima input form ubah user.
		7. Jika data belum sesuai, maka menampilkan error.
		8. Jika semua data telah sesuai, maka menampilkan berhasil mengubah user.
		9. Selesai.

j. • Menghapus user

Dalam proses ini, seorang Admin dapat menghapus akun user yang terdaftar didalam aplikasi ini.

Spesifikasi dari kegiatan menghapus user dapat dilihat pada **Tabel 4.10** dibawah ini.

Tabel 4.10 Spesifikasi *use case* menghapus user

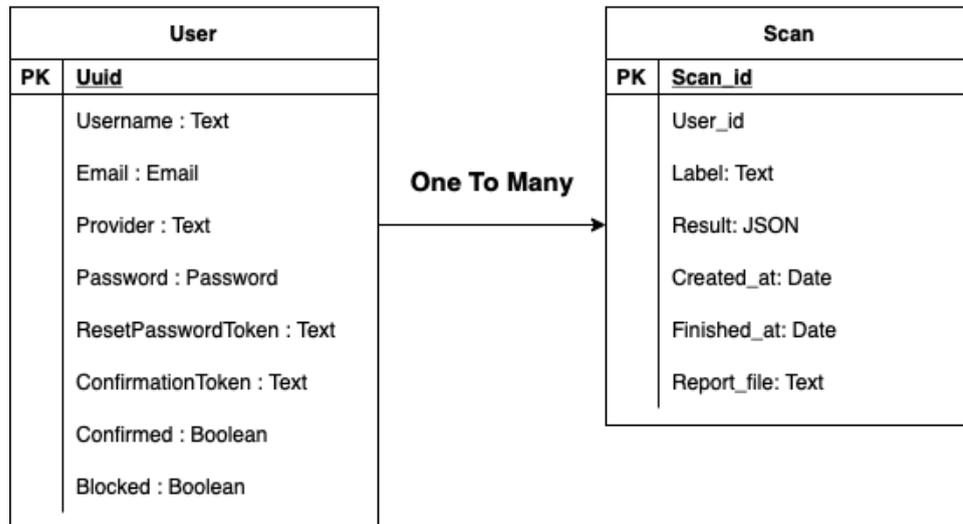
Use Case Name	Menghapus user	
Actors	Admin	
Triggers	Admin ingin menghapus user	
Precondition	Admin harus berhasil melakukan <i>login</i>	
Post Condition	Admin berhasil menghapus user	
Success Scenario	Actor	System
	1. Admin mengakses daftar user yang terdaftar	2. Sistem menampilkan daftar user yang telah terdaftar.
	3. Admin memilih user yang ingin dihapus.	4. Sistem berhasil menghapus user sesuai permintaan.
		5. Selesai.

4.2.3 Class Diagram

Pada umumnya, *class diagram* adalah sebuah diagram yang memetakan sebuah struktur sistem dan mencari hubungan antar-struktur tersebut. Dalam

aplikasi pemindaian kerentanan keamanan website ini, penulis membuat sebuah *class diagram* sederhana yang cukup untuk memenuhi kebutuhan sistem.

Relasi Database Aplikasi Pemindaian Kerentanan Website



Gambar 4.2 Class Diagram Aplikasi Pemindaian Kerentanan Keamanan Website

4.2.4 Activity Diagram

Activity diagram adalah sebuah diagram yang menjelaskan alur dari kegiatan di aplikasi tersebut. *Activity diagram* tersebut berguna untuk mengetahui alur untuk melakukan kegiatan tersebut, sehingga orang yang akan membuat aplikasi dapat memahami alur dari aplikasi tersebut

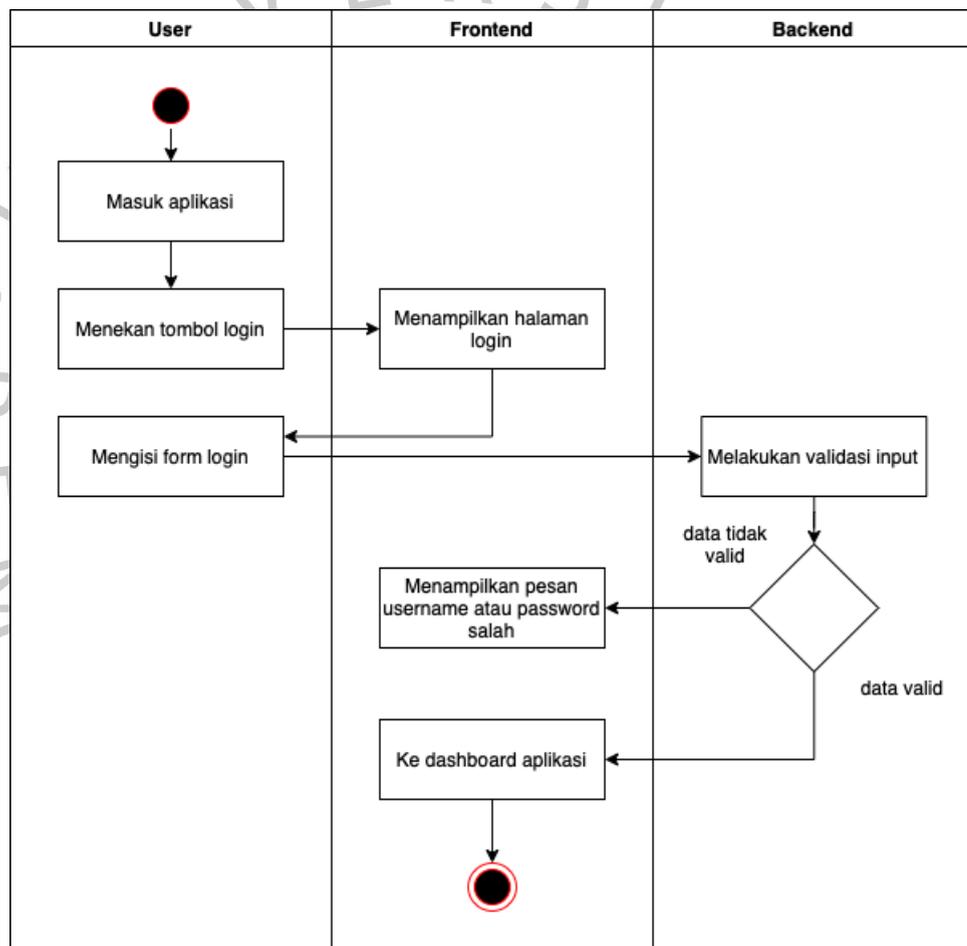
Activity diagram dari aplikasi pemindaian kerentanan keamanan website yang penulis buat terdapat 10 diagram, dimulai dari kegiatan *login*, registrasi, memindai kerentanan website, melihat proses pemindaian yang sedang berjalan, mengunduh laporan hasil audit, melihat daftar website yang telah dipindai hingga diagram manajemen user (CRUD).

a. *Login*

Dalam kegiatan *login*, yang pertama kali user lakukan ialah masuk ke aplikasi lalu menekan tombol *login* dan aplikasi akan menampilkan halaman *login*.

Setelah itu user mengisi form *login* berupa *username*, dan *password* yang akan dilakukan validasi oleh backend valid atau tidaknya data tersebut.

Jika valid, masa user akan terontetikasi masuk ke dashboard aplikasi. Jika yang data yang telah dimasukkan tidak valid, maka akan langsung menampilkan pesan username/password salah.



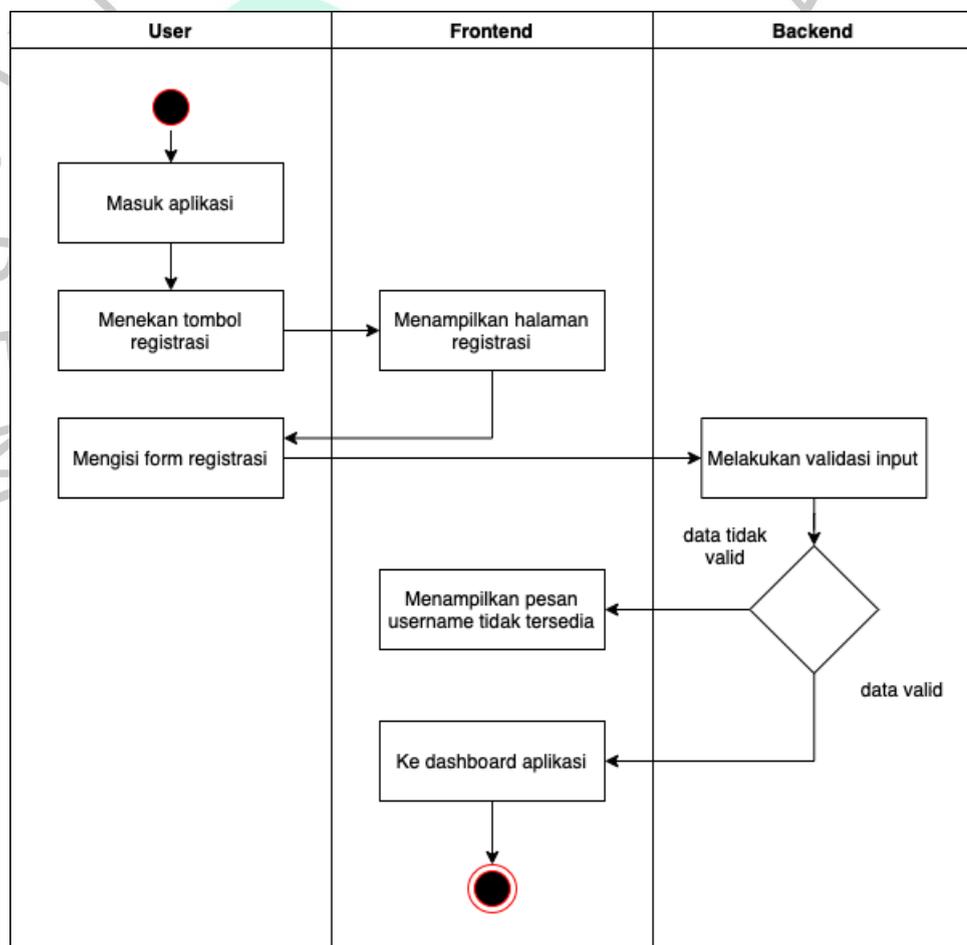
Gambar 4.3 Activity Diagram Login

b. Registrasi

Dalam kegiatan registrasi, yang pertama kali user lakukan ialah masuk ke aplikasi lalu menekan tombol registrasi dan aplikasi akan menampilkan halaman registrasi.

Setelah itu user mengisi form registrasi berupa *username*, *email* dan *password* yang akan dilakukan validasi oleh backend valid atau tidaknya data tersebut.

Jika valid, masa user akan terotentikasi terdaftar dan masuk ke dashboard aplikasi. Jika data yang telah dimasukkan tersebut tidak valid, maka akan langsung menampilkan pesan username tidak tersedia.



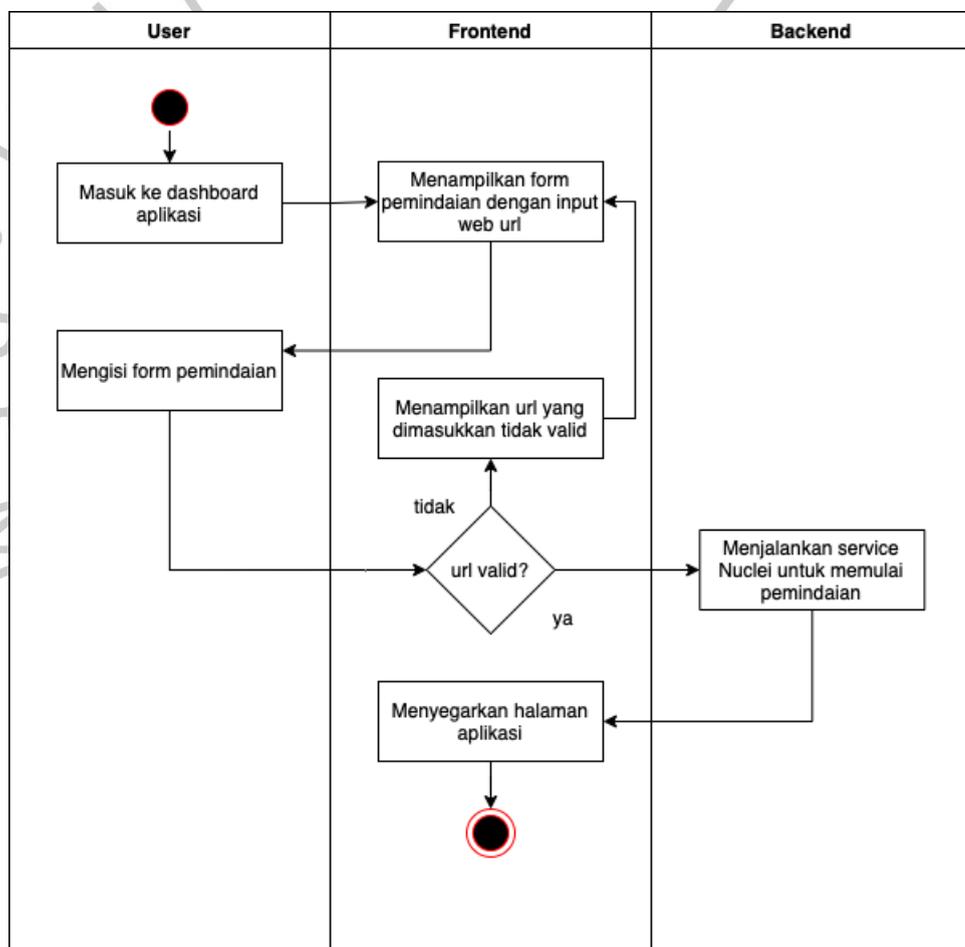
Gambar 4.4 Activity Diagram Registrasi

c. Memindai kerentanan website

Dalam kegiatan memindai kerentanan website, user dalam keadaan sudah *login* dan masuk didalam dashboard aplikasi dan aplikasi akan menampilkan form pemindaian yang dapat diisi dengan web url.

Jika form yang dimasukkan valid, maka backend akan menjalankan service Nuclei untuk melakukan aktivitas pemindaian dan menyegarkan halaman aplikasi.

Jika web url yang telah dimasukkan itu tidak valid, maka akan langsung menampilkan pesan url yang dimasukkan tidak valid.

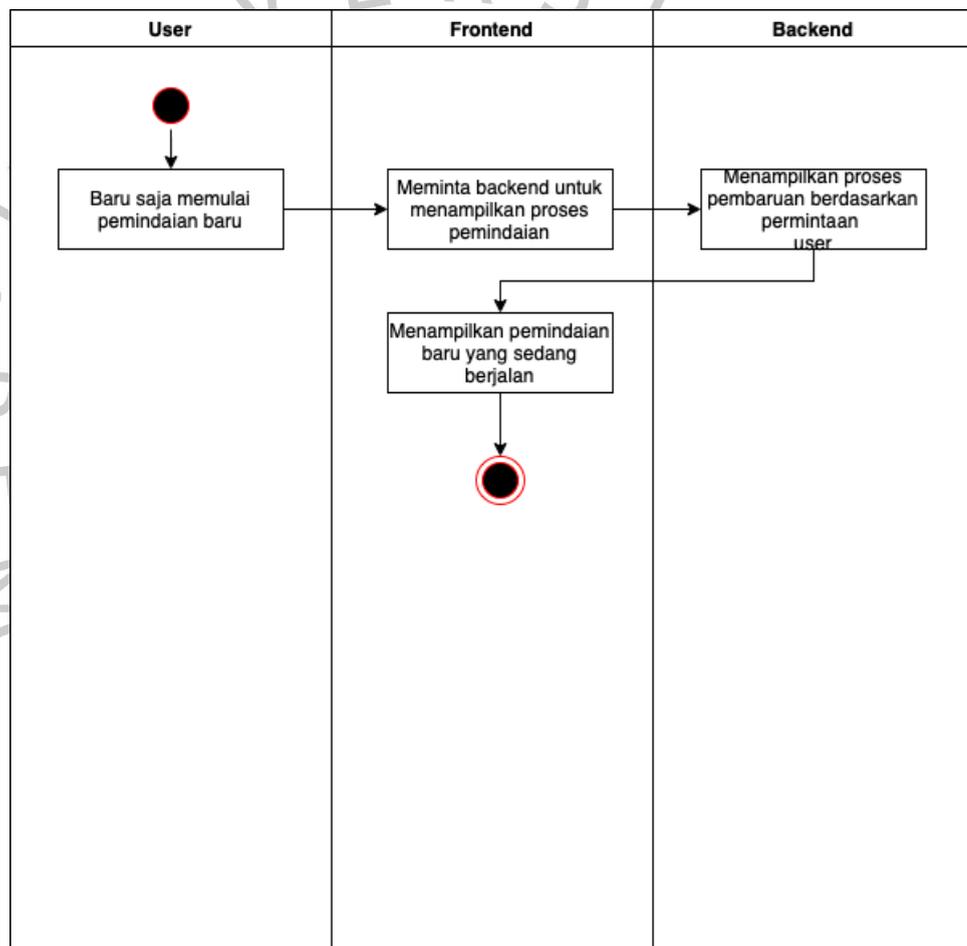


Gambar 4.5 Activity Diagram Memindai kerentanan website

d. Melihat proses pemindaian yang sedang berjalan

Dalam kegiatan melihat proses pemindaian yang sedang berjalan, user dalam kegiatan baru saja melakukan pemindaian baru dan aplikasi meminta backend untuk menampilkan proses berlangsungnya pemindaian.

Setelah itu, backend menampilkan sebuah data proses pembaruan berdasarkan permintaan user dan diteruskan ke aplikasi untuk menampilkan proses pemindaian baru yang sedang berjalan.



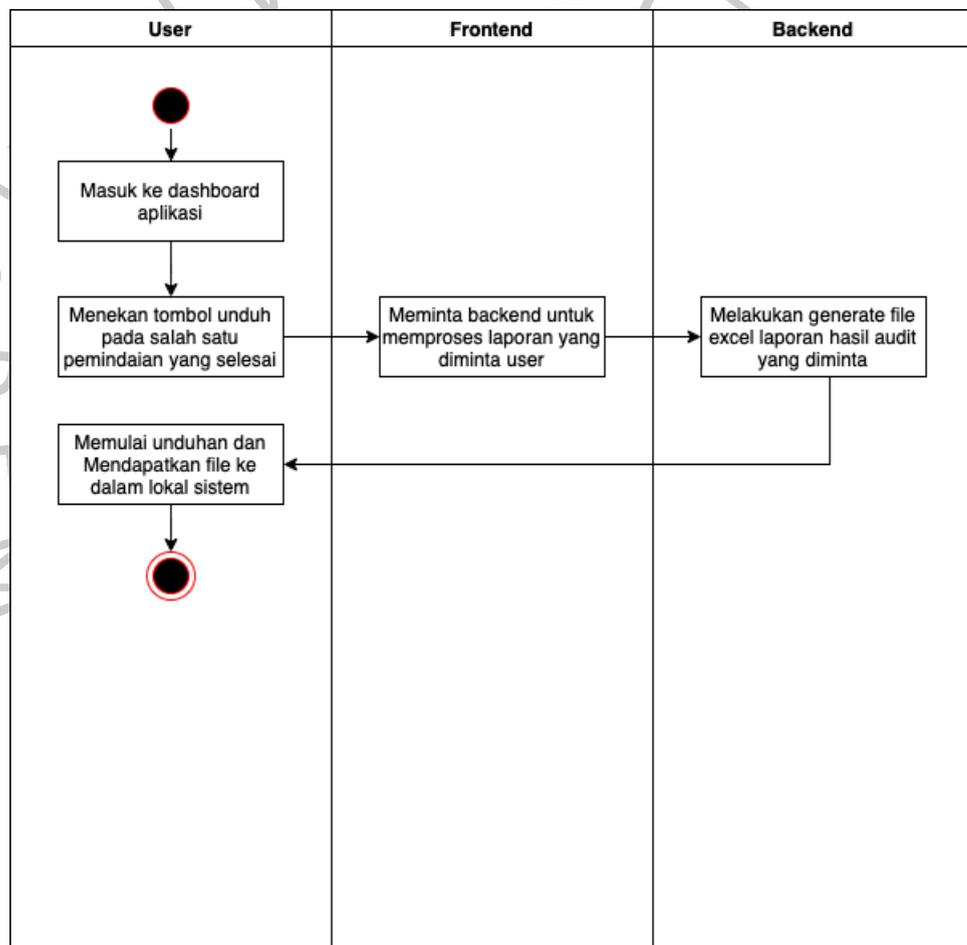
Gambar 4.6 Activity Diagram Melihat proses pemindaian yang sedang berjalan

e. Mengunduh laporan hasil audit

Dalam kegiatan mengunduh laporan hasil audit, user sudah dalam keadaan *login* dan sudah pernah melakukan proses pemindaian yang sudah selesai.

User dapat menekan tombol unduh pada salah satu pemindaian yang telah selesai, lalu aplikasi meminta backend untuk memproses laporan yang diminta user.

Backend melakukan *generate file excel* laporan hasil audit yang diminta untuk dapat diunduh oleh user dan disimpan didalam sistem lokalnya user.

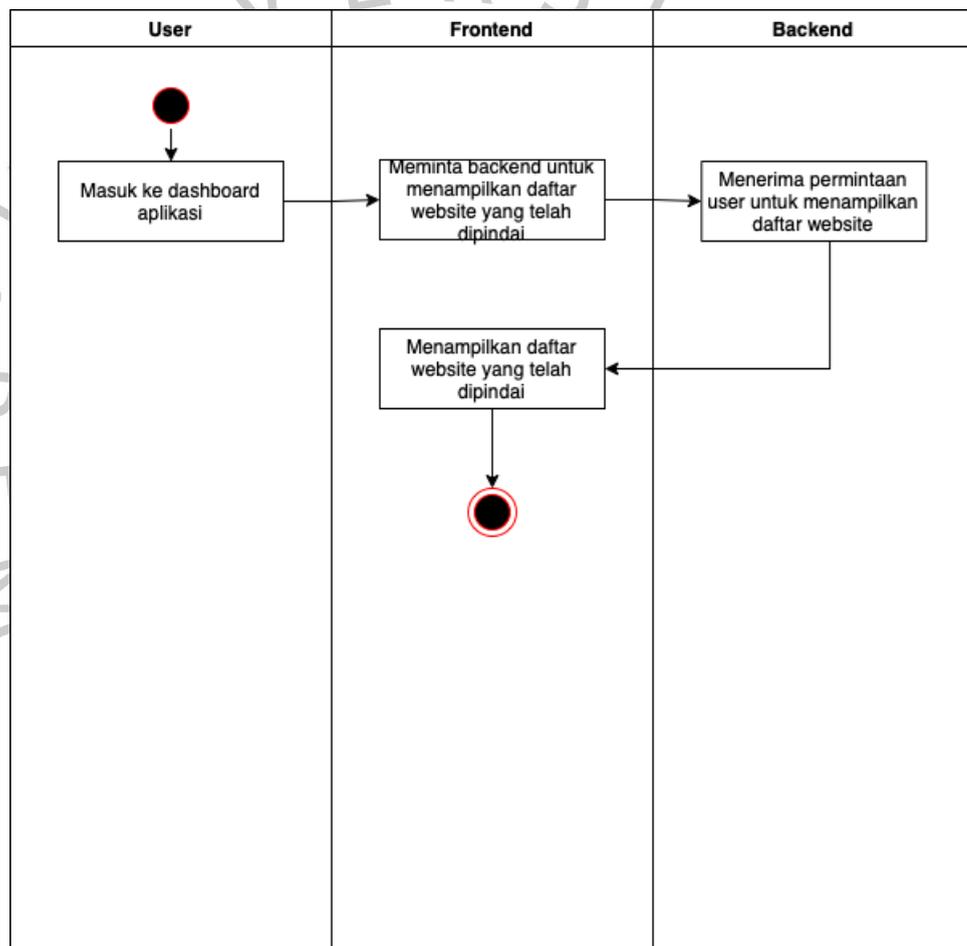


Gambar 4.7 Activity Diagram Mengunduh laporan hasil audit

f. Melihat daftar website yang telah dipindai

Dalam kegiatan melihat daftar website yang telah dipindai, user sudah dalam keadaan *login* dan masuk kedalam dashboard aplikasi, lalu aplikasi akan meminta backend untuk menampilkan daftar website yang telah dipindai.

Setelah backend menerima permintaan user sebuah data daftar website yang telah user pindai, lalu diteruskan ke aplikasi untuk ditampilkan daftar website yang telah dipindai.

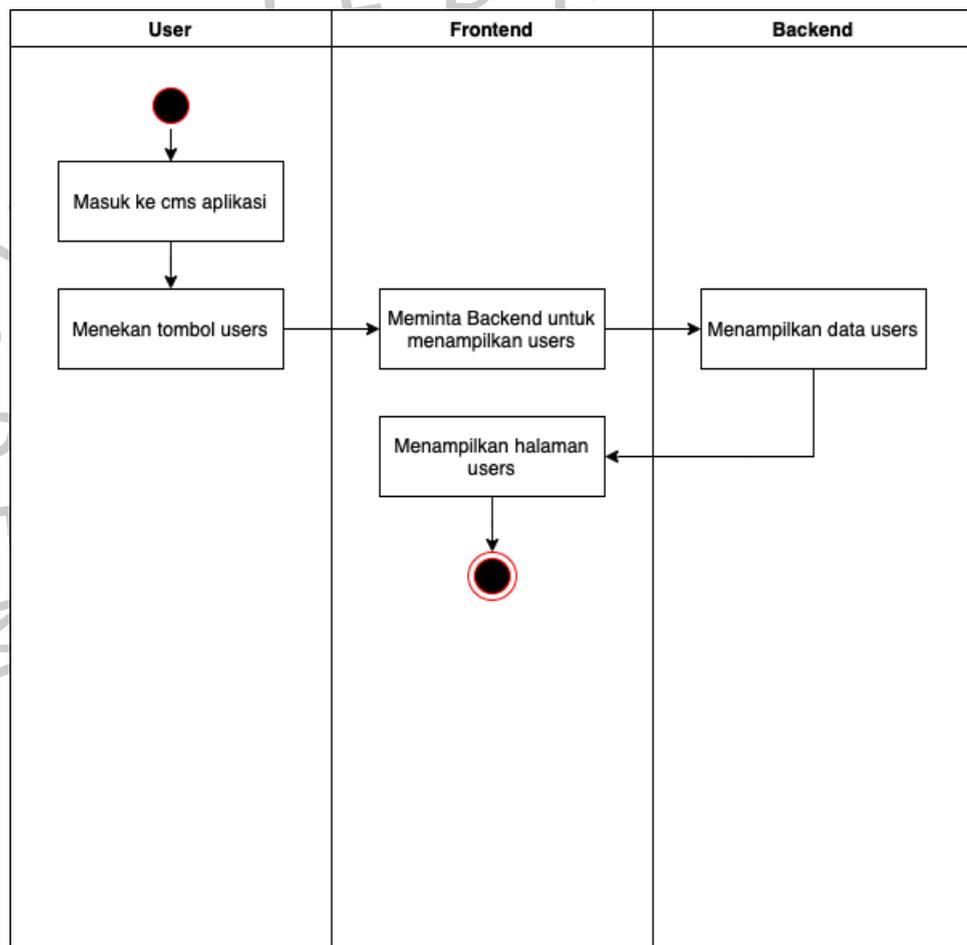


Gambar 4.8 Activity Diagram Melihat daftar website yang telah dipindai

g. Melihat user

Dalam kegiatan melihat daftar user, admin sudah dalam keadaan *login* dan masuk ke cms aplikasi dan menekan tombol users.

Lalu aplikasi akan meminta sebuah data daftar user ke backend dan setelah itu aplikasi akan menampilkan halaman daftar user berdasarkan data yang diterima tersebut.



Gambar 4.9 Activity Diagram Melihat user

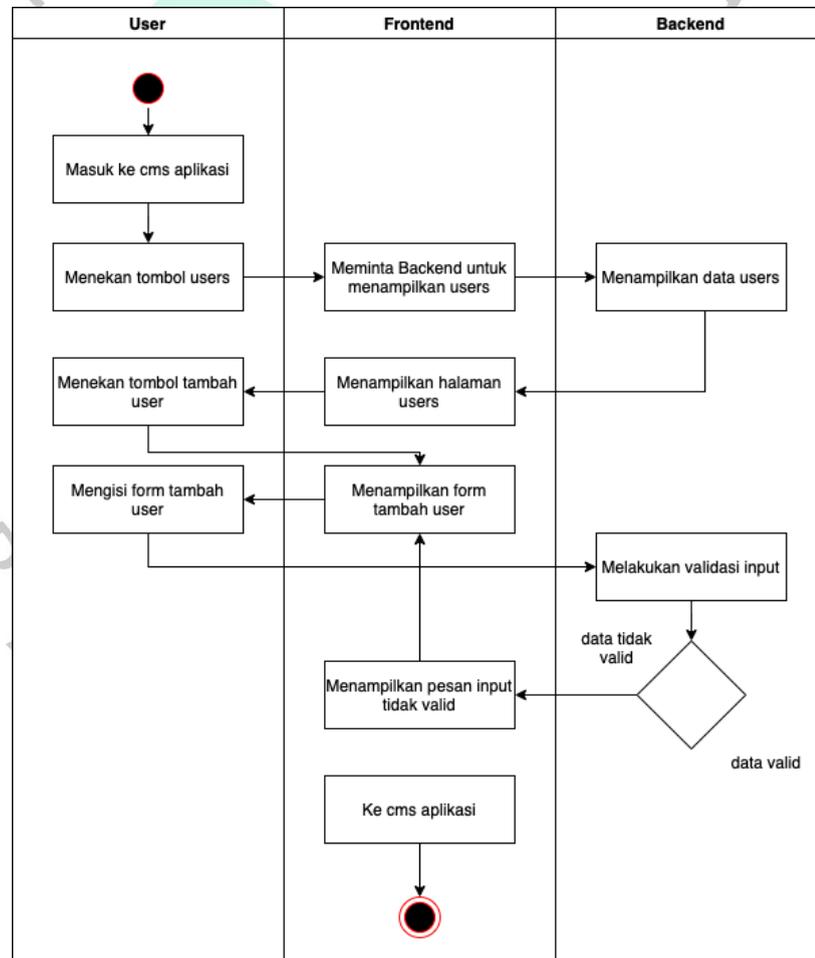
h. Menambahkan user

Dalam kegiatan melihat daftar user, admin sudah dalam keadaan *login* dan masuk ke cms aplikasi dan menekan tombol users.

Lalu aplikasi akan meminta sebuah data daftar user ke backend dan setelah itu aplikasi akan menampilkan halaman daftar user berdasarkan data yang diterima tersebut.

Setelah itu admin dapat menekan tombol tambah user dan mengisi form tambah user yang ditampilkan.

Backend akan melakukan validasi input, jika valid maka akan diarahkan ke halaman daftar user. Jika yang telah dimasukkan tidak valid, maka akan langsung ditampilkan pesan error.



Gambar 4.10 Activity Diagram Menambahkan user

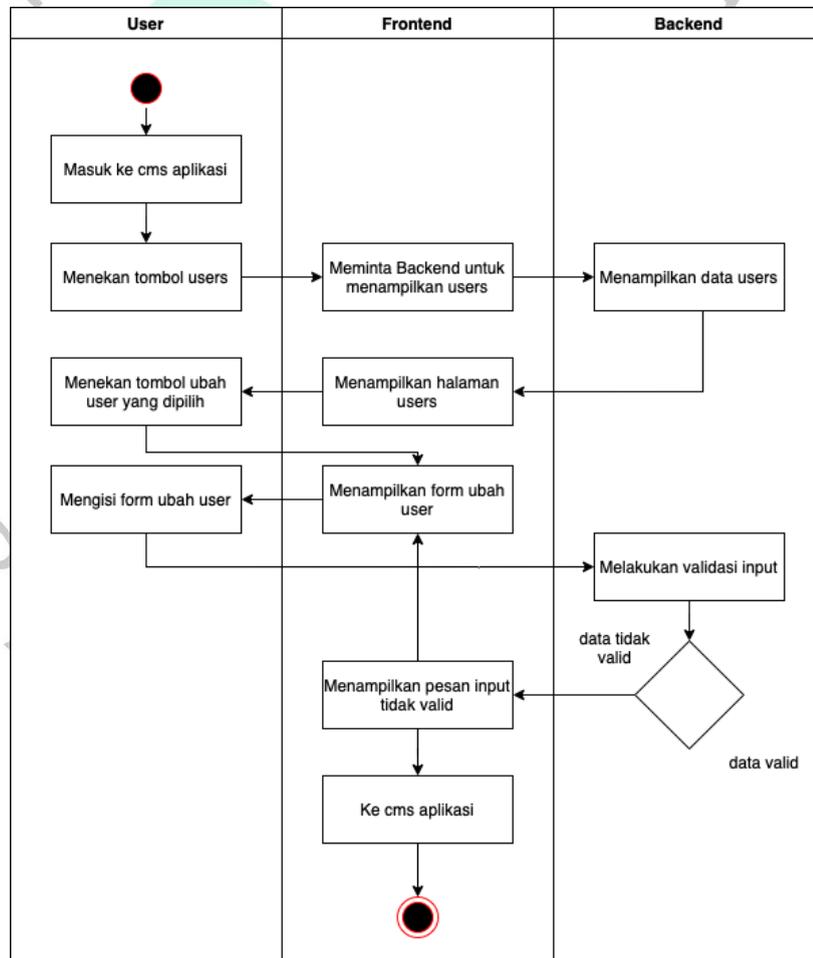
i. Mengubah user

Dalam kegiatan melihat daftar user, admin sudah dalam keadaan *login* dan masuk ke cms aplikasi dan menekan tombol users.

Lalu aplikasi akan meminta sebuah data daftar user ke backend dan setelah itu aplikasi akan menampilkan halaman daftar user berdasarkan data yang diterima tersebut.

Setelah itu admin dapat menekan tombol ubah user yang dipilih dan mengubah form ubahuser yang ditampilkan.

Backend akan melakukan validasi input, jika valid maka akan diarahkan ke halaman daftar user. Jika yang dimasukkan tidak valid, maka akan ditampilkan pesan error.



Gambar 4.11 Activity Diagram Mengubah user

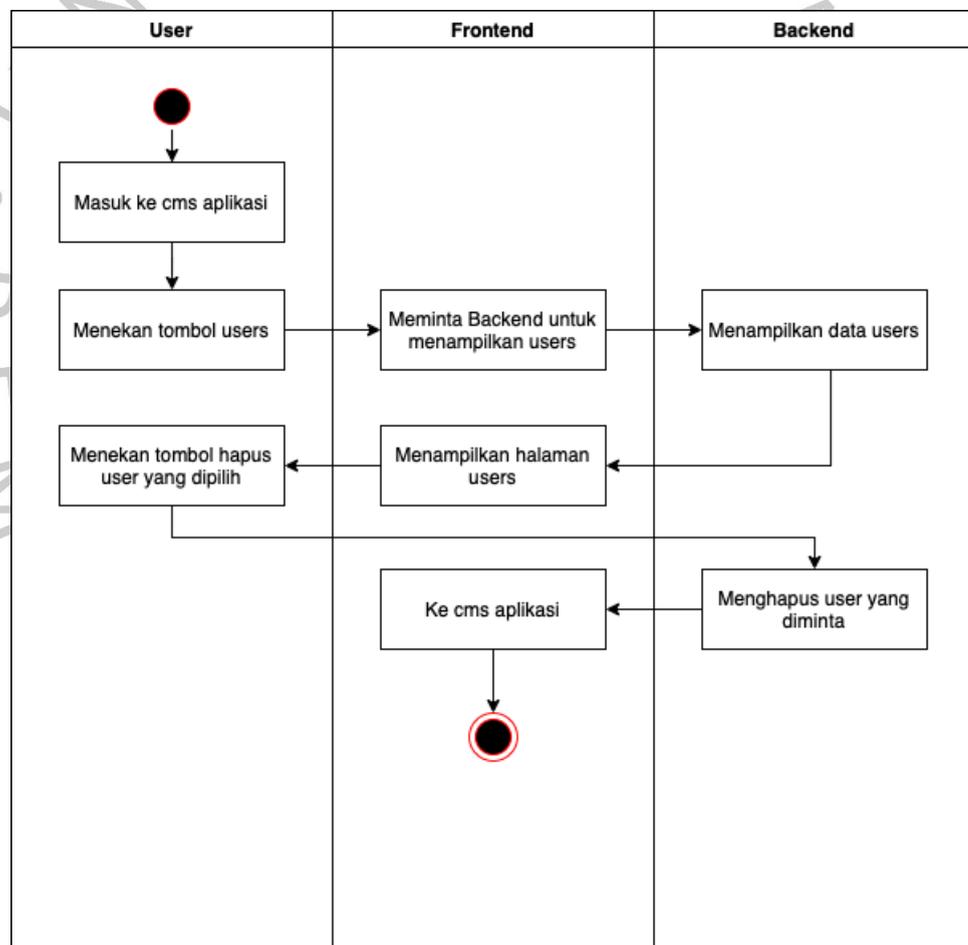
j. Menghapus user

Dalam kegiatan melihat daftar user, admin sudah dalam keadaan *login* dan masuk ke cms aplikasi dan menekan tombol users.

Lalu aplikasi akan meminta sebuah data daftar user ke backend dan setelah itu aplikasi akan menampilkan halaman daftar user berdasarkan data yang diterima tersebut.

Sehabis itu, admin dapat melakukan penghapusan *user* yang telah dipilihnya dengan menekan tombol hapus yang tersedia ditampilan.

Dan akan diteruskan oleh backend, setelah selesai halaman akan diarahkan ke halaman daftar user.



Gambar 4.12 Activity Diagram Menghapus user

4.2.5 Sequence Diagram

Sequence diagram adalah sebuah diagram yang menjelaskan secara rinci bagaimana proses tersebut bekerja. Serupa dengan *activity diagram*, namun *sequence diagram* lebih menjelaskan tentang alur dari data tersebut. Banyak dari *sequence diagram* ini dilihat dari berapa banyak *use case* yang digunakan dalam proses tersebut.

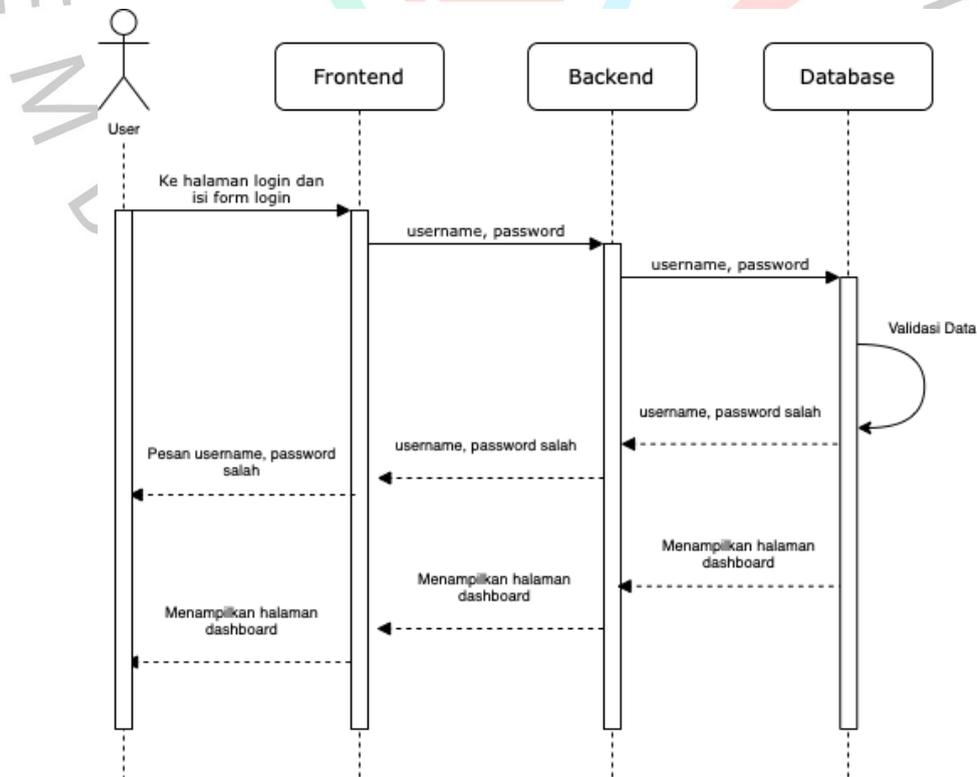
Sequence diagram dari aplikasi pemindaian kerentanan keamanan website yang penulis buat terdapat 10 diagram, dimulai dari kegiatan *login*, registrasi, memindai kerentanan website, melihat proses pemindaian yang sedang berjalan, mengunduh laporan hasil audit, melihat daftar website yang telah dipindai hingga diagram manajemen user (CRUD).

a. *Login*

Dalam kegiatan *login*, yang pertama kali user lakukan ialah masuk ke aplikasi lalu menekan tombol *login* dan aplikasi akan menampilkan halaman *login*.

• Setelah itu user mengisi form *login* berupa *username*, dan *password* yang akan dilakukan validasi oleh backend valid atau tidaknya data tersebut.

Jika valid, masa user akan terontetikasi masuk ke dashboard aplikasi. Jika data yang telah dimasukkan tersebut tidaklah valid, maka akan langsung menampilkan pesan *username/password* salah.



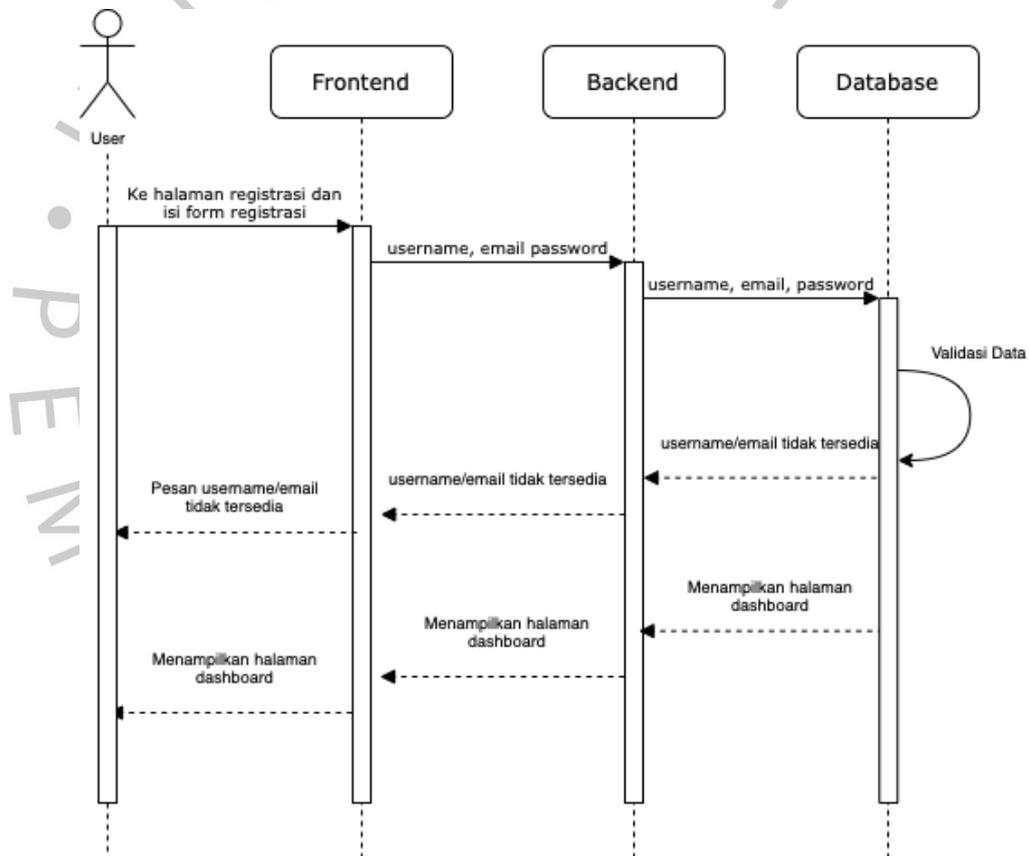
Gambar 4.13 *Sequence Diagram Login*

b. Registrasi

Dalam kegiatan registrasi, yang pertama kali user lakukan ialah masuk ke aplikasi lalu menekan tombol registrasi dan aplikasi akan menampilkan halaman registrasi.

Setelah itu user mengisi form registrasi berupa *username*, *email* dan *password* yang akan dilakukan validasi oleh backend valid atau tidaknya data tersebut.

Jika valid, masa user akan terontetikasi terdaftar dan masuk ke dashboard aplikasi. Jika sebuah data yang dimasukkan tidak valid, maka akan langsung menampilkan pesan username tidak tersedia.



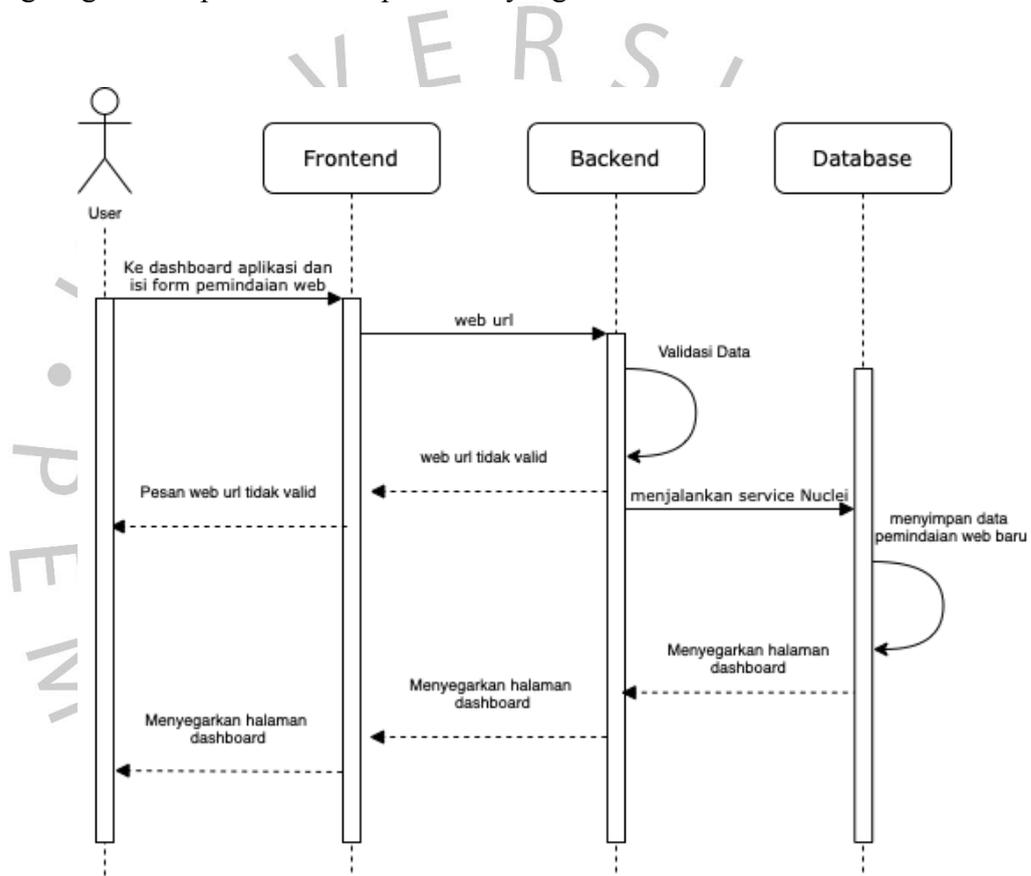
Gambar 4.14 Sequence Diagram Registrasi

c. Memindai kerentanan website

Dalam kegiatan memindai kerentanan website, user dalam keadaan sudah *login* dan masuk didalam dashboard aplikasi dan aplikasi akan menampilkan form pemindaian yang dapat diisi dengan web url.

Jika form yang dimasukkan valid, maka backend akan menjalankan service Nuclei untuk melakukan aktivitas pemindaian dan menyegarkan halaman aplikasi.

Jika pada web url yang telah dimasukkan itu tidak valid, maka akan langsung menampilkan bahwa pesan url yang telah dimasukkan tidak valid.

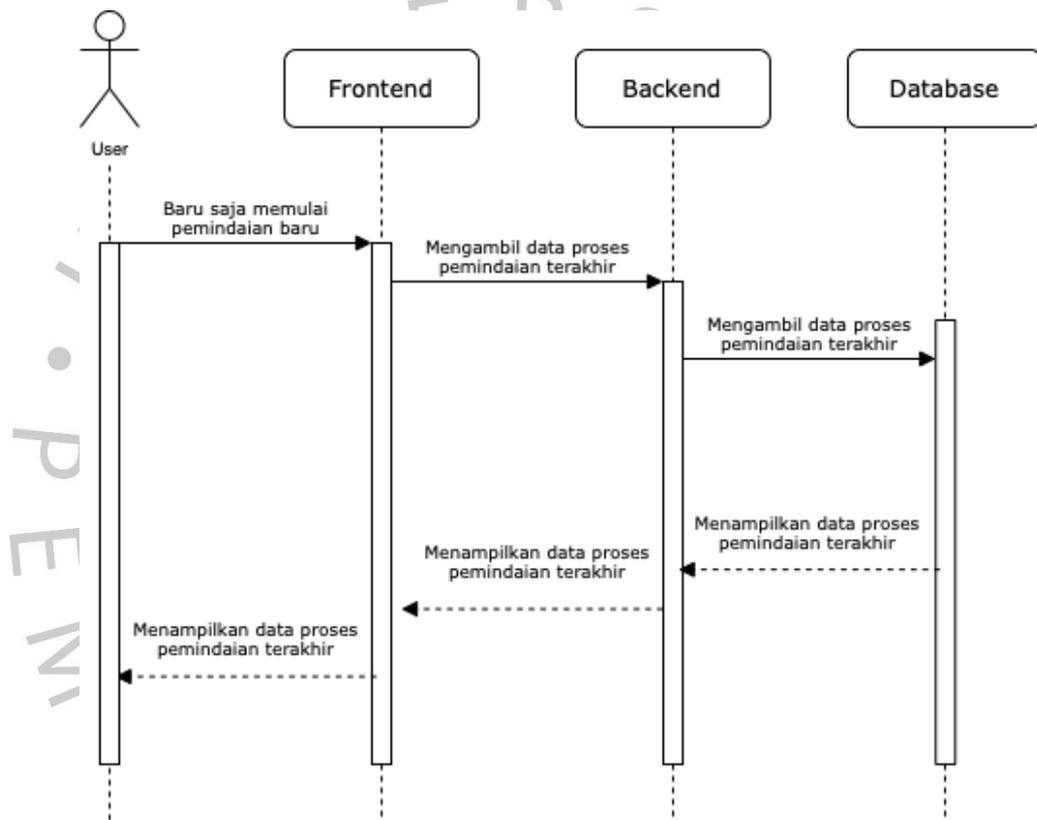


Gambar 4.15 Sequence Diagram Memindai kerentanan website

d. Melihat proses pemindaian yang sedang berjalan

Dalam kegiatan melihat proses pemindaian yang sedang berjalan, user dalam kegiatan baru saja melakukan pemindaian baru dan aplikasi meminta backend untuk menampilkan proses berlangsungnya pemindaian.

Setelah itu, backend menampilkan sebuah data proses pembaruan berdasarkan permintaan user dan diteruskan ke aplikasi untuk menampilkan proses pemindaian baru yang sedang berjalan.



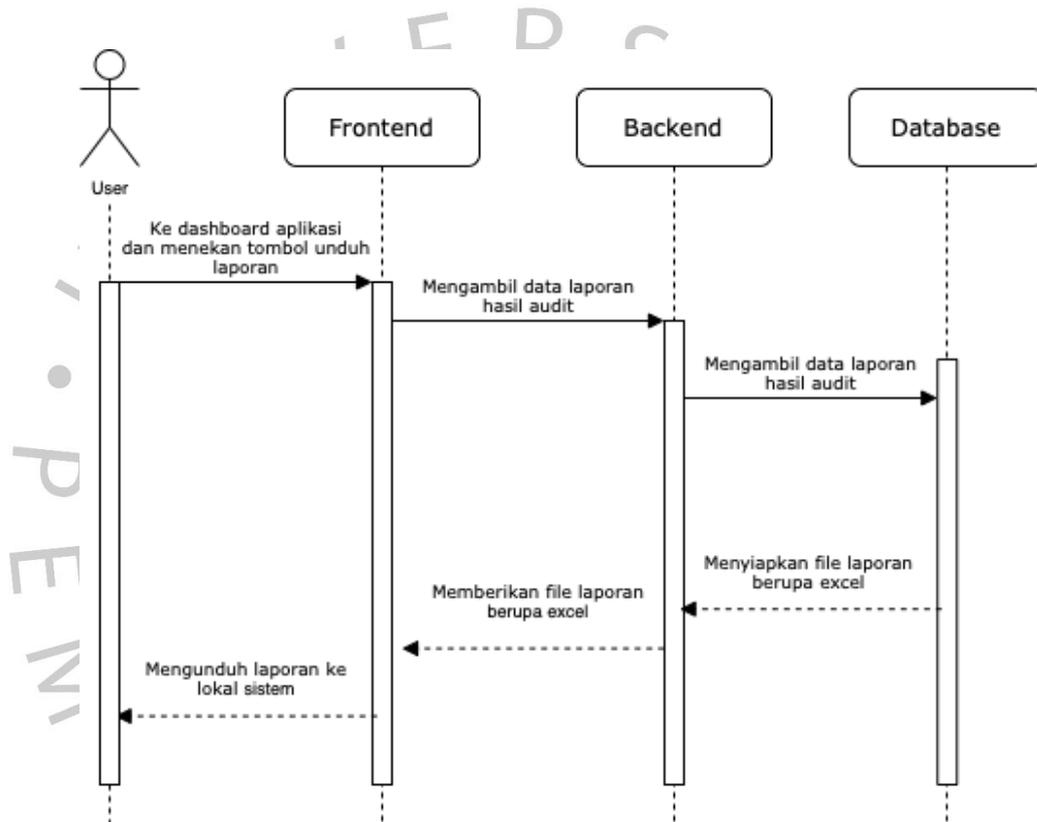
Gambar 4.16 *Sequence* Diagram Melihat proses pemindaian yang sedang berjalan

e. Mengunduh laporan hasil audit

Dalam kegiatan mengunduh laporan hasil audit, user sudah dalam keadaan *login* dan sudah pernah melakukan proses pemindaian yang sudah selesai.

User dapat menekan tombol unduh pada salah satu pemindaian yang telah selesai, lalu aplikasi meminta backend untuk memproses laporan yang diminta user.

Backend melakukan *generate file excel* laporan hasil audit yang diminta untuk dapat diunduh oleh user dan disimpan didalam sistem lokalnya user.

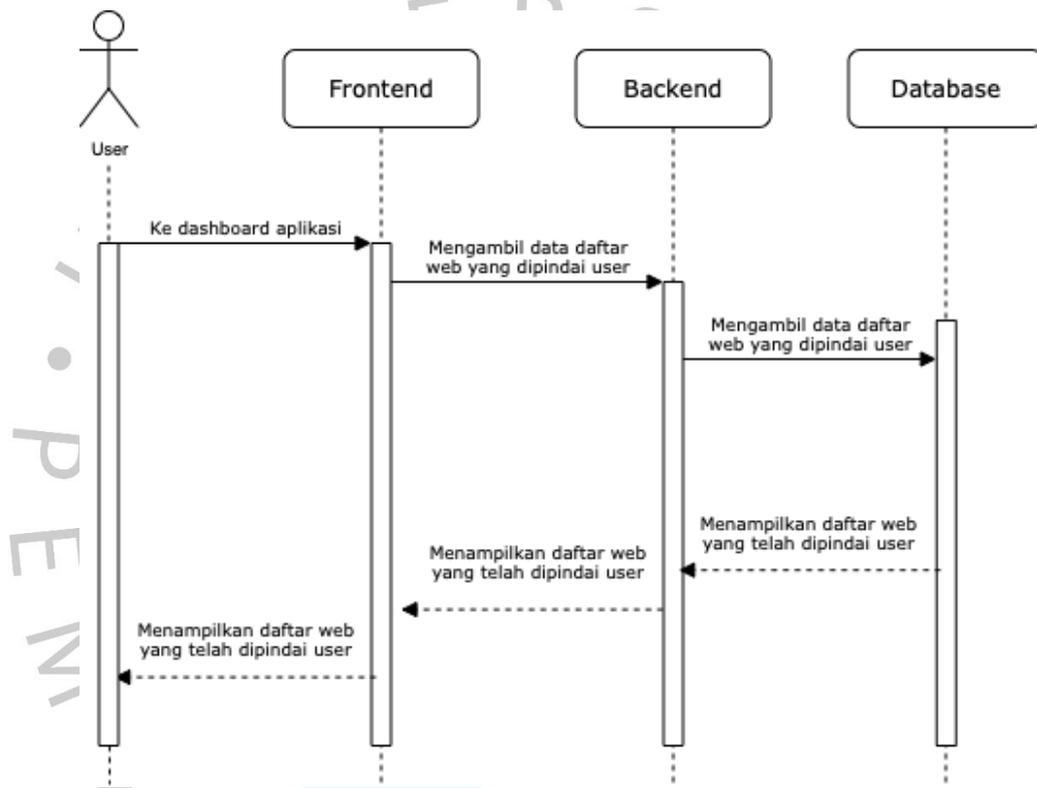


Gambar 4.17 Sequence Diagram Mengunduh laporan hasil audit

f. Melihat daftar website yang telah dipindai

Dalam kegiatan melihat daftar website yang telah dipindai, user sudah dalam keadaan *login* dan masuk kedalam dashboard aplikasi, lalu aplikasi akan meminta backend untuk menampilkan daftar website yang telah dipindai.

Setelah backend menerima permintaan user sebuah data daftar website yang telah user pindai, lalu diteruskan ke aplikasi untuk ditampilkan daftar website yang telah dipindai.

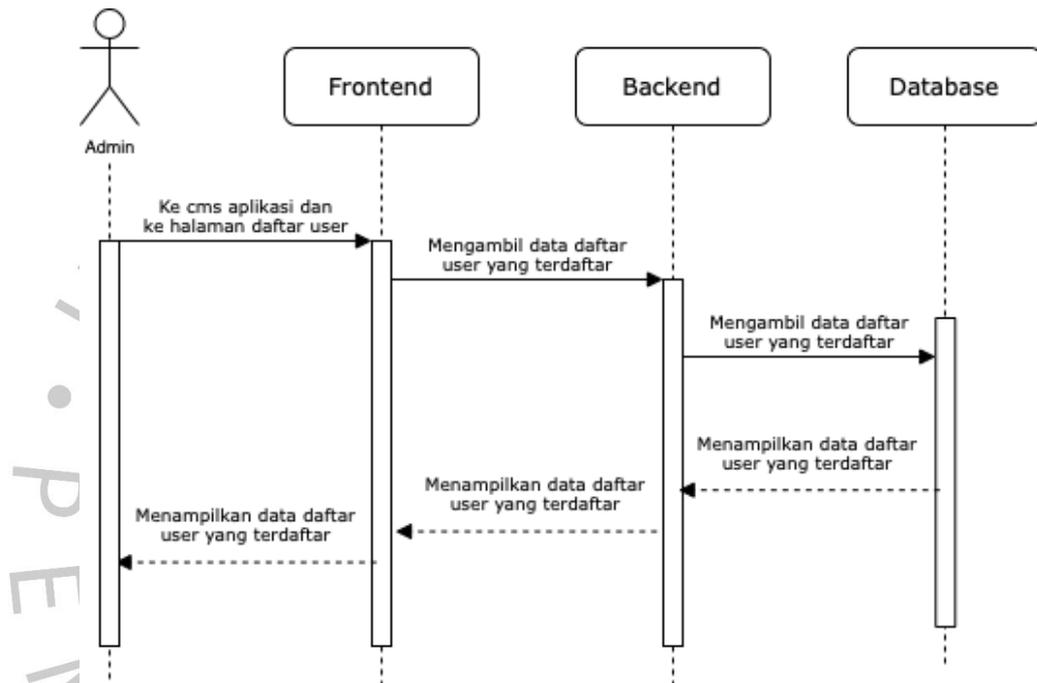


Gambar 4.18 *Sequence Diagram* Melihat daftar website yang telah dipindai

g. Melihat user

Dalam kegiatan melihat daftar user, admin sudah dalam keadaan *login* dan masuk ke cms aplikasi dan menekan tombol users.

Lalu aplikasi akan meminta sebuah data daftar user ke backend dan setelah itu aplikasi akan menampilkan halaman daftar user berdasarkan data yang diterima tersebut.



Gambar 4.19 Sequence Diagram Melihat user

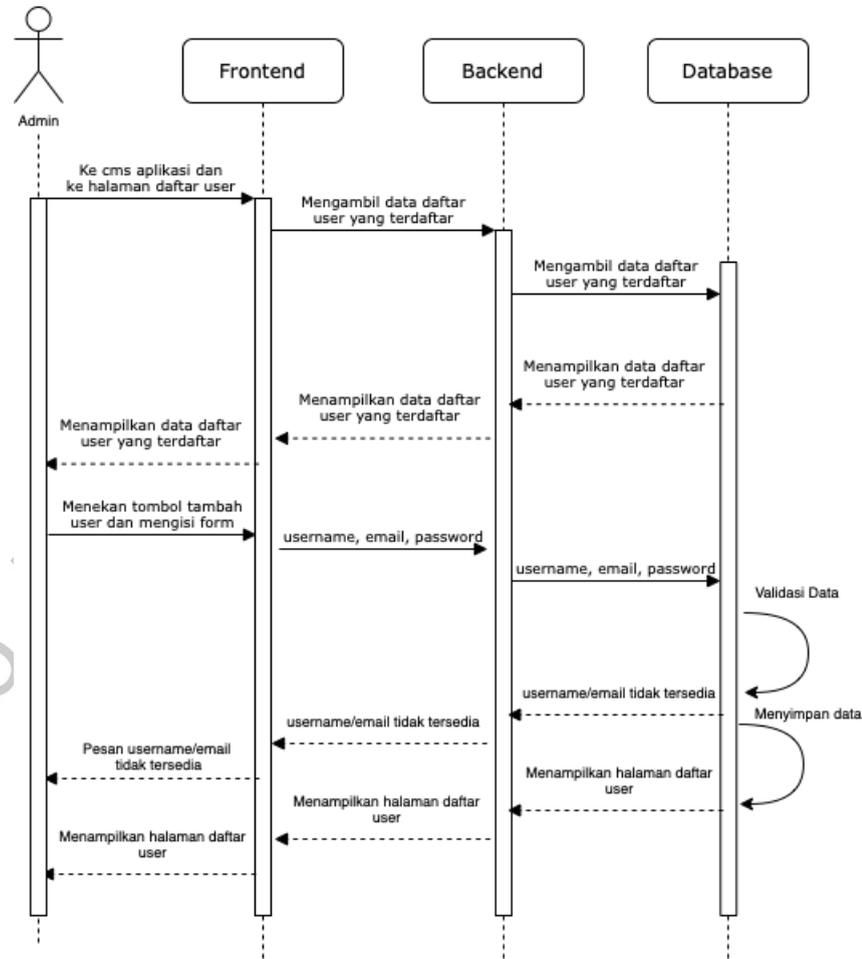
h. Menambahkan user

Dalam kegiatan melihat daftar user, admin sudah dalam keadaan *login* dan masuk ke cms aplikasi dan menekan tombol users.

Lalu aplikasi akan meminta sebuah data daftar user ke backend dan setelah itu aplikasi akan menampilkan halaman daftar user berdasarkan data yang diterima tersebut.

Setelah itu admin dapat menekan tombol tambah user dan mengisi form tambah user yang ditampilkan.

Backend akan melakukan validasi input, jika valid maka akan diarahkan ke halaman daftar user. Jika yang dimasukkan menghasilkan tidak valid, maka akan langsung ditampilkan pesan error.



Gambar 4.20 Sequence Diagram Menambahkan user

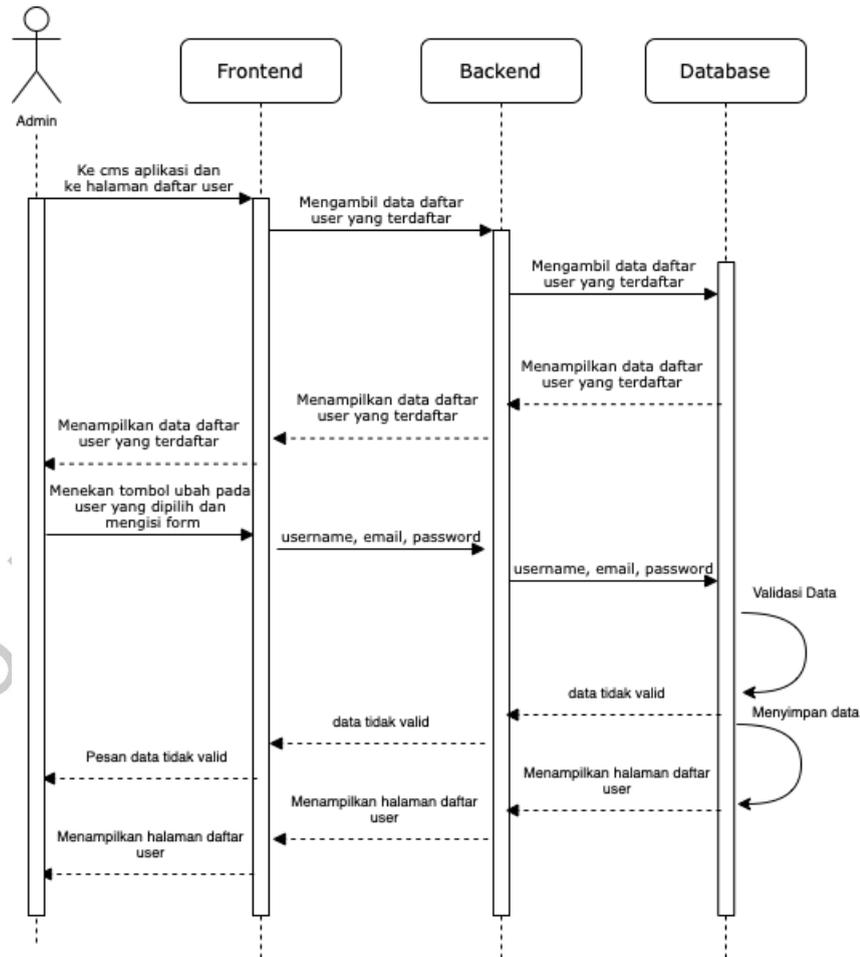
i. Mengubah user

Dalam kegiatan melihat daftar user, admin sudah dalam keadaan *login* dan masuk ke cms aplikasi dan menekan tombol users.

Lalu aplikasi akan meminta sebuah data daftar user ke backend dan setelah itu aplikasi akan menampilkan halaman daftar user berdasarkan data yang diterima tersebut.

Setelah itu admin dapat menekan tombol ubah user yang dipilih dan mengubah form ubahuser yang ditampilkan.

Backend akan melakukan validasi input, jika valid maka akan diarahkan ke halaman daftar user. Jika yang dimasukkan tidak valid, maka akan langsung ditampilkan pesan error.



Gambar 4.21 Sequence Diagram Mengubah user

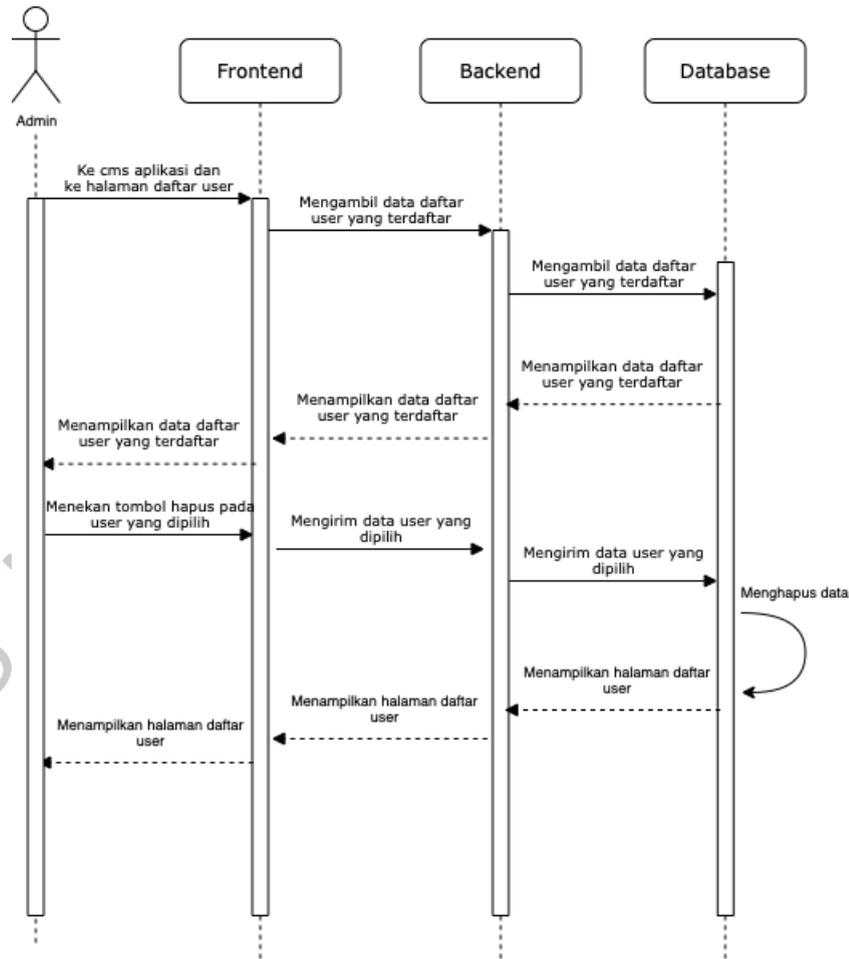
j. Menghapus user

Dalam kegiatan melihat daftar user, admin sudah dalam keadaan *login* dan masuk ke cms aplikasi dan menekan tombol users.

Lalu aplikasi akan meminta sebuah data daftar user ke backend dan setelah itu aplikasi akan menampilkan halaman daftar user berdasarkan data yang diterima tersebut.

Setelah itu, admin dapat melakukan penghapusan *user* yang telah dipilihnya dengan dapat menekan tombol hapus yang telah tersedia.

Dan akan diteruskan oleh backend, setelah selesai halaman akan diarahkan ke halaman daftar user.



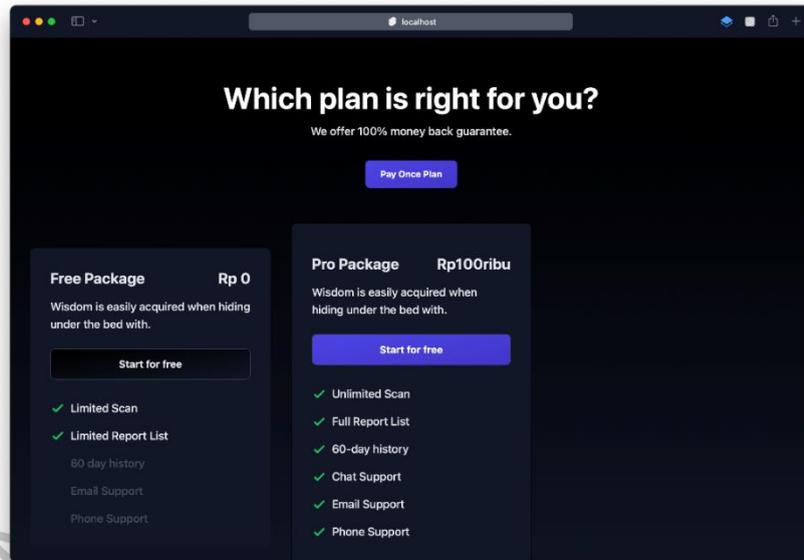
Gambar 4.22 Sequence Diagram Menghapus user

4.3 Hasil Antar Muka

4.3.1 Hasil Antar Muka Pengguna

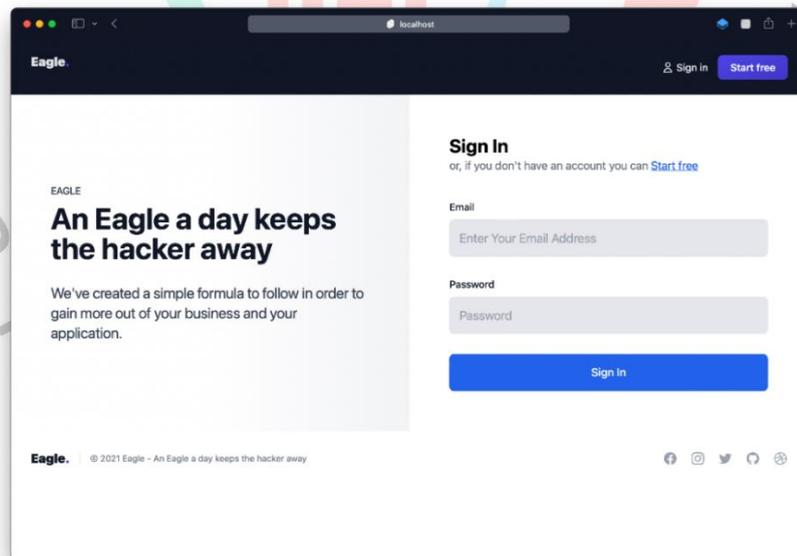
Perancangan antar muka pengguna menjelaskan tentang bagaimana tampilan dari setiap modul yang ada di aplikasi tersebut. Perancangan tersebut dilakukan agar si *programmer* dapat mendesain aplikasi tersebut sesuai dengan yang diperlukan oleh si pengguna.

Pada perancangan aplikasi pemindai kerentanan keamanan website ini, penulis menamakan aplikasi ini bernama *Eagle* atau Bahasa Indonesianya adalah Elang. Kenapa dinamakan *Eagle*, karena burung elang terkenal mempunyai mata yang sangat tajam sehingga pada penamaan aplikasi ini ditujukan pemindaian kerentanan keamanan website seperti mata elang yang sangat tajam.



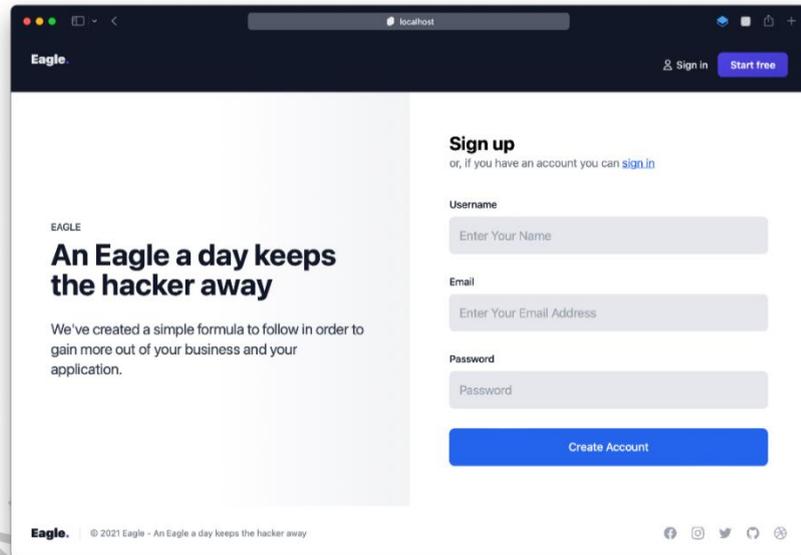
Gambar 4.24 Tampilan utama bagian harga

Pada tampilan *login*, aplikasi menampilkan form *login* yang dapat diisi dengan *username*, dan *password* untuk dapat diarahkan masuk ke dalam dashboard aplikasi jika data yang dimasukkan valid.



Gambar 4.25 Tampilan *login*

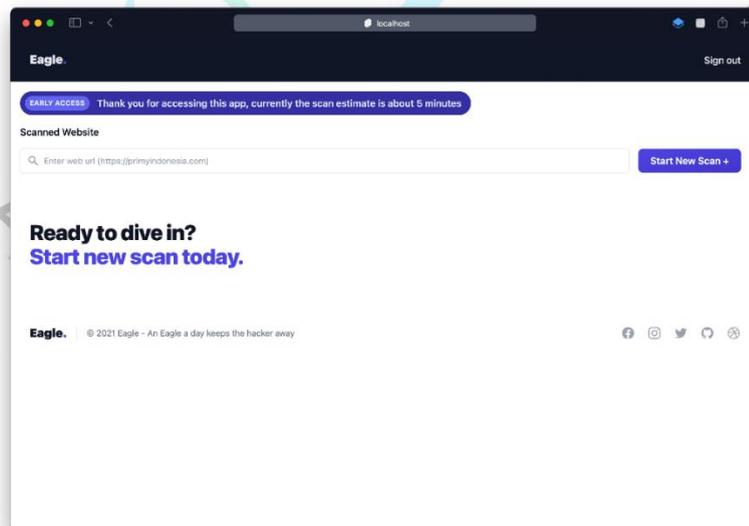
Pada tampilan registrasi, aplikasi menampilkan form registrasi yang dapat diisi dengan *username*, *email* dan *password* untuk melakukan pembuatan akun dalam menikmati fitur yang ada.



Gambar 4.26 Tampilan registrasi

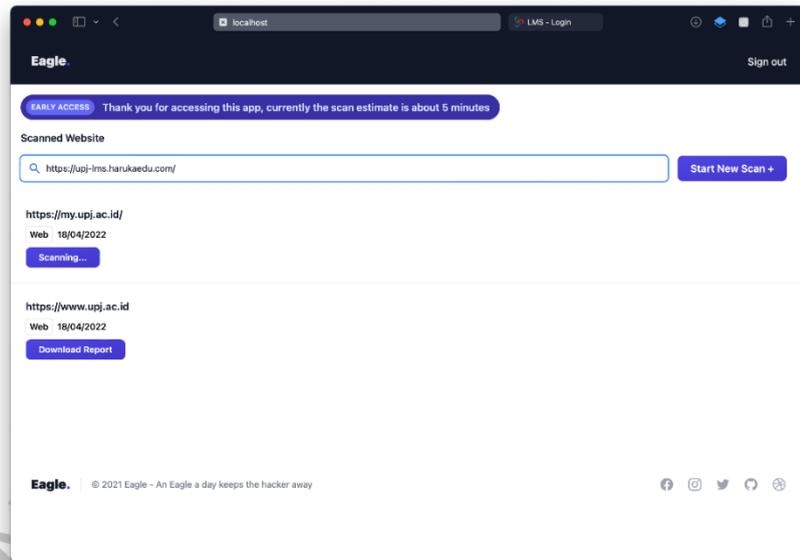
Pada tampilan *dashboard* terdapat fitur utama yaitu pemindaian kerentanan keamanan website berdasarkan input web url yang dimasukkan user pada bagian atas halaman *dashboard*.

Serta jika *user* belum pernah melakukan pemindaian, maka diarahkan untuk melakukan pemindaian baru dengan kalimat memulai pemindaian baru.



Gambar 4.27 Tampilan *dashboard*

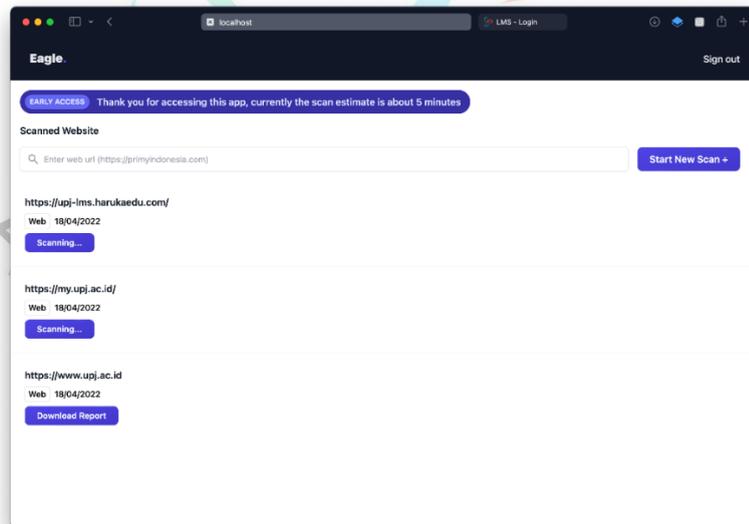
Jika *user* sudah melakukan pemindaian sebelumnya, maka akan ditampilkan daftar website yang telah dipindai beserta status proses pemindaian yang sedang berlangsung.



Gambar 4.28 Tampilan dashboard memulai pemindaian baru

Masih pada tampilan yang sama yaitu di dashboard, *user* dapat mengisi web url yang dijadikan target untuk dilakukan pemindaian kerentanan keamanan website dengan menekan tombol *Start New Scan*.

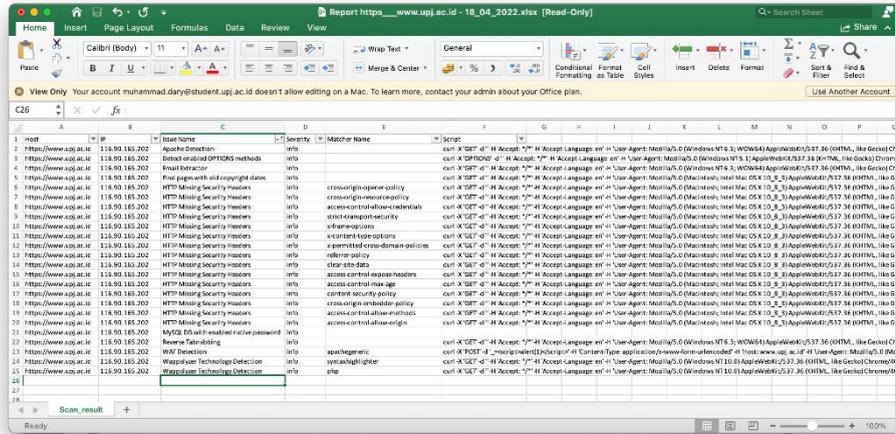
Jika web url yang dimasukkan valid, maka pemindaian akan dimulai di latar belakang Backend untuk menjalankan *service* Nuclei dan di aplikasi menampilkan daftar terbaru pemindaian website dengan status *Scanning*.



Gambar 4.29 Tampilan dashboard daftar pemindaian yang berjalan dan selesai

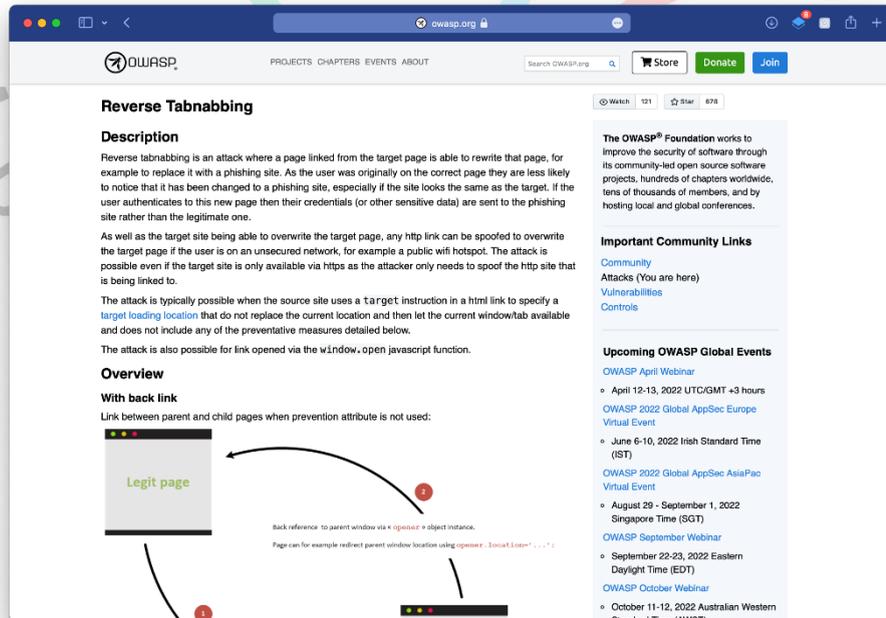
Pada pemindaian kerentanan keamanan website yang sudah selesai, *user* dapat mengunduh laporan hasil audit dengan menekan tombol *Download Report*.

Lalu *user* akan menerima file laporan hasil audit pemindaian kerentanan keamanan websitenya berupa excel yang dapat diunduh kedalam sistem lokalnya.



Gambar 4.30 Tampilan laporan hasil audit kerentanan keamanan *website*

● Pada laporan hasil audit kerentanan keamanan website ini, terdapat informasi berupa daftar kerentanan meliputi nama isu kerentanan, tingkat kerentanan, detail mengenai isu kerentanan (*matcher name*) dan lainnya yang dapat membantu untuk dilakukannya perbaikan penambalan kerentanan keamanan website.



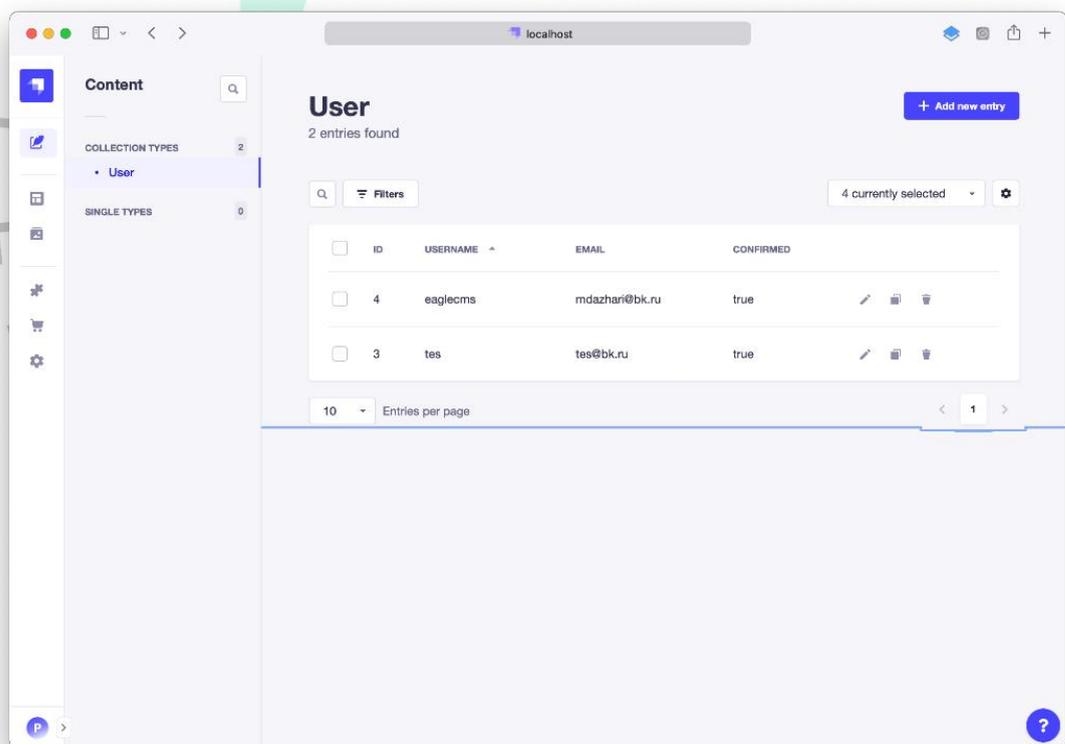
Gambar 4.31 Tampilan referensi OWASP meninjau isu kerentanan keamanan *website*

Pada salah satu isu keamanan kerentanan website yang terjadi pada **Gambar 4.30** yaitu isu *Reverse Tabnabbing*, informasi isu ini dapat juga ditemukan didalam website OWASP (*Open Web App Security Project*) yang membantu kita mengetahui tentang informasi detail suatu isu kerentanan keamanan website yang terjadi.

4.3.2 Hasil Antar Muka Admin

Pada perancangan antar muka admin dikhususkan untuk memfasilitasi admin dalam mengelola konten (*Content Management System*) khususnya pengelolaan data pengguna aplikasi pemindaian kerentanan website yang meliputi membuat, menghapus, mengedit *user*.

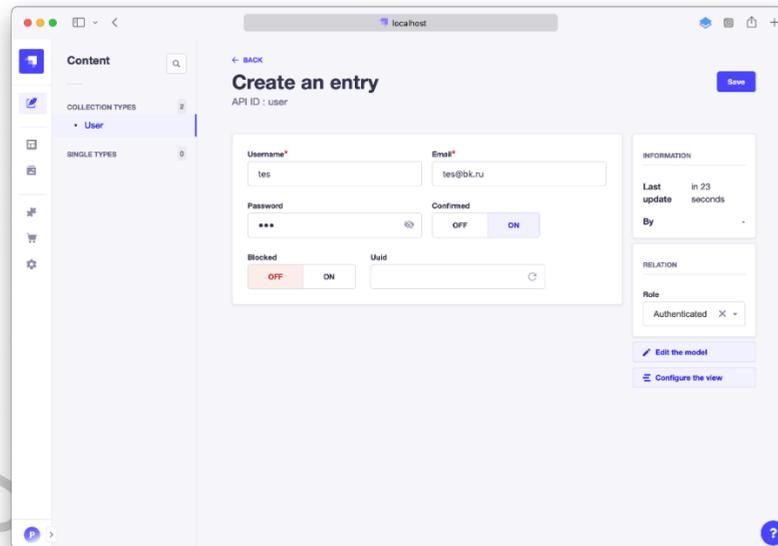
Dalam mempermudah membangun pembuatan perancangan antar muka admin ini menggunakan *headless-cms* bernama Strapi yang berbasis Node.js.



Gambar 4.32 Tampilan admin daftar *user* yang terdaftar

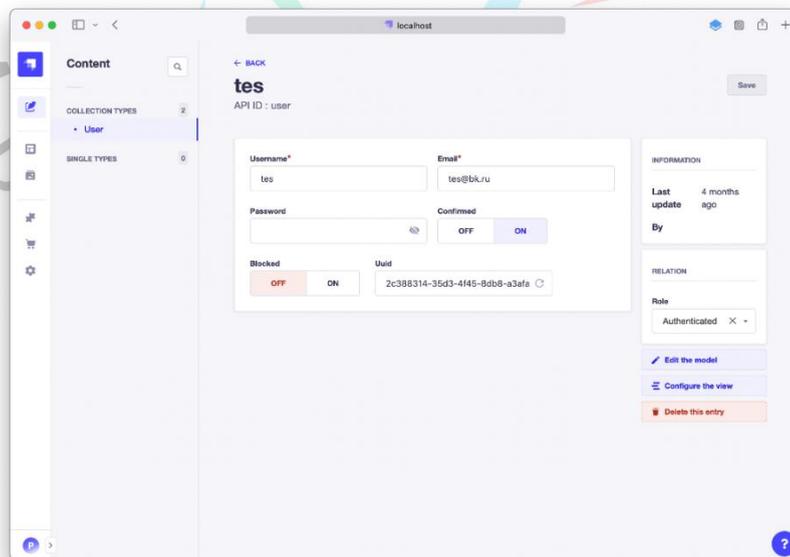
Pada tampilan daftar *user* yang terdaftar, admin dapat mengelola user segala keperluan CRUD (*Create, Read, Update & Delete*).

Pada tampilan ini admin dapat menekan tombol *Add new entry* untuk membuat *user* baru, dan admin juga dapat menghapus atau mengedit *user* tertentu dengan menekan *icon* edit atau hapus.



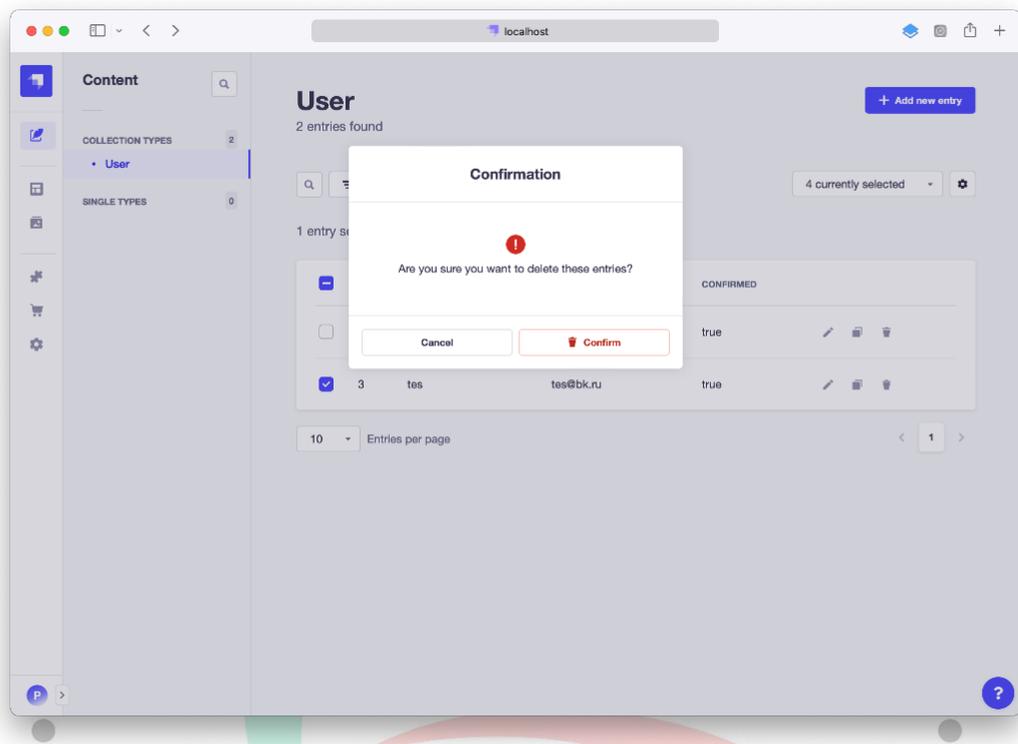
Gambar 4.33 Tampilan admin membuat *user* baru

Pada tampilan pembuatan *user*, admin dapat membuat *user* dengan mengisi form yang ada dengan lengkap. Jika valid maka *user* baru akan terbuat.



Gambar 4.34 Tampilan admin merubah *user* yang sudah ada

Pada tampilan edit *user*, admin dapat mengedit atribut *user* dengan mengisi form yang ada dengan lengkap. Jika valid maka data *user* akan diperbarui.



Gambar 4.35 Tampilan admin menghapus *user*

Pada tampilan hapus *user*, admin dapat menghapus *user* yang dipilihnya dengan menekan icon tong sampah dan menekan tombol *Confirm* untuk mengonfirmasi bahwa admin ingin benar-benar melakukan aksi penghapusan suatu akun/*user*.

4.4. Perancangan Implementasi

Perancangan Implementasi merupakan sebuah penyatuan semua rancangan yang sudah dibuat oleh Peneliti untuk membuat aplikasi Eagle tersebut.

Semua rancangan tersebut dibuat berdasarkan analisis yang sudah dilakukan penulis, diagram yang sudah disesuaikan dengan analisis perancangan aplikasi Eagle, dan tampilan yang sesuai dengan aplikasi tersebut.

Selain itu, penulis juga melakukan pengujian pada perancangan aplikasi yang telah dibuat berjalan pada semestinya, pengujian dilakukan dari aplikasi end-user hingga aplikasi untuk admin. Hasil dari percobaan aplikasi tersebut dapat dilihat pada **Tabel 4.11** dan **Tabel 4.12** dibawah.

4.4.1 Perancangan Implementasi Aplikasi Pengguna

Pada **Tabel 4.11**, dilakukannya pengujian pada aplikasi end-user pemindaian kerentanan keamanan website yang diakses oleh pengguna.

Tabel 4.11 pengujian aplikasi *end-user* pemindaian kerentanan keamanan *website*

No.	Test Name	Steps	Result
1.	Melakukan <i>Login</i> ke Aplikasi	<ol style="list-style-type: none"> 1. Buka browser 2. Buka web aplikasi Eagle 3. Menekan tombol Sign in 4. Mengisi form <i>login</i> 	Berhasil melakukan <i>login</i>
2.	Melakukan Registrasi	<ol style="list-style-type: none"> 1. Buka browser 2. Buka web aplikasi Eagle 3. Menekan tombol Start free 4. Mengisi form registrasi 	Berhasil melakukan registrasi
3.	Melakukan Keluar	<ol style="list-style-type: none"> 1. <i>Login</i> 2. Menekan tombol Sign out 	Berhasil melakukan keluar dari aplikasi
4.	Melihat daftar website yang sudah dipindai	<ol style="list-style-type: none"> 1. <i>Login</i> 2. Masuk ke halaman dashboard 	Berhasil melihat daftar website yang sudah dipindai
5.	Melakukan pemindaian baru	<ol style="list-style-type: none"> 1. <i>Login</i> 2. Masuk ke halaman dashboard 3. Mengisi form pemindaian baru dengan input web url yang dijadikan target 	Berhasil melakukan pemindaian baru
6.	Melihat status perkembangan pemindaian	<ol style="list-style-type: none"> 1. <i>Login</i> 2. Masuk ke halaman dashboard 	Berhasil melihat status perkembangan pemindaian yang sedang berjalan maupun sudah selesai
7.	Mengunduh laporan hasil audit pemindaian	<ol style="list-style-type: none"> 1. <i>Login</i> 2. Masuk ke halaman dashboard 3. Menekan tombol Download Report 	Berhasil mengunduh laporan hasil audit pemindaian berupa file excel

4.4.1 Perancangan Implementasi Aplikasi Admin

Pada **Tabel 4.12** dilakukannya pengujian pada aplikasi admin pemindaian kerentanan keamanan website yang diakses oleh admin.

Tabel 4.12 pengujian aplikasi admin pemindaian kerentanan keamanan *website*

No.	Test Name	Steps	Result
1.	Melakukan <i>Login</i> ke CMS	<ol style="list-style-type: none">1. Buka browser2. Buka web CMS Eagle3. Mengisi form <i>Login</i>	Berhasil melakukan <i>login</i>
2.	Melakukan Keluar	<ol style="list-style-type: none">1. <i>Login</i>2. Mengakses side menu3. Menekan tombol Logout	Berhasil melakukan keluar dari CMS aplikasi
3.	Melihat daftar <i>user</i> yang terdaftar	<ol style="list-style-type: none">1. <i>Login</i>2. Ke menu Users	Berhasil melihat daftar <i>user</i> yang terdaftar
4.	Membuat <i>user</i> baru	<ol style="list-style-type: none">1. <i>Login</i>2. Ke menu Users3. Menekan tombol Add new entry4. Mengisi form pembuatan <i>user</i> baru	Berhasil membuat <i>user</i> baru
5.	Mengedit <i>user</i> lama	<ol style="list-style-type: none">1. <i>Login</i>2. Ke menu Users3. Menekan icon edit pada <i>user</i> yang dipilih4. Mengisi form pengeditan <i>user</i>	Berhasil mengedit <i>user</i> lama
6.	Menghapus <i>user</i>	<ol style="list-style-type: none">1. <i>Login</i>2. Ke menu Users3. Menekan icon hapus pada <i>user</i> yang dipilih4. Menekan tombol konfirmasi penghapusan <i>user</i>	Berhasil menghapus <i>user</i>

