

BAB 4

HASIL PENELITIAN

4.1 Ringkasan Hasil Penelitian

Penelitian rancang bangun aplikasi *mobile vulnerability assessment* sesuai judul dirancang menggunakan *framework* Flutter dan bahasa pemrograman Dart serta database PostgreSQL ini membantu user untuk melakukan *vulnerability assessment* dengan baik dan mudah.

Infrastruktur aplikasi memisahkan antara *Frontend*, *Backend* dan *Server*. Di mana *Frontend* memiliki 3 halaman yaitu login, home dan settings. Bagian backend sendiri terdapat beberapa *routes* untuk mengelolah *API* yang dibutuhkan oleh *Frontend*, sementara *Server* melakukan proses *scanning* dan *reporting* kemudian dikirimkan ke supabase.

Berdasarkan analisis kebutuhan sistem yang dibangun, user sudah bisa melakukan tambah domain target yang nanti akan dilakukan proses *scanning* di perangkat masing-masing. Setelah proses *scanning* selesai kemudian data akan ditampilkan di halaman home, *user* bisa melihat hasil yang telah selesai serta bisa mendownload jika perlu.

4.2 Analisis Kebutuhan

Analisis ini bertujuan untuk mengidentifikasi fitur-fitur penting yang harus tersedia dalam aplikasi sesuai dengan tujuan penggunaan. Berikut adalah hasil analisis kebutuhan pengguna:

Tabel 4. 1 Daftar Kebutuhan Pengguna

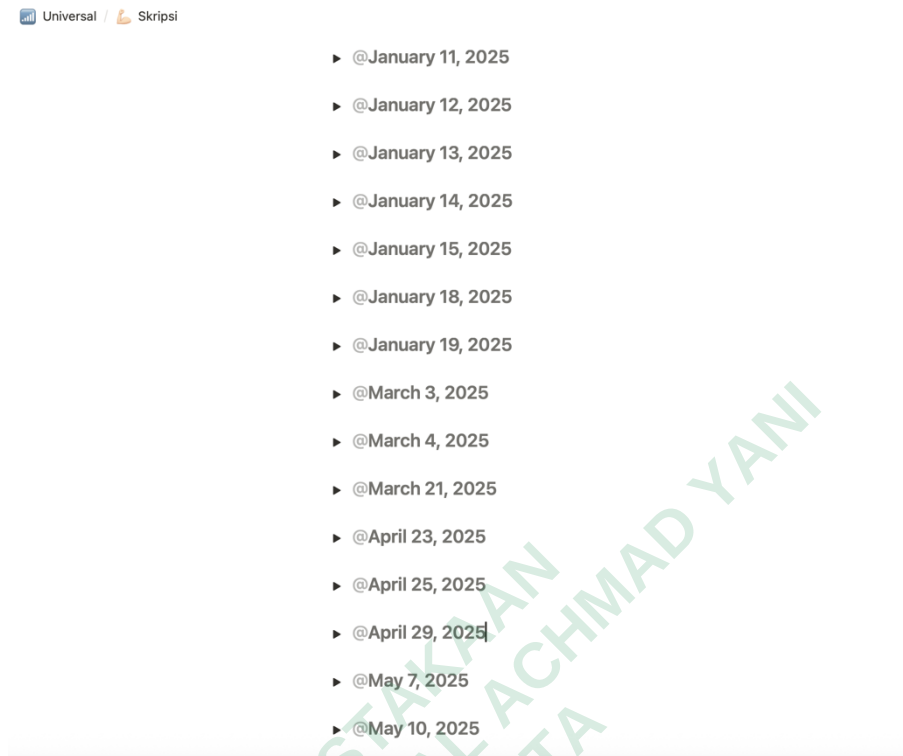
No	Kebutuhan pengguna	Deskripsi	Fitur
1	Pemindaian otomatis terhadap domain target	Pengguna membutuhkan cara cepat untuk memindai keamanan domain tanpa konfigurasi manual.	Pemindaian berjalan otomatis setelah input domain.
2	Laporan hasil deteksi kerentanan	Pengguna ingin melihat hasil pemindaian dalam format yang mudah	Hasil scan ditampilkan langsung di

		dipahami dan terdokumentasi.	aplikasi dan dapat diunduh dalam format PDF.
3	Integrasi dengan tool keamanan (Sn1per)	Pengguna menginginkan sistem terintegrasi dengan tools yang telah terbukti efektif.	Backend terhubung otomatis dengan Sn1per di server VPS.
4	Akses cepat melalui perangkat mobile	Pengguna membutuhkan kemudahan akses di mana saja.	Aplikasi tersedia di platform Android dan iOS.
5	Otentikasi yang aman dan praktis	Pengguna tidak ingin membuat akun baru untuk masuk ke aplikasi.	Fitur login dengan akun Google menggunakan OAuth2.

4.3 Dokumentasi Iteratif

Penelitian ini menggunakan metode pengembangan perangkat lunak Agile, yang bersifat iteratif dan fleksibel. Dalam implementasinya, proses pengembangan dilakukan secara bertahap melalui beberapa iterasi, di mana setiap iterasi difokuskan pada penyelesaian fitur tertentu dan menghasilkan versi aplikasi yang dapat langsung diuji dan dievaluasi.

Berikut adalah dokumentasi tahapan iteratif yang dilakukan selama proses pembuatan aplikasi:



Gambar 4. 1 Dokumentasi Iteratif

1. Iterasi Pertama

Fokus pada proses awal aplikasi, yaitu mengirimkan domain target dari client ke server, mulai dari tanggal 11 Januari – 15 Januari 2025.

- a. Pembuatan halaman input domain di aplikasi mobile (Flutter).
- b. Koneksi awal Flutter ke broker *MQTT*.
- c. Pembuatan endpoint di backend Express.js untuk menerima data dan meneruskannya ke server.
- d. Pengujian pengiriman pesan *MQTT* antara client dan backend.

2. Iterasi Kedua

Fokus pada integrasi proses scanning menggunakan Sn1per, mulai dari tanggal 18 Januari – 3 Maret 2025.

- a. Konfigurasi Sn1per pada server VPS untuk dijalankan otomatis.
- b. Pengiriman hasil scanning ke backend, kemudian dikirim ke aplikasi melalui *MQTT*.
- c. Tampilan awal hasil scanning di aplikasi.

3. Iterasi Ketiga

Fokus pada pengelolaan data hasil scanning dan fitur tambahan, mulai dari tanggal 4 Maret – 25 April 2025.

- a. Implementasi fitur *download* laporan hasil *scanning* (HTML ke PDF).
- b. Penyimpanan hasil scanning ke database Supabase.
- c. Penambahan fitur Google Sign-In untuk autentikasi pengguna.
- d. Pembuatan halaman pengaturan (Settings).

4. Iterasi Keempat

Tahap terakhir fokus pada validasi aplikasi mulai dari tanggal 29 April – 10 Mei 2025.

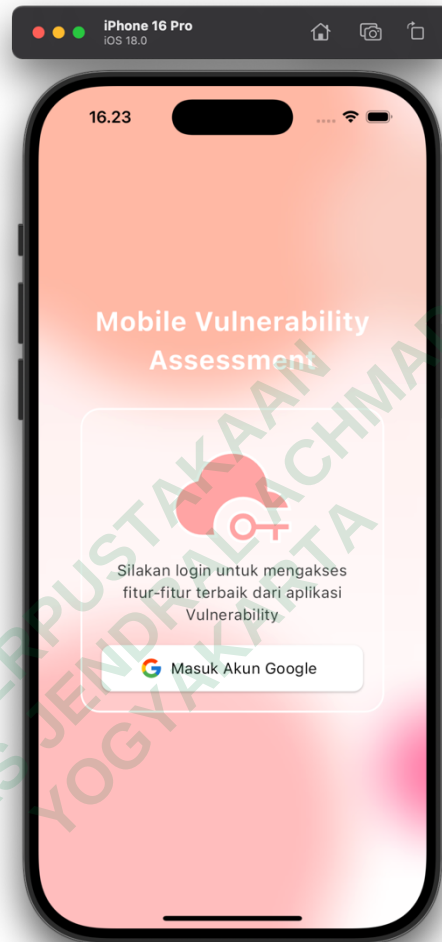
- a. Pengujian *Black-box Testing* terhadap seluruh fungsionalitas utama.
- b. Pengujian koneksi *MQTT* dalam berbagai kondisi jaringan.
- c. Pengumpulan umpan balik dari pengguna non-teknis terkait usability.
- d. Penyempurnaan UI/UX berdasarkan hasil pengujian.

4.4 Implementasi Desain Antarmuka

Implementasi antar muka merupakan tahapan yang penting dalam memenuhi kebutuhan user. Implementasi antar muka menerapkan tampilan yang akan dilihat oleh user dilakukan dengan cara membuat desain antar muka pada perangkat mobile yang akan digunakan. Antar muka yang ditampilkan dengan baik dapat membantu pemakai dalam memahami proses yang sedang dilakukan sistem tersebut. Berikut ini implementasi sistem antar muka:

1. Tampilan Halaman Login

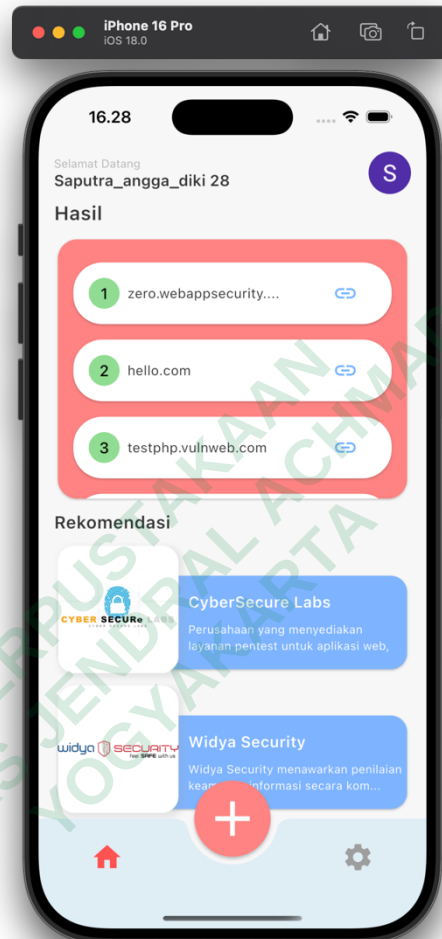
Gambar 4. 2 ini merupakan hasil implementasi halaman login yang telah dikerjakan setelah proses rancangan.



Gambar 4. 2 Implementasi Tampilan Halaman Login

2. Tampilan Halaman Home

Gambar 4. 3 ini merupakan hasil implementasi halaman home yang telah dikerjakan setelah proses rancangan.



Gambar 4. 3 Implementasi Tampilan Halaman Home

3. Tampilan Menambah Target Scan

Gambar 4. 4 ini merupakan hasil implementasi menambah target yang telah dikerjakan setelah proses rancangan.



Gambar 4. 4 Implementasi Form Tambah Domain Target

4. Tampilan Menghapus Data Hasil Scan

Gambar 4. 5 ini merupakan hasil implementasi menghapus data hasil scan yang telah dikerjakan setelah proses rancangan.

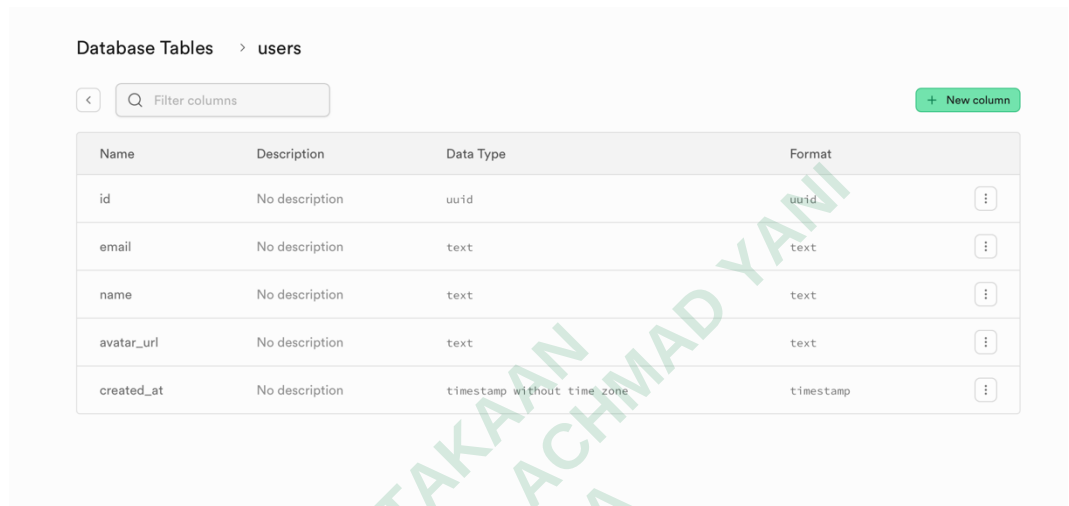


Gambar 4. 5 Implementasi Hapus Target Scan

4.5 Basis Data

Implementasi database untuk penelitian ini menggunakan bantuan *Supabase* yang berbasis Postgresql.

1. Tabel Users



Database Tables > users

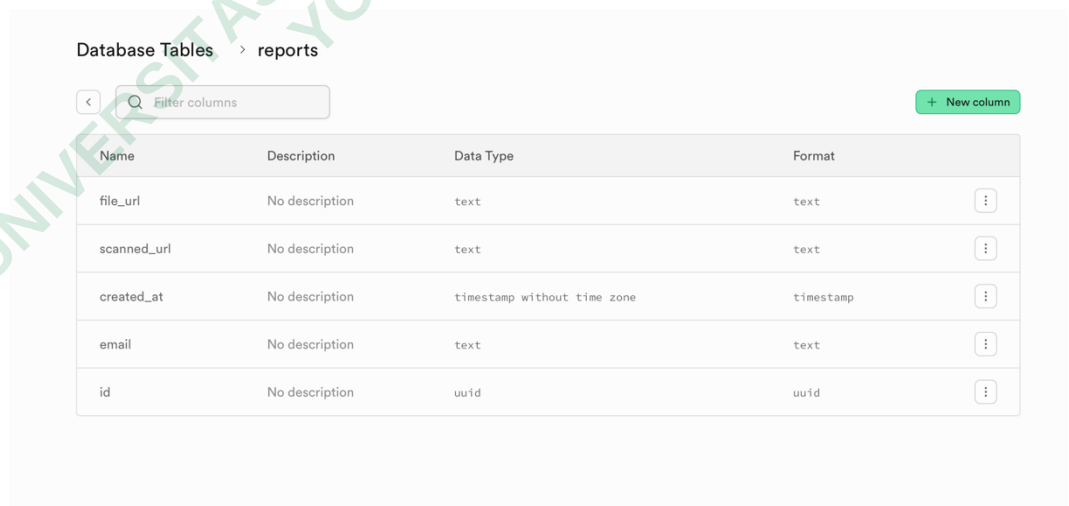
< Filter columns + New column

Name	Description	Data Type	Format
id	No description	uuid	uuid
email	No description	text	text
name	No description	text	text
avatar_url	No description	text	text
created_at	No description	timestamp without time zone	timestamp

Gambar 4. 7 Tabel Users di Supabase

Berisikan id, email, name, avatar_url, created_at yang menyimpan data dari user ketika login.

2. Tabel Reports



Database Tables > reports

< Filter columns + New column

Name	Description	Data Type	Format
file_url	No description	text	text
scanned_url	No description	text	text
created_at	No description	timestamp without time zone	timestamp
email	No description	text	text
id	No description	uuid	uuid

Gambar 4. 8 Tabel Reports di Supabase

Berisikan id, file_url, scanned_url, created_at, email yang menyimpan hasil scan domain target yang di input oleh masing masing user berdasarkan email.

3. Tabel Rekomendasi

Database Tables > rekomendasi

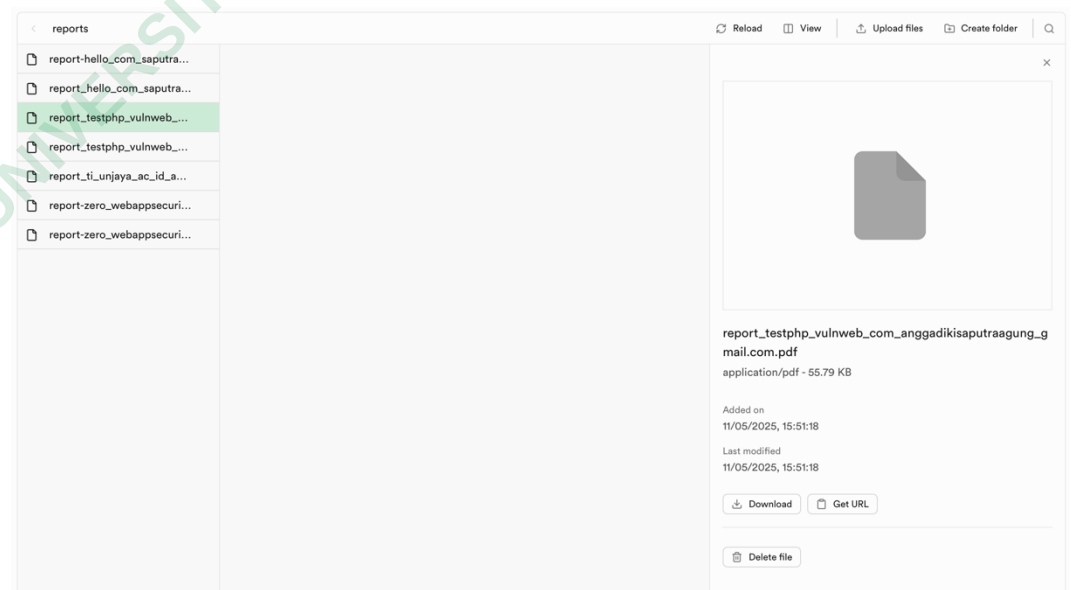
< Filter columns + New column

Name	Description	Data Type	Format
id	No description	uuid	uuid
gambar	No description	text	text
nama_perusahaan	No description	text	text
deskripsi	No description	text	text
link_perusahaan	No description	text	text
created_at	No description	timestamp with time zone	timestampz

Gambar 4. 9 Tabel Rekomendasi di Supabase

Berisikan id, gambar, nama_perusahaan, deskripsi, link_perusahaan, created_at yang menyimpan rekomendasi start up atau perusahaan yang menyediakan jasa *vulnerability assessment* secara lebih professional.

4. Bucket Storage

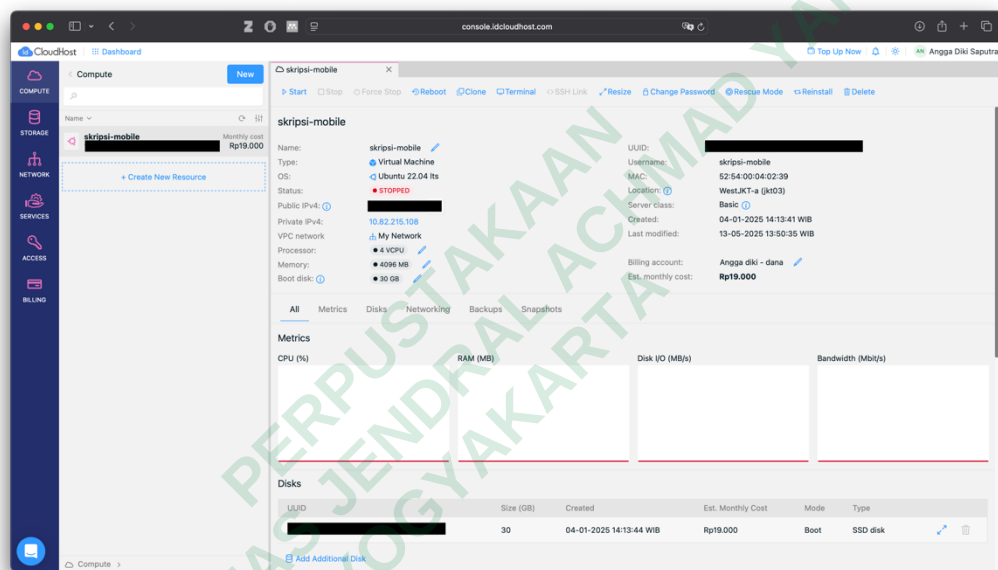


Gambar 4. 10 Storage Untuk Menyimpan Pdf Hasil Scan

Berisikan Bucket reports yang digunakan untuk menyimpan dan mengakses *public url* hasil *scanning* untuk di simpan ke dalam tabel *reports*.

4.6 Implementasi Vps

Implementasi untuk memproses domain yang telah dikirim dari perangkat mobile dibutuhkan *VPS* untuk mengolah proses tersebut. Mulai dari *install* dan konfigurasi *MQTT*, membuat skrip untuk proses otomatis menerima dan mengirim hasil scan dari *tools Snlper*.

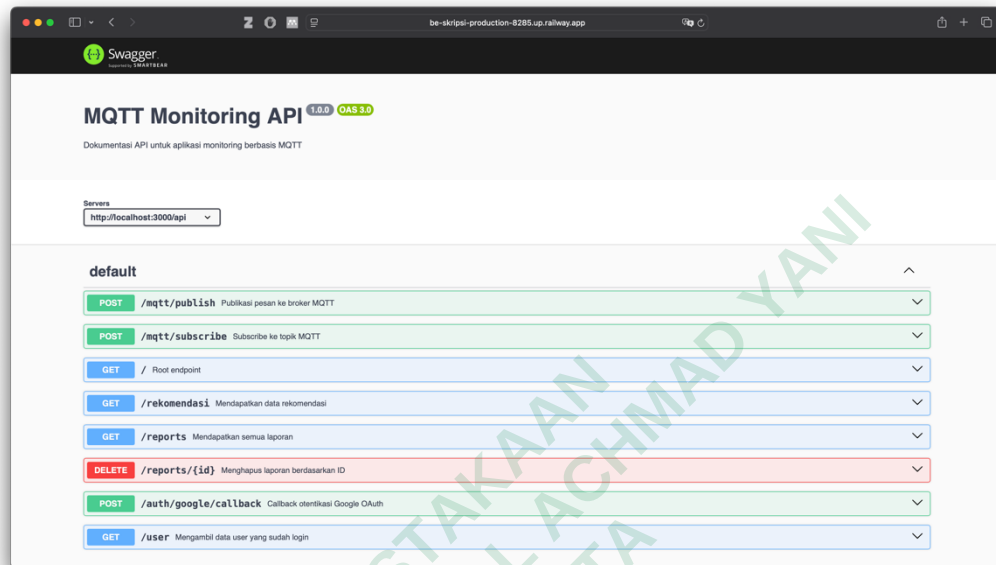


Gambar 4. 11 Vps Untuk Memproses Data

4.7 Implementasi Backend

Di sisi backend, keamanan juga menjadi fokus utama dalam implementasi sistem. Setiap permintaan dari *frontend* yang masuk ke *backend* Express.js dilindungi menggunakan token autentikasi yang diperoleh dari proses *login* Google. *Token* ini diverifikasi pada setiap permintaan untuk memastikan bahwa hanya pengguna yang telah terautentikasi yang dapat mengakses endpoint penting, seperti pengiriman domain dan pengambilan hasil scan. Selain itu, koneksi antara *frontend*, *backend*, dan *broker MQTT* telah diamankan menggunakan TLS untuk mengenkripsi data yang dikirimkan. Variabel sensitif seperti kredensial koneksi, *URL* Supabase, dan konfigurasi token disimpan menggunakan file *.env* untuk mencegah kebocoran informasi penting dalam kode sumber. Seluruh backend

dideploy menggunakan Railway, yang juga mendukung manajemen environment secara terpisah antara pengembangan dan produksi, sehingga keamanan API dan stabilitas layanan tetap terjaga.



Gambar 4. 12 Dokumentasi Api Menggunakan Swagger

Gambar di atas merupakan dokumentasi *endpoint* apa saja yang dibuat untuk memenuhi kebutuhan *Frontend*.

4.8 Pengujian Black-Box

Pengujian *Black-Box* memungkinkan untuk memperoleh sekumpulan kondisi yang akan menjalankan semua kebutuhan fungsionalitas untuk sebuah sistem. Pengujian dilakukan terhadap tiga halaman utama pada perangkat Android dan iOS, yaitu halaman Login, Home, dan Settings.

1. Pengujian halaman login

Hasil pengujian pada halaman login, dapat dilihat pada tabel di bawah

Tabel 4. 2 Hasil Pengujian Login Page

No	Fungsi yang Diuji	Cara Pengujian	Hasil yang Diharapkan	Keterangan
1	Load data halaman login	Masuk ke halaman login	Berhasil menampilkan halaman login	Berhasil

No	Fungsi yang Diuji	Cara Pengujian	Hasil yang Diharapkan	Keterangan
2	Login Google	Melakukan klik pada button google	Berhasil masuk ke dalam sistem	Berhasil

2. Pengujian halaman home

Hasil pengujian pada halaman home, dapat dilihat pada tabel di bawah

Tabel 4. 3 Hasil Pengujian Home Page

No	Fungsi yang Diuji	Cara Pengujian	Hasil yang Diharapkan	Keterangan
1	Load data halaman home	Masuk ke halaman home pengguna baru	Berhasil masuk ke halaman home dan data hasil scan kosong	Berhasil
2	Load data halaman home	Masuk ke halaman home pengguna lama	Berhasil masuk ke halaman home dan data hasil scan tampil	Berhasil
3	Tambah data target scan	Klik tombol dan masukan nama domain dengan valid	Berhasil mengirim data dan muncul <i>floating message</i>	Berhasil
4	Tambah data target scan	Klik tombol dan masukan nama domain dengan tidak valid	Muncul warning nama domain tidak valid	Berhasil
5	Lihat data hasil scan	Klik tombol link hasil scan	Berhasil mengarah ke url pdf hasil scan	Berhasil
6	Hapus data target scan	Tahan satu item data hasil scan	Muncul allert message yakin ingin menghapus data	Berhasil
7	Rekomendasi perusahaan	Klik tombol card rekomendasi	Berhasil mengarahkan ke url perusahaan	Berhasil

3. Pengujian halaman settings

Hasil pengujian pada halaman settings, dapat dilihat pada tabel di bawah

Tabel 4. 4 Hasil Pengujian Settings Page

No	Fungsi yang Diuji	Cara Pengujian	Hasil yang Diharapkan	Keterangan
1	Load data halaman settings	Masuk ke halaman settings	Berhasil masuk ke halaman settings	Berhasil
2	Mode malam	Klik tombol mode malam	Berhasil merubah tema menjadi gelap	Berhasil
3	Notifikasi hasil scanning	Klik tombol hasil scanning	Muncul notifikasi hasil scanning terakhir	Berhasil
4	Log out	Melakukan klik tombol log out	Berhasil melakukan logout	Berhasil

4.9 Pembahasan

Setelah melakukan proses implementasi sistem, akan dilakukan pembahasan lebih lanjut hasil penelitian Rancang Bangun Aplikasi *Mobile Vulnerability Assessment* Menggunakan *Framework* Flutter Dan Protokol *MQTT*.

4.9.1 Rancangan Aplikasi

Perancangan aplikasi dimulai dengan penentuan arsitektur sistem yang memanfaatkan pendekatan *modular client-server*. Aplikasi mobile dibangun menggunakan Flutter sebagai teknologi *frontend* untuk menghasilkan antarmuka pengguna yang responsif dan lintas platform. Komunikasi antara aplikasi dan *server* dilakukan menggunakan protokol *MQTT*, yang dipilih karena sifatnya yang ringan dan cocok untuk pengiriman data secara real-time dalam konteks pemindaian kerentanan. Komponen utama dalam perancangan mencakup halaman autentikasi, halaman utama untuk input domain, halaman hasil pemindaian, dan halaman pengaturan.

Desain antarmuka aplikasi dibuat intuitif dengan mempertimbangkan *user experience*, seperti penggunaan ikon, warna, dan navigasi yang konsisten. Proses pemindaian dirancang mengalir dari *input domain*, pengiriman data via *MQTT*, pemrosesan di server menggunakan *SnIper*, dan pengembalian hasil ke *client* dalam bentuk ringkasan kerentanan.

4.9.2 Implementasi Sistem

Implementasi diawali dengan integrasi Google Sign-In melalui Google Cloud Authentication untuk mengelola login pengguna secara aman dan tanpa registrasi manual. Setelah autentikasi berhasil, pengguna diarahkan ke halaman Home, yang memungkinkan mereka menambahkan domain untuk dianalisis. Data dikirimkan ke server menggunakan protokol *MQTT* dan diproses oleh mesin *scanning*.

Setelah proses scan selesai, hasil dikirim ke *storage* Supabase dan ditampilkan pada halaman Hasil, lengkap dengan informasi tingkat kerentanan (*Critical, High, Medium, Low*), teknologi yang terdeteksi, serta port dan service yang terbuka. Selain itu, halaman Settings disediakan untuk pengaturan akun, termasuk logout, preferensi notifikasi, serta opsi bantuan.

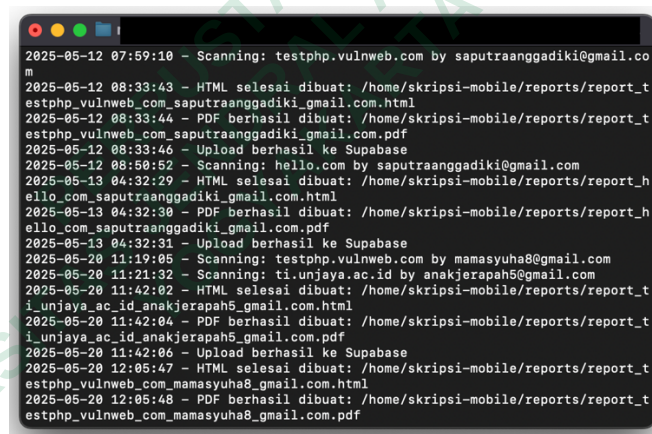
Untuk menjaga keamanan dan privasi pengguna, aplikasi ini mengadopsi pendekatan autentikasi berbasis OAuth 2.0 melalui Google Sign-In, yang menyediakan lapisan proteksi tambahan dengan verifikasi token yang terenkripsi. Penggunaan autentikasi eksternal ini tidak hanya menghindari penyimpanan kredensial di sisi *server*, tetapi juga meminimalkan risiko serangan seperti pencurian password atau *brute-force login*. Setiap komunikasi data antara aplikasi dan server juga dilakukan melalui protokol *MQTT* yang dikonfigurasi agar hanya menerima koneksi dari klien yang telah terautentikasi. Dengan pendekatan ini, akses terhadap fitur sensitif seperti pengiriman domain dan pengambilan hasil scanning menjadi lebih aman, menjaga integritas data, dan melindungi pengguna dari akses tidak sah.

4.9.3 Pengujian dan Evaluasi

Pengujian dilakukan dengan metode Black-Box Testing untuk memverifikasi semua fungsionalitas utama aplikasi pada perangkat Android dan iOS. Pengujian mencakup:

1. Login menggunakan akun Google = berhasil dan stabil
2. Penambahan domain dan pengiriman via *MQTT* = data diterima oleh *server* tanpa *error*.
3. Pemrosesan hasil dan pengembalian data = data berhasil ditampilkan di aplikasi
4. Navigasi antar halaman (login, home, settings) = berjalan tanpa hambatan

Kemudian dilakukan pengujian koneksi mqtt dengan cara mengirimkan dan menerima pesan melalui topik menggunakan client *mosquitto_pub* dan *mosquitto_sub*. Hasil pengujian menunjukkan bahwa:



Gambar 4. 13 Log Pengujian MQTT Di Server

1. Client berhasil terhubung ke broker MQTT pada IP VPS.
2. Tidak terdapat error atau delay signifikan saat proses komunikasi.
3. Broker menerima koneksi secara konsisten dan stabil selama pengujian berlangsung.

Sebagai tambahan, dilakukan pengujian perbandingan hasil pemindaian terhadap 2 website berbeda:

1. <http://testphp.vulnweb.com> menunjukkan beberapa kerentanan, dengan banyak informasi sensitif terbuka, dapat dilihat dari Gambar 4. 14 di bawah.

Laporan Hasil Scan Website	
Informasi Umum	
URL:	testphp.vulnweb.com
Tanggal Scan:	12 May 2025
Email:	saputraanggadiki@gmail.com
Server:	nginx/1.19.0
TLS Support:	Tidak ditemukan
Ringkasan Kerentanan	
Tingkat	Jumlah
Critical	0
High	4
Medium	164
Low	0
Info	97
Total	265
Detail Kerentanan	

Gambar 4. 14 Laporan Pdf Hasil Scanning (<http://testphp.vulnweb.com>)

2. <https://hello.com> menunjukkan sedikit kerentanan terkait di Gambar 4.15

Laporan Hasil Scan Website	
Informasi Umum	
URL:	hello.com
Tanggal Scan:	13 May 2025
Email:	saputraanggadiki@gmail.com
Server:	Google Frontend
TLS Support:	Tidak ditemukan
Ringkasan Kerentanan	
Tingkat	Jumlah
Critical	0
High	0
Medium	0
Low	8
Info	176
Total	184
Detail Kerentanan	

Gambar 4. 15 Laporan Pdf Hasil Scanning (<https://hello.com>)

Hasil pengujian menunjukkan bahwa aplikasi mampu menampilkan perbedaan tingkat kerentanan tergantung pada karakteristik masing-masing website. Ini membuktikan bahwa sistem bekerja secara dinamis dan sesuai dengan konteks target yang dipindai. Selain itu, pengujian pada perangkat Android dilakukan pada 5 pengguna nyata dan menunjukkan tingkat keberhasilan 100% dalam seluruh uji fungsionalitas, termasuk login, pengiriman domain, pemrosesan hasil scan, dan navigasi antar halaman. Untuk pengguna iOS, pengujian dilakukan melalui virtual device dan juga menunjukkan hasil keberhasilan 100%. Hal ini mengindikasikan bahwa aplikasi berjalan stabil dan konsisten di kedua platform utama.

PERPUSTAKAAN
UNIVERSITAS JENDRAL ACHMAD YOHANES
YOGYAKARTA