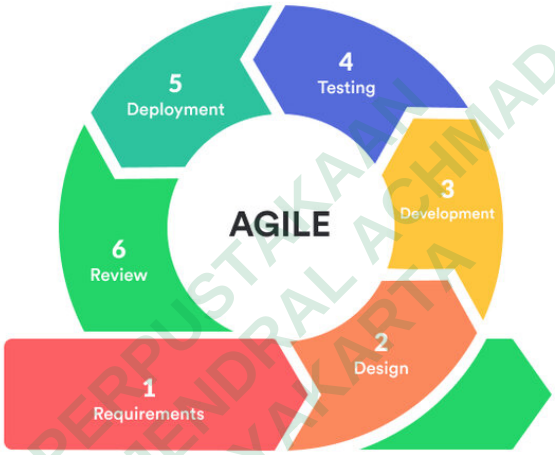
BAB 3
METODE PENELITIAN

Metode penelitian yang digunakan dalam penelitian ini adalah metode *SDLC (Software Development Life Cycle)* dengan model *Agile*. ini dipilih karena bersifat iteratif dan memungkinkan pengembangan aplikasi dilakukan secara bertahap dengan evaluasi berkelanjutan. Tahapan pengembangan meliputi:



Gambar 3. 1 Metode Agile

1. Analisis Kebutuhan:
   a. Menggali kebutuhan pengguna melalui studi literatur dan evaluasi alat bantu yang sudah ada.
   b. Mengidentifikasi fitur penting, seperti pemindaian otomatis, laporan hasil deteksi, dan integrasi dengan *Sn1per*.
2. Perancangan Sistem:
   a. Mendesain arsitektur sistem, termasuk struktur komunikasi antara aplikasi dan server menggunakan protokol *MQTT*.
   b. Membuat diagram UML, perancangan database dan rancangan desain antar muka.

3. Implementasi:
    a. Mengembangkan aplikasi menggunakan *Flutter* untuk antarmuka pengguna dan mengintegrasikan *Sn1per* untuk proses *vulnerability assessment*.
    b. Menghubungkan aplikasi dengan server VPS melalui protokol *MQTT* untuk menerima hasil pemindaian secara *real-time*. Ketika ada request *user* langsung akan diproses dan setelah selesai *report* dihasilkan.
4. Pengujian:
    a. Melakukan uji fungsionalitas untuk memastikan fitur utama bekerja sesuai harapan.
    b. Menggunakan metode *black-box testing* untuk menguji setiap komponen aplikasi tanpa melihat kode sumber.
5. Evaluasi dan Penyempurnaan:
    a. Mengumpulkan umpan balik dari pengujian dan memperbaiki kekurangan yang ditemukan.
    b. Melakukan pengujian ulang setelah perbaikan untuk memastikan peningkatan performa dan akurasi aplikasi.

3.1 Bahan dan Alat Penelitian

Bahan yang diperlukan dalam penelitian ini adalah website target yang telah diinput oleh *user* kemudian dilakukan proses *scanning*. Alat yang diperlukan dalam penelitian ini yaitu perangkat keras dan perangkat lunak yang mendukung agar setiap tahapan berjalan optimal. Perangkat yang digunakan mencakup kebutuhan untuk mengembangkan fitur, menguji keamanan, serta memastikan performa aplikasi stabil di berbagai platform. Berikut adalah rincian perangkat keras dan perangkat lunak yang digunakan:

Tabel 3. 1 Perangkat Keras

No	Kategori	Penggunaan	Deskripsi
1.	Komputer PC/Laptop	Pengembangan	macOS Sequoia 15.02 atau lebih baru

No	Kategori	Penggunaan	Deskripsi
2.	Perangkat Mobile	Pengujian	Smartphone Android dan iOS untuk pengujian aplikasi
3.	Server	Pengujian	Server VPS Ubuntu untuk menjalankan Sn1per

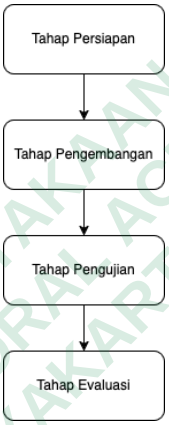
Tabel 3. 2 Perangkat Lunak

No	Kategori	Penggunaan	Deskripsi
1.	Framework	Pengembangan	Flutter sebagai kerangka kerja pengembangan aplikasi lintas platform
2.	Bahasa Pemrograman	Pengembangan	Dart sebagai bahasa pemrograman utama aplikasi
3.	Backend Framework	Pengembangan	Express.js sebagai kerangka kerja untuk backend
4.	Bahasa Pemrograman	Pengembangan	JavaScript sebagai bahasa pemrograman backend
5.	Editor Kode	Pengembangan	Visual Studio Code untuk menulis code aplikasi
6.	Protokol Komunikasi	Pengembangan	*MQTT* sebagai protokol komunikasi data ringan
7.	Penilaian Keamanan	Pengujian	*Sn1per* untuk *vulnerability assessment* otomatis
8.	Pengujian *API*	Pengujian	Postman untuk pengujian *API*

No	Kategori	Penggunaan	Deskripsi
9.	Basis Data	Pengembangan & Pengujian	Supabase sebagai basis data lokal maupun *production*

3.2 Jalan Penelitian

Jalan Penelitian ini terdiri dari beberapa tahap,yang bisa dilihat dalam gambar di bawah, yaitu sebagai berikut:



Gambar 3. 2 Jalan Penelitian

1. Tahap Persiapan:
   a. Studi literatur tentang *vulnerability assessment*, *Flutter*, *MQTT*, dan *Sn1per*.
   b. Pengumpulan data terkait kebutuhan pengguna dan fitur yang diharapkan.
   c. Menentukan fitur utama, menyusun backlog, dan merancang arsitektur awal.
2. Tahap Pengembangan:
   a. Perancangan, implementasi, dan pengujian aplikasi secara bertahap menggunakan pendekatan Agile.
   b. Melakukan uji internal dan menerima masukan untuk perbaikan iterasi berikutnya.

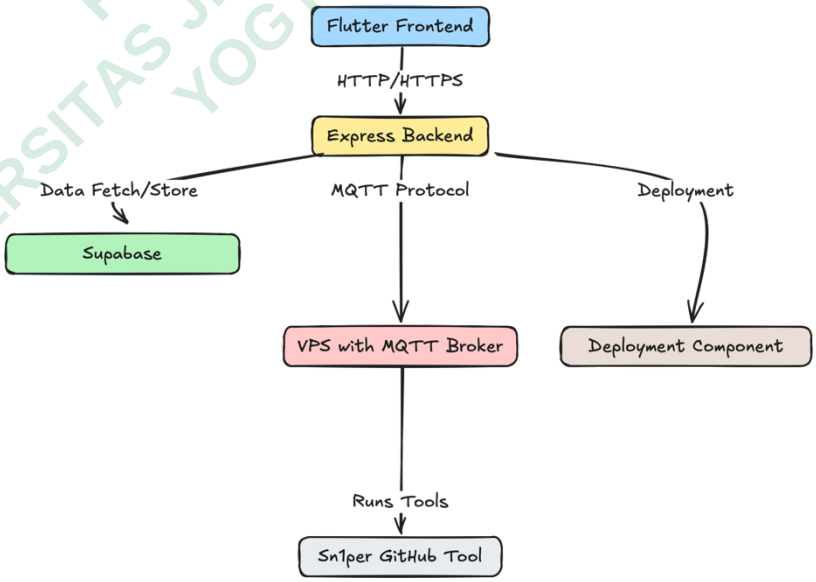
3. Tahap Pengujian
    a. Pengujian terakhir dilakukan oleh *end user* dengan menguji aplikasi yang telah melawati fase *development.*
4. Tahap Evaluasi dan Penyusunan Laporan:
    a. Evaluasi performa aplikasi berdasarkan kecepatan pemrosesan, dan kemudahan penggunaan.
    b. Penyusunan laporan hasil penelitian, termasuk dokumentasi system dan hasil pengujian.

3.3 Perancangan Sistem

Perancangan sistem yang akan dibuat oleh penulis ini sesuai dengan analisa yang telah dilakukan. Berdasarkan analisis tersebut dapat dibuat perancangan sistem sebagai berikut yang diharapkan dapat menjadi solusi dari analisa masalah yang ada.

Perancangan ini dibantu menggunakan UML sebagai model sistem yang nantinya dibangun.

1. Rancangan Arsitektur Sistem



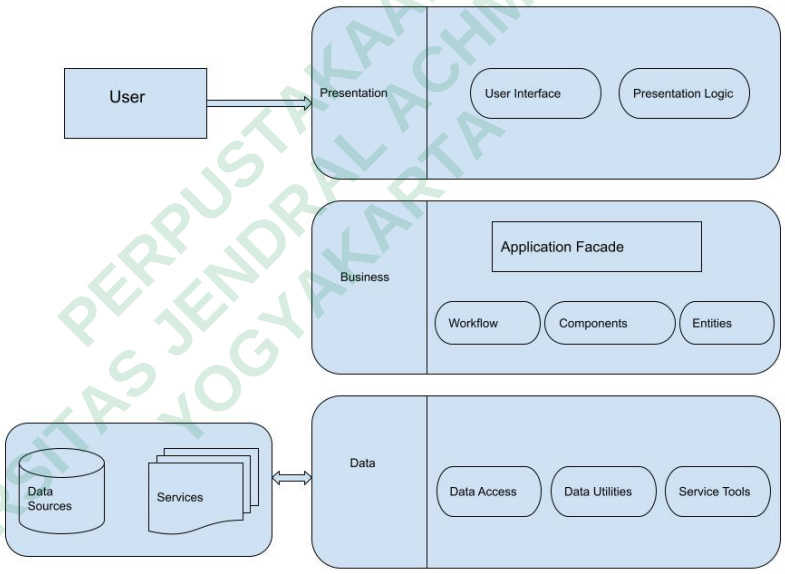
Gambar 3. 3 Rancangan Arsitektur Sistem

Gambar di atas menunjukkan arsitektur sistem yang mengintegrasikan Flutter sebagai frontend, Express sebagai

backend, dan Supabase sebagai database opsional. *Backend* berkomunikasi dengan VPS menggunakan protokol *MQTT*, yang berperan sebagai broker untuk mengelola pesan. VPS ini juga menjalankan *Sn1per GitHub Tool* sebagai alat utama untuk aktivitas pra-pentest. Selain itu, ada komponen deployment yang menghubungkan backend untuk proses otomatisasi.

2. Diagram Arsitektur Aplikasi

   Setelah membuat rancangan arsitektur sistem, maka Langkah selanjutnya membuat diagram arsitektur aplikasi,yang secara umum terdiri dari 3 lapisan yaitu:



Gambar 3. 4 Diagram Arsitektur Aplikasi

Client Layer, atau lapisan presentasi, merupakan bagian dari aplikasi mobile yang berfungsi sebagai antarmuka pengguna. Di sini, pengguna dapat berinteraksi langsung dengan aplikasi melalui berbagai halaman dan fitur. Beberapa contoh halaman yang termasuk dalam lapisan ini adalah Halaman Depan (yang mungkin berfungsi sebagai halaman awal), Login Google (untuk proses autentikasi menggunakan akun Google), dan Main Menu (menu

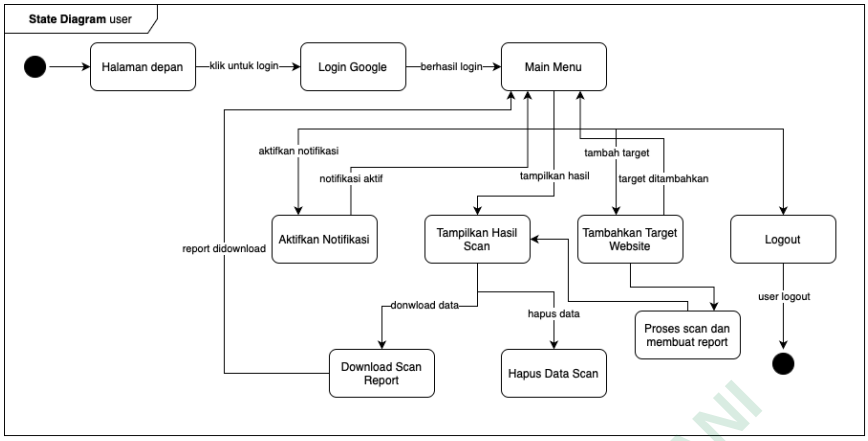
utama aplikasi). Selain itu, terdapat juga halaman seperti Tampilkan Hasil Scan (untuk menampilkan hasil scan), Tambahkan Target Website (untuk memasukkan target website yang akan di-scan), Download Scan Report (opsi mengunduh laporan), dan Hapus Data Scan (opsi menghapus data scan yang telah dilakukan).

Application Layer, atau lapisan logika bisnis, bertanggung jawab untuk menangani semua proses dan logika inti dari aplikasi. Lapisan ini mencakup berbagai layanan seperti Authentication Service, yang mengelola proses login termasuk integrasi dengan Google. Selain itu, terdapat Scan Service yang bertugas memproses logika bisnis terkait scanning website, termasuk menghasilkan dan memproses hasil scan. Report Service juga termasuk dalam lapisan ini, yang mengelola pembuatan dan pengunduhan laporan scan. Terakhir, Data Management Service berperan dalam menghapus data scan serta mengelola data lainnya yang terkait dengan aplikasi.

Data Layer merupakan lapisan yang bertugas menyimpan dan mengelola semua data yang digunakan oleh aplikasi. Lapisan ini terdiri dari beberapa komponen penting, seperti User Database yang menyimpan informasi pengguna, termasuk data login dan preferensi. Scan Database digunakan untuk menyimpan data hasil scan serta target website yang telah dimasukkan oleh pengguna. Selain itu, terdapat Report Storage yang berfungsi sebagai penyimpanan untuk laporan scan yang dapat diunduh oleh pengguna. Lapisan ini memastikan bahwa semua data aplikasi tersimpan dengan aman dan dapat diakses ketika dibutuhkan.

3. State Diagram

   Berdasarkan rancangan arsitektur yang telah dibuat,maka state diagram dari aktivitas yang ada di aplikasi *vulnerability assessment* yaitu sebagai berikut:



Gambar 3. 5 State Diagram User

State diagram ini diawali dengan Initial State yang ditunjukkan oleh simbol lingkaran hitam penuh. Pada tahap ini, pengguna berada dalam kondisi awal, yaitu pada state "Halaman Depan". Dari sini, pengguna dapat memulai proses dengan mengklik tombol login, yang merupakan sebuah event pertama. Event ini membawa pengguna ke state "Login Google", tempat pengguna diminta untuk melakukan autentikasi menggunakan akun Google. Setelah proses login berhasil, pengguna akan berpindah ke state "Main Menu" yang berfungsi sebagai pusat navigasi ke berbagai fitur dalam aplikasi.

Dalam state "Main Menu", pengguna memiliki beberapa opsi transisi ke state lain berdasarkan event yang terjadi. Misalnya, ketika pengguna memilih untuk menambahkan target website, maka transisi menuju state "Tambahkan Target Website" akan terjadi, dan jika proses penambahan target selesai, pengguna kembali ke "Main Menu". Pengguna juga dapat mengaktifkan notifikasi untuk mendapatkan pemberitahuan hasil scan, yang membawa pengguna ke state "Aktifkan Notifikasi". Ketika notifikasi berhasil diaktifkan, pengguna dapat melihat hasil scan melalui state "Tampilkan Hasil Scan". Dari state ini, pengguna juga memiliki opsi untuk mengunduh laporan hasil scan pada state "Download Scan Report",

atau menghapus data hasil scan yang sudah tidak dibutuhkan pada state “Hapus Data Scan”.

Proses utama dari aplikasi, yaitu pemindaian keamanan dan pembuatan laporan, dilakukan pada state “Proses Scan dan Membuat Report”, yang menjadi pusat dari fitur *vulnerability assessment*. Seluruh proses ini dapat dijalankan secara iteratif sesuai kebutuhan pengguna. Jika pengguna ingin mengakhiri sesi penggunaan aplikasi, maka mereka dapat memilih opsi logout, yang membawa sistem menuju Final State, yang ditandai dengan lingkaran hitam dengan garis lingkar luar putih. Final State ini menunjukkan bahwa siklus interaksi pengguna telah selesai dan aplikasi tidak lagi berada dalam kondisi aktif.

Secara keseluruhan, *state diagram* ini memberikan gambaran menyeluruh mengenai alur kerja pengguna dalam aplikasi, mulai dari login hingga logout, termasuk berbagai fitur yang tersedia. Diagram ini juga menunjukkan bagaimana *event*atau aksi pengguna menentukan arah navigasi dalam sistem. Pemodelan seperti ini sangat penting dalam pengembangan aplikasi berbasis Agile karena membantu peneliti memahami skenario penggunaan secara menyeluruh dan mendukung perencanaan iterasi pengembangan yang lebih terstruktur dan responsif terhadap perubahan kebutuhan pengguna.

4. Desain Database

Setelah proses di atas selanjutnya adalah perancangan database yang terdiri dari beberapa tabel yaitu :

Tabel 3. 3 Tabel User

Nama	Tipe Data
id	uuid
email	text
name	text
avatar_url	text

Nama	Tipe Data
created_at	timestamp

Tabel 3. 4 Tabel Reports

Nama	Tipe Data
id	uuid
file_url	text
scanned_url	text
email	text
created_at	timestamp

Tabel 3. 5 Tabel Rekomendasi

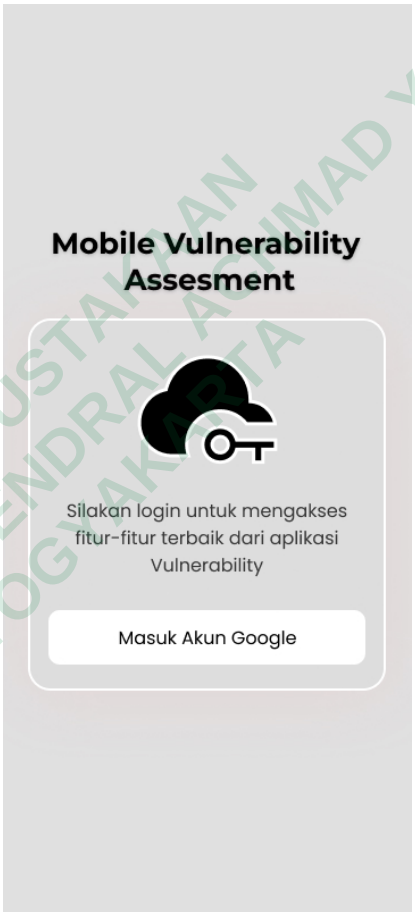
Nama	Tipe Data
id	uuid
gambar	text
nama_perusahaan	text
deskripsi_perusahaan	text
link_perusahaan	text
created_at	timestamp

5. Desain Antar Muka

   Desain Antar Muka adalah rancangan tampilan yang akan dibuat agar mempermudah dalam melakukan desain sistem, berikut beberapa antar muka yang telah dibuat:

a. Halaman Login

Gambar 3. 6 merupakan rancangan desain antar muka untuk halaman login,yaitu sebagai berikut:



Gambar 3. 6 Login Page

Tampilan halaman awal dari aplikasi *Mobile Vulnerability Assessment*, yang didesain dengan antarmuka sederhana dan estetis menggunakan gradasi warna lembut. Komponen utama yang ditampilkan terdiri dari teks judul yang menyatakan nama aplikasi di bagian atas, ikon awan dengan kunci yang merepresentasikan keamanan dan privasi sebagai tema utama aplikasi, serta teks

deskriptif yang mengajak pengguna untuk login agar dapat mengakses fitur-fitur terbaik aplikasi. Di bagian bawah, terdapat tombol “Masuk Akun Google” yang dilengkapi dengan logo Google, berfungsi sebagai pintu masuk autentikasi menggunakan akun Google pengguna. Tombol ini memanfaatkan integrasi *OAuth2* yang umum digunakan pada aplikasi Flutter modern untuk memudahkan proses login secara aman dan cepat. Keseluruhan elemen didesain agar mudah dipahami oleh pengguna dan mendukung pengalaman pengguna (UX) yang intuitif.

b. Halaman Home

Gambar 3. 7 merupakan rancangan desain antar muka untuk halaman home,yaitu sebagai berikut:

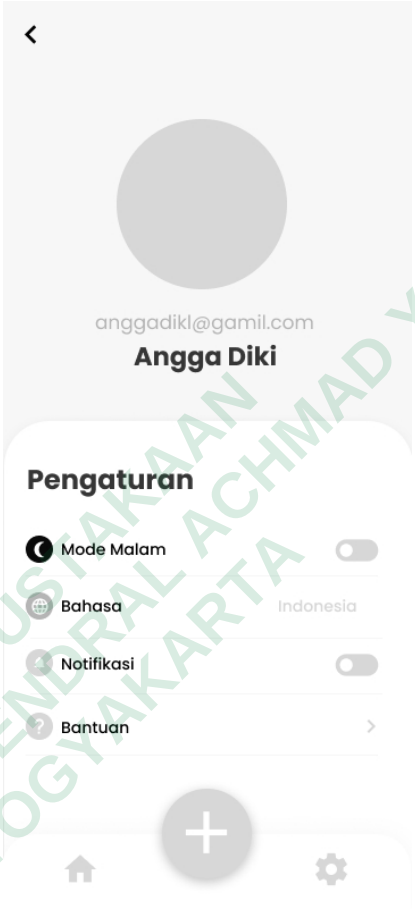
Gambar 3. 7 Home Page

Gambar 3. 7 yang ditampilkan merupakan tampilan utama setelah pengguna berhasil login ke dalam aplikasi *Mobile Vulnerability Assessment*. Di bagian atas, terdapat ucapan selamat datang yang dipersonalisasi dengan nama pengguna, dalam hal ini "Angga Diki", menandakan bahwa aplikasi telah berhasil mengautentikasi akun. Bagian selanjutnya menampilkan hasil pemindaian website dalam bentuk daftar yang menunjukkan identitas atau status, diikuti oleh URL target, serta ikon unduh di sisi kanan yang memungkinkan pengguna untuk mengunduh laporan hasil pemindaian.

Selanjutnya, terdapat bagian "Rekomendasi" yang menampilkan konten dalam bentuk kartu, masing-masing berisi gambar, judul, dan deskripsi singkat, berfungsi memberikan saran atau informasi tambahan dari pihak ketiga seperti "Sentra Data" dan "Widya Security".

c. Halaman Settings

Gambar 3. 8 merupakan rancangan desain ui untuk halaman settings,yaitu sebagai berikut:



Gambar 3. 8 Settings Page

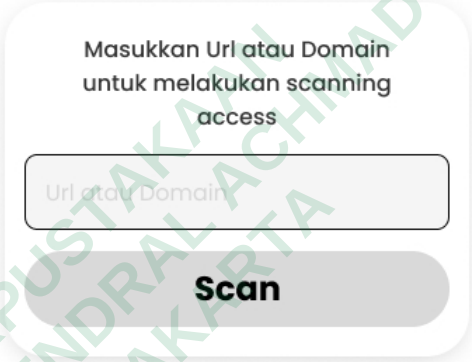
Gambar di atas merupakan tampilan halaman Pengaturan (Settings) dalam aplikasi *Mobile Vulnerability Assessment*. Di bagian atas, terdapat informasi akun pengguna berupa avatar ilustratif, alamat email dan nama pengguna "Angga Diki". Tampilan ini memudahkan pengguna untuk mengenali identitas yang sedang aktif.

Di bawah bagian identitas pengguna, terdapat beberapa pengaturan utama aplikasi, yaitu Mode Malam (dengan switch toggle untuk mengaktifkan tema gelap), Bahasa yang saat ini disetel ke "Indonesia", serta Notifikasi yang juga dapat diaktifkan atau

dinonaktifkan dengan switch toggle. Terdapat juga menu Bantuan yang kemungkinan membawa pengguna kepanduan atau pusat bantuan aplikasi. Seluruh menu disusun secara rapi dan mudah diakses, dilengkapi ikon yang konsisten. Tampilan ini menunjukkan perhatian terhadap pengalaman pengguna (UX) dengan pendekatan yang intuitif dan minimalis.

d. Card Tambah Data

Gambar 3. 9 merupakan rancangan desain card untuk menambahkan nama domain, yaitu sebagai berikut:



Gambar 3. 9 Card Tambah Data

Gambar diatas menunjukkan komponen input sederhana yang digunakan untuk memasukkan *URL* atau domain yang akan di scan dalam aplikasi *Mobile Vulnerability Assessment*. Pengguna diminta untuk mengetik alamat domain atau *URL* ke dalam *field input* berlabel "*Url* atau Domain", lalu menekan tombol "*Scan*" berwarna merah muda dengan teks tebal berwarna putih. Komponen ini dirancang untuk langsung memfasilitasi proses scanning akses dengan tampilan minimalis namun jelas, sehingga pengguna dapat dengan cepat memahami fungsinya dan melakukan tindakan tanpa kebingungan.

e. Card Hapus Data

Gambar 3. 10 merupakan rancangan desain card untuk menambahkan nama domain, yaitu sebagai berikut:

Gambar 3. 10 Card Hapus Data

Gambar diatas menampilkan komponen dialog konfirmasi untuk menghapus *URL* atau domain dalam aplikasi. Di dalamnya terdapat pertanyaan singkat "Hapus *URL* atau domain" yang disertai dua tombol aksi yaitu "Batal" berwarna abu-abu untuk membatalkan tindakan, dan "Iya" untuk menyetujui penghapusan. Desain ini memberikan kejelasan dan kemudahan bagi pengguna dalam mengambil keputusan, sekaligus mencegah penghapusan data secara tidak sengaja.