Journal of Applied Informatics and Computing (JAIC)
Vol.x, No.x, Juli 201x, pp. xx~xx
e-ISSN: 2548-6861

PEMBUATAN SISTEM DETEKSI HARDCODE KREDENSIAL PADA REPOSITORY

Bill Jeferson Nababan[1], Antoni Haikal,S.ST.,M.T[2]

[1]Program Studi Teknik Informatika, [2]Politeknik Negeri Batam E-mail:
billnbbn@gmail.com [1], antoni@polibatam.ac.id [2]

Article Info

Article history:

Received …
Revised …
Accepted …

Keyword:

Hardcoded Credential, Truffleho JavaScript; Prototyping

ABSTRACT

Hardcoded Credential is the practice of embedding authentication information, such as usernames and passwords, directly into the source code of software or applications. This means that the credential information is not stored separately or managed securely, but rather integrated into the program code. This practice poses significant security risks, one of which is the difficulty of changing credentials, making modifications to the source code impractical and increasing security risks. This research proposes a web-based Hardcoded Credential Detection System that can detect Hardcoded Credentials in the Repository on Github, by implementing security tools in the form of Trufflehog to the website, the system can see the results of Hardcoded Credential detection after the detection process is complete. By using the Prototyping method which is one approach in software development by following a series of stages that are carried out sequentially and completed one by one before entering the next stage. The technologies used include ReactJs as a library for making Front-end, ExpressJs as a Framework for making Back-end with Javascript as a Programming Language, and MYSQL as a database. The results of this system can help in maintaining the security of Github repositories by providing the use of tools that can identify potential leaks of sensitive credentials. Thus, developers and security teams can take action to remove or secure those accidental credentials

I. PENDAHULUAN

Keamanan perangkat lunak telah menjadi fokus terpenting dalam dunia teknologi informasi. Dengan meningkatnya kompleksitas dan ukuran perangkat lunak, risiko dan kemungkinan kebocoran kredensial semakin menjadi ancaman yang serius. Salah satu bentuk risiko ini adalah penggunaan *hardcoded credentials*, di mana informasi seperti login, *username*, *password*, *aws-key*, *API*, atau kredensial lainnya secara langsung ada di dalam kode sumber. [1]

Kredensial yang tertanam, juga bisa disebut dengan *hardcoded credentials*, yang merupakan teks biasa yang bersifat rahasia tertanam didalam kode sumber. Kredensial mengacu pada praktik menyematkan kredensial teks biasa(tidak terenkripsi), kunci SSH, nama pengguna, E-mail, kata sandi, *API*, *key*, *aws-key* dan lainnya. Dalam beberapa kasus, pelaku ancaman dapat memasukkan kredensial yang dikodekan untuk membuat backdoor yang memungkinkan pelaku mengakses perangkat, aplikasi, atau sistem terus-menerus [2]. Pentingnya deteksi dini *hardcoded credentials* sangat krusial, terutama dalam konteks pengembangan perangkat lunak kolaboratif menggunakan sistem kontrol

e-ISSN: 2548-6861

versi seperti Git. Repositori perangkat lunak khususnya github yang mengandung hardcoded credentials dapat menjadi sumber potensi kerentanan keamanan yang serius, menyediakan akses langsung ke sumber daya yang dapat dimanfaatkan oleh pihak yang tidak sah [3]. Namun, mengidentifikasi *hardcoded credentials* secara manual dalam repositori yang besar atau kompleks dapat menjadi tugas yang sangat sulit dan rentan terhadap kesalahan manusia . *Hardcoded credentials* dapat dijadikan solusi sementara, walaupun terkadang pengembang tidak menyadari bahwa git sebenarnya melacak *secret* dalam kredensial. [4] Oleh karena itu, perlu adanya sistem otomatis untuk mendeteksi *hardcoded credentials* dalam repositori pada github. Sistem ini tidak hanya akan membantu pengembang dalam menerapkan praktik keamanan terbaik tetapi juga dapat mengurangi potensi risiko keamanan yang dapat timbul dari kelalaian dan penyebab lainnya.

Mengangkat masalah kebocoran data dari *Cyber Blitz* Direktorat Keamanan Siber, terdapat kebocoran data berupa *secret* kredensial di *source code* pada Neopets sebesar 69 juta data pribadi, dari 69 juta tersebut didalam source code terdapat data pribadi berupa nama pengguna, alamat E-mail, jenis-kelamin, kata sandi, dan hal pribadi lainnya. Hal serupa juga terjadi pada aplikasi Kesehatan Mental Fellyou yang mengalami kebocoran data 76.000 email pengguna [5].

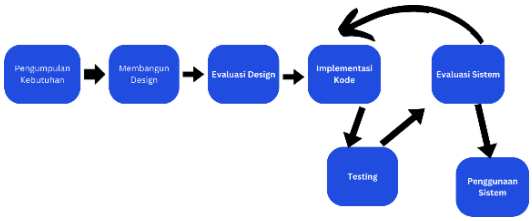
Maka dari uraian masalah diatas, Pembuatan sistem deteksi *hardcoded kredensial* pada *repository* akan menjadi solusi positif dan bisa direalisasikan untuk menghadapi tantangan keamanan ini. Sistem ini diharapkan dapat mengintegrasikan teknik-teknik deteksi kredensial pada *repository* github dengan *trufflehog* dan analisis sintaksis, untuk secara efisien dan akurat mengidentifikasi *hardcoded* kredensial. *TruffleHog* nantinya akan digunakan untuk mendeteksi *API*, nama pengguna, kata sandi, *secret* yang dikommit kedalam git. [6] Setelah proses deteksi hardcoded kredensial selesai, akan dilakukan validasi valid atau tidak valid nya kredensial dari hasil validasi sebelumnya, yang Dimana jika valid user dapat meng-*Export* hasil validasi ke format Excel.

Dengan implementasi sistem ini, pengembang akan mendapatkan keuntungan dari lingkungan pengembangan yang lebih aman, sementara organisasi akan dapat menjaga reputasi dan kepercayaan pengguna mereka. Oleh karena itu, tugas akhir ini bertujuan untuk mengembangkan sistem deteksi *hardcoded* kredensial dengan bantuan *tools Trufflehog* yang mudah di integrasikan, dan efisien, yang dapat diterapkan dalam berbagai konteks pengembangan perangkat lunak.

Implementasi sistem ini akan memberikan kontribusi signifikan terhadap praktik keamanan perangkat lunak dan membantu melindungi informasi kredensial seperti kata sandi, nama pengguna dari risiko kebocoran dan penyalahgunaan. [7]

II. METODE

Penelitian ini dibuat dengan tujuan untuk mendeteksi kebocoran hardcoded kredensial pada repository di github. Oleh karena itu beberapa tahapan yang digunakan untuk mengembangkan sistem deteksi ini adalah sebagai berikut:



Gambar 3. 1 Ilustrasi Protyping Pada system

Metode *Prototyping* ini bertujuan mengumpulkan informasi dari user atau client sehingga client dapat berinteraksi dan memakai website dengan model prototype yang akan dibangun secara keseluruhan [8]

1. Tahapan Pengumpulan Kebutuhan, teknik pengumpulan data yang akan digunakan untuk kebutuhan sistem adalah mengidentifikasikan semua kebutuhan dan garis besar sistem yang akan dibuat. Salah satunya adalah pengumpulan repositori pada github, yang nantinya penulis akan mengumpulkan banyak link *repository* dari github yang akan dideteksi kredensialnya melalui *trufflehog* pada website.
2. Membangun Desain, membangun Desain yang bertujuan memberikan gambaran lengkap tentang bagian apa saja yang harus dikerjakan dan bagaimana tampilan dari sebuah sistem, sehingga membantu peneliti dalam mendefinisikan arsitektur sistem yang akan dibuat secara keseluruhan. Tahapan Desain meliputi desain UI (*User Interface*) menggunakan aplikasi Figma, dan perancangan database menggunakan aplikasi Mysql Workbench.

3. Evaluasi Desain, Evaluasi desain dilakukan oleh client apakah desian atau *prototyping* yang sudah dibuat sesuai dengan keinginan client
4. Implementasikan Kode, rancangan desain yang sudah dirancang tadi akan diimplementasikan kedalam sebuah Bahasa pemrograman. Bahasa pemrograman yang akan penulis gunakan nantinya adalah untuk bagian *Front-End* nanti menggunakan Bahasa pemrograman javascript dan menggunakan Framework Vite dan untuk di *Back- End* sendiri akan menggunakan Framework *Express-js*, Node JS dan database berupa Mysql.
5. Testing, setelah pengembang selesai membangun website sampai ke tahap dimana *truffleHog* juga sudah bisa digunakan di dalam website, akan dilakukan pengujian pada sistem bersama dengan client untuk mencari kesalahan atau kekurangan yang ada dalam website yang akan diperbaiki untuk hasil yang lebih baik.
6. Evaluasi Sistem, di tahapan ini akan membutuhkan bantuan dari client untuk membantu dalam mengevaluasi sistem. Apakah sistem atau website ini sudah sesuai kebutuhan client dan dapat digunakan dengan baik. Dan jika tidak terpenuhi akandilakukan perbaikan oleh pengembang.
7. Penggunaan Sistem, Sistem yang sudah di testing dan disetujui oleh client sudah bisa digunakan.

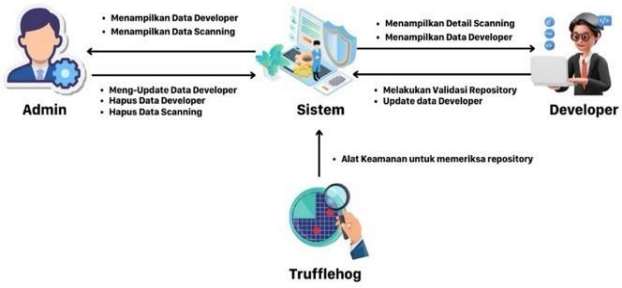
III. HASIL DAN PEMBAHASAN

A. Tahapan Requirement

Sistem Deteksi *Hardcoded Credentials* pada repository ini dibuat dengan menggunakan arsitektur web yang dapat diakses melalui web browser. Analisis dilakukan berdasarkan dokumentasi yang ada serta pemahaman mendalam terhadap kebutuhan sistem berupa:

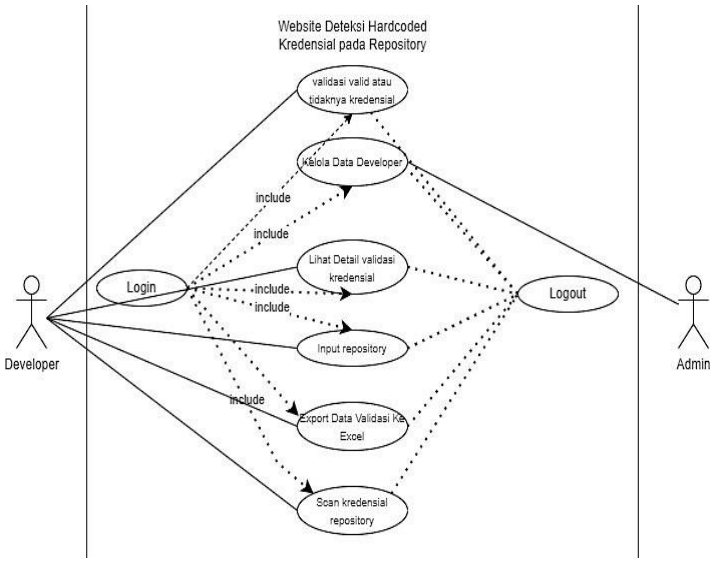
a. Developer dapat melakukan login ke dalam website.
b. Developer melakukan pemeriksaan *repository* melalui web dengan bantuan *Truffalehog*.
c. Hasil *scanning* akan ditampilkan dan diperiksa secara manual oleh Developer.

Pada sistem ini memiliki 2 roles yaitu ada Admin dan Developer dan tools yang kita sebut sebagai *trufflehog*. Untuk roles Admin nantinya akan digunakan oleh tim keamanan perusahaan, dan Developer akan digunakan oleh pengguna platform github secara umum



Gambar 3. 2 Gambaran Umum Sistem

2. Use Case Diagram



Gambar 3. 3 *Use Case Diagram*

Diagram use case di atas menunjukkan interaksi antara dua aktor utama, yaitu Admin dan Developer, dengan sistem deteksi kredensial hardcoded pada repository. Berikut adalah penjelasan mengenai alur dalam diagram tersebut:

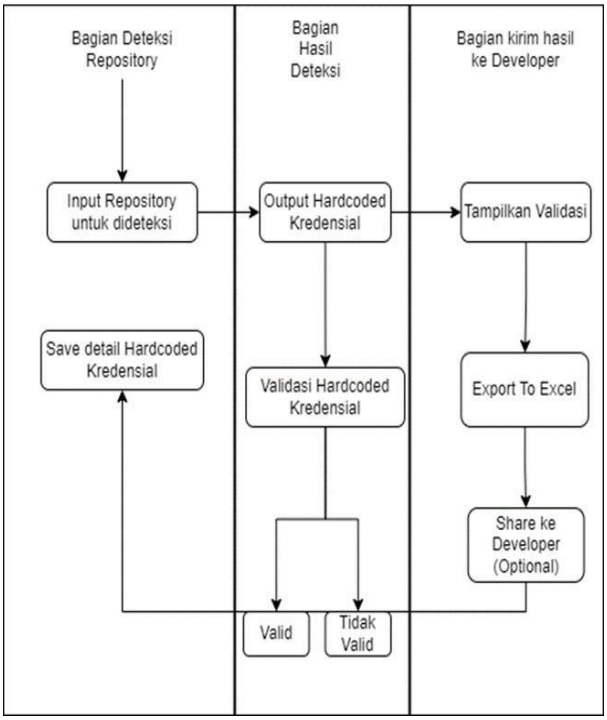
a. Admin dan Developer harus melakukan login terlebih dahulu untuk mengakses fitur-fitur yang tersedia dalam sistem. Setelah login, Developer dapat melakukan scanning terhadap repository yang ada di GitHub menggunakan alat yang disebut *Trufflehog*. *Trufflehog* adalah alat keamanan yang digunakan untuk memeriksa repository dan mencari kredensial seperti email, password, username, dan informasi rahasia lainnya yang mungkin terdapat dalam kode sumber.
b. Setelah proses scanning dilakukan oleh *Trufflehog*, Developer memeriksa hasil scanning tersebut secara manual untuk menentukan apakah kredensial yang ditemukan valid atau tidak valid. Jika valid, Jika valid,

e-ISSN: 2548-6861

maka kredensial tersebut masih aktif dan perlu ditangani dengan hati-hati. Jika tidak valid, berarti kredensial tersebut sudah tidak digunakan atau tidak aktif lagi.

c. Admin memiliki tugas utama untuk mengelola data pengguna dan data repository. Tugas ini mencakup melihat detail validasi kredensial, meng-update, dan menghapus data yang diperlukan. Admin tidak terlibat langsung dalam proses scanning, tetapi memastikan bahwa data yang dikelola tetap akurat dan up-to-date.
d. Setelah semua proses selesai, baik Admin maupun Developer dapat logout dari sistem

3. Activity Diagram



Gambar 3. 4 Alur *Activity Diagram*

Activity diagram merupakan rancangan aliran aktivitas yang digunakan pada sebuah sistem yang dijalankan.

Di dalamnya terdapat komponen dengan bentuk tertentu yang dihubungkan melalui tanda panah. Kemudian panah itu mengarah ke urutan aktivitas yang akan dilakukan dari awal sampai akhir. Berikut gambaran activity diagram dan penjelasannya.

Activity diagram di atas menggambarkan proses deteksi dan validasi kredensial hardcoded dalam sebuah repository, serta penyampaian hasil deteksi tersebut ke developer. Proses ini terdiri dari tiga bagian utama.

A. Deteksi Repository:

1. Input Repository untuk Dideteksi: Langkah pertama adalah memasukkan repository yang ingin diperiksa untuk kredensial hardcoded.
2. Save Detail Hardcoded Kredensial: Setelah deteksi dilakukan, detail dari kredensial yang ditemukan disimpan untuk langkah-langkah selanjutnya.

B. Hasil Deteksi:

1. Output Hardcoded Kredensial: Kredensial yang ditemukan kemudian dikeluarkan sebagai output.
2. Validasi Hardcoded Kredensial: Kredensial tersebut kemudian divalidasi untuk memastikan apakah masih berlaku atau sudah tidak diragukan lagi.
3. Valid: Jika kredensial valid, berarti kredensial tersebut masih aktif dan perlu ditangani dengan hati-hati.
4. Tidak Valid: Jika tidak valid, berarti kredensial tersebut sudah tidak aktif atau tidak lagi digunakan.

C. Kirim Hasil ke Developer:

1. Tampilkan Validasi: Hasil dari proses validasi ditampilkan agar bisa dilihat.
2. Export to Excel: Hasil ini kemudian diekspor ke dalam file Excel untuk memudahkan analisis dan dokumentasi.
3. Share ke Developer (Optional): Akhirnya, hasil ini bisa dibagikan kepada developer yang bertanggung jawab untuk memperbaiki atau menghapus kredensial hardcoded dari kode.

4. Kebutuhan Fungsional

a. Functional Requirement (FR)

Functional Requirement adalah kebutuhan yang berisi proses atau layanan yang nantinya akan disediakan oleh sistem. Berikut adalah hasil analisis mengenai Functional Requirement pada Sistem Informasi Dana Amal Polibatam:

Table 3. 2

NO	KEBUTUHAN FUNGSIONAL
F001	Pengguna (Developer dan Admin) dapat melakukan login
F002	Developer dapat melakukan Register jika belum terdaftar
F003	Admin dapat melakukan delete/update pada data akun Developer
F004	Developer dapat melakukan Update User
F005	Developer dapat melakukan input link repository github ke dalam form trufflehog
F006	Developer dapat melihat detail dari validasi yang berupa valid atau tidak saat di deteksi
F006	Developer dapat melihat detail dari validasi yang berupa valid atau tidak saat di deteksi
F007	Developer hanya dapat melihat hasil validasi dari link yang Developer input, developer lain tidak dapat melihat hasil validasi developer yang lainnya
F008	Developer dapat meng-*Export* hasil validasi dalam format Excel
F009	Admin dapat menghapus data hasil validasi *Repository*
F010	Admin dapat menyimpan data akhir dari validasi kredensial kedalam database.

b. Non Functional Requirement (NFR)

Non Functional Requirement adalah kebutuhan yang menitik beratkan pada *property* pendukung yang dimiliki oleh sistem.

Table 3. 2

KRITERIA	PARAMETER
Availability	Sistem dapat di akses apabila terhubung ke jaringan internet Sistem mampu berjalan selama 24 jam non-stop, kecuali apabila ada perawatan sistem atau pembaharuan sistem Sistem dapat dijalankan dimana saja dan kapan saja
Ergonomy	Sistem harus dapat digunakan dengan mudah atau *user friendly*
Bahasa	Menggunakan Bahasa Inggris
Safety	Sistem dapat memastikan bahwa data yang digunakan harus terlindungi dari akses yang tidak berwenang

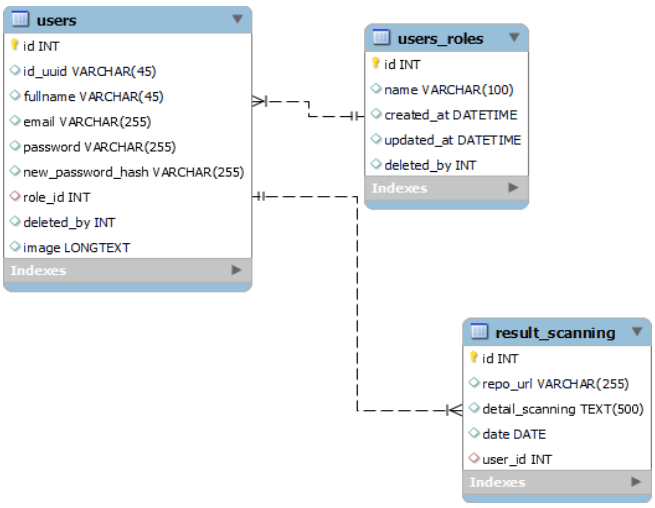
.

B. Membangun Desain

Desain User *Interface* (UI) adalah proses merancang elemen-elemen visual dan interaktif dari suatu sistem atau produk, dengan tujuan utama untuk meningkatkan pengalaman pengguna*[9]*.

1. Entity Relationship Diagram

Entity Relationship Diagram (ERD) adalah alat visual yang digunakan dalam rekayasa perangkat lunak dan desain sistem database. ERD digunakan untuk menggambarkan dan memodelkan hubungan antara table pada database. Untuk Erd dibawah ini saya buat menggunakan Model di Aplikasi Mysql Workbench.



Gambar 3. 5 ERD

e-ISSN: 2548-6861

Diagram di atas menggambarkan struktur basis data untuk sistem deteksi kredensial hardcoded. Terdapat tiga tabel utama: *users*, *users_roles*, dan *result_scanning*.

a. Tabel *users*: Tabel ini menyimpan informasi tentang pengguna sistem. Kolom-kolom di dalamnya mencakup:
   1. id: Nomor identifikasi unik untuk setiap pengguna.
   2. id_uuid: ID unik berbentuk string.
   3. fullname: Nama lengkap pengguna.
   4. email: Alamat email pengguna.
   5. password: Kata sandi pengguna yang dienkripsi.
   6. new_password_hash: Hash dari kata sandi baru jika pengguna mengganti kata sandi.
   7. role_id: Nomor identifikasi peran pengguna yang menghubungkan ke tabel *users_roles*

b. Tabel *users_roles*: Tabel ini menyimpan informasi tentang peran atau posisi pengguna dalam sistem. Kolom-kolom di dalamnya mencakup:
   1. id: Nomor identifikasi unik untuk setiap peran.
   2. name: Nama peran, seperti Admin atau Developer.
   3. created_at: Tanggal dan waktu ketika peran dibuat.
   4. updated_at: Tanggal dan waktu ketika peran terakhir diperbarui.
   5. deleted_by: ID pengguna yang menghapus data peran (jika ada).

c. Tabel *result_scanning*: Tabel ini menyimpan hasil dari proses scanning repository untuk kredensial hardcoded. Kolom-kolom di dalamnya mencakup:
   1. id: Nomor identifikasi unik untuk setiap hasil scanning.
   2. repo_url: URL dari repository yang di scan.
   3. detail_scanning: Detail dari hasil scanning, seperti kredensial yang ditemukan.
   4. date: Tanggal scanning dilakukan.
   5. user_id: Nomor identifikasi pengguna yang melakukan scanning, menghubungkan ke tabel users.

C. Implementasi Kode

1. Implementasi AntarMuka

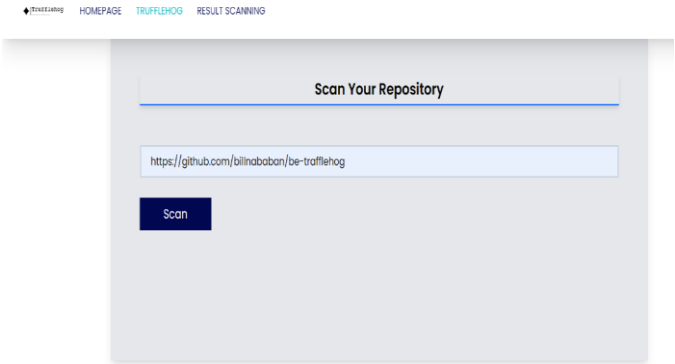
Implementasi halaman landing page halaman landing page adalah halaman yang dapat diakses secara bebas oleh semua orang tanpa harus login. Pada halaman ini menampilkan seputar informasi umum tentang *Hardcoded Credential*, Pengenalan *Trufflehog* dan Keamanan lainnya.Implementasi halaman landing page ini dapat dilihat pada gambar dibawah ini.

Gambar 3. 6 *Landing Page*

Gambar 3. 6 *Landing Page*

2. Implementasi Halaman Validasi Repositori

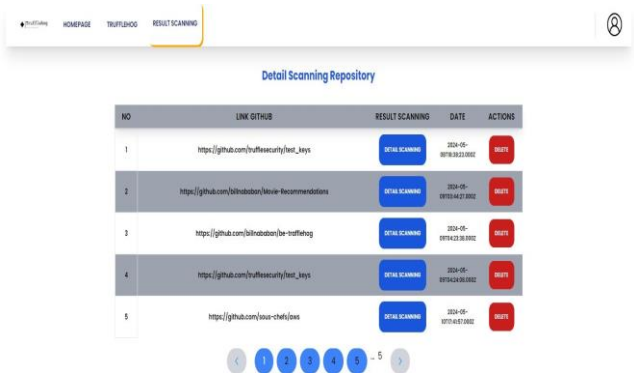
Halaman ini hanya dapat diakses jika developer sudah melakukan login terlebih dahulu. Pada halaman ini, Developer akan menginput link repository github mereka untuk divalidasi oleh Trufflehog untuk validasi keamanan repositori.



Gambar 3. 7 Form Input Link *repository*

3. Implementasi Halaman *History* Hasil validasi

Di halaman *History* ini, user dapat melihat hasil dari validasi *repository* yang sudah selesai di deteksi oleh *Trufflehog*, dan untuk *history* ini juga sudah dipisah untuk data tiap developer, para developer hanya bisa melihat data *history* dari Link yang mereka input dan Developer tidak bisa melihat *history* Developer yang lainnya.



Gambar 3. 8 Halaman *Detail Scanning*

D. Testing

Testing pada sistem ini bertujuan untuk memverifikasi bahwa sistem yang telah dikembangkan dapat berfungsi secara optimal sesuai dengan kebutuhan fungsionalnya. Proses pengujian sistem ini dilakukan menggunakan pendekatan black box, yang menitik beratkan pada penilaian kualitas perangkat lunak dengan cara mengevaluasi fungsionalitasnya untuk mengidentifikasi ketidakcocokan fungsi, kesalahan antarmuka, dan potensi kesalahan lainnya. [10]

Berikut hasil dari testing pada sistem ini :

Tabel 3. 3 Testing

No	Fungsional	Skenario Pengujian	Role	Indikator Keberhasilan	Hasil Pengujian
1	Login	Pengguna memasukan usernamebenar dan password benar	Developer dan Admin	Sistem memberikan pesan Selamat dating dan Pengguna berhasil login	Sesuai
		Pengguna memasukan email benar dan password salah		Sistem memberikan pesan Password salah	Sesuai
		Pengguna memasukan email salah dan password benar		Sistem memberikan pesan Email yang anda masukkan salah	Sesuai
		Pengguna memasukan email salah dan password salah		Sistem memberikan pesan Akun anda tidak terdaftar silahkan registrasi	Sesuai
2	Register	Pengguna memasukkan username, email, password, dan konfirmasi password	Developer	Sistem akan memberikan pesan register berhasil dan langsung diarahkan ke halaman login	Sesuai
		Pengguna memasukkan password tidak berisi simbol, angka dan huruf kapital		Sistem akan memberikan pesan, silahkan tambahkan simbol, angka dan huruf kapital untuk keamanan data anda	Sesuai
		Pengguna memasukkan password yang berisi simbol, angka dan huruf kapital		Sistem akan memberikan pesan, Password benar silahkan lakukan register	Sesuai
		Admin Memasukkan Email dan Full Name data user yang ingin dirubah		Sistem akan menampilkanpesan Ubah data user berhasil	Sesuai
3	Mengubah data user	Admin memasukan data user dengan Username yang sudah digunakan	Admin	Sistem akan menampilkanpesan Username sudah ada sebelumnya	Sesuai

e-ISSN: 2548-6861

		Admin memasukkan data user dengan Email yang digunakan		Sistem akan menampilkan pesan Email sudah digunakan	Sesuai
		Admin memasukkan data user dengan Username dan Email yang sudah digunakan		Sistem akan menampilkan pesan Username dan email sudah digunakan	Sesuai
4	Menghapus user	Menekan tombol hapus pada user yang ingin dihapus	Admin	Sistem akan menampilkan pesan Akun berhasil dihapus	Sesuai
5	melakukan update profile	Pengguna memasukkan data baru	Developer	Sistem akan menampilkan pesan Akun berhasil diupdate	Sesuai
		Pengguna memasukan username yang sudah digunakan oleh user lain		Sistem akan menampilkan pesan Username sudah digunakan	Sesuai
		Pengguna memasukan email yang sudah digunakan oleh user lain		Sistem akan menampilkan pesan Email sudah digunakan	Sesuai

6	dapat melihat hasil validasi dari trufflehog	melihat hasil validasi	Developer	Sistem menampilkan Modal pop up untuk melihat hasil	Sesuai

7	Data tiap user sudah dipisah	Di halaman History developer, hanya berisi data dari hasil validasi yang dilakukan oleh developer tersebut	Developer	Sistem menampilkan form history yang terpisah sesuai user yang login	sesuai
8	Menghapus Hasil validasi	Mengklik button untuk menghapus validasi	Developer and Admin	Sistem menampilkan pesan Data sudah dihapus	sesuai
9	Trufflehog validasi	Link github yang dikirim dari Front end akan di proses ke backend dan dieksekusi oleh trufflehog	Trufflehog	Trufflehog akan melempar kembali ke server dan hasil validasi akan dikembalikan dan disimpan ke database dalam format JSON	sesuai
10	Navigasi Halaman Data	Pengguna mengklik Button next dan previous	Developer dan Admin	Sistem berpindah halaman	sesuai

11	Export Excel	untuk berpindah atau kembali ke halaman Pengguna melihat detail hasil validasi yang masih berantakan di modal Detail	Developer	Sistem akan mendownload otomatis hasil validasi tersebut dan disimpan dalam format Excel tentunya dengan format penulisan yang lebih rapi	sesuai

E. Evaluasi Sistem

Pada tahapan ini akan dilakukan evaluasi sistem secara berkala, khususnya untuk melakukan update-update tertentu, jika sudah selesai nantinya website akan dapat digunakan oleh para developer.

IV. KESIMPULAN

Berdasarkan hasil dari perancangan dan implementasi telah dibangun Sistem deteksi *Hardcoded Credential* pada *repository* Public di Github Berbasis Web dengan metode *Prototyping*, dapat disimpulkan bahwa sistem ini memberikan kemudahan bagi developer dalam memeriksa keamanan repositori Github, melihat validasi dari *trufflehog*. Berdasarkan hasil testing dengan metode Black Box sistem ini sudah sesuai dan dapat digunakan dengan baik. Namun sistem ini masih belum sepenuhnya sempurna, dikarenakan proses validasi dari *Trufflehog* sedikit lama dikarenakan *trufflehog* memeriksa lewat commit secara keseluruhan. Oleh karena itu diperlukan pengembangan selanjutnya untuk menambahkan fitur-fitur lain yang masih belum tersedia.

DAFTAR PUSTAKA

[1] M. Ashari, "Kebocoran Data Kredensial," 22 Maret 2022. [Online].Available: https://penerbitdeepublish.com/menulis-buku-membuat- sitasi-dengan-mudah/. [Accessed 20 Februari 2024].

[2] M. Miller, "Hardcoded and Embedded Credentials are an IT Security Hazard," BeyondTrust, 26 Februari 2019.[Online].Available: https://www.beyondtrust.com/blog/entry/hardcoded - and- embedded-credentials-are-an-it-security-ha zard- heres-what-you-need-to-know. [Accessed 23 Februari 2024].

[3] C. Company, "HARD-CODED CREDENTIALS," HSSEDI, 10 February 2023. [Online]. Available: https://cwe.mitre.org/data/definitions/798.html. [Accessed 12 Maret 2024].

[4] T. Segura, "Detect hardcoded secrets with GitGuardian," GitGuardian, 26 Agustus 2022. [Online]. Available: https://circleci.com/blog/detect-hardcoded-secrets-with-gitguardian/. [Accessed 15 Maret 2024].

[5] C. Blitz, "Data Kebocoran-kebocoran data kredensial," Kementerian Keuangan Republik Indonesia, 2022. [Online]. Available:https://www.djkn.kemenkeu.go.id/artikel/bac a/14838/Belajar-Dari-Kebocoran-Data-Kredensial- Data-Yang Paling-Berharga-adalah-Data-Pribadi.html. [Accessed 16 Maret 2024].

[6] Ayrey, "TruffleHog v3," Truffle Security Co, 04 April 2022. [Online]. Available: https://trufflesecurity.com/blog/introducing-trufflehog- v3. [Accessed 4 November 2023].

[7] A. Developers, "Secret Kriptografis yang Di-hardcode," 07 11 2023. [Online].CISCO, "What Is Cybersecurity?," 2023. https://developer.android.com/privacy-and security/risks/hardcoded-cryptographic-secrets?hl=id. [Accessed 22 Maret 2024].

[8] Herlawati, "Prototype Mesin Presensi Berbasis E-Mail Pada SMP Assyairiyah Attahiriyah Jakarta," *Jurnal JSRCS 1 (2): 179-190 (November 2020)* .

[9] D. Ridzky, "User Interface Modelling for SIBI (Sistem Isyarat Bahasa Indonesia/Indonesian Sign Language System) learning applications using the User-Centered Desain Method," *Journal of Physics: Conference Series*, 2018.

[10] I.Permatasari, "Pengujian Black Box Menggunakan Metode Analisis Nilai Batas pada Aplikasi DANA," vol. Vol. 3 No. 2, p. 15, Desember 2023.

[11] B. Unnes, "TruffleHog v3 Dirilis, Kini Tambah 600 Deteksi Celah Keamanan," 13 April 2022.

[12] CISCO, "What Is Cybersecurity?," 2023.