

Perancangan *Backend* dengan *Framework* Laravel Pada *Website* Pengelolaan Kunjungan Di PuTI Universitas Telkom

1st Wenner Frederikus Silalahi
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
wenner@student.telkomuniversity.ac.id

2nd Uke Kurniawan Usman
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
ukeusman@telkomuniversity.ac.id

3rd Widi Tri Yuwono
Direktorat Pusat Teknologi Informasi
Universitas Telkom
Bandung, Indonesia
widi@telkomuniversity.ac.id

Abstrak — Peningkatan pengelolaan kunjungan menjadi salah satu keberhasilan sebuah institusi dalam melakukan pengelolaan kunjungan. Perkembangan era teknologi yang pesat mampu mengubah pengelolaan kunjungan secara manual menjadi pengelolaan kunjungan berbasis *website*. Peningkatan pengelolaan kunjungan ini mampu diterapkan pada Direktorat Pusat Teknologi Informasi (PuTI) Universitas Telkom sehingga memudahkan Direktorat PuTI dalam melakukan pengelolaan data kunjungan. Pengelolaan kunjungan berbasis *website* ini mampu diciptakan menggunakan *framework* yang tepat. Laravel merupakan salah satu *framework* terkemuka yang menawarkan berbagai fitur yang mendukung pengembangan pengelolaan kunjungan berbasis *website* yang handal dan aman. Laravel mampu menangani berbagai aspek pengembangan backend seperti routing, autentikasi, dan manajemen database. Perancangan backend menggunakan *framework* Laravel mampu menciptakan pengelolaan kunjungan yang lebih efisien dan terintegrasi. Penerapan sistem manajemen pengunjung berbasis *website* menggunakan *framework* Laravel akan menciptakan sistem yang memberikan kemudahan dalam pengelolaan kunjungan. Berdasarkan hasil perancangan backend yang telah diselesaikan maka dilakukan pengujian terhadap sistem keamanan backend menggunakan Nessus Essential dan Pentest tools. Hasil pengujian diperoleh bahwa sistem keamanan *website* yang dirancang menggunakan *framework* Laravel berada pada level medium

Kata kunci— PuTI, Laravel, Back End, *Website*

I. PENDAHULUAN

Universitas Telkom adalah salah satu Perguruan Tinggi Swasta (PTS) terbaik yang berlokasi di Bandung, Jawa Barat, Indonesia. Keberhasilan Universitas Telkom menjadi salah satu perguruan tinggi terbaik tidak terlepas dari layanan teknologi informasi yang menjadi penopang kemajuan Universitas Telkom. Direktorat PuTI merupakan salah satu direktorat yang berperan dalam bidang teknologi dan layanan informasi serta menjadi salah satu tumpuan keberhasilan Universitas Telkom dalam bersaing dengan institusi lain. Universitas Telkom memiliki data center yang dikelola oleh Direktorat PuTI, kepemilikan data center menjadi direktorat PuTI salah satu direktorat yang memiliki banyak kunjungan dari berbagai pihak. Akan tetapi, hingga kini manajemen kunjungan pada PuTI masih dikelola secara manual. Hal tersebut kurang relevan pada era teknologi saat ini. Perkembangan teknologi mampu menggantikan sistem pengelolaan kunjungan secara manual dengan solusi digital

yang lebih canggih. Digitalisasi pengelolaan kunjungan mampu meningkatkan keakuratan data, mengurangi beban administratif, dan mempercepat proses pengolahan informasi. Laravel menjadi salah satu *framework* yang relevan dalam menciptakan pengelolaan kunjungan berbasis *website*. Laravel menyediakan berbagai fitur yang mendukung penciptaan pengelolaan kunjungan berbasis *website*. Penggunaan Laravel dalam merancang backend mampu menciptakan pengelolaan kunjungan yang lebih efisien dan terintegrasi.

II. KAJIAN TEORI

A. *Website*

Website adalah layanan penyebaran informasi yang dapat diakses melalui jaringan internet. *Website* adalah elemen elemen yang mencakup tulisan, audio, dan gambar yang menarik dalam penyebaran informasi sehingga *website* menjadi media yang menarik dan mudah untuk diakses. Ada berbagai jenis *website* yaitu : *website* statis, *website* dinamis, *website* interaktif[1].

B. Laravel

Laravel adalah salah satu *framework* perangkat lunak untuk mengembangkan aplikasi web menggunakan PHP[2]. Laravel mengadakan seperangkat fitur dan alat yang membuat penciptaan *website* menjadi lebih mudah. Laravel mampu menyederhanakan sebuah pengembangan dengan struktur yang teroganisir, sintaks yang sederhana dan ekspresif. Penggunaan *framework* Laravel mempermudah dikarenakan menggunakan konsep *Model-View-Controller* (MVC) yang memisahkan model, tampilan (*view*), dan pengendali (*controller*). Konsep MVC merupakan konsep yang populer dikarenakan mampu meningkatkan skabilitas aplikasi[3].

C. *Front End*

Front end adalah tampilan antar muka *website* yang terlihat oleh pengguna *website*. Front end dapat diciptakan menggunakan HTML, Javascript, CSS[2]. Javascript berfungsi mengontrol fungsionalitas elemen dan CSS berfungsi mengatur tata letak tampilan.

D. Back End

Pada sebuah *website* tentu memiliki back end yang berfungsi menyimpan, mengolah, dan mengirimkan data kepada tampilan muka *website*. Back end merupakan tampilan yang tidak dapat dilihat oleh pengguna. Back end dapat diciptakan menggunakan bahasa PHP. Back end umumnya mencakup server web yang berfungsi melakukan hosting aplikasi[2]. Penggunaan PHP pada back end bertujuan berkomunikasi dengan database untuk menyajikan informasi kepada pengguna.

III. METODE

Perancangan backend menggunakan *framework* Laravel membutuhkan beberapa metode perancangan yang bertujuan dalam menyelesaikan perancangan backend menggunakan *framework* Laravel.

A. Perancangan Use Case Diagram

Perancangan *use case diagram* didasarkan oleh kebutuhan sistem yang menjelaskan hubungan antara pengunjung dan admin [4].

B. Perancangan Activity Diagram

Setelah melakukan perancangan *use case diagram*, maka hal berikut yang perlu dilakukan adalah perancangan *activity diagram*. *Activity diagram* berfungsi menjalankan fungsi sebuah sistem.[4]

C. Perancangan Database

Perancangan database dibutuhkan setelah dilakukan perancangan *use case diagram* dan perancangan *activity diagram*. Perancangan database berfungsi menyimpan data pengguna[4].

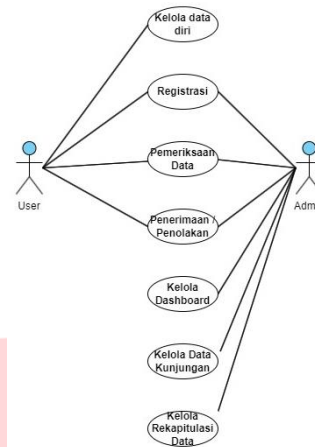
D. Implementasi

Setelah dilakukan berbagai perancangan maka tahap berikut yang dilakukan adalah implementasi menggunakan *framework* Laravel menggunakan Bahasa pemrograman PHP[4].

E. Pengujian

Setelah diselesaikannya perancangan *backend* dengan *framework* Laravel pada *website* maka diperlukan sebuah pengujian yang bertujuan memeriksa sistem *website* berjalan dengan baik dan dapat digunakan. Pengujian dilakukan dengan menggunakan *Nessus Essentials* dan *Pentest Tools*. Pengujian bertujuan memeriksa sistem keamanan *website*.

yang berfungsi melakukan pemeriksaan data pengunjung, pengelolaan data pengunjung, dan pengelolaan data pengunjung. Dua aktor yaitu user dan admin saling keterkaitan. .



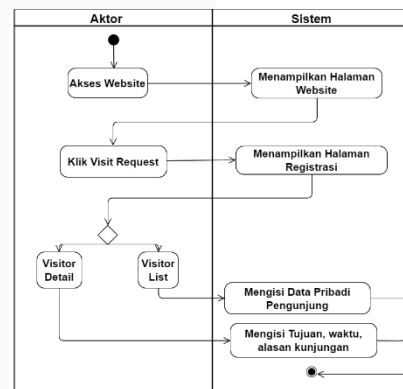
GAMBAR 1
Use Case Diagram

B. Rancangan Activity Diagram

Rancangan *activity diagram* merupakan kegiatan sebuah sistem yang menjelaskan berbagai proses yang terjadi pada sebuah sistem. *Activity diagram* merupakan tindak lanjut dari *use case diagram* yang berfokus kepada sistem bukan kepada pengguna[5]. Pada perancangan ini *activity diagram* dibagi menjadi dua yaitu :

1. Activity Diagram Pada User (Pengunjung)

Activity diagram pada user menjelaskan sistem yang terjadi pada bagian user.



GAMBAR 2
Activity Diagram Pada User

IV. HASIL DAN PEMBAHASAN

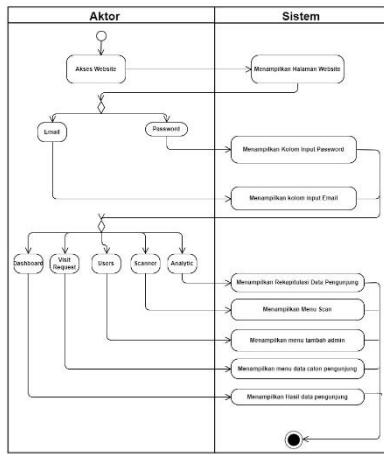
Setelah dilakukan berbagai metode perancangan backend menggunakan *framework* Laravel maka berikut hasil perancangan dan hasil pengujian dari perancangan backend menggunakan *framework* Laravel.

A. Rancangan Use Case Diagram

Rancangan *use case diagram* bertujuan menjelaskan hubungan antara pengunjung dan juga PuTI. Pada rancangan ini terdapat dua aktor yang berperan yaitu *user* (pengunjung) yang berfungsi melakukan registrasi kunjungan dan admin

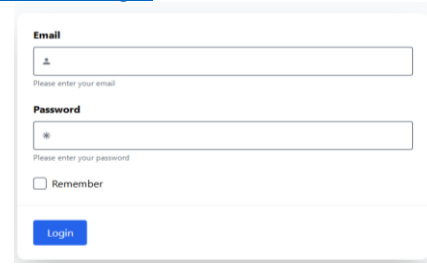
2. Activity Diagram Pada Admin

Activity diagram pada admin menjelaskan sistem yang terjadi pada bagian admin.



GAMBAR 3 Activity Diagram Pada Admin

Admin dapat mengakses *website* pada <https://saw-visitor-management.com/login>

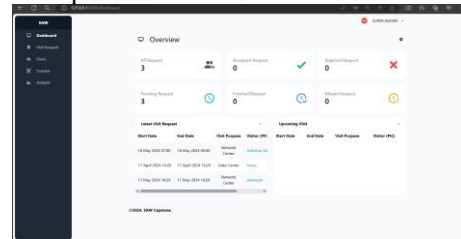


GAMBAR 7 Tampilan Halaman Login

Pada halaman *login* akan muncul kolom email dan *password* yang wajib diisi oleh admin untuk dalam akses masuk ke menu taskbar admin.

d. Tampilan Halaman *Dashboard*

Kemudian admin akan beralih ke menu *dashboard*. Pada halaman *dashboard* akan menampilkan tentang catatan kunjungan yang berisi total kunjungan, kunjungan yang diterima, kunjungan yang tolak, kunjungan yang ditunda, kunjungan yang telah selesai, kunjungan yang terlewatkan, data pengunjung terakhir yang melakukan kunjungan, dan data pengunjung yang akan melakukan kunjungan. Pada halaman *dashboard* akan memberikan informasi yang signifikan kepada admin. Informasi diambil dari database kemudian ditampilkan pada halaman *dashboard* menggunakan perintah *backend*.



GAMBAR 8 Tampilan Halaman Dashboard

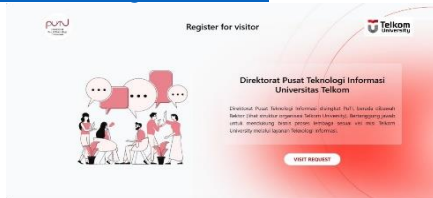
C. Implementasi Website

Implementasi *website* dibagi menjadi dua bagian pada satu *website* yaitu pada bagian user (pengunjung) dan pada bagian admin.

1. Tampilan Website Pada User (Pengunjung)

a. Tampilan Landing Page Website

Pengunjung dapat melakukan akses *website* pada <https://saw-visitor-management.com/>

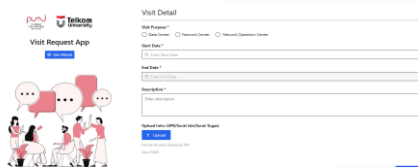


GAMBAR 4 Tampilan Landing Page

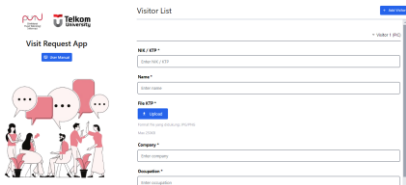
Jika pengunjung ingin melakukan kunjungan maka pengunjung melakukan klik pada kolom "Visit request"

b. Tampilan Halaman Registrasi

Kemudian pengunjung akan dialihkan kepada halaman registrasi.



GAMBAR 5 Tampilan Halaman Registrasi Bagian 1



GAMBAR 6 Tampilan Halaman Registrasi Bagian 2

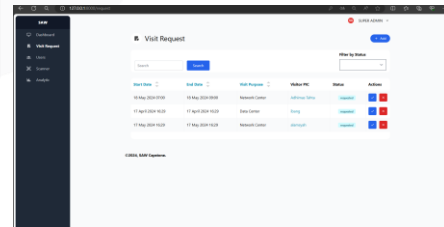
Pada halaman registrasi ini pengunjung melakukan pengisian data pribadi pengunjung.

2. Tampilan Website Pada Admin

c. Tampilan Halaman Login

e. Tampilan Halaman Visit request

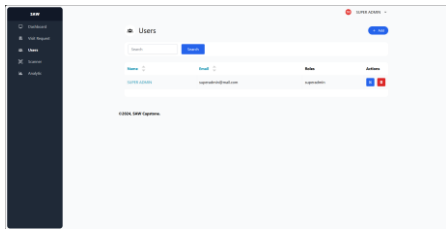
Halaman *visit request* berfungsi melakukan penerimaan atau penolakan kunjungan yang ditentukan oleh admin. Data pada halaman *visit request* diambil dari database yang telah dikelompokkan kemudian ditampilkan pada halaman *visit request*.



GAMBAR 9 Tampilan Halaman Visit request

f. Tampilan Halaman Users

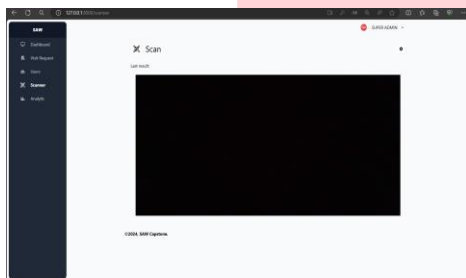
Halaman users dikhususkan kepada superadmin yang bertujuan menambahkan admin baru. Penambahan admin baru akan disimpan pada database.



GAMBAR 10
Tampilan Halaman Users

g. Tampilan Halaman Scanner

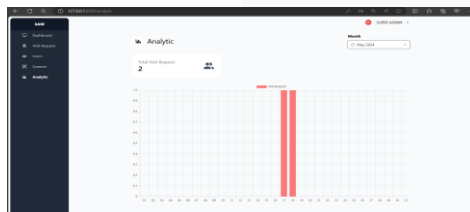
Pada halaman scanner berfungsi melakukan aktivitas scan QR Code yang telah dikirimkan kepada pengunjung etika tiba waktu kunjungan. Pemeriksaan data akan dilakukan ketika melakukan aktivitas scan QR Code. Apabila data tersedia pada database maka scan QR Code berhasil akan tetapi apabila data tidak tersedia pada database maka hasil scan QR Code tidak berhasil.



GAMBAR 11
Tampilan Halaman Scanner

h. Tampilan Halaman Analytic

Pada halaman analytic berfungsi menampilkan hasil rekapitulasi data pengunjung yang telah melakukan pendaftaran kunjungan. Hasil rekapitulasi kunjungan pada halaman analytic juga dapat diekspor ke microsoft excel.



GAMBAR 12
Tampilan Halaman Analytic

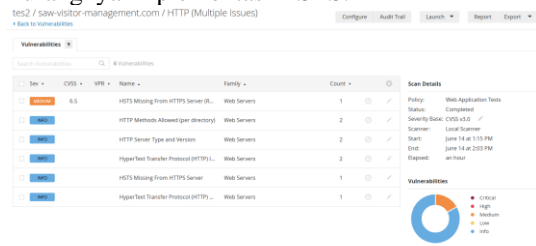
D. Hasil Pengujian

Pengujian pada perancangan backend menggunakan framework pada website bertujuan menemukan kelemahan dari hasil rancangan. Pengujian dilakukan dengan memeriksa sistem keamanan yang berfungsi melakukan serangan pada sistem keamanan. Pengujian dilakukan dengan dua alat yaitu menggunakan Nessus Essential dan Pentest Tools.

1. Nessus Essentials

Nessus berfungsi untuk sebagai pemindai keamanan yang akan memindai jaringan yang dimaksud, yang akan menemukan kelemahan dari jaringan tersebut. Nessus juga menyediakan penilaian tingkat kerentanan yang mencakup mulai dari Kritis, Tinggi, Sedang, Rendah, dan terakhir Informasi, berdasarkan Skor Keparahannya CVSSv2. Pengujian menggunakan Nessus dilakukan sebanyak 4 kali. Hasil

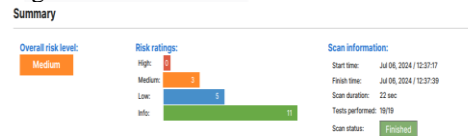
pengujian diperoleh bahwa tingkat keamanan berada pada level medium. Kelemahan sistem keamanan yang ditemukan yaitu kurangnya implementasi HSTS.



GAMBAR 13
Hasil Pengujian dengan Nessus Essentials

2. Pentest Tools

Pengujian sistem keamanan tidak hanya dilakukan dengan Nessus essentials. Pengujian sistem keamanan juga menggunakan pentest tools yang bertujuan sama dengan nessus essential. Dengan menggunakan pentest tools, pada website dapat ditemukan celah sebelum penyerangan sebenarnya dilakukan, sehingga memungkinkan untuk memperkuat keamanan dan mencegah potensi pelanggaran data pada website sebelum website resmi digunakan. Pengujian menggunakan pentest tool dilakukan sebanyak 4 kali dan berdasarkan hasil pengujian diperoleh bahwa tingkat keamanan berada pada level medium dan menemukan kelemahan pada *Insecure cookie setting: missing Secure flag*. Hal ini menunjukkan bahwa cookie dikirim tanpa tanda aman, sehingga rentan terhadap serangan melalui koneksi HTTP yang tidak aman.



GAMBAR 14
Hasil Pengujian dengan Pentest Tools

V. KESIMPULAN

Perancangan backend menggunakan framework Laravel pada website pengelolaan kunjungan di PuTI Universitas Telkom mempercepat waktu perancangan dikarenakan telah tersedia berbagai fitur yang mempermudah. Pemanfaatan framework laravel menggunakan konsep Model-View-Controller (MVC) menggunakan Bahasa pemrograman PHP memudahkan perancangan backend dan memudahkan terhubungnya dengan database dan menampilkan informasi kepada tampilan website. Pengelolaan kunjungan menggunakan website memudahkan dalam melakukan pengelolaan data dan mempermudah penyimpanan data dengan membangun sebuah web menggunakan bahasa pemrograman PHP dengan framework Laravel yang memiliki konsep MVC dan menggunakan SQLite sebagai database.

REFERENSI

- [1] K. N. Ramadha, "WEBSITE NOVEL KOMEDI WEBNOVEL MENGGUNAKAN PHP DAN MYSQL," vol. 1, no. 4, 2022.
- [2] "LARAVEL PHP FRAMEWORK AND PROJECT CREATED USING LARAVEL," *International Journal of Progressive Research in Engineering Management and Science*, Apr 2023, doi: 10.58257/ijprems30976.
- [3] R. Kumala Zai, N. Ambarsari, dan S. Suakanto, "PERANCANGAN BACK END WEBSITE MITRA.ID PADA MODUL UMKM DENGAN METODE ITERATIVE INCREMENTAL."
- [4] R. Pangestika dan R. T. Dirgahayu, "Pengembangan Back-end Sistem Informasi Pendataan Sekolah Desa Komunitas Pendar Foundation Yogyakarta."
- [5] F. Rahmansyah dan I. Nuryasin, "Pembangunan Back-End Untuk Meningkatkan Fungsionalitas Buku Sekolah Elektronik (BSE) Tematik 8 Menggunakan Application Programming Interface (API)," *REPOSITOR*, vol. 4, no. 2, hlm. 173–184, 2022.