

PENGEMBANGAN *DASHBOARD TRIVY* BERBASIS WEBSITE MENGGUNAKAN REACT JS DAN GOLANG

Aquenov Alexandro Kroons ¹, Christine Dewi ^{2*}.

^{1,2*} Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana, Kota Salatiga, Provinsi Jawa Tengah, Indonesia.

Email: 672019161@student.uksw.edu ¹, christine.dewi@uksw.edu ^{2*}

Histori Artikel:

Dikirim 1 Juni 2023; *Diterima dalam bentuk revisi* 19 Juni 2023; *Diterima* 2 Juli 2023; *Diterbitkan* 10 September 2023. Semua hak dilindungi oleh Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) STMIK Indonesia Banda Aceh.

Abstrak

Container image dapat membantu developer dalam pengembangan aplikasi. Dalam pembuatan container image, diperlukan untuk melakukan pemeriksaan keamanan sebelum dapat digunakan oleh orang lain. Hal ini membuat pentingnya bagi tim untuk menemukan solusi dengan mendeteksi kerentanan dari container image yang digunakan. Trivy bertujuan untuk memindai kerentanan container image dan menggunakan database Common Vulnerabilities and Exposures (CVE) sebagai informasi dari kerentanan yang ditemukan pada container image. Penelitian ini mencoba untuk merancang sebuah dashboard menggunakan React JS dan Golang sebagai bahasa pemrograman untuk membangun sisi frontend dan backend serta menggunakan Trivy yang merupakan tool berbasis command line interface untuk melakukan pemeriksaan kerentanan dari container image. Penelitian dilakukan dengan menggunakan metode Software Development Life Cycle (SDLC). Penelitian ini menghasilkan sebuah dashboard berbasis website yang memiliki antarmuka user-friendly dan dapat digunakan tanpa ada batasan sistem operasi.

Kata Kunci: Container Image; React JS; Golang; Trivy.

Abstract

Container images can help developers in application development. In creating a container image, it is necessary to perform security checks before it can be used by others. This makes it important for the team to find a solution by detecting the vulnerability of the container image used. Trivy aims to scan container image vulnerabilities and uses the Common Vulnerabilities and Exposures (CVE) database as information from the vulnerabilities found in the container image. This research tries to design a dashboard using React JS and Golang as programming languages to build the frontend and backend sides and use Trivy which is a command line interface-based tool to check the vulnerability of container images. The research was conducted using the Software Development Life Cycle (SDLC) method. This research produces a website-based dashboard that has a user-friendly interface and can be used without any operating system restrictions.

Keyword: Container Image; React JS; Golang; Trivy.

1. Pendahuluan

Seiring dengan perkembangan teknologi, hidup manusia menjadi lebih mudah dan terbantu dikarenakan berbagai macam teknologi yang ada [1]. Dengan adanya teknologi, manusia dapat berkomunikasi antara satu dengan yang lain secara langsung dari tempat yang berbeda [2], menggunakan teknologi sebagai media pembelajaran [3], mendapatkan berbagai informasi yang ada [4], dan sebagainya. Hal ini memungkinkan manusia dapat melakukan apa pun dimana saja dan kapan saja hanya dengan menggunakan teknologi.

Aplikasi merupakan salah satu contoh dari perkembangan teknologi. Pada umumnya aplikasi dibangun menggunakan arsitektur *monolithic* dan *microservices* [5]. Arsitektur *monolithic* mengembangkan aplikasi secara *single unit* dimana semua bagian berada di dalam satu sistem sehingga proses untuk mengubah satu komponen diharuskan untuk menggarap seluruh aplikasi sedangkan arsitektur *microservices* mengembangkan aplikasi dengan cara memecahkan bagian – bagian atau *services* yang lebih kecil dan disusun menjadi satu aplikasi sehingga skalabilitas yang dimiliki oleh arsitektur *microservices* lebih baik dibandingkan dengan arsitektur *monolithic* [6]. Dengan arsitektur *microservices*, *developer* dapat lebih mudah dan bebas untuk bekerja secara tim karena fitur – fitur yang ada di dalamnya dapat bekerja sendiri tanpa terhubung dengan fitur – fitur lainnya atau membuat *environment* sendiri yang berbeda dari *environment* utama dengan adanya *container image*. Tim *developer* membuat *container image* dan menggunakannya untuk menggarap aplikasi yang dibuat.

Dengan menggunakan *container image*, terdapat dampak besar pada keamanan karena banyaknya *container image* yang memiliki kerentanan dan terus bertambah seiring waktu [7]. Hal ini membuat pentingnya bagi tim untuk menemukan solusi dengan mendeteksi kerentanan dari *container image* yang digunakan. Dengan adanya masalah tersebut maka diciptakanlah Trivy.

Trivy merupakan alat pemindaian yang bertujuan untuk memindai kerentanan *container image* dan menggunakan *database Common Vulnerabilities and Exposures (CVE)* sebagai informasi dari kerentanan yang ditemukan pada *container image* [8], [9]. Trivy merupakan alat berbasis *open-source* sehingga dapat dimodifikasi dan dikembangkan secara bebas [10]. Proses pemindaian yang dilakukan oleh Trivy dimulai dari membaca *metadata* dari sebuah *package manager* di dalam *container image*. Trivy akan menemukan nama *package* beserta dengan versinya yang kemudian akan dibandingkan dengan informasi kerentanan yang terdapat pada *database CVE*. [11]

Dashboard merupakan sebuah model aplikasi yang disediakan bagi para manager untuk memvisualisasikan data untuk mencapai tujuan secara efektif dan efisien [12]. Pengembangan *dashboard* berguna untuk memberikan informasi kepada pengguna dengan cepat dan akurat sesuai kebutuhan [13]. *Dashboard* penting dalam suatu website karena menampilkan data yang singkat untuk membantu pengguna agar dapat mengambil keputusan dengan cepat [14]. Pengembangan sebuah *dashboard website* bisa menggunakan bahasa pemrograman seperti Golang dan library seperti React JS. React JS merupakan *library Javascript* yang berguna untuk membangun antarmuka pengguna [15]. React JS memudahkan *developer* dalam mengembangkan sebuah *web app* karena berbasis komponen yang interaktif, *stateful*, dan *reusable* [16]. Dengan menggunakan teknik *single page application (SPA)*, React JS dapat melakukan proses *rendering* halaman lebih cepat dikarenakan React JS hanya akan memuat data tertentu yang dibutuhkan oleh pengguna [17]. Go atau yang sering disebut sebagai golang merupakan bahasa pemrograman yang dibuat di google pada tahun 2009 [18]. Golang dikembangkan dengan tujuan agar menjadi bahasa pemrograman yang memiliki keunggulan dari sisi keamanan, kesederhanaan, dan skalabilitas [19]. Golang adalah bahasa yang dibangun menggunakan bahasa pemrograman C dan C++ serta merupakan salah bahasa yang dapat dikompilasi. [20]

Trivy berbasis *command line interface* dan hanya dapat diinstall di beberapa sistem operasi seperti MacOS dan Linux. Hal ini membuat pengguna sistem operasi lainnya seperti Windows tidak dapat menggunakan trivy untuk melakukan pemindaian *container image*. Oleh karena itu, dalam penelitian ini dikembangkan *dashboard* trivy berbasis *web* yang bertujuan agar pemindaian *container image* dapat dilakukan tidak terbatas pada beberapa sistem operasi saja dan memiliki antarmuka *user-friendly*. Untuk mewujudkan solusi tersebut, maka digunakan bahasa pemrograman React JS untuk sisi *frontend* dan Golang untuk sisi *backend*.

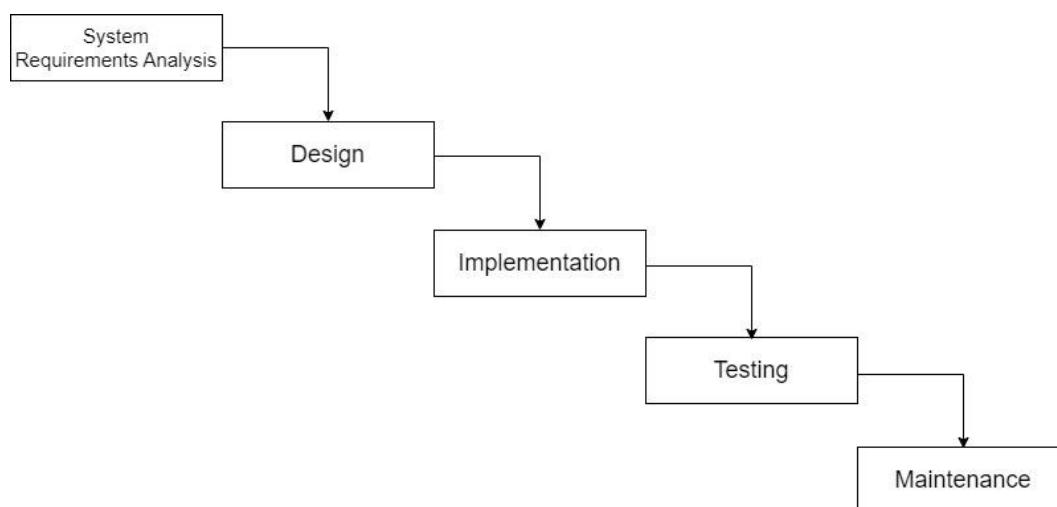
Pada penelitian yang berjudul Perancangan Dashboard Monitoring Painting Defect Berbasis Website, membahas tentang perancangan *dashboard* yang bertujuan untuk melakukan pengumpulan, peninjauan ulang dan pelaporan dari temuan *defect* hasil pengerjaan menggunakan PHP dan HTML. Hasil pada penelitian ini menunjukkan bahwa sistem *dashboard* yang dibuat dapat memberikan informasi suatu laporan yang bersifat sistematis [21]. Penelitian di atas memiliki keselarasan terhadap penelitian ini, dimana proses menampilkan data di implementasikan dalam bentuk *dashboard*. Namun penelitian tersebut menggunakan bahasa pemrograman PHP sebagai *backend* dan HTML sebagai *frontend* dari *dashboard website* yang dibangun sedangkan penelitian ini menggunakan Golang dan React JS.

Pada penelitian yang berjudul Teknik Single Page Application (SPA) Layout Web Dengan Menggunakan React Js Dan Bootstrap, membahas tentang perancangan *layout web* menggunakan React JS. Hasil pada penelitian ini menunjukkan bahwa perancangan *layout web* menggunakan *library* React JS dengan teknik SPA dapat memaksimalkan kinerja *browser* agar proses menampilkan isi dari keseluruhan halaman dengan sempurna [22]. Penelitian di atas memiliki keselarasan terhadap penelitian ini, dimana penelitian tersebut menggunakan React JS sebagai *library* untuk sisi *frontend* karena mengadopsi teknik SPA yang memiliki kelebihan dari segi kecepatan. Namun penelitian tersebut menggunakan Bootstrap yang merupakan framework dari CSS untuk mengatur *layout* dari *web* yang dibangun, berbeda dengan penelitian ini yang menggunakan Material UI yang merupakan *library* dari React JS.

Pada penelitian yang berjudul Designing website vaccine booking system using golang programming language and framework react JS, membahas tentang pengembangan *website* yang dapat membantu masyarakat selama masa pandemi dengan tujuan agar tidak terjadinya antrian yang dapat menambah jumlah kasus Covid-19 yang berada di Indonesia. Hasil penelitian ini menunjukkan bahwa pengembangan *website* yang dibuat dapat menjadi alternatif untuk mempermudah masyarakat dalam melakukan pendaftaran vaksinasi [23]. Penelitian di atas memiliki keselarasan terhadap penelitian ini, dimana pengembangan *website* yang dibuat menggunakan React JS dan Golang. Namun, penelitian tersebut bertujuan untuk membantu masyarakat selama pandemi sedangkan penelitian ini bertujuan untuk membantu *developer* agar dapat melakukan pemindaian *container image* tanpa terbatas oleh beberapa sistem operasi.

2. Metode Penelitian

Metode pengembangan perangkat lunak atau biasanya disebut sebagai *Software Development Life Cycle* (SDLC) merupakan suatu tahapan analisis aplikasi yang digunakan oleh perusahaan *vendor* aplikasi untuk mengembangkan aplikasi yang akan diproduksi. Proses yang dilakukan dalam SDLC dimulai dari analisa kebutuhan, perancangan, pengujian, sampai ke proses pemeliharaan [24]. Metode Pengembangan Perangkat Lunak yang digunakan dalam penelitian ini adalah *Waterfall*. Metode ini merupakan metode yang dilakukan dengan cara menyelesaikan suatu tahap sebelum tahap sebelumnya diselesaikan [25]. Metode ini dipilih karena cocok untuk melakukan pengembangan sistem baru [26]. Metode *Waterfall* memberikan kerangka kerja yang terstruktur dan linear dalam pengembangan perangkat lunak. Namun, metode ini kurang fleksibel terhadap perubahan kebutuhan yang mungkin terjadi selama proses pengembangan. Oleh karena itu, penting untuk melakukan analisis kebutuhan yang teliti dan menyediakan waktu yang cukup untuk perencanaan sebelum memulai setiap tahap dalam metode *Waterfall*. Metode *waterfall* adalah metode yang memiliki beberapa tahapan sebagai berikut:



Gambar 1 Metode Waterfall

1) *System Requirements Analysis*

Pada tahapan yang pertama metode waterfall ini, dilakukan analisa segala kebutuhan yang diperlukan dilihat dari sisi kebutuhan pengguna dengan melakukan studi literatur terkait dengan alat - alat yang digunakan dalam mengembangkan *dashboard* trivy.

2) *Design*

Kemudian pada tahapan desain, dibuat rancangan sistem berupa *Unified Modelling Language* (UML) yang dapat membantu memberikan gambaran mengenai apa yang akan dikerjakan dan kemudian akan diimplementasikan pada tahap berikutnya.

3) *Implementation*

Jika tahapan desain telah selesai merancang *tools system* yang dibutuhkan, sehingga tahapan selanjutnya adalah untuk mentranslasikan desain tersebut ke dalam kode pemrograman.

4) *Testing*

Setelah proses implementasi selesai, selanjutnya akan dilakukan pemeriksaan dan pengujian program secara keseluruhan untuk menemukan kemungkinan adanya kegagalan atau kesalahan pada sistem.

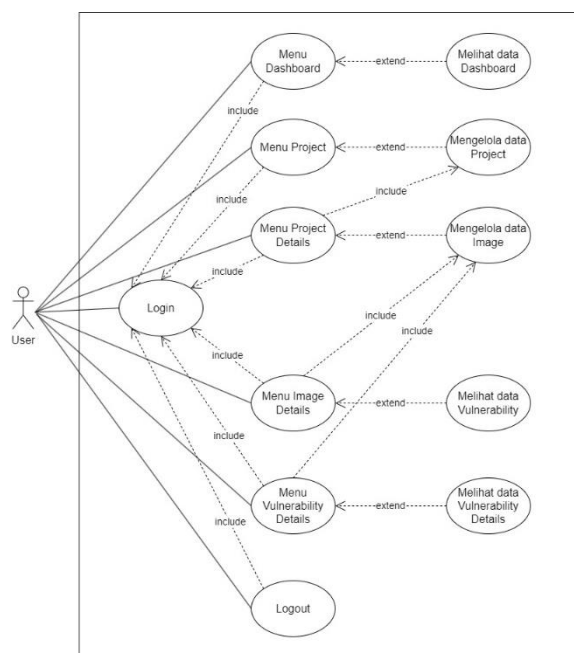
5) *Maintenance*

Tahap terakhir dalam metode ini adalah dengan melakukan pemeliharaan sistem yang bertujuan untuk memperbaiki kesalahan – kesalahan yang ada pada sistem tersebut sehingga sistem yang baru dapat berjalan lebih baik daripada sistem sebelumnya.

3. Hasil dan Pembahasan

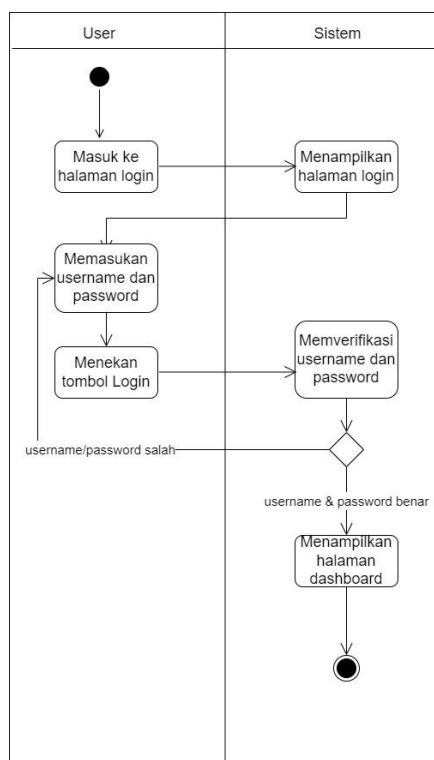
3.1 Design

Bagian ini membahas tentang pembuatan *Unified Modelling Language* (UML) yang meliputi *use case diagram* dan *activity diagram*. *Use case diagram* pada gambar 2 menjelaskan tentang struktur dari seorang *user* (pengguna) yang akan menggunakan website yang dirancang. *User* dapat mengelola *project*, melakukan pemindaian *container image*, dan melihat kerentanan dari sebuah *container image* yang telah dipindai.

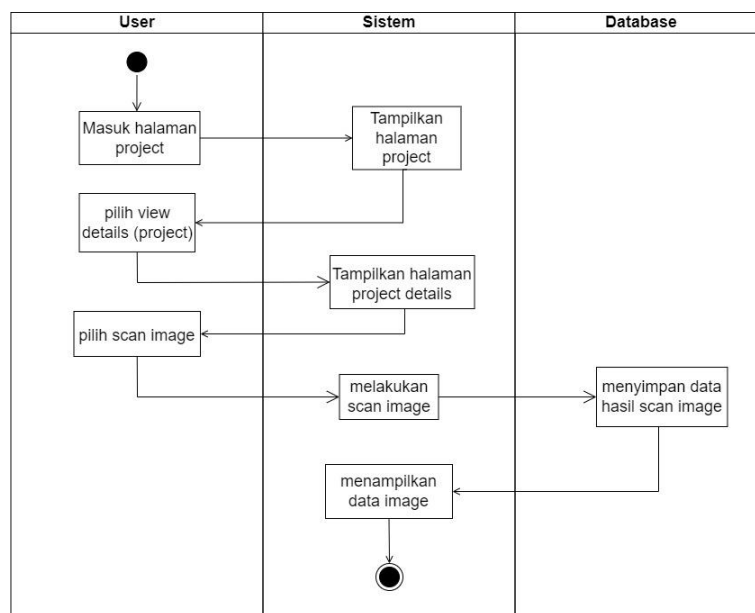


Gambar 2. Use Case Diagram

Activity diagram dibuat untuk menggambarkan proses – proses yang terjadi dari titik awal sampai akhir pada sebuah sistem [27]. Proses *login* yang dilakukan oleh user telah digambarkan pada sebuah *activity diagram* yang dapat dilihat pada gambar 3. Proses pemindaian *image* juga telah digambarkan pada sebuah *activity diagram* yang dapat dilihat pada gambar 4.



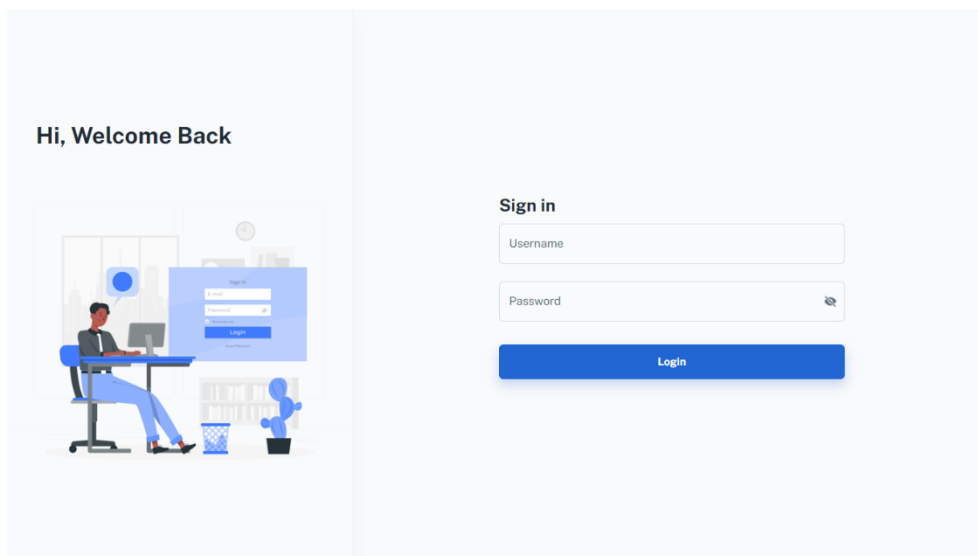
Gambar 3. Activity Diagram Login



Gambar 4. Activity Diagram Image Scan

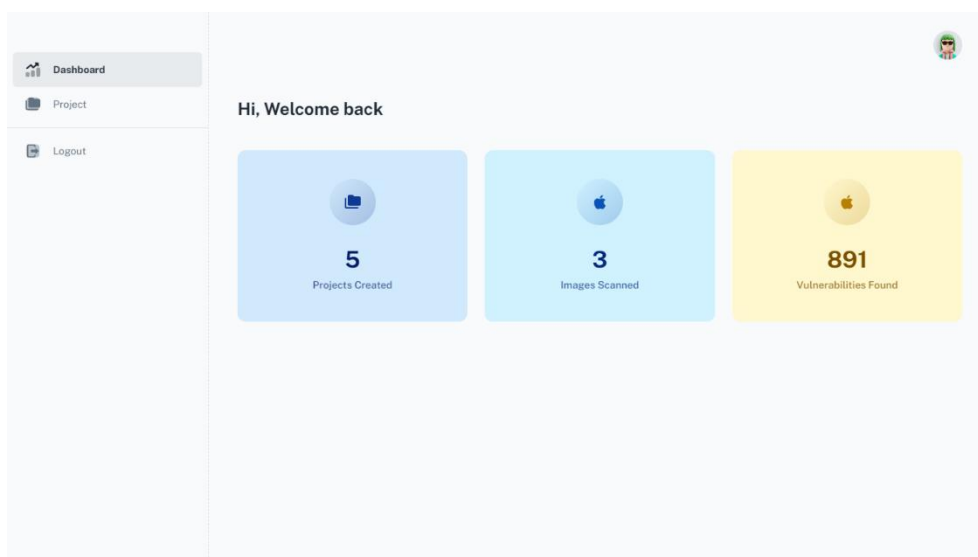
3.2 Implementation

Bagian ini membahas tentang bentuk sebuah *website* yang dirancang menggunakan React JS dan Golang. Halaman *login* merupakan halaman yang mengharuskan pengguna untuk melakukan proses *login* dengan cara memasukkan *username* dan *password* terlebih dahulu sebelum dapat menggunakan *website* tersebut sebagaimana ditampilkan pada gambar 5.

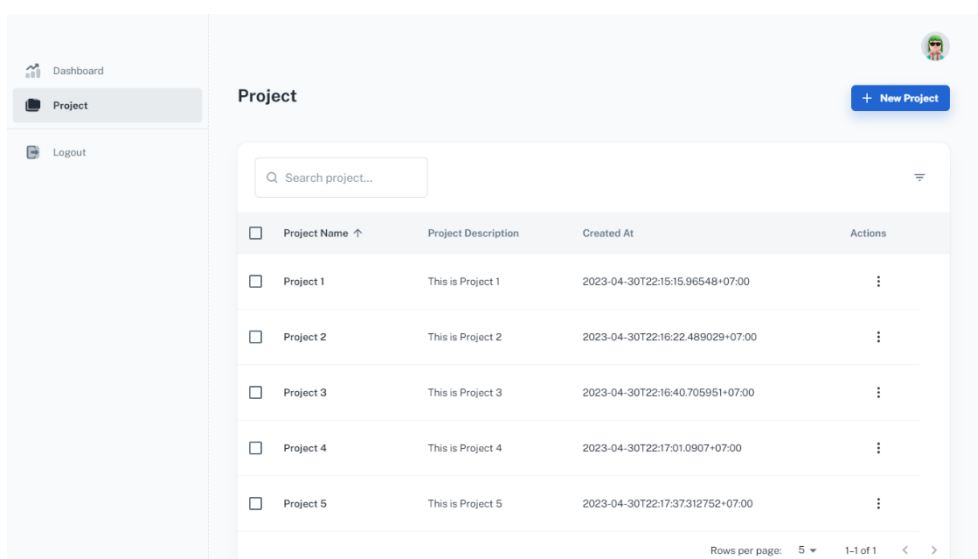


Gambar 5. Tampilan Halaman Login

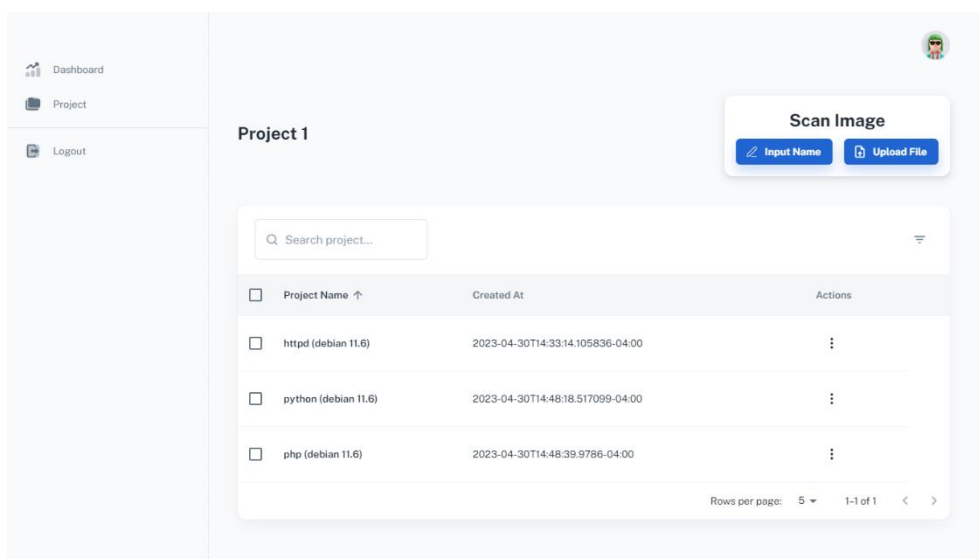
Halaman *dashboard* merupakan halaman yang menampilkan jumlah data secara garis besar seperti jumlah *project*, jumlah *image*, dan jumlah *vulnerability* yang sebagaimana ditampilkan pada gambar 6.

Gambar 6. Tampilan Halaman *Dashboard*

Halaman *project* merupakan halaman utama dari *website* yang dikembangkan. Halaman ini menampilkan seluruh *project* yang dibuat oleh pengguna. Halaman ini memungkinkan pengguna dapat melakukan beberapa aktivitas seperti menambahkan *project*, *edit project*, hapus *project*, dan masuk ke halaman *project details* untuk melihat detail dari masing – masing *project* sebagaimana ditampilkan pada gambar 7.

Gambar 7. Tampilan Halaman *Project*

Halaman *project details* merupakan halaman yang dapat melakukan proses pemindaian *container image* dengan cara memasukkan nama *image* atau mengunggah *image* dalam bentuk *file*. Seluruh *image* yang telah dipindai akan ditampilkan ke dalam halaman ini. Halaman *project details* juga memungkinkan pengguna sehingga dapat menghapus *image* yang telah dipindai atau melihat detail dari *image* yang dipilih sebagaimana ditampilkan pada gambar 8.



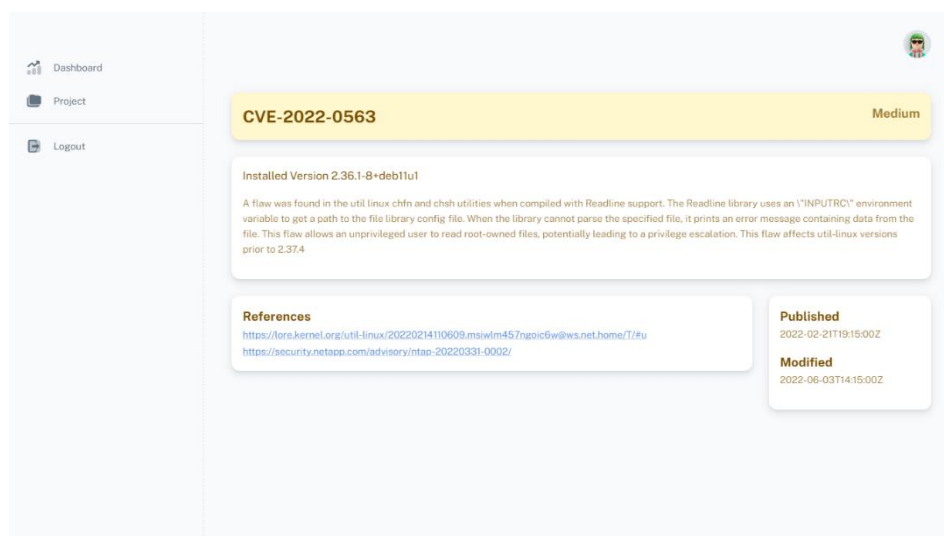
Gambar 8. Tampilan Halaman *Project Details*

Halaman *image details* merupakan halaman yang menampilkan hasil pemindaian dari sebuah *image*. Hasil pemindaian berupa *vulnerability* atau kerentanan yang ditemukan pada sebuah *image* yang berisi informasi seperti *Vulnerability ID*, versi yang diinstall, tingkat kerentanan, dan deskripsi dari kerentanan tersebut. Informasi kerentanan secara detail dapat dilihat dengan menekan tombol *View Details* sebagaimana ditampilkan pada gambar 9.



Gambar 9. Tampilan Halaman *Image Details*

Halaman *vulnerability details* merupakan halaman yang menampilkan informasi terperinci dari sebuah kerentanan. Halaman ini menampilkan informasi yang sama seperti halaman sebelumnya tetapi memiliki sedikit tambahan seperti referensi dari sebuah kerentanan dan tanggal terbit serta tanggal modifikasi dari informasi yang diberikan sebagaimana ditampilkan pada gambar 10.

Gambar 10. Tampilan Halaman *Vulnerability Details***Kode Program 1 Kode Program untuk Unggah *Image File***

```
const acceptedFileItems = acceptedFiles.map(file => {
  console.log(file);
  return (
    <Card sx={{m: 2, height: 50, width: 400, display: 'flex', alignItems:
      'center', justifyContent: 'flex-start'}}>
      <InsertDriveFile color='primary' sx={{ m: 1, fontSize: 30 }} />
      {file.name}
    </Card>
  )
});

const fileRejectionItems = fileRejections.map(({ file, errors }) => {
  console.log(file, errors);
  return (<li key={file.path}>
    {file.path} - {file.size} bytes
    <ul>
      {errors.map(e => (
        <li key={e.code}> {e.message} </li>
      ))}
    </ul>
  </li>)
});

const handleSubmitFile1 = (event) => {
  event.preventDefault()

  axios.post(`${process.env.REACT_APP_BACKEND}/uploadFile`, formData,
    config)
    .then(response => {
      console.log(response);
    })
    .catch(err => {
      console.log(err.response.data.Message)
    });
}

useEffect(() => {
  setSelectedFiles(acceptedFiles[0])
}, [acceptedFiles])
```

Kode program 1 merupakan fungsi untuk melakukan pengunggahan *image file*. Fungsi ini akan melakukan pengecekan *file* yang diunggah oleh pengguna. *File* yang dengan ekstensi *.tar* merupakan ekstensi *file* yang dapat digunakan untuk melakukan proses pemindaian *image*. *File* yang diunggah akan dikirim ke sisi *backend* dengan menggunakan *method post* untuk melakukan proses pemindaian.

Kode Program 2 Kode Program untuk *Image Scan*

```
func ScanbyTar(c *gin.Context) {
    file, err := c.FormFile("file")
    loggers := logrus.New()
    file_name := file.Filename
    path := "app/repository/tar_files/" + file_name
    _, file_exist := os.Stat(path)
    if file_exist != nil {
        if err := c.SaveUploadedFile(file, path); err != nil {
            JSON(http.StatusInternalServerError, gin.H{
                "message": "Unable to save the file",
            })
        }
    }
    id := uuid.New().String()
    image_name := file_name
    date := time.Now().Format("01-02-2006")
    title, version, variants := helper.SplitString(image_name)
    tipe := "tar"
    var history entity.History
    if db_file := database.PostgreSQL.First(&history, "image = ?",
    image_name).Error; db_file == nil {
        database.PostgreSQL.First(&history, "image = ?", image_name)
        path_file := history.Path
        result, genInfo := repository.ParseJSON(path_file)
        loggers.Info("This image is already exist. Return the json...")
        JSON(http.StatusOK, gin.H{
            "Info Scanned": map[string]string{
                "date scan": history.Date,
                "time scan": history.Time,
            },
            "General Info": genInfo,
            "Result": result})
    } else if err := database.PostgreSQL.First(&history, "image = ?",
    image_name).Error; err != nil {
        path_out := helper.CreateDir(date, title, tipe)
        full_path_out := path_out + image_name + ".json"
        error := exec.Command("trivy", "image", "-f", "json", "-o",
        full_path_out, "--input", path).Run()
        if error != nil {
            fmt.Println(error.Error())
        }
    }
}
```

Kode program 2 merupakan fungsi melakukan pemindaian *image*. Fungsi ini akan melakukan pengecekan terlebih dahulu dimana jika ada *image file* yang baru diunggah dan merupakan *image file* yang sama dengan *image* yang telah ada, maka data *image* akan menggunakan data dari *image* yang telah ada sebelumnya. Jika *image file* yang baru diunggah belum ada di dalam database, maka *file* tersebut akan dilakukan proses pemindaian.

3.3 Testing

Metode pengujian yang digunakan adalah metode *Black Box*. Metode ini hanya melakukan pengujian dari sisi pengguna atau berfokus pada fungsionalitas dari sebuah program sehingga dapat

melakukan pengujian yang tidak dapat diuji dengan metode *White Box* [28]. Hasil pengujian dapat dilihat pada Tabel 1 dimana fungsi dari website yang dibangun dapat berjalan sesuai yang diharapkan.

Tabel 1. Hasil Pengujian Black Box

No	Aktivitas	Hasil yang diharapkan	Hasil Pengujian
1	User Masuk ke halaman <i>Login</i>	Sistem dapat menampilkan halaman <i>Login</i>	Sistem dapat menampilkan halaman <i>Login</i>
2	User Masuk ke halaman <i>Dashboard</i>	Sistem dapat menampilkan halaman <i>Dashboard</i>	Sistem dapat menampilkan halaman <i>Dashboard</i>
3	User Masuk ke halaman <i>Project</i>	Sistem dapat menampilkan halaman <i>Project</i>	Sistem dapat menampilkan halaman <i>Project</i>
4	User Masuk ke halaman <i>Project Details</i>	Sistem dapat menampilkan halaman <i>Project Details</i>	Sistem dapat menampilkan halaman <i>Project Details</i>
5	User Masuk ke halaman <i>Image Details</i>	Sistem dapat menampilkan halaman <i>Image Details</i>	Sistem dapat menampilkan halaman <i>Image Details</i>
6	User Masuk ke halaman <i>Vulnerability Details</i>	Sistem dapat menampilkan halaman <i>Vulnerability Details</i>	Sistem dapat menampilkan halaman <i>Vulnerability Details</i>
7	User melakukan proses <i>login</i>	User dapat masuk ke halaman <i>dashboard</i>	User dapat masuk ke halaman <i>dashboard</i>
8	User mengelola data <i>project</i>	Sistem dapat menjalankan proses pengelolaan data <i>project</i>	Sistem dapat menjalankan proses pengelolaan data <i>project</i>
9	User melakukan pemindaian <i>image</i>	Sistem dapat melakukan proses pemindaian <i>image</i>	Sistem dapat melakukan proses pemindaian <i>image</i>

4. Kesimpulan

Berdasarkan hasil yang diperoleh dari pembahasan Pengembangan Dashboard Trivy Berbasis Website Menggunakan React JS dan Golang maka dapat disimpulkan bahwa pengembangan *dashboard* trivy menggunakan React JS sebagai *frontend* dan Golang sebagai *backend*. Pengembangan sistem dibuat dalam bentuk *website* sehingga memiliki antarmuka yang lebih *user-friendly* dibandingkan dengan *command line interface* (CLI) dan proses pemeriksaan kerentanan pada *container image* yang awalnya hanya bisa dilakukan pada beberapa sistem operasi seperti Linux dan MacOS, sekarang dapat dilakukan pada sistem operasi lainnya seperti Windows. Untuk kedepannya, dashboard berbasis website ini kiranya dapat dikembangkan dalam segi fitur sehingga dapat lebih memudahkan pengguna dalam melakukan pemindaian *container image*.

5. Daftar Pustaka

- [1] Aminullah, M. (2020). Konsep Pengembangan Diri Dalam Menghadapi Perkembangan Teknologi Komunikasi Era 4.0. *Komunike*. 12(1), pp. 1–23. DOI: <https://doi.org/10.20414/jurkom.v12i1.2243>
- [2] Mahayoni, N. M. S. (2020). PENGGUNAAN APLIKASI ZOOM MEETING PADA PEMBELAJARAN AGAMA HINDU DI MASA PANDEMI. *Jurnal Widya Sastra Pendidikan Agama Hindu*. 3(1), pp. 47–53. DOI: <https://doi.org/10.36663/wspah.v3i1.67>
- [3] Mailoa, E. *et al.* (2023). Pendampingan Pengintegrasian Teknologi Dalam Pembelajaran Di SMAN Saengga, Kabupaten Teluk Bintuni, Papua Barat. 03(02), pp. 353–363.

- [4] Novi Yona Sidratul Munti, & Dwi Asril Syaifuddin. (2020). Analisa Dampak Perkembangan Teknologi Informasi Dan Komunikasi Dalam Bidang Pendidikan. *Jurnal Pendidikan Tambusai*. 4(2), pp. 1799–1805. Retrieved from <https://jptam.org/index.php/jptam/article/view/655>
- [5] Siagian, N., Tamba, T. E., Situmorang, H. H. O., & Samosir, H. (2021). Aplikasi Apotek Berbasis Web Menggunakan Arsitektur Microservices (Studi Kasus Apotek Glen, Kab.Toba). *Journal of Applied Technology and Informatics Indonesia*. 1(2), pp. 22–28. DOI: <https://doi.org/10.54074/jati.v1i2.35>
- [6] Pramakrisna, F. D., Adhinata, F. D., & Tanjung, N. A. F. (2022). Sistem Penilaian Inovasi Karyawan Digital Amoeba Menggunakan Desain Arsitektur Microservice Pada Aplikasi Mobile. *Jurnal Media Informatika Budidarma*. 6(3), pp. 1640–1648. DOI: <https://doi.org/10.30865/mib.v6i3.4187>
- [7] Sultan, S., Ahmad, I., & Dimitriou, T. (2019). Container Security: Issues, Challenges, and the Road Ahead. *IEEE Access*. 7, pp. 52976–52996. DOI: <https://doi.org/10.1109/ACCESS.2019.2911732>
- [8] Berkovich, S., Kam, J., & Wurster, G. (2020). UBCIS: Ultimate Benchmark for Container Image Scanning. *CSET 2020 - 13th USENIX Workshop on Cyber Security Experimentation and Test, co-located with USENIX Security 2020*.
- [9] Ahmed, F. A., & Fatih, D. (2022). Security Analysis of Code Bloat in Machine Learning Systems.
- [10] Jaisinghani, G. (2022). VULNERABILITY MANAGEMENT IN THE AGE OF CONTAINERS – A REVIEW. *International Journal of Information Security*. 1(1), pp. 1–5. DOI: <https://doi.org/10.17605/OSF.IO/WJSV6>
- [11] Doan, T. P., & Jung, S. (2022). DAVS: Dockerfile Analysis for Container Image Vulnerability Scanning. *Computers, Materials and Continua*. 72(1), pp. 1699–1711. DOI: <https://doi.org/10.32604/cmc.2022.025096>
- [12] Rahardja, U., Aini, Q., & Khoirunisa, A. (2019). Monitoring Kinerja User Akuntan Menggunakan Dashboard Pada Web Based Accounting Online Di Perguruan Tinggi. *SATIN - Sains dan Teknologi Informasi*. 4(2), pp. 58–62. DOI: <https://doi.org/10.33372/stn.v4i2.406>
- [13] Jurnal, H., Yulhendri, Y., Faridho, F., & Kurniawan, D. (2022). Implementasi Digital Dashboard Untuk Mengontrol Wilayah Rt/Rw. *Jurnal Informatika dan Teknologi Komputer*. 2(1), pp. 43–54. DOI: <https://doi.org/https://doi.org/10.55606/jitek.v2i1.186>
- [14] Maghfira, S., Sagirani, T., & Amelia, T. (2022). Prototipe Dashboard Aplikasi POSTASY Berbasis Website Menggunakan Metode Pureshare Untuk Meningkatkan Kemudahan Layanan Posyandu. *Jurnal Buana Informatika*. 13(02), pp. 155–164. DOI: <https://doi.org/10.24002/jbi.v13i02.6117>
- [15] Bhalla, A., Garg, S., & Singh, P. (2020). Present Day Web-Development Using ReactJS. *International Research Journal of Engineering and Technology (IRJET)*. 7(5), pp. 1154–1157.

- [16] Aldo, D., Richo, & Munir, Z. (2021). Aplikasi Pelayanan Pada Klinik Kemina Dental Care Berbasis React.Js Dan Database NoSQL. *Jurnal Ilmiah Ilmu Komputer*. 16(2), pp. 139–147. DOI: <https://doi.org/10.30872/jim.v16i2.6532>
- [17] Nasution, & Iswari, L. (2021). Penerapan React JS Pada Pengembangan FrontEnd Aplikasi Startup Ubaform. *Jurnal Automata - UII*. 2(2), pp. 193–200.
- [18] Suwarno, & Yulandi, A. P. (2023). Analisis Performa Backend Framework : Studi Komparasi Framework Golang Dan Node . Js. *Jurnal Riset Sistem Informasi Dan Teknik Informatika*. 8(1), pp. 155–168. DOI: <https://doi.org/http://dx.doi.org/10.30645/jurasik.v8i1.551.g529>
- [19] Pranoto, D., Handojo, A., & Satiabudhi, G. (2022). Aplikasi War Game Pada Mobile Device Menggunakan Sensor Gyroscope Dan Accelerometer. *Jurnal Infra*. 10(2), pp. 1–7.
- [20] Lusita, M. D., Hurnianingsih, H., & Rihyanti, E. (2020). Aplikasi Bot Akademik BAAK STMIK Jakarta STI&K Platform Line Messenger Menggunakan Go Languages. *Jurnal Teknologi Sistem Informasi dan Aplikasi*. 3(1), pp. 1–8. DOI: <https://doi.org/10.32493/jtsi.v3i1.4130>
- [21] Rohmanianti, B. W., & Haerudin, H. (2022). Perancangan Dashboard Monitoring Painting Defect Berbasis Website. *OKTAL: Jurnal Ilmu Komputer dan Science*. 1(10), pp. 1629–1637.
22. Santoso, M. F. (2021). TEKNIK SINGLE PAGE APPLICATION (SPA) LAYOUT WEB DENGAN MENGGUNAKAN REACT JS DAN BOOTSTRAP. *JURNAL KHATULISTIWA INFORMATIKA*. 9(2), pp. 107–114.
- [23] Sari, A. S., & Hidayat, R. (2022). Designing Website Vaccine Booking System Using Golang Programming Language and Framework React JS. *Journal of Information System* 6(1), pp. 22–39. DOI: <https://doi.org/10.52362/jisicom.v6i1.760>
- [24] Hasanah, N., & Indriawan, M. N. (2021). Rancangan Aplikasi Batam Travel Menggunakan Metode Software Development Life Cycle (SDLC). *CoMBInES - Conference on Management, Business, Innovation, Education and Social Sciences*. 1(1), pp. 925–938. Retrieved from <https://journal.uib.ac.id/index.php/combin/es/article/view/4524>
- [25] Nur, H. (2019). Penggunaan Metode Waterfall Dalam Rancang Bangun Sistem Informasi Penjualan. *Generation Journal*. 3(1), pp. 1–10. DOI: <https://doi.org/10.29407/gj.v3i1.12642>
- [26] Aceng Abdul Wahid. (2020). Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi. *Jurnal Ilmu-ilmu Informatika dan Manajemen STMIK*. (November), pp. 1–5.
- [27] Sornkliang, W., & Phetkaew, T. (2021). Performance Analysis of Test Path Generation Techniques Based on Complex Activity Diagrams. *Informatika (Slovenia)*. 45(2), pp. 231–242. DOI: <https://doi.org/10.31449/inf.v45i2.3049>
- [28] Nurudin, M., Jayanti, W., Saputro, R. D., Saputra, M. P., & Yulianti, Y. (2019). Pengujian Black Box Pada Aplikasi Penjualan Berbasis Web Menggunakan Teknik Boundary Value Analysis. *Jurnal Informatika Universitas Pamulang*. 4(4), pp. 143–148. DOI: <https://doi.org/10.32493/informatika.v4i4.3841>