

Pengembangan Aplikasi Manajemen Citra MRI Berbasis Framework Laravel

Lutfiadji Mazda Pradenta, Eko Pramunanto, Arta Kusuma Hernanda

Dept. Teknik Komputer Institut Teknologi Sepuluh Nopember Surabaya, Indonesia

Email: lutfiadji13@mhs.te.its.ac.id, ekopram@te.its.ac.id, artakusuma@its.ac.id

Abstrak

Penggunaan gambar medis sebagai dasar pemeriksaan suatu penyakit semakin meningkat dalam bidang medis, sebagian besar ekstensi citra dari alat-alat medis *Magnetic Resonance Imaging* (MRI) adalah *Digital Imaging and Communications in Medicine* (DICOM), sebagai standar dalam dunia medical engineering untuk menyimpan, mencetak dan bertukar informasi dalam dunia pencitraan biomedis. Pada file DICOM terdapat informasi tentang nama pasien, jenis scan, dimensi gambar, serta semua data citra lainnya, sehingga memudahkan komunikasi antar paramedis.

Secara komputasi, file DICOM ini tidak di-support secara otomatis oleh sistem operasi. Sehingga dibutuhkan sebuah sistem untuk membaca file DICOM maka dikembangkan aplikasi manajemen citra MRI berbasis framework Laravel. Melalui aplikasi tersebut akan menampilkan hasil file DICOM untuk melihat gambar citra medis. Pengembangan aplikasi manajemen citra MRI dikembangkan aplikasi berbasis web dan memungkinkan pengguna untuk mengupload citra DICOM, dan akan dilakukan proses capture dan penambahan informasi pasien yang selanjutnya akan ditampung di server MySQL Orthanc untuk proses manajemen citra.

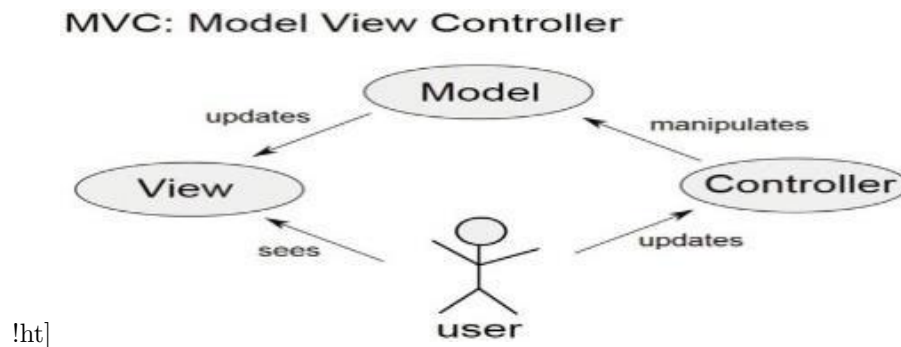
Kata kunci: DICOM, Citra MRI, Aplikasi berbasis framework Laravel

Diterima Redaksi: 1-Nov-2024 Selesai Revisi: 15-Nov-2024 Diterbitkan Online: 15-Des-2024
DOI: DOI: <https://doi.org/10.59378/jcenim.v2i3.58>

I. PENDAHULUAN

Unit radiologi pada sebuah rumah sakit sangatlah penting [7]. Namun meningkatnya jumlah pasien dan keterbatasan dokter ahli radiologi menyebabkan unit ini kesulitan dalam memberikan pelayanan ke pasien sebagai pendiagnosa penyakit [8]. *Magnetic Resonance Imaging* merupakan salah satu radiodiagnostik dengan menggunakan teknologi magnet dan gelombang radio untuk mendapatkan hasil gambar organ, tulang, dan jaringan di dalam tubuh secara rinci dan mendalam. MRI otak adalah pemeriksaan bersifat diagnostik yang paling umum digunakan untuk mendeteksi adanya tumor otak, kanker otak atau gangguan lainnya. Selama pemeriksaan MRI akan memungkinkan molekul-molekul dalam tubuh bergerak untuk membentuk sinyal-sinyal. Sinyal ini akan ditangkap oleh antena dan dikirimkan ke komputer untuk diproses dan ditampilkan di layar monitor menjadi sebuah gambaran yang jelas dari struktur rongga tubuh bagian dalam [10].

Dalam mengidentifikasi penyakit dalam, biasanya dokter atau radiologis menganalisa citra hasil Resonansi Magnetik yang disimpan dalam format *Digital Imaging Communication In Medicine* (DICOM) [3]. Hasil Resonansi Magnetik yang disimpan dalam format *Digital Imaging Communication In Medicine* (DICOM) karena pada file DICOM tunggal berisi sebuah header yang menyimpan informasi tentang nama pasien, jenis scan, dimensi gambar, serta semua data citra lainnya, sehingga memudahkan komunikasi antar paramedis. Dengan melihat rutinitas dokter atau radiologis yang sering berpindah tempat untuk memberikan pelayanan kepada pasien menjadi penyebab kurangnya efektifitas kerja menyebabkan keterlambatan diagnosa dan ketidakakuratan interpretasi dari file citra medis padahal kesempatan untuk bertukar pendapat mengenai gambar medis sangat dibutuhkan dan ketersediaan berkas gambar medis sangatlah diperlukan [7]. Dengan adanya permasalahan ini maka mendasari perlunya sebuah pengembangan aplikasi manajemen Citra MRI berbasis framework Laravel yang memiliki fungsi utama untuk dapat mengupload citra DICOM sehingga membantu ahli radiologi dalam melakukan analisa atau pembacaan citra, proses capture untuk melihat informasi file DICOM dan menambahkan informasi pasien untuk mempermudah manajemen file DICOM tersebut [2].



Gambar 1: Model View Controller

II. KAJIAN PUSTAKA

A. Magnetic Resonance Imaging

Magnetic Resonance Imaging (MRI) adalah suatu alat kedokteran di bidang pemeriksaan diagnostik radiologi, yang menghasilkan rekaman gambar potongan penampang tubuh atau organ manusia dengan menggunakan medan magnet berkekuatan antara 0.064–1.5 tesla dan resonansi getaran terhadap inti atom hydrogen [10]. MRI dapat memberikan gambaran struktur tubuh yang tidak bisa didapatkan pada tes lain, seperti Rontgen, USG, atau CT scan. Salah satu kelebihan pencitraan MRI adalah, menurut pengetahuan pengobatan masa kini, tidak berbahaya kepada orang yang sakit. Secara ringkas, proses terbentuknya citra *Magnetic Resonance Imaging* (MRI) dapat digambarkan bila tubuh pasien diposisikan dalam medan magnet yang kuat.

B. Picture Archiving and Communication System

PACS (*Picture Archiving and Communication System*) merupakan sistem yang memungkinkan penyimpanan, pengambilan, dan penampilan sebuah citra medis, lebih khususnya untuk rumah sakit berskala besar dan klinik [4]. Selain itu PACS merupakan sistem dengan konsep filmless atau tidak berbentuk fisik dengan metode komputerisasi komunikasi dan dapat menyimpan citra medis yang berasal dari modalitas. PACS harus menjamin bahwa citra yang disimpan dalam jangka waktu panjang dapat memenuhi aturan atau kewajiban hukum yang berlaku di negara setempat. Sehingga dapat digunakan untuk interpretasi pada saat dibutuhkan. Kebutuhan ini dipenuhi oleh PACS core, yaitu: (1) sistem manajemen database, (2) media penyimpanan, (3) software control, serta (4) antarmuka RIS (*radiology information system*).

Sistem manajemen database merupakan jantung dari PACS, hubungan antara citra dan lokasi penyimpanannya disimpan dan dikelola dalam database bersama dengan seluruh data yang relevan yang dibutuhkan untuk mengambil citra, sistem manajemen database yang digunakan harus mampu untuk mengambil citra medis milik pasien yang sekarang ataupun yang sebelumnya ketika RIS atau sistem lainnya melakukan permintaan. Permintaan yang direspon oleh database sistem ini didefinisikan sebagai DICOM (*Digital Imaging and Communication in Medicine*) [1].

C. Framework Laravel

Seperangkat struktur dan pedoman konseptual, yang digunakan untuk membangun sesuatu yang bermanfaat. Dalam istilah pembuatan web, framework merupakan software untuk memudahkan para programmer dalam membuat sebuah aplikasi web. Dalam framework terdapat *Software framework* adalah struktur yang dapat digunakan dalam membangun sesuatu. *Software framework* memungkinkan untuk menggunakan jenis komponen yang berbeda, berkomunikasi dengan API eksternal dan menentukan struktur aplikasi. Di dalam framework terdapat plugin dan konsep dengan sistem yang terstruktur sehingga kita bebas dalam hal pemrograman [2].

Laravel Framework ini mengikuti struktur MVC (*Model View Controller*). MVC adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi. Dengan menggunakan struktur MVC maka membuat laravel mudah untuk dipelajari dan mempercepat proses pembuatan prototipe aplikasi web. MVC memisahkan aplikasi berdasarkan komponen-komponen aplikasi, seperti: manipulasi data, controller, dan user interface.

D. MySQL

MySQL adalah merupakan salah satu perangkat lunak system manajemen basis data SQL atau *Database Management System* (DBMS). Sedangkan *Relation Database Management System* (RDBMS) merupakan aplikasi yang berfokus pada relasi antara data.

E. Standar DICOM

DICOM (*Digital Imaging and Communication in Medicine*) merupakan standar yang digunakan dalam PACS untuk komunikasi gambar dan data medis [3]. Standar ini memiliki dua komponen, yaitu protokol komunikasi dan format data gambar. Standar DICOM memungkinkan komunikasi digital antara peralatan medis dari tiap vendor yang berbeda. Saat ini semua peralatan sistem pencitraan medis modern (*Imaging Modalities*) seperti Sinar-X, Ultrasound, CT (*Computed Tomography*), dan MRI (*Magnetic Resonance Imaging*) mendukung DICOM dan menggunakannya secara luas. Dengan standar ini, maka praktisi dan vendor dapat dengan mudah bekerja sama untuk membangun sistem medis tanpa adanya kendala .

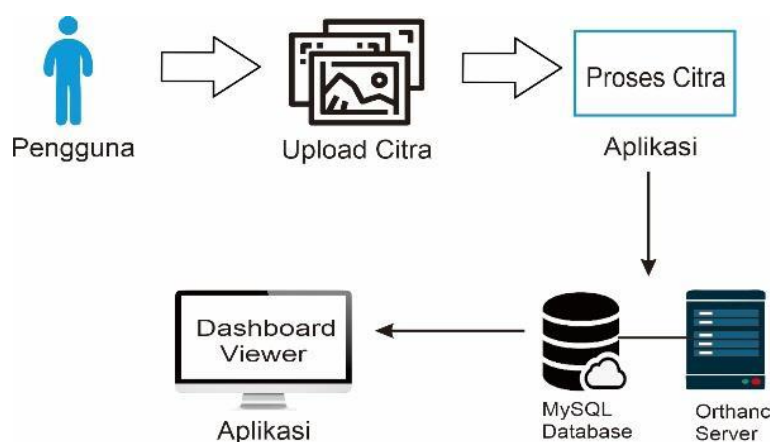
F. Orthanc Server

Orthanc secara inheren adalah VNA (*Vendor Neutral Archive*) yang dapat menerima, menyimpan, mengindeks, dan mengirim gambar medis sesuai dengan standar DICOM [6]. Orthanc bertujuan menyediakan server DICOM mandiri yang sederhana namun kuat. Orthanc memungkinkan penggunaanya fokus pada konten file DICOM, meminimalkan kompleksitas format DICOM dan protokol DICOM. Orthanc dapat menjadikan komputer mana pun yang menjalankan Windows, Linux atau OS X menjadi sistem mini-PACS. Arsitekturnya sederhana dan tidak diperlukan administrasi basis data yang rumit maupun instalasi dependensi pihak ketiga.

III. DESAIN SISTEM

A. Desain Sistem

Proses dimulai dengan pengguna melakukan input citra melalui dashboard aplikasi. Citra yang diinput akan diproses satunya menggunakan library connerstone untuk membaca pixel dari sebuah citra medis, setelah citra dikenali sebagai objek citra medis maka aplikasi akan menampilkan sebuah header atau beberapa metadata yang ada pada citra dicom beserta objek citra yang ditampilkan. Beberapa informasi dapat dirubah atau ditambahkan sebelum citra disimpan ke dalam database.



Gambar 2: Gambaran Umum Kerja Sistem

1. Input Citra

Input atau upload data gambar Citra MRI yang digunakan untuk diagnosa pada aplikasi dan agar dapat diproses dalam program yang telah dibuat maka file citra berbentuk DICOM (*Digital Imaging and Communications in Medicine*).

2. Pemrosesan Data Citra

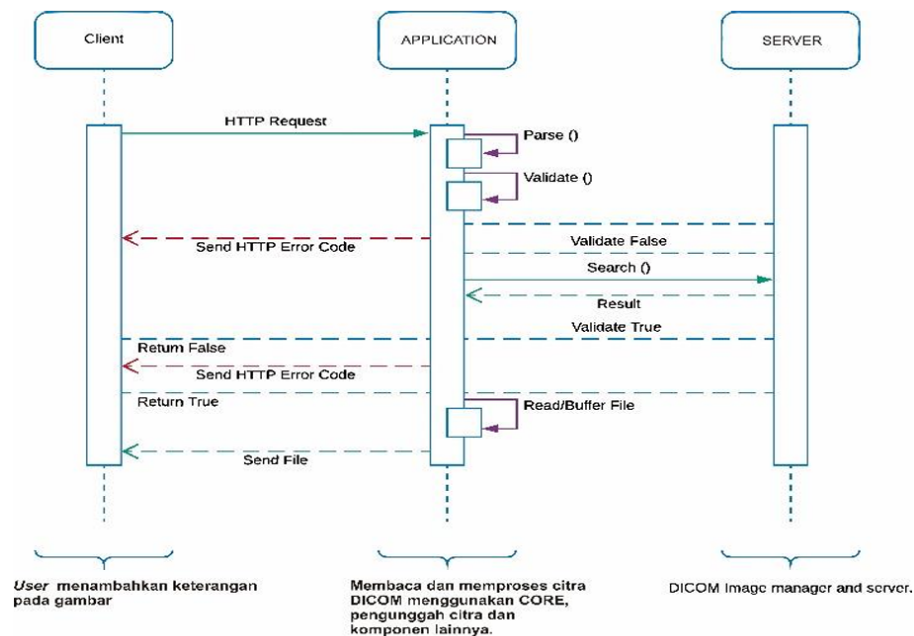
Data citra yang di capture dan dianalisa akan menampilkan informasi file DICOM berisi sebuah header yang menyimpan informasi tentang informasi file dan beberapa metadata citra dicom.

3. Menampilkan Citra Medis

Citra medis akan ditampilkan atau divisualisasikan secara langsung setelah citra DICOM berhasil diproses.

B. Workflow Arsitektur Aplikasi

Sistem yang dikerjakan digambarkan ke dalam diagram yang ditunjukan pada Gambar 3, dengan implementasi yang terdiri dari (i) client service, (ii) application service, dan (iii) orthanc. Bagian dari standar DICOM harus diterima sebagai layanan dari server citra dan menjadi respon dari layanan aplikasi untuk menentukan parameter. Desain workflow arsitektur sistem ditunjukan pada gambar.



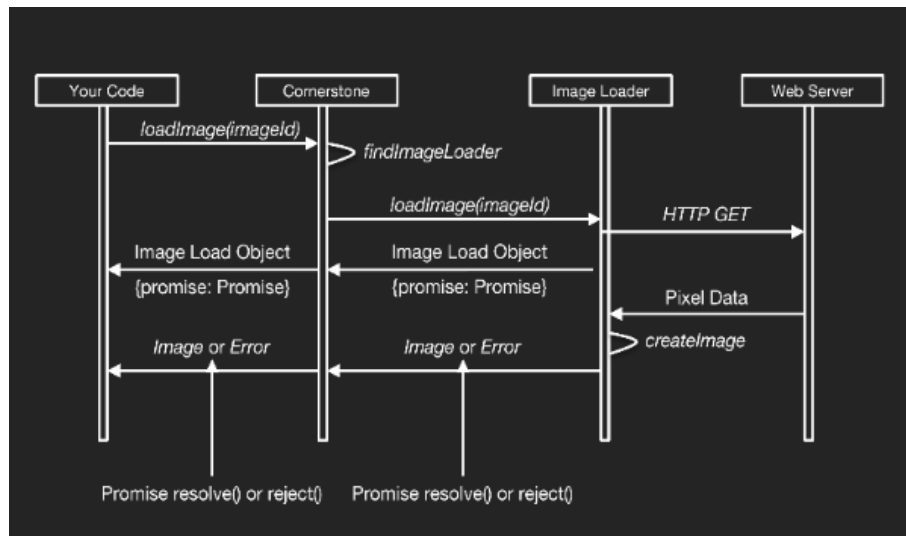
Gambar 3: Urutan permintaan UML ke layanan

Dalam penelitian ini client akan mengunggah citra DICOM melalui sistem client menuju layanan aplikasi dengan menggunakan library yang mendefinisikan parameter yang diperlukan oleh server untuk mengenali objek DICOM dengan jelas. Semua parameter tersebut mengacu pada modifikasi citra atau format data dimana server melakukan validasi parameter saat melakukan pencarian. Gambar 3 menunjukkan urutan UML yang sesuai dengan permintaan HTTP pada layanan aplikasi.

Dengan menggunakan komponen library connerstone yang HTML5 dan JavaScript sehingga dapat mendukung manipulasi, dan akses ke informasi objek Dicom.

C. Workflow Menampilkan Citra Medis

Dalam menjalankan proses untuk menampilkan data citra medis aplikasi menggunakan library connerstone yang berisikan sebuah fungsi *Image Loader* yang bertanggung jawab untuk mengambil id dan meta data gambar agar gambar citra dapat ditampilkan. Objek Citra akan memuat sebuah objek promise yang berfungsi untuk melakukan komputasi asynchronous karena memuat gambar biasanya membutuhkan panggilan ke server. Connerstone membutuhkan fungsi *Image Loader* untuk mengembalikan sebuah objek yang berisi promise yang akan digunakan untuk menerima objek citra secara asynchronous atau mengembalikan nilai error jika terjadi kesalahan. Proses untuk menampilkan citra medis menggunakan *Image Loader* ditunjukan pada Gambar 4.



Gambar 4: Workflow Image Loader

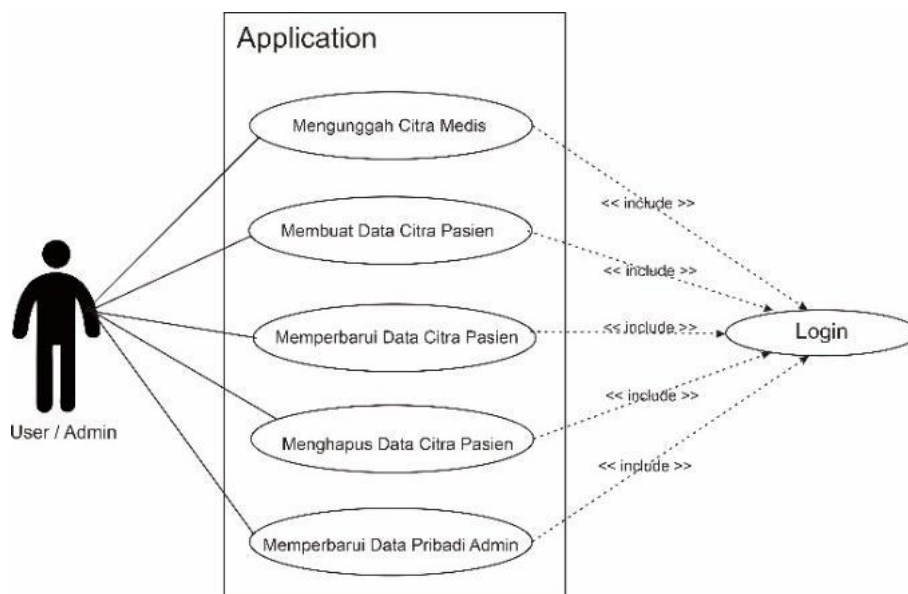
Untuk mengetahui tahapan alur kerja proses menampilkan citra akan dibaca oleh cornerstone untuk mendapatkan hirarki id dimana lokasi citra dimuat dan *Image Loader* mendaftarkan tiap gambar yang dibaca dengan tujuan untuk memuat skema URL ImageId dan mengembalikan sebuah Objek berisi promise dan kemudian akan ditampilkan dengan *displayImage* API.

D. Auxiliary Files

Sistem memiliki sumber file tambahan (auxiliary files) yang berinteraksi langsung dengan beberapa modul penting untuk sistem pada aplikasi. Beberapa file tambahan ini terkait dengan gambar 3 dan gambar 4. Terdiri dari *Image Loader*, Promises, Image Rendering, Viewport, dan DICOM Parser yang digunakan untuk mengekstrak data piksel dari file DICOM dan menampilkan citra pada user interface.

E. Use Case Diagram

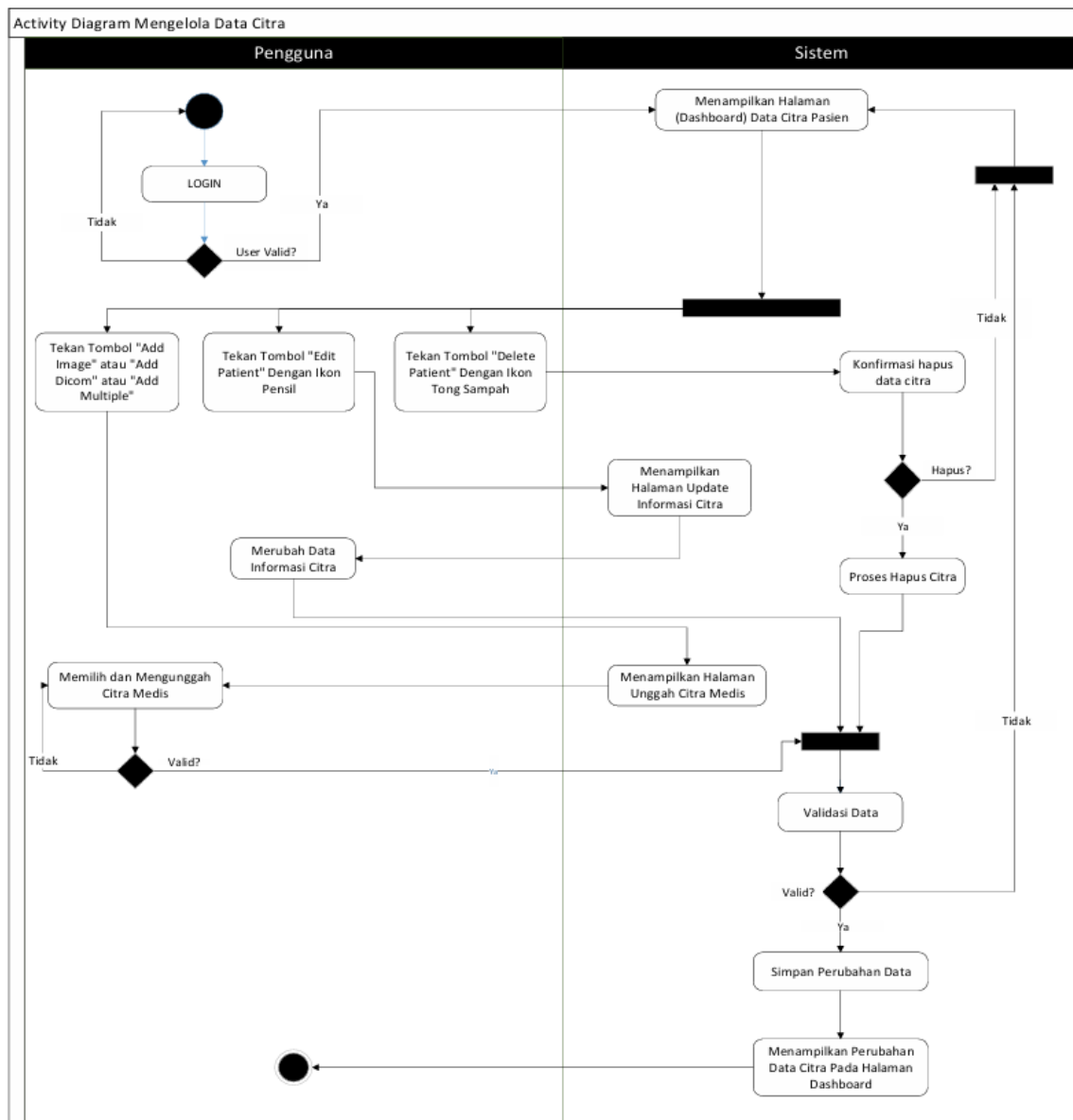
Berdasarkan workflow sistem aplikasi yang dikerjakan, dibuat use case diagram yang menggambarkan interaksi aktor dan sistem yang dapat dilihat pada Gambar 5.



Gambar 5: Use case diagram

F. Activity Diagram

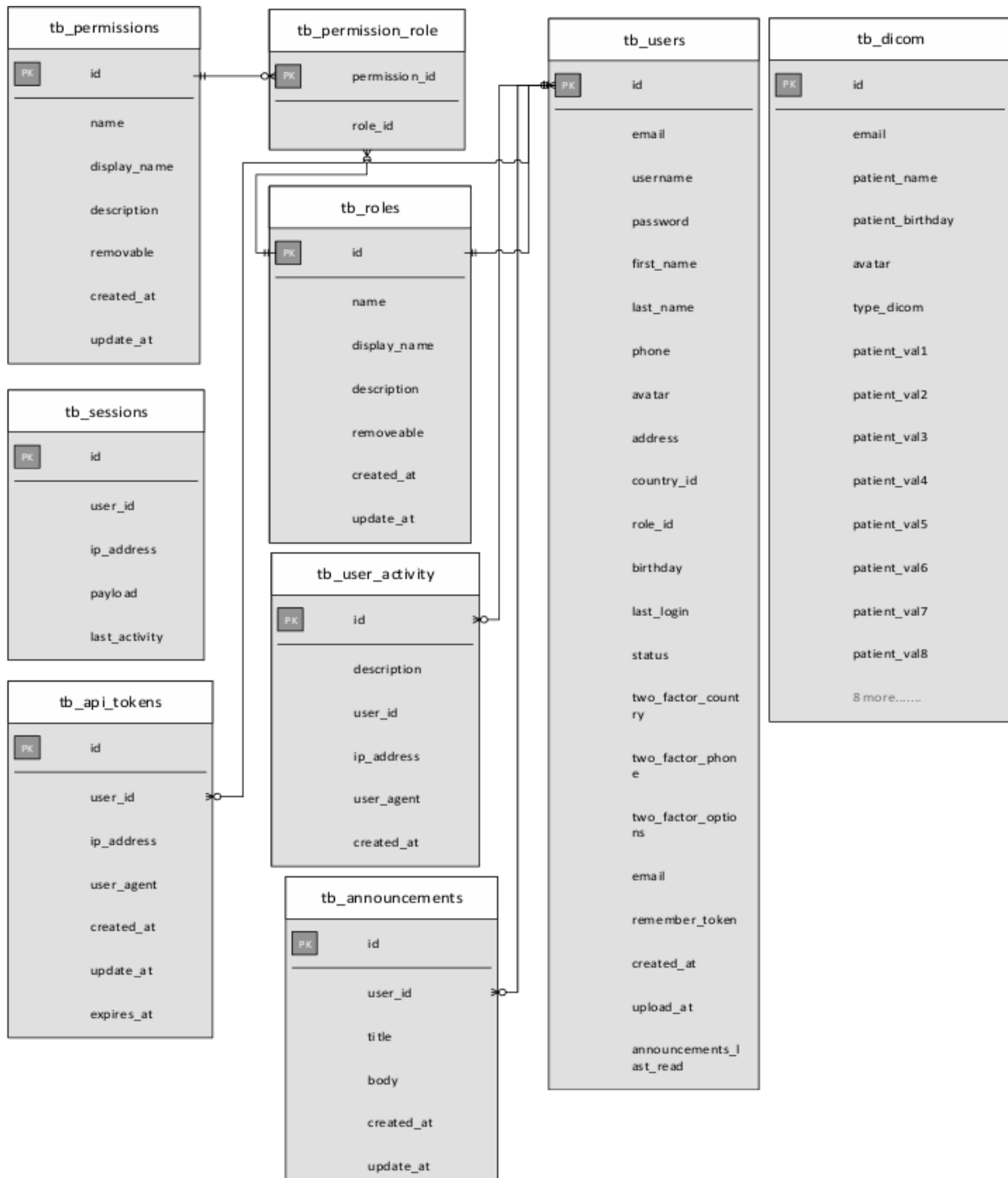
Activity diagram menggambarkan tentang aktifitas yang akan terjadi dalam sistem aplikasi yang akan dibuat. Dari awal hingga akhir, diagram ini menunjukkan langkah-langkah dalam proses kerja sistem yang dibuat. Pada Gambar 6 digambarkan bahwa sistem diawali dengan proses login pada sistem layanan aplikasi. Sistem akan melakukan validasi pada informasi login pengguna untuk menentukan apakah pengguna sudah terdaftar di dalam aplikasi. Apabila pengguna telah terdaftar, maka sistem akan menampilkan dashboard sesuai hak akses pengguna.



Gambar 6: Activity diagram mengelola data citra

G. Entity Relationship Diagram

Entity relationship diagram adalah suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi dapat dilihat pada Gambar 7.



Gambar 7: Entity Relationship Diagram

IV. PENGUJIAN DAN HASIL

Pada bagian ini akan dilakukan pengujian untuk mengetahui apakah fungsionalitas dari sistem yang dibuat dan memastikan pengiriman data berjalan dengan baik sehingga dapat menarik kesimpulan dari pengujian yang telah dibuat. Uji coba dilakukan dengan menggunakan lokal server yang sudah terinstal Orthanc dengan spesifikasi yang ditampilkan pada Tabel 1.

A. Pengujian Kesesuaian Sistem Aplikasi

Pada pengujian kesesuaian sistem aplikasi, hal yang diperhatikan adalah untuk mengetahui apakah sistem aplikasi berjalan dengan desain sistem yang diinginkan. Pengujian dilakukan sebanyak lebih dari

Tabel 1: Spesifikasi Server

| Komponen | Spesifikasi |
|----------------|--|
| Processor | Intel Core i7-9750H (2.6 GHz, 12M Cache) |
| RAM | 8 GB DDR4 |
| Storage | 512GB INTELSSD PEKNW512G8 |
| OS | Windows 64-bit |
| Server | 127.0.0.1 via TCP/IP |
| Server Version | 10.4.8-MariaDB - mariadb.org |

100 kali uji scenario untuk mengecek seluruh proses dalam sistem aplikasi dan fitur yang ada apakah berjalan dengan baik atau tidak.

Tabel 2: Pengujian Tampilan Citra Medis

| Kelas Uji | Skenario Uji | Skenario Dapat Berjalan |
|------------------------|-------------------------|-------------------------|
| Penampilan Citra Medis | Validasi modality DICOM | Ya |
| | Memilih Citra Medis | Ya |
| | Menampilkan Citra Medis | Ya |

B. Pengujian Waktu Unggah Citra Medis

Pada pengujian ini dilakukan pengiriman file dari sebuah komputer atau perangkat ke sebuah sistem server sehingga data yang telah diunggah dapat dilihat dan diambil oleh pengguna lain. Pengujian waktu unggah citra medis dilakukan dengan melakukan unggahan terhadap file citra medis dalam satuan atau jumlah tertentu.

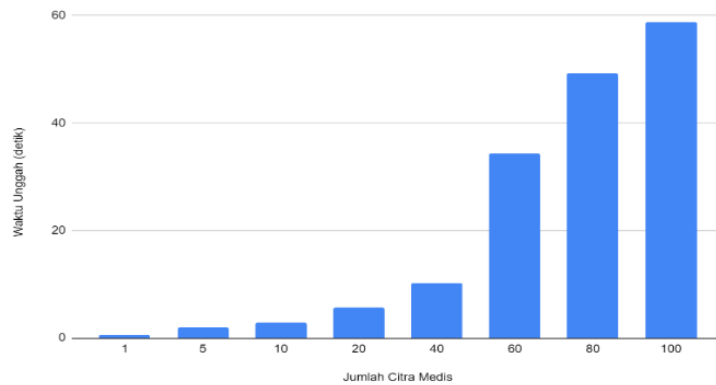
Tabel 3: Pengujian Waktu Unggah Citra Medis

| No | Jumlah Citra Medis | Waktu Unggah (Detik) |
|----|--------------------|----------------------|
| 1 | 1 | 0.46 |
| 2 | 5 | 1.91 |
| 3 | 10 | 2.8 |
| 4 | 20 | 5.7 |
| 5 | 40 | 10.2 |
| 6 | 60 | 34.3 |
| 7 | 80 | 49.2 |
| 8 | 100 | 58.7 |

Dari Tabel 3 menunjukkan bahwa dari 8 kali percobaan unggah citra didapatkan hasil kecepatan waktu unggahan citra medis akan lebih efektif dilakukan dengan jumlah 40 gambar karena pada jumlah upload citra di atas 40, waktu unggah citra menjadi lebih lambat bila dibandingkan unggah 1 gambar citra yang dilakukan lebih dari 40 kali.

C. Pengujian Waktu Render Citra Medis

Pada pengujian waktu render citra medis dilakukan dengan menampilkan sebuah series atau data citra medis dalam serangkaian percobaan. Hasil pengujian waktu render citra medis dari sistem aplikasi yang telah dirancang adalah sebagai berikut seperti yang ditampilkan pada Tabel 4.



Gambar 8: Grafik pengujian waktu unggah citra medis

Tabel 4: Pengujian Waktu Render Citra Medis

| Percobaan ke | Status | Interval Waktu (Detik) |
|--------------------------|-------------|------------------------|
| 1 | Ditampilkan | 0.27 |
| 2 | Ditampilkan | 0.71 |
| 3 | Ditampilkan | 0.38 |
| 4 | Ditampilkan | 0.53 |
| 5 | Ditampilkan | 0.31 |
| 6 | Ditampilkan | 0.64 |
| 7 | Ditampilkan | 0.33 |
| 8 | Ditampilkan | 0.32 |
| 9 | Ditampilkan | 0.28 |
| 10 | Ditampilkan | 0.37 |
| Rata-rata interval waktu | | 0.414 |

Berdasarkan hasil pengujian dari Tabel 4, ditunjukan bahwa waktu tercepat yang dapat diperoleh ketika citra dapat divisualisasikan oleh sistem aplikasi adalah 0.27 detik, dengan rata-rata interval 0.414 detik.

D. Pengujian Fungsionalitas Antarmuka Sistem

Pada pengujian ini dilakukan uji coba fungsionalitas antarmuka untuk mengetahui bahwa dari setiap aktivitas yang dapat dilakukan oleh pengguna dapat berjalan dengan baik. Berdasarkan hasil ujicoba yang dilakukan didapatkan bahwa seluruh skenario dan kelas uji berhasil dilakukan. Sehingga antarmuka aplikasi dapat berjalan secara fungsional.

V. KESIMPULAN DAN SARAN

Dari pengujian yang telah dilakukan, penulis berhasil mengembangkan aplikasi untuk manajemen citra MRI menggunakan framework Laravel untuk informasi file DICOM, namun untuk unggah citra belum dapat optimal jika jumlah citra lebih dari 40 karena dari hasil yang diperoleh pada pengujian membutuhkan waktu sekitar 3.36 kali bila melakukan upload dengan jumlah 60 data citra atau lebih.

Saran pengembangan selanjutnya adalah perlu peningkatan performa agar pengunggahan citra medis secara multiple/lebih banyak lagi, minimal 100 gambar citra sekali upload lebih cepat dan mampu memvisualisasikan data series agar lebih jelas dan akurat.

Daftar Pustaka

- [1] T. Ranga, "Implementasi pacs orthanc berbasis postgresql dan radiology information system," 2019.
- [2] Y. Neha, R. S. Dharmveer, and D. K. Shri, "Laravel: A php framework for e-commerce website," 2019.
- [3] National Electrical Manufacturers Association, *Digital imaging and communications in medicine (DICOM)*, Std., 1998.
- [4] H. K. Huang and Faimbe, *PACS and imaging informatics*. John Wiley, 2012.
- [5] N. Jacinto and G. Daniel, "Medical imaging multimodality breast cancer diagnosis user interface," 2017.
- [6] S. Jodogne *et al.*, "Orthanc-a lightweight, restful dicom server for healthcare and medical research," in *IEEE 10th International Symposium on Biomedical Imaging*, 2013.
- [7] R. J. Sachs and C. Long, "Process for managing and optimizing radiology work flow in the electronic health record environment," *Journal of Digital Imaging*, vol. 28, no. 2, pp. 123–133, 2015.
- [8] N. Fadhilah and F. Ayubi, "Lean six sigma approach for wait time analysis in radiology services at krakatau medika hospital," *Devotion Journal of Research and Community Service*, vol. 7, no. 1, pp. 45–60, 2023.
- [9] M. R. Pala, S. Khadem, and A. Tosun, "Patient volume forecasting in the radiology department using deep learning and statistical-based models," *Preprint*, 2022.
- [10] S. M. Wani, A. M. Lodhi, and P. K. Patil, "Improvement in radiographic quality workflow due to use of digital imaging and picture archiving and communication system," *International Journal of Advances in Medicine*, vol. 6, no. 9, pp. 2086–2091, 2019.
- [11] D. Deng, "The prevention and management of the coronavirus disease 2019 (covid-19) outbreak in radiology departments in epidemic areas," *Japanese Journal of Radiology*, vol. 38, no. 5, pp. 431–437, 2020.