Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR – EC 184801

PENGEMBANGAN APLIKASI MANAJEMEN CITRA MRI BERBASIS *FRAMEWORK LARAVEL*

Lutfiadji Mazda Pradenta
NRP 07211340000020

Dosen Pembimbing
Dr. I. Ketut Eddy Purnama, S.T., M.T.
Dr. Supeno Mardi Susiki Nugroho, S.T., M.T.

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019

TUGAS AKHIR – EC 184801

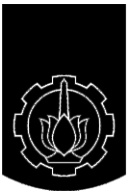
PENGEMBANGAN APLIKASI MANAJEMEN CITRA MRI BERBASIS *FRAMEWORK LARAVEL*

Lutfiadji Mazda Pradenta
NRP 07211340000020

Dosen Pembimbing
Dr. I. Ketut Eddy Purnama, S.T., M.T.
Dr. Supeno Mardi Susiki Nugroho, S.T., M.T.

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019

Halaman ini sengaja dikosongkan



FINAL PROJECT – EC 184801

DEVELOPMENT OF MRI IMAGE MANAGEMENT APPLICATION BASED ON LARAVEL FRAMEWORK

Lutfiadji Mazda Pradenta
NRP 07211340000020

Advisor
Dr. I. Ketut Eddy Purnama, S.T., M.T.
Dr. Supeno Mardi Susiki Nugroho, S.T., M.T.

DEPARTMENT OF COMPUTER ENGINEERING
Faculty of Electrical Technology
Sepuluh Nopember Institute of Technology
Surabaya 2019

Halaman ini sengaja dikosongkan

PERNYATAAN KEASLIAN
TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul **"Pengembangan Aplikasi Manajemen Citra MRI Berbasis *Framework Laravel*"** adalah benar-benar hasil karya ntelektual sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya orang lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun durujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Surabaya, Juli 2020

Lutfiadji Mazda Pradenta
NRP. 07211340000020

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN

Pengembangan Aplikasi Manajemen Citra MRI
Berbasis *Framework Laravel*

Tugas Akhir ini disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana Teknik di Institut Teknologi Sepuluh Nopember Surabaya

Oleh: Ludfiadji Mazda Pradenta (NRP: 07211340000020)

Tanggal Ujian: 07 Juli 2020 Periode Wisuda: September 2020

Disetujui oleh:

Dr. I Ketut Eddy Purnama, S.T., M.T. (Pembimbing I)
NIP: 196907301995121001

Dr. Supeno Mardi Susiki N, S.T., M.T. (Pembimbing II)
NIP: 197003131995121001

Susi Juniastuti, S.T., M. Eng. (Penguji I)
NIP: 196506181999032001

Mochamad Hariadi, S.T., M.Sc., Ph.D. (Penguji II)
NIP: 196912091997031002

Dr. Diah Puspito Wulandari, S.T., M.Sc. (Penguji III)
NIP: 198012192005012001

Mengetahui
Kepala Departemen Teknik Komputer

Dr. Supeno Mardi Susiki Nugroho, S.T., M.T.
NIP. 197003131995121001

Halaman ini sengaja dikosongkan

ABSTRAK

Nama Mahasiswa	:	Lutfiadji Mazda Pradenta
Judul Tugas Akhir	:	Pengembangan Aplikasi Manajemen Citra MRI Berbasis *Framework Laravel*
Pembimbing	:	1. Dr. I Ketut Eddy Purnama, S.T., M.T.
.		2. Dr. Supeno Mardi Susiki N., S.T., M.T.

Penggunaan gambar medis sebagai dasar pemeriksaan suatu penyakit semakin meningkat dan dalam bidang medis, sebagian besar ekstensi citra dari alat-alat medis *Magnetic resonance imaging* (*MRI)* adalah Digital Imaging and Communications in Medicine (DICOM). Dicom merupakan suatu 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, DICOM memiliki extension file .dcm. File ini tidak di-support secara otomatis oleh sistem operasi yang ada seperti Windows atau Linux. Untuk membaca file DICOM maka dikembangkan aplikasi manajemen citra MRI berbasis framework Laravel. Melalui aplikasi tersebut akan menampilkan hasil file header dari file DICOM yang diupload untuk melihat gambar citra medis. Pengembangan aplikasi manajemen citra MRI tersebut akan lebih membantu ketika dikembangkan aplikasi berbasis web. Perancangan aplikasi ini memungkinkan pengguna untuk mengupload citra DICOM, dan akan dilakukan proses capture dan penambahan informasi pasien yang selanjutnya akan ditampung di server mysql untuk proses manajemen citra.

Kata Kunci: DICOM, Citra MRI, Aplikasi berbasis *framework laravel.*

Halaman ini sengaja dikosongkan

ABSTRACT

Name	:	Lutfiadji Mazda Pradenta
Title	:	Development of MRI Image Management Application Based On Laravel Framework
Advisor	:	1. Dr. I Ketut Eddy Purnama, S.T., M.T. 2. Dr. Supeno Mardi Susiki N., S.T., M.T..

.

The use of medical images as a basis for examining a disease is increasing and in the medical field, most of the image extensions from magnetic resonance imaging (MRI) medical devices are Digital Imaging and Communications in Medicine (DICOM). Dicom is a standard in the world of medical engineering for storing, printing and exchanging information in the world of biomedical imaging. In the DICOM file there is information about the patient's name, scan type, image dimensions, and all other image data, making it easier for communication between paramedics. In computing, DICOM has a .dcm file extension. This file is not supported automatically by existing operating systems such as Windows or Linux. To read the DICOM file an MRI image management application based on the Laravel framework was developed. Through this application we will get an introduction to the results of MRI images in more detail to view medical images. The development of MRI image management applications will be more helpful when developing web-based applications. The design of this application allows users to upload DICOM images, and capture and adding patient information will be carried out which will then be accommodated on the mysql server for image management processes.

Keywords: DICOM, MRI image, an Application based on Laravel Framework.

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji dan syukur kehadirat Allah SWT atas segala limpahan berkah dan hidayah-Nya, penulis dapat menyelesaikan penelitian ini dengan judul "**Pengembangan Aplikasi Manajemen Citra MRI Berbasis *Framework Laravel***" untuk membuat rancangan online berupa aplikasi untuk manajemen citra MRI dengan kerangka kerja laravel.

Penelitian ini disusun dalam rangka pemenuhan bidang riset di Departemen Teknik Komputer, FTE-ITS serta digunakan sebagai persyaratan menyelesaikan pendidikan S1. Penelitian ini dapat ter- selesaikan tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Keluarga, Ibu, Bapak dan Saudara tercinta yang telah memberikan dorongan spiritual dan material dalam penyelesaian buku penelitian ini.
2. Bapak Dr. I Ketut Eddy Purnama, S.T., M.T. selaku Kepala Departemen Teknik Komputer, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember.
3. Secara khusus penulis mengucapkan terima kasih yang sebesar- besarnya kepada Bapak Dr. I Ketut Eddy Purnama, S.T., M.T dan Dr. Supeno Mardi Susiki N., S.T., M.T. atas bimbingan selama pengerjaan tugas akhir ini.
4. Bapak-ibu dosen pengajar Departemen Teknik Komputer atas pengajaran, bimbingan, serta perhatian yang diberikan kepa- da penulis selama ini.
5. Dan teman-teman yang selalu memberikan dukungan dan yang tidak dapat saya sebutkan satu-persatu.

Kesempurnaan hanya milik Allah SWT, untuk itu penulis memohon segenap kritik dan saran yang membangun. Semoga penelitian ini dapat memberikan manfaat bagi kita semua. Amin.

Surabaya, Juni 2020

Lutfiadji Mazda Pradenta

Halaman ini sengaja dikosongkan

DAFTAR ISI

Abstrak i

Abstract iii

KATA PENGANTAR v

DAFTAR ISI vii

DAFTAR GAMBAR xi

DAFTAR TABEL xv

SINGKATAN xvii

1 PENDAHULUAN 1
- 1.1 Latar belakang 1
- 1.2 Permasalahan 2
- 1.3 Tujuan 3
- 1.4 Batasan Masalah 3
- 1.5 Sistematika Penulisan 3
- 1.6 Relevansi 4

2 TINJAUAN PUSTAKA 5
- 2.1 Magnetic Resonance Imaging 5
- 2.2 Picture Archieving and Communication System 6
- 2.3 Standar DICOM 8
- 2.4 Framework 9
- 2.5 Laravel 10
- 2.6 Hypertext Preprocessor 13
- 2.7 Database 14
- 2.8 MySQL 15
  - 2.8.1 Client Service Model 16
  - 2.8.2 Kerja Sistem MySQL 16
- 2.9 Orthanc Server 19

3 DESAIN DAN IMPLEMENTASI SISTEM 21
3.1 Desain Sistem 21
3.1.1 Workflow Arsitek Aplikasi 25
3.1.2 Workflow Menampilkan Citra Medis 27
3.1.3 Auxiliary Files 28
3.1.4 Use Case Diagram 29
3.1.5 Activity Diagram 33
3.2 Entity Relationship Diagram 35
3.3 Implementasi Orthanc Sebagai PACS Server 37
3.3.1 Konfigurasi Jaringan 38
3.3.2 Implementasi MySQL Pada Orthanc 38
3.3.3 Implementasi Orthanc Pada Laravel 40
3.4 Rancangan User Interface 40
3.4.1 Desain Halaman Beranda Aplikasi 40
3.4.2 Desain Halaman Login 41
3.4.3 Desain Halaman Daftar Citra Pasien 42
3.4.4 Desain Halaman Penampil Citra pasien 43
3.4.5 Desain Halaman Profil Admin 43

4 PENGUJIAN DAN ANALISA 44
4.1 Pengujian Kesesuain Sistem Aplikasi 45
4.2 Pengujian Perform Sistem 45
4.2.1 Pengujian waktu unggah citra medis 46
4.2.2 Pengujian waktu render citra medis 47
4.3 Pengujian Fungsionalitas Antarmuka Sistem 48
4.3.1 Pengujian Halaman Login 49
4.3.2 Pengujian Halaman Citra Pasien 50
4.3.3 Pengujian Halaman Unggah Citra Pasien 51
4.3.4 Pengujian Halaman Penampil Citra Pasien 53

5 PENUTUP 55
5.1 Kesimpulan 55
5.2 Saran 55

DAFTAR PUSTAKA 57

LAMPIRAN 59

Biografi Penulis 61

DAFTAR GAMBAR

2.1 Bagian MRI 6
2.2 Ilustrasi pemanfaatan PACS 7
2.3 *Model View Controller* 11
2.4 Cara kerja MySQL 17

3.1 Gambaran umum kerja sistem 22
3.2 CT Mon brain 23
3.3 Informasi Metadata Citra 23
3.4 Metodologi Proses Citra 24
3.5 Urutan permintaan UML ke layanan 26
3.6 *Workflow Image Loader* 27
3.7 *Use case diagram* aplikasi 29
3.8 *Activity diagram* menglola data citra pasien 34
3.9 *Entity relationship diagram* aplikasi 35
3.10 Tampilan antarmuka Orthanc 37
3.11 Konfigurasi File pada Orthanc 38
3.12 *Database* Orhanc pada MySQL 39
3.13 Halaman beranda aplikasi 41
3.14 Halaman login aplikasi 41
3.15 Halaman daftar citra pasien 42
3.16 Halaman penampil citra pasien 43
3.17 Halaman profil admin 43

4.1 Grafik pengujian waktu unggah citra medis 47
4.2 Pengujian halaman login aplikasi 49
4.3 Notifikasi informasi login yang tidak sesuai 50
4.4 Pengujian halaman citra pasien 50
4.5 Pengujian pencarian data citra pasien 51
4.6 Pengujian halaman unggah data citra *single* baru 52
4.7 Pengujian unggah data citra *multiple* baru 52
4.8 Hasil Pengujian unggah data citra baru 53
4.9 Pengujian halaman penampil citra pasien 54

1 Citra medis yang diterima pada PACS Server 59
2 File citra yang berhasil disimpan melalui aplikasi 59
3 Waktu render citra medis 60

Halaman ini sengaja dikosongkan

DAFTAR TABEL

3.1 Skenario *use case* mengunggah citra medis 30
3.2 Skenario *use case* membuar data citra pasien 31
3.3 Skenario *use case* memperbarui data citra pasien 31
3.4 Skenario *use case* menghapus data citra pasien 32
3.5 Skenario *use case* memperbarui data pribadi admin 33

4.1 Spesifikasi *Server* 44
4.2 Pengujian tampilan citra medis 45
4.3 Spesifikasi Komputer 46
4.4 Pengujian waktu unggah citra medis 46
4.5 Pengujian waktu render citra medis 48

Halaman ini sengaja dikosongkan

SINGKATAN

MRI	: Magnetic Resonance Imaging
MVC	: Model View Controller
ML	: Machine Learning
MLP	: Multi Layer Perception
CNN	: Central Neural Network
DICOM	: Digital Imaging and Communication in Medicine
SQL	: Struktured Query Language
PHP	: Personal HomePage
RDBMS	: Relational Database Management
PACS	: Picture Archieving and Communication System
UML	: Unified Modeling Language
ERD	: Entitas Relationship Diagram
API	: Application Programming Interface
ANN	: Artificial Neural Networks
CT Scan	: Computed Tomography Scan

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

Penelitian ini di latar belakangi oleh berbagai kondisi yang menjadi acuan. Selain itu juga terdapat beberapa permasalahan yang akan dijawab sebagai luaran dari penelitian.

1.1 Latar Belakang

Unit Radiologi merupakan unit penunjang medis yang mempunyai peran penting dalam pelayanan pasien baik sebagai pendiagnosa suatu penyakit maupun sebagai acuan pemberian arah pengobatan bagi para klinisi dalam sebuah rumah sakit. Kemajuan teknologi di bidang kesehatan yang ada pada saat ini memberi kemudahan bagi para praktisi kesehatan untuk mendiagnosa penyakit serta menentukan jenis pengobatan bagi pasien. Salah satu bentuk kemajuan tersebut adalah penggunaan alat MRI (Magnetic Resonance Imaging) untuk melakukan pencitraan diagnosa penyakit pasien.

MRI (*Magnetic Resonance Imaging*) merupakan suatu alat diagnostik mutakhir untuk memeriksa dan mendeteksi tubuh dengan menggunakan medan magnet yang besar dan gelombang frekuensi radio, tanpa operasi, penggunaan sinar X, atau pun bahan radioaktif. Selama pemeriksan MRI akan memungkinkan molekul-molekul dalam tubuh bergerak dan bergabung 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. Dalam mengidentifikasi penyakit dalam, biasanya dokter atau radiologis menganalisa image hasil Resonansi Magnetik yang disimpan dalam format *Digital Imaging Communication In Medicine* (DICOM). Dalam menganalisa file tersebut dibutuhkan keahlian dan pengalaman yang cukup agar diagnosa yang diberikan tepat dan akurat. Saat ini dunia teknologi kesehatan sudah bertransformasi menjadi digital dan penggunaan gambar medis sebagai dasar pemeriksaan suatu penyakit semakin meningkat tetapi hal ini tidak disertai dengan meningkatnya tenaga ahli dibidang radiologi.

Pada bagian radiologi khususnya, pencitraan medis sudah berubah dari sistem analog menjadi digital sehingga manajemen file MRI sangat dibutuhkan untuk membantu dokter atau radiologis dalam menganalisa *image* 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. Transformasi pencitraan kedalam bentuk digital juga telah meningkatkan mutu pelayanan medis terutama dalam peningkatan kualitas, kecepatan, komunikasi serta mempermudah peyimpanan data medis. Dengan melihat rutinitas dokter atau radiologis yang sering berpindah tempat untuk memberikan pelayanan kepada pasien menjadi penyebab kurangnya efektifitas kerja menyebabkan keterlambatan diagnosa dan ketidak akuratan interprestasi dari file citra medis padahal kesempatan untuk bertukar pendapat mengenai gambar medis sangat dibutuhkan dan ketersediaan berkas gambar medis sangatlah diperlukan. 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 majajemen file DICOM tersebut. Hasil dari aplikasi manajemen Citra MRI akan ditampung di server orthanc untuk mempermudah proses manajemen citra.

1.2 Permasalahan

Penggunaan MRI (Magnetic Resonance Imaging) untuk melakukan pencitraan penyakit pada pasien yang mengalami gangguan otak semakin dibutuhkan sebagai penunjang diagnosa utama. Seiring kemajuan teknologi yang sedang berkembang pesat saat ini, internet merupakan hal wajib bagi berbagai kalangan untuk mendapatkan banyak kemudahan. Dalam menunjang kemudahan akses oleh berbagai pengguna terkait hal tersebut, dibutuhkan pengembangan manajemen citra MRI berbasis online, yang memungkinkan perluasan akses dan dari berbagai device. Demikian juga tuntutan dibidang radiologi yang menunjang diagnosa ganguan otak pada pasien melalui citra MRI.

Dalam mengembangkan aplikasi untuk manajemen citra MRI tersebut dibutuhkan suatu *framework* yang memungkinkan untuk terus dikembangkan.Framework yang sesuai tersebut adalah framework laravel yang akan diintegrasikan dengan mysql dan orthanc server dalam menyimpan file citra MRI.

1.3 Tujuan

Dalam tugas akhir ini akan dikembangkan aplikasi manajemen citra MRI untuk menampilkan informasi di file DICOM. User dapat melakukan secara banyak *(multiple) upload* file citra MRI untuk mempermudah mengelola file citra MRI. Pengembangan aplikasi tersebut menggunakan *framework laravel* yang merupakan kerangka kerja terbaik, framework tersebut dirancang untuk memudahkan pekerjaan-pekerjaan umum yang sering digunakan pada mayoritas *web developer*. Disamping itu framework laravel menyediakan fitur yang memungkinkan integrasi dengan bahasa pemrograman lainnya termasuk server database.

1.4 Batasan Masalah

Untuk memfokuskan permasalahan yang diangkat, maka dilakukan pembatasan masalah. Batasan-batasan masalah tersebut adalah sebagai berikut:

1. Menggunakan data citra MRI DICOM yang akan diolah pada aplikasi.
2. Aplikasi dibuat menggunakan framework laravel yang merupakan salah satu framework PHP terbaik.
3. Aplikasi akan menggunakan MySQL dan orthanc sebagai sistem data yang digunakan dalam menyimpan data citra.

1.5 Sistematika Penulisan

Laporan penelitian Tugas akhir ini tersusun dalam sistematika dan terstruktur sehingga mudah dipahami dan dipelajari oleh pembaca maupun seseorang yang ingin melanjutkan penelitian ini. Alur sistematika penulisan laporan penelitian ini yaitu :

1. BAB I Pendahuluan
   Bab ini berisi uraian tentang latar belakang permasalahan, tujuan, batasan masalah, sistematika, penulisan, dan relevansi.
2. BAB II Dasar Teori
   Pada bab ini berisi tentang uraian secara sistematis teori-teori yang berhubungan dengan permasalahan yang dibahas pada pengerjaan tugas akhir ini. Teori-teori ini digunakan sebagai dasar dalam penelitian, yaitu framework, Laravel, Data Citra MRI, algoritma Deep Learning dan teori-teori penunjang lainya.
3. BAB III Perancangan Sistem dan Impementasi
   Bab ini berisi tentang penjelasan-penjelasan proses membuat pengembangan aplikasi sistem online yang mendukung manajemen citra MRI.
4. BAB IV Pengujian dan Analisa
   Bab ini menjelaskan tentang pengujian yang dilakukan terhadap pengembangan aplikasi yang telah dibuat untuk mendukung manajemen citra MRI. Hasil dari tugas akhir ini adalah menganalisa system tersebut. Spesifikasi perangkat keras dan perangkat lunak yang digunakan juga disebutkan dalam bab ini. Sehingga ketika akan dikembangkan lebih jauh, spesifikasi perlengkapannya bisa dipenuhi tanpa harus melakukan uji coba perangkat lunak maupun perangkat keras lagi.
5. BAB V Penutup
   Bab ini merupakan penutup yang berisi kesimpulan yang diambil dari penelitian dan pengujian yang telah dilakukan. Saran dan kritik yang membangun untuk pengembangkan lebih lanjut juga dituliskan pada bab ini.

1.6 Relevansi

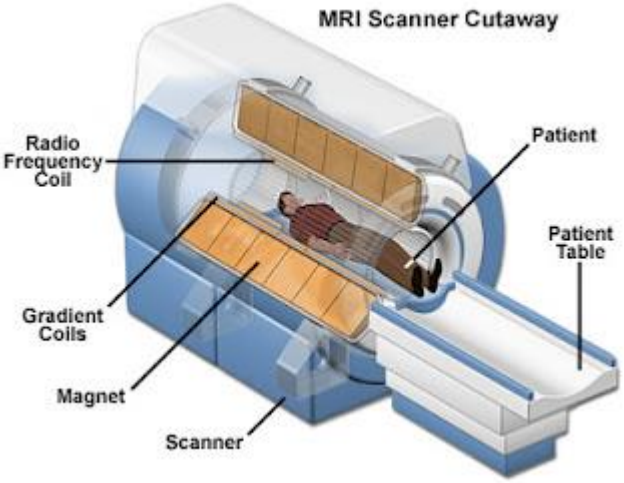
Penelitian mengenai pengembangan aplikasi manajemen Citra MRI berbasis framework laravel ini diharapkan sejalan dengan kebutuhan citra MRI untuk mendukung diagnosa gangguan otak dan bisa dikembangkan lebih lanjut untuk analisa lebih mendalam dalam memberikan diagnosa yang lebih tepat kepada pasien.

BAB 2
TINJAUAN PUSTAKA

Demi mendukung penelitian ini, dibutuhkan beberapa teori penunjang sebagai bahan acuan dan referensi. Dengan demikian penelitian ini menjadi lebih terarah.

2.1 Magnetic Resonance Imaging

Magnetic Resonance Imaging (MRI) adalah suatu alat kedok-teran di bidang pemeriksaan diagnostik radiologi ,yang menghasilkanrekaman gambar potongan penampang tubuh atau organ manusia dengan menggunakan medan magnet berkekuatan antara 0,0641,5 tesla dan resonansi getaran terhadap inti atom hydrogen. MRI dapat memberikan gambaran struktur tubuh yang tidak bisa dida- patkan pada tes lain, seperti Rontgen, USG, atau CT scan. Salah satu kelebihan pencitraan MRI adalah, menurut pengetahuan pe- ngobatan masa kini, tidak berbahaya kepada orang yang sakit. Dibandingkan dengan CT scans "computed axial tomography" yang menggunakan aksial tomogra berkomputer yang melibatkan dosis radiasi tertentu, MRI hanya menggunakan medan magnet kuat dan pancarannya tidak mengion dalam jalur frekuensi radio. Bagaimanapun, perlu diketahui bahwa pasien yang membawa benda asing logam (seperti serpihan peluru) atau implant tertanam (seperti tulang Titanium buatan, atau pacemaker) tidak boleh dipindai di dalam mesin MRI, disebabkan penggunaan medan megnet yang kuat. Sistem Kerja Magnetic Resonance Imaging (MRI), MRI berbentuk suatu tabung silinder yang ditengahnya terdapat ruang kosong dimana nantinya sang pasien akan dimasukkan untuk di ambil gambaran jaringan-jaringan yang diperlukan oleh dokter. Bagian dari Magnetic Resonance Imaging (MRI) adalah sebagai berikut.



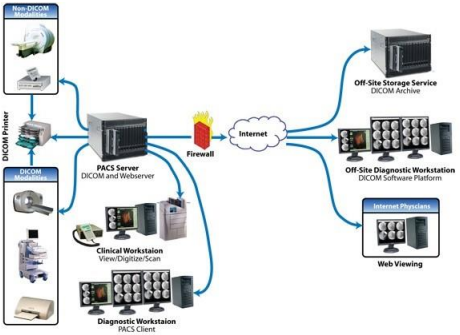
Gambar 2.1: Bagian MRI

Secara ringkas, proses terbentuknya citra Magnetic Resonance Imaging (MRI) dapat digambarkan sebagai berikut: bila tubuh pasien diposisikan dalam medan magnet yang kuat, inti-inti Hidrogen tubuh akan searah dan berotasi mengelilingi arah/vektor medan magnet. Bila signal frekuensi radio dipancarkan melalui tubuh, beberapa inti Hidrogenakan menyerap energi dari frekuensi radio tersebut dan mengubah arah, atau dengan kata lain mengadakan resonansi. Bila signal frekuensi radio dihentikan pancarannya, inti-inti tersebut akan kembali pada posisi semula, melepaskan energi yang telah diserap dan menimbulkan signal yang ditangkap oleh antena dan kemudian diproses komputer dalam bentuk radiograf. Pemeriksaan dalam dengan menggunakan MRI dapat diklasifikasikan aman sebab pada penggunaan Magnetic Resonance Imaging (MRI)ini pasien tidak terkena radiasi yang mungkin dapat membahayakan tubuh dalam jangka panjang.

2.2 Picture Archiving and Communication System

PACS (*Picture Archiving and Communication System*) merupakan sistem yang memungkinkan penyimpanan, pengambilan, dan menampilkan

sebuah citra medis, lebih khususnya untuk rumah sakit berskala besar dan klinik . Selain itu PACS merupakan sistem dengan konsep *filmless* atau tidak berbentuk fisik dengan metode komputerisasi komunikasi dan dapat menyimpan citra medis seperti *computed radiographic, digital radiographic, computed tomographic, ultrasound, fluoroscopic, magnetic resonance* dan lain-lain . Sebelumnya interpretasi citra medis bergantung pada pengalaman dan riwayat kesehatan pasien untuk memperoleh diagnosis yang tepat. Sebelum citra medis dapat memasuki PACS, diperlukan akuisisi citra yang dilakukan oleh berbagai *modality* (*computed radiographic, digital radiographic, computed tomographic, ultrasound, fluorosco- pic, magnetic resonance* dan lain-lain). Secara umum, ilustrasi pe- manfaatan PACS ditampilkan pada gambar 2.3.



Gambar 2.2: Ilustrasi pemanfaatan PACS

Terdapat dua metode untuk melakukan akuisisi citra, yaitu (1)*Direct Capture* dan (2)*Frame Grabbing*. Pada metode direct capture, antarmuka direct digital memungkinkan penangkapan dan transmisi sebuah data citra dari sebuah modality dengan resolusi spasial penuh dengan kedalaman bit atau gray scale yang berasal dari *modality*. ementara frame grabber analog (video) mendigitali- sasi sinyal output voltase video pada tampilan gambar, seperti sebuah scanner console monitor. Pada metode frame grabbing, seperti proses mencetak citra ke film, kualitas citra yang dihasilkan dibatasi pada proses, yaitu hanya 8 bits (atau 256 gray values). Seperti yang sudah dijelaskan bahwa seluruh proses akuisisi ini dapat di- lakukan oleh *modality*. PACS akan mengelola seluruh citra yang telah diakuisisi, sehingga dapat disimpan, diambil dan ditampilkan sesuai kebutuhan klinis tanpa adanya kesalahan.

Ketika citra sudah berhasil diakuisisi, citra harus dikelola dengan sesuai untuk memastikan bahwa penyimpanan, pengambilan dan pengiriman berlangsung tanpa adanya masalah. PACS harus menjamin bahwa citra yang disimpan dalam jangka waktu panjang dapat memenuhi aturan atau kewajiban hukum yang berla- ku di negara setempat. Sehingga dapat digunakan untuk interpre- tasi pada waktu yang dibutuhkan. Kebutuhan ini dipenuhi oleh PACS core, yaitu: (1) sistem manajemen database (Oracle, MS- SQL, Sybase), (2) media penyimpanan (RAID, Jukebox), (3) soft- ware kontrol (image manager), serta (4) antarmuka RIS. Sistem manajemen database merupakan jantung dari PACS, hubungan antara citra dan lokasi penyimpanannya disimpan dan dikelola dalam data- base bersama dengan seluruh data yang relevan yang dibutuhkan untuk mengambil citra. sistem manajemen database yang diguna- kan harus mampu untuk mengambil citra medis milik pasien yang sekarang ataupun yang sebelumnya ketika RIS atau sistem lainnya melakukan query. Query yang dapat direspon oleh database sis- tem ini didefinisikan sebagai *Digital Imaging and Communications in Medicine* (DICOM). Arsitektur database yang digunakan pada PACS biasanya bersifat relasional, seperti Oracle atau Microsoft SQL Server.

2.3. Standar DICOM

DICOM (Digital Imaging and Communication in Medicine) merupakan standar yang digunakan dalam PACS untuk komunikasi gambar dan data medis. 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 Ima- ging)* 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. Pemanfaatan standar DICOM dapat memberikan keuntungan, yaitu :

1. Memudahkan koneksi antar peralatan medis. kemudahan praktisi medis dalam memilih vendor, karena standar DICOM berlaku secara internasional.
2. Memudahkan praktisi untuk memilih piranti lunak yang digu-

nakan dalam menangani citra medis.
3. Memudahkan penyimpanan citra medis.

Secara komputasi, DICOM memiliki extension file .dcm. File ini tidak di-support secara otomatis oleh sistem operasi yang ada seperti Windows atau Linux. Untuk melihat isi file DICOM, kita harus menggunakan aplikasi DICOM Viewer. Dengan menggunakan aplikasi DICOM Viewer, kita bisa mendapatkan visualisasi dari obyek yang kita lihat secara 3 dimensi Visualisasi tersebut dilengkapi dengan data-data penunjang seperti data pasien, data obyek yang sedang dilihat, dan catatan-catatan penting lainnya.

2.4 Framework

Framework adalah seperangkat struktur dan pedoman kon- septual, yang digunakan untuk membangun sesuatu yang bermanfaat. Dalam istilah pembuatan web, framework merupakan software untuk memudahkan para programer 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 coding. Berdasarkan pengertian framework diatas, sudah tampak bahwa framework memiliki fungsi utama untuk membantu dan memudahkan para developer dalam menyelesaikan suatu proyek pengembangan software atau aplikasi. Selain itu, ada beberapa fungsi framework lainnya yaitu :

1. Menghemat Waktu Pengembangan
   Penggunaan *framework* dalam pengembangan suatu *software* akan mengurangi beban kerja developer, sehingga tidak ada waktu yang terbuang untuk memikirkan fungsi-fungsi umum yang akan digunakan. Selain itu, developer akan lebih fokus pada alur cerita pada aplikasi seperti yang dibutuhkan oleh pengguna. Dengan begitu, waktu pengembangan *software* akan berjalan lebih cepat dan dapat diserahkan kepada pengguna sesuai dengan waktu yang telah disepakati bersama.
2. Pemrograman menjadi Lebih Terstruktur
   Ketika developer menghadapi suatu proyek pengembangan *software* yang besar, maka akan terdapat banyak program yang ditulis

didalamnya. Terkadang, semakin banyaknya program akan menjadikan proses *debugging* semakin lambat. Selain itu, untuk mengecek kode program yang menjadi penyebab terjadinya *error* akan semakin sulit karena program tidak terstruktur dengan baik. Oleh karena itu, penggunaan framework dapat menjadikan pemrograman menjadi lebih terstruktur sehingga Anda dapat dengan mudah menemukan bagian-bagian dari kode yang perlu diperbaiki. Apalagi, ada framework yang menerapkan konsep *MVC (Model View Control)* yang memudahkan developer untuk memisahkan dan menyusun program berdasarkan bagiannya, yaitu Model, View ataupun Control.

3. Pengulangan Kode
   Seperti yang sudah disinggung pada poin pertama bahwa pembuatan software membutuhkan waktu yang lama, apalagi Anda berperan sebagai programmer tunggal. Maka Anda dapat menggunakan framework untuk mengurangi beban tersebut. Sebab, framework sudah menanamkan berbagai fungsi-fungsi umum yang bisa digunakan tanpa harus mengulangi pembuatan kode dari awal. Pastinya akan memudahkan juga untuk menggunakannya kembali di proyek-proyek berikutnya.
4. Meningkatkan Keamanan
   Keamanan menjadi suatu hal yang sangat vital dalam pengembangan software atau aplikasi. Apalagi, *software* yang memuat data pengguna yang privasinya harus dilindungi. Disini, framework terus memperbarui versinya yaitu menawarkan fitur yang handal dalam menangani berbagai jenis ancaman yang menyerang sistem keamanan.

2.5 Laravel

Laravel merupakan salah satu Framework PHP yang paling popular dan paling banyak digunakan di seluruh dunia dalam membangun aplikasi web mulai dari proyek kecil hingga besar. Laravel ini memiliki banyak keunggulan dari sisi kinerja, tur, dan skalabilitas nya, berikut adalah manfaat dari laravel:

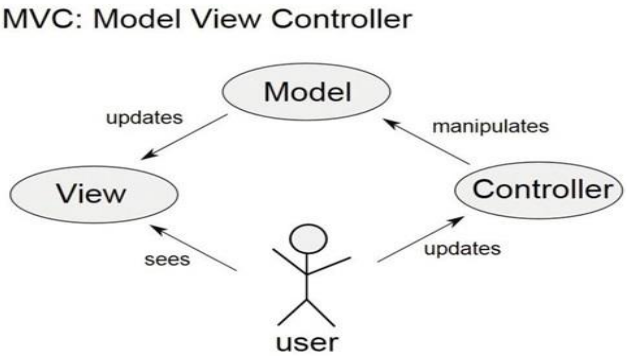
1. Laravel adalah Framework yang membantu pekerjaan programmer dengan membuat website menjadi lebih mudah untuk dikembangkan (Scalable).
2. Dalam upaya pengembangan website, Laravel memberikan kemampuan Framework untuk mengembangkan lebih cepat. Sehingga

menghemat waktu karena adanya kombinasi dengan komponen lain dari Framework lain.

3. Tampilan Framework yang mempermudah untuk digunakan dalam organisir dan mengatur sumber daya di dalam website.
4. Laravel bisa digunakan di hampir semua browser di berbagai perangkat dengan baik sehingga bisa digunakan untuk meningkatkan kunjungan website
5. Laravel adalah Framework yang memiliki sistem keamanan yang cukup kuat dengan menerapkan mekanisme password dengan metode hashed dan salted. Metode ini memung- kinkan password akan di enkripsi dalam database dan tidak akan tertembus ancaman SQL Injection. melengkapi keunggu- lannya dengan menyediakan library dan modular lebih banyak dari Framework PHP yang lain.

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*. Berikut ini adalah gambaran mode MVC model.



Gambar 2.3: *Model View Controller*

Biasanya model berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data dan lain-lain.

1. *View* adalah bagian yang mengatur tampilan ke pengguna. Bisa dikatakan berupa halaman web.
2. Controller merupakan bagian yang menjembatani model dan view.

Laravel mempunyai sintaks yang ekspresif, jelas dan menghemat waktu. *Framework* ini dikembangkan dengan tujuan bahwa pengembangan web harus dapat dinikmati dan penuh kreatifitas.

Pengembangan web dengan *Laravel* mempermudah proses pengembangan web dengan mempermudah tugas-tugas yang umum seperti *routing, authentication, sessions,* dan *caching.*

Pengembangan Web Menggunakan *framework Laravel* mempunyai kelebihan sebagai berikut:

1. Waktu yang dibutuhkan untuk mengembangan projek website dengan menggunakan *framework* ini menjadi lebih cepat.
2. Dapat meningkatkan pengunjung webiste karena teknologi *framework* ini dapat digunakan di segala *browser* dan berbagai perangkat dengan baik.
3. Laravel dilengkapi dengan utilitas pemrograman untuk membantu proses pengembangan aplikasi web dan juga moderasi dengan cara terbaik. Ini dikemas dengan *Modular Packaging System* (MPS) dengan pengaturan ketergantungan yang lengkap.

Beberapa fitur yang terdapat di Laravel :

1. Bundles, yaitu sebuah fitur dengan sistem pengemasan modular dan tersedia beragam di aplikasi.
2. Eloquent ORM, merupakan penerapan PHP lanjutan menyediakan metode internal dari pola “active record” yang menagatasi masalah pada hubungan objek database.
3. Application Logic, merupakan bagian dari aplikasi, menggunakan controller atau bagian Route.
4. Reverse Routing, mendefinisikan relasi atau hubungan antara Link dan Route.
5. Restful controllers, memisahkan logika dalam melayani HTTP GET and POST.
6. Class Auto Loading, menyediakan loading otomatis untuk class PHP.
7. *View Composer* adalah kode unit logikal yang dapat dieksekusi ketika view sedang loading.

8. IoC Container, memungkin obyek baru dihasilkan dengan pembalikan controller.
9. Migration, menyediakan sistem kontrol untuk skema database.
10. Unit Testing, banyak tes untuk mendeteksi dan mencegah regresi.

Untuk menggunakan Laravel harus melakukan penginstalan sebuah *composer*. *Composer* adalah alat manajemen *dependency* pada PHP seperti npm(Node.js) dan *Bundler(Ruby).Composer* memungkinkan untuk membuat library pada *project* anda dan *composer* sendiri akan meninstall atau mengupdate secara otomatis tanpa anda harus menginstall manual.
Kelebihan dalam menggunakan *composer*:

1. Membuat proses *coding* menggunakan PHP lebih terstruktur karena mengikuti konsep MVC.
2. Tidak diperlukan meng-include semua file PHP atau *class* PHP yang dibutuhkan,sudah ada *autoload* yang akan menangani fungsi tersebut.
3. *Package* yang dibutuhkan akan otomatis terpasang pada
4. Dengan menggunakan *pakagist,* kita dapat menggunakan *ribuan package*.

2.6 *Hypertext Preprocessor*

Hypertext preprocessor atau PHP adalah bahasa pemrograman script server-side yang didesain untuk pengembangan web. Selain itu, PHP juga bisa digunakan sebagai bahasa pemrograman umum (wikipedia). PHP di kembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan sekarang dikelola oleh The PHP Group. Situs resmi PHP beralamat di http://www.php.net. PHP disebut bahasa pemrograman *server side* karena PHP diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman *client-side* seperti *JavaScript* yang diproses pada web browser (client). Pada awalnya PHP merupakan singkatan dari *Personal Home Page.* Sesuai dengan namanya, PHP digunakan untuk membuat website pribadi. Dalam beberapa tahun perkembangannya, PHP menjelma menjadi bahasa pemrograman web yang powerful dan tidak hanya digunakan untuk membuat halaman web sederhana, tetapi juga website populer yang digunakan oleh jutaan orang seperti wikipedia, wordpress, joomla, dan sebagainya. PHP dapat digunakan dengan gratis (free) dan bersifat *Open Source*. PHP dirilis

dalam lisensi *PHP License*, sedikit berbeda dengan lisensi *GNU General Public License (GPL)* yang biasa digunakan untuk proyek *Open Source*. PHP memiliki beberapa kelebihan jika dibandingkan dengan bahasa pemrograman yang lain, diantaranya adalah:

4. Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
5. Web Server yang mendukung PHP dapat ditemukan dimanamana dari mulai Apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah. 12
6. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan developer yang siap membantu dalam pengembangan.
7. Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak.
8. PHP adalah bahasa open source yang dapat digunakan berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah sistem.
9. PHP memiliki delapan tipe data, yaitu : Boolean, Integer, Float/Double, String, Array, Object, Resource.

2.7 *Database*

Basis data (database) adalah suatu kumpulan data yang disusun dalam bentuk tabel-tabel yang saling berkaitan maupun berdiri sendiri dan disimpan secara bersama-sama pada suatu media. Database dapat digunakan oleh satu atau lebih program aplikasi secara optimal, data disimpan tanpa mengalami ketergantungan pada program yang akan menggunakannya. Sebuah Database dapat memiliki beberapa basis data. Setiap basis data dapat berisi atau memiliki sejumlah objek basis data seperi file atau tabel. Untuk mengelola database diperlukan suatu perangkat lunak yang disebut DBMS (Database Management System). DBMS merupakan suatu sistem perangkat lunak yang memungkinkan user (pengguna) untuk membuat, memelihara, mengontrol, dan mengaksesdatabase secara praktis dan efisien. Dengan DBMS, user akan lebih mudah mengontrol dan memanipulasi data yang ada. DBMS merupakan salah satu sistem dalam mengakses database yang menggunakan bahasa SQL.

Sedangkan RDBMS (Relationship Database Management System)

merupakan salah satu jenis DBMS yang mendukung adanya relationship atau hubungan antar tabel. Disamping RDBMS, terdapat jenis DBMS lain, misalnya Hierarchy DBMS, Object Oriented DBMS. Database mempunyai dua varian model, yaitu model Post-relational database dan model Object database.

2.7.1 *Post-relational database models*

Sebuah produk yang menawarkan model data yang lebih umum dari model relasional dan dikenal sebagai post-relational. Model data dalam produk tersebut mencakup hubungan namun tidak dibatasi oleh Prinsip Informasi yang mana mewakili semua informasi dengan nilai-nilai data dalam kaitannya dengan hal itu. Sebagian dari perluasan ini ke model relasional benar-benar mengintegrasikan konsep-konsep dari teknologi yang tanggal pre-date the relational model.

2.7.2 *Object database models*

Dalam beberapa tahun terakhir, paradigma yang berorientasi pada obyek telah diterapkan dalam bidang-bidang seperti teknik dan spasial database, telekomunikasi dan ilmu pilmiah lainnya. Para konglomerasi pemrograman berorientasi objek dan teknologi database mengarah pada model pemrograman baru yang dikenal sebagai Object database. Database ini berusaha untuk membawa dunia database dan aplikasi-dunia pemrograman lebih dekat bersama-sama, khususnya dengan memastikan bahwa database menggunakan jenis system yang sama seperti program aplikasi.

2.8 MySQL

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

2.8.1. Client Server Model

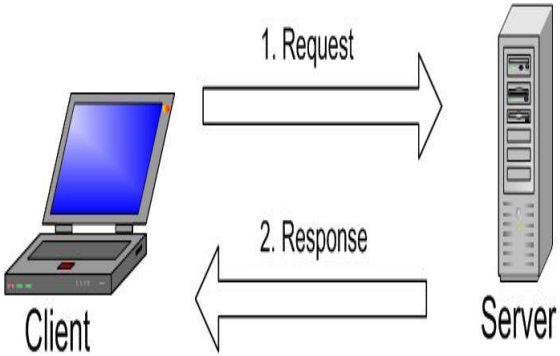
Komputer yang memasang dan menjalankan software RDBMS disebut sebagai *client*. Agar bisa mengakses data, komputer harus terhubung dengan server RDBMS terlebih dulu. Keadaan seperti inilah yang disebut client-server.MySQL adalah salah satu pilihan software RDBMS. Terkadang RDBMS dan MySQL dianggap sama karena popularitas MySQL. Pada awalnya MySQL dibuat untuk penggunaan terbatas saja, tapi sekarang software ini sudah kompatibel dengan berbagai platform *computing*, seperti Linux, macOS, Microsoft Windows, dan Ubuntu.

2.8.2. Kerja Sistem MySQL

MySQL dan SQL adalah dua software yang berbeda. MySQL merupakan salah satu nama brand terpopuler dari software RDBMS yang menerapkan *client-server* model. Lalu, bagaimana client dan server berkomunikasi di dalam ruang lingkup RDBMS? Jadi, baik client maupun server, keduanya menggunakan bahasa spesifik domain *Structured Query Language* (SQL). Jika Anda pernah melihat atau membaca beberapa nama yang dikombinasikan dengan SQL, misalnya P*ostgreSQL* dan *Microsoft SQL*, maka server tersebut biasanya menggunakan syntax SQL. Walaupun terkadang ditulis dalam bahasa pemrograman yang lain, software RDBMS selalu menggunakan SQL sebagai bahasa utama untuk berinteraksi dengan database. MySQL sendiri ditulis dalam C dan C++. Agar lebih mudah dipahami, kita mengambil negara-negara di Amerika Selatan sebagai contohnya. Secara geografis, negara-negara tersebut tidaklah sama, bahkan sejarahnya pun berbeda. Namun, masyarakat di semua negara di Amerika Selatan menggunakan bahasa Spanyol untuk berkomunikasi pada awal tahun 1970-an, seorang ahli komputer, Ted Codd, mengembangkan SQL dengan IBM berbasis model relasional. Pada tahun 1974, SQL mulai banyak digunakan dan dengan cepat menggantikan posisi bahasa yang sudah *outdated*, yakni ISAM dan VISAM. Tugas SQL adalah untuk memberitahukan server tentang apa yang harus dilakukannya terhadap data. Penggambaran umumnya

seperti password atau kode *WordPress*. Anda memasukkan password atau kode tersebut ke sistem untuk mendapatkan akses agar bisa login ke dashboard. Dalam hal ini, SQL statement menginstruksikan server untuk menjalankan operasi tertentu:

1. Data query: meminta informasi yang spesifik dari database yang sudah ada.
2. Manipulasi data: menambahkan, menghapus, mengubah, menyortir, melakukan operasi lainnya untuk memodifikasi data, value, atau visual.
3. Identitas data (data identity): menentukan tipe data, misalnya mengubah data numerik menjadi data integer. Selain itu, juga menentukan schema atau hubungan dari masing-masing tabel yang ada di database.
4. Data access control: menyediakan metode keamanan untuk melindungi data, termasuk dalam menentukan siapa yang boleh melihat atau menggunakan informasi yang tersimpan di database.



Gambar 2.4: Cara kerja MySQL

Gambar di atas menjelaskan struktur dasar dari client-server. Satu atau banyak perangkat terhubung ke server melalui network atau jaringan khusus. Setiap client dapat membuat permintaan (request) dari antarmuka pengguna grafis atau graphical user interface (GUI) di layar, dan server

akan membuat output yang diinginkan, sepanjang server dan juga client memahami instruksi dengan benar. Idealnya, proses utama yang terjadi di ruang lingkup MySQL sama, yaitu:

1. MySQL membuat database untuk menyimpan dan memanipulasi data, serta menentukan keterkaitan antara masing-masing tabel.
2. Client membuat permintaan (request) dengan mengetikkan pernyataan SQL yang spesifik di MySQL.
3. Aplikasi server akan merespons dengan memberikan informasi yang diminta. Informasi ini nantinya muncul di sisi klien.

Dari sisi client, biasanya akan diberitahukan MySQL GUI mana yang harus digunakan. Semakin ringan dan *user friendly* suatu GUI, maka semakin cepat dan mudah aktivitas manajemen data yang dimilikinya. Sebagian MySQL GUI yang terkenal adalah *MySQL WorkBench, SequelPro, DBVisualizer, dan Navicat DB Admin Tool*. Beberapa MySQL GUI terpopuler ada yang gratis dan ada juga yang berbayar, ada yang dijalankan secara eksklusif di macOS dan ada juga yang kompatibel dengan sistem operasi lainnya. Client memilih GUI berdasarkan pada kebutuhannya. Untuk manajemen database, termasuk situs WordPress, GUI yang paling sesuai adalah php MyAdmin.

Kelebihan MySQL Sehingga Banyak Digunakan:

1. Fleksibilitas dan kemudahan penggunaan
   Anda dapat memodifikasi *source code* sesuai dengan keinginan tanpa perlu mengkhawatirkan adanya batasan, termasuk opsi untuk mengupgrade paket saat ini ke versi premium berbayar. Proses installnya relatif sederhana dan tidak membutuhkan waktu lebih dari 30 menit.
2. Performa terbaik
   Ada banyak cluster server yang mensupport MySQL. Dengan performa dan kecepatan yang optimal, software ini akan membantu Anda baik dalam menyimpan sejumlah data e-Commerce berukuran besar maupun ketika melakukan kegiatan bisnis berat lainnya.
3. Memiliki standar industri
   Banyak industri yang telah menggunakan MySQL bertahun-tahun lamanya, dan itu berarti ada sejumlah resource yang dikelola oleh developer berpengalaman. Dengan demikian, user akan mendapatkan

software MySQL yang terus-menerus diperbarui. Software ini dapat dikembangkan oleh siapa pun, bahkan freelance developer demi memperoleh uang saku.

4. Aman
   Keamanan data menjadi salah satu prioritas utama software RDBMS. Dengan *Access Privilege System dan User Account Management* yang dimilikinya, MySQL menetapkan level keamanan tingkat tinggi. Verifikasi berbasis *host* dan enkripsi password juga tersedia.

2.6. Orthanc *Server*

Orthanc secara inheren adalah VNA (Vendor Neutral Archive) yang dapat menerima, menyimpan, mengindeks, dan mengirim gambar medis sesuai dengan standar DICOM. Othanc bertujuan menyediakan server DICOM mandiri yang sederhana namun kuat. Orthanc memungkinkan penggunanya 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 isntalasi dependensi pihak ketiga.

Halaman ini sengaja dikosongkan

BAB 3
DESAIN DAN IMPLEMENTASI SISTEM

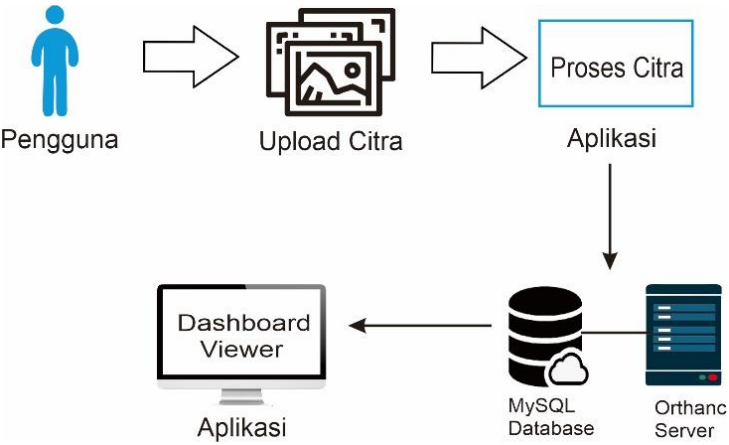
Tugas akhir ini memiliki tujuan untuk melakukan penyimpanan citra medis MRI melalui aplikasi agar dapat diakses oleh dokter atau pengguna yang tedaftar di aplikasi sehingga dapat digunakan berbagai unit di rumah sakit atau lembaga tertentu yang tidak terbatas pada satu *workstation* saja. Sekaligus memproses dan menampilkan informasi dari file citra medis MRI sebagai kebutuhan pengembangan sistem analisa citra agar dapat dipelajari lebih lanjut.

Dalam penelitian ini, gambaran umum kerja sistem ditampilkan pada gambar 3.1. Sistem dalam penelitian ini dimulai dari pengguna yang sebelumnya telah terdaftar dalam aplikasi melakukan upload citra medis kemudian citra akan proses oleh aplikasi dan disimpan dalam database MySQL. Implementasi proses deep learning dalam penelitian ini hanya sebagai rancangan untuk pengembangan aplikasi tahap berikutnya. PACS sebagai sistem yang memungkinkan penyimpanan, pengambilan, dan menampilkan sebuah citra medis sebagai kebutuhan interpretasi medis akan digantikan oleh lokal server yang berfungsi untuk penyimpanan data citra medis saja. Seluruh data citra medis yang tersimpan pada server akan dapat ditampilkan dan diupdate sesuai kebutuhan klinis.

Citra medis yang di simpan dalam server menggunakan protokol DICOM, yaitu standar protokol yang biasa digunakan di dalam PACS. Kemudian citra medis yang tersimpan akan di tampilkan di aplikasi sehingga dapat membantu dokter atau pengguna untuk melakukan interpretasi citra medis dan mendiagnosa lebih lanjut.

3.1 Desain Sistem

Secara garis besar, terdapat tujuan proses yang ada pada bagian ini atau sesuai dengan desain sistem yang digambarkan melalui *workflow*, *unifed modeling language* (UML), dan *use case diagram*, berikut dengan implementasinya.

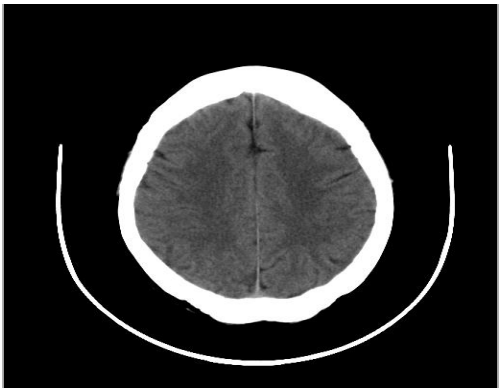


Gambar 3.1: Alur Kerja Sistem

Penelitian ini dilakukan sesuai dengan desain sistem ditunjukan pada gambar 3.1. Sehingga Konsep dari pembuatan dan perancangan infrastruktur diwujudkan dari proses citra menuju ke aplikasi (dashboard viewer).

1. Input Data Citra MRI

Pada penelitian ini datanya adalah soft copy hasil pencitraan MRI abdomen berupa gambar/citra 2 D berurutan yang dihasilkan dari alat MRI. Data tersebut bisa diambil atau dicopy langsung dari perangkat pengolah komputer pengolah data yang ada di MRI. Pada tahapan ini pengguna akan melakukan input/upload data gambar Citra MRI yang akan digunakan untuk diagnosa pada aplikasi. Agar dapat diproses dalam program yang telah dibuat maka file citra harus berbentuk DICOM (*Digital Imaging and Communications in Medicine*).



Gambar 3.2: CT Mono brain

2. Pemrosesan Data Citra

Data citra yang di capture dan dianalisa akan menampilkan informasi file DICOM tunggal berisi sebuah header yang menyimpan informasi tentang informasi file dan beberapa metadata. Metadata citra dapat berisikan data yang berhubungan dengan pasien atau citra yang diunggah, antara lain, nama pasien, jenis kelamin, tanggal lahir, *study case*, dan akuisisi data citra lainnya. File DICOM juga dapat dikompresi (*encapsulated*) untuk mengurangi ukuran gambar.

File Information

Name:	CT-MONO2-16-brain.dcm
File type:	DICOM Image (.dcm)
MIME Type:	image/x-dicom
Opens with:	File Viewer Plus 3
Size:	513 KB (525.968 bytes)
Location:	C:\Users\admin\Documents\luthfi\gambar\MicroDICOM\
Created:	May 12, 2020 10:24 AM
Modified:	September 23, 2019 7:22 PM
Accessed:	May 12, 2020 10:24 AM
Read-only:	No
Hidden:	No
System:	No
MD5:	a21649de46e5f5609ef6e227999a1dc8

Gambar 3.3: Informasi Metadata Citra

Gambar 3.3 dilanjutkan dari halaman sebelumnya

Dalam contoh gambar 3.3 ukuran file adalah 513 kb, ukuran dari header ini bervariasi tergantung pada seberapa banyak informasi header yang disimpan. Informasi header dan data gambar disimpan dalam file yang sama.

Gambar 3.4: Metodologi Proses Citra

Selain mendapati citra medis, gambar yang sudah diambil juga akan menampilkan beberapa informasi yang belum terdapat pada objek citra sehingga pengguna dapat mengubah atau menambahkan informasi sebelum data dibuat dan disimpan di dalam database.

3. Penampilan Citra Medis

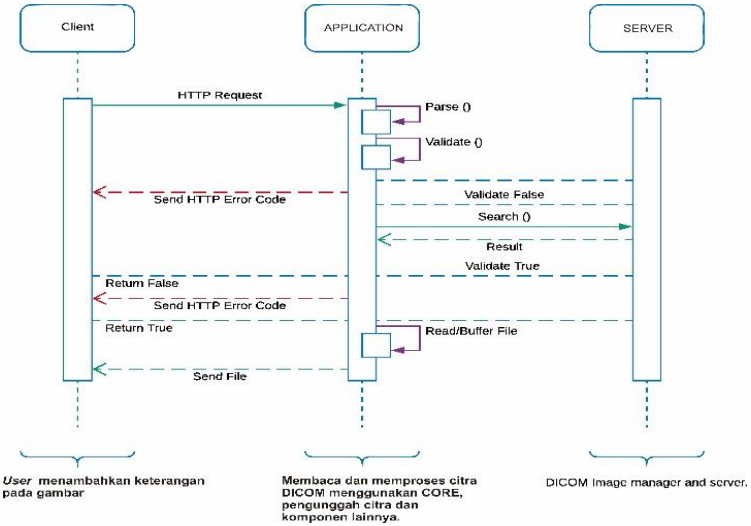
Aplikasi akan menampilkan citra medis yang sudah melalui proses pemrosesan data dan menampilkan citra DICOM yang divisualisasikan secara langsung untuk mempermudah dokter dalam memberikan diagnosa lebih lanjut berdasarkan gangguan otak pasien

3.1.1 Workflow Arsitektur Aplikasi

Sistem yang dikerjakan digambarkan ke dalam diagram yang ditunjukan pada gambar 3.5, 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.

Dalam penelitian ini clien akan mengunggah citra DICOM melalui sistem client menuju layanan aplikasi dengan menggunakan *library* yang mendefinisikan parameter yang diperlukan oleh server untuk mengenditifikasikan objek DICOM dengan jelas. Semua parameter tersebut mengacu pada modifikasi citra atau format data dimana server melakukan validasi parameter saat melakukan pencarian. Gambar 3.5 menunjukan 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 di browser web (klien).

Setelah menerima Permintaan HTTP, layanan aplikasi akan mengumpulkan parameter dan menautkannya ke model data. Jika permintaan divalidasi dengan benar, layanan mencari lokasi objek di server dan jika tersedia menanggapi Permintaan HTTP dengan mengembalikan data objek. Jika tidak, permintaan dijawab dengan kode kesalahan HTTP.



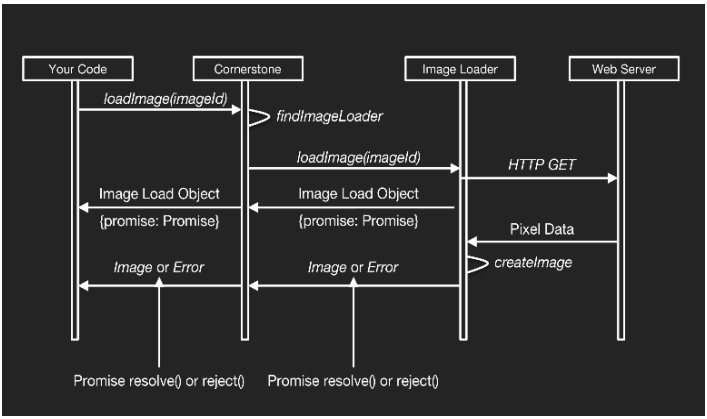
Gambar 3.5: Urutan permintaan UML ke layanan

Cornerstone adalah pustaka JavaScript yang dirancang untuk menampilkan gambar medis di lingkungan web dengan menggunakan elemen Canvas HTML5. Cornerstone Core tidak dimaksudkan untuk menjadi aplikasi yang lengkap, melainkan komponen yang menggunakan bagian dari aplikasi yang lebih kompleks dan lebih besar, dan digunakan untuk menyimpan piksel citra dan untuk mendapatkan data citra. Connerstone Core tidak memiliki kemampuan membaca atau menguraikan citra selain bergantung pada fungsi ImageLoader, dicom-Parser (Validate() dan Parse() dari gambar 3.5). Tujuannya adalah untuk membatasi pengembang untuk bekerja dalam satu wadah dan menyimpan gambar dalam berbagai format citra. Dengan pembatasan tersebut dapat mendapatkan performa tampilan citra yang lebih baik karena tidak membutuhkan konversi yang mengenai wadah atau alternatif lain.

3.1.2 *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.

Pada Penelitian ini dibuat sebuah *workflow* proses untuk menampilkan citra medis menggunakan *Image Loader* yang ditunjukan pada gambar 3.6.



Gambar 3.6: *Workflow Image Loader*

Untuk mengetahui tahapan alur kerja proses menampilkan citra, maka dibuat tahapan workflow dan dijelaskan sesuai dengan gambar 3.6, yaitu:

1. Citra akan dibaca oleh cornerstone untuk mendapatkan hirarki id citra (*ImageId*) untuk mendapatkan format url di mana lokasi citra dimuat.
2. *Image Loader* mendaftarkan tiap gambar yang dibaca dengan tujuan untuk memuat skema URL *ImageId* tertentu.

3. Connerstone akan mengirim permintaan untuk memuat gambar ke Image Loader yang terdaftar dengan scema URL dari *ImageId* dan diteruskan ke load Image.
4. Image Loader akan mengembalikan sebuah Objek *Image Load* yang berisi sebuah *promise* yang akan diselesaikan dengan objek citra yang sesuai setelah memperoleh data piksel.
5. Untuk mendapatkan data piksel setidaknya memerlukan permintaan ke server menggunakan XMLHttpRequest, dekompresi data piksel (missal dari JPEG 2000) dan dikonversi ke format yang dapat dipahami connerstone.
6. Objek citra akan dikembalikan oleh promise yang telah diselesiakan kemudian ditampilkan dengan displayImage API.

3.1.3 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 3.5 dan gambar 3.6. File-file tambahan tercantum sebagai berikut:

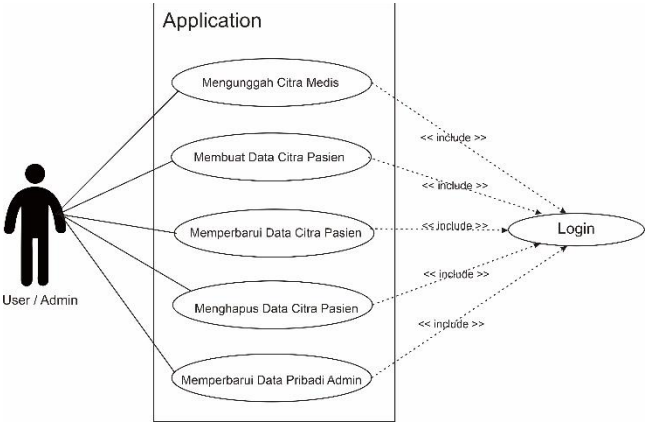
1. *Image Loader*
   Merupakan objek javascript yang diimplementasikan sebagai plugin yang berfungsi untuk mengambil citra DICOM sebagai ImageId dan mengembalikan data piksel yang sesuai untuk citra yang sama. Sementara untuk memuat citra DICOM sering membutuhkan panggilan ke server Orthanc. Sementara bila menggunakan API citra harus diproses secara.
2. *Promises*
   Menangani proses asynchronous (HTTP *Request)*. Sebuah Promise akan menerima data pixel secara *asynchronous* dan mengembalikan nilai kesalahan bila terjadi error.
3. *Image Rendering*
   Event yang digunakan untuk merender citra setiap kali citra dipanggil dan mencakup setiap fitur menggambar atau anotasi yang digunakan di dalam citra dengan menggunakan HTML5 *canvas*. Dan dapat diberbagai dengan berbagai property viewport seperti *zoom, window level* and *window width.*
4. *Viewport*
   Setiap elemen yang diaktifkan memiliki yang penjelesan

bagaimana gambar DICOM seharusnya direnderatau ditampilkan. Meliputi dimensi lebar, tinggi citra dalam pixel.

5. DICOM *Parser*
   Setiap objek citra akan diparsing ke dalam HTML5 dan node.js tidak memerlukan dependensi eksternal. Metode ImageLoader menggunakan library ini untuk mengekstrak data piksel dari file DICOM dan menampilkan citra pada user interface aplikasi.

3.1.4 *Use Case Diagram*

Use Case Diagram merupakan gambaran *graphical* dari beberapa atau actor, use case, dan interaksi diantaranya terlibat dalam suatu sistem. *Use case diagram* tidak menjelaskan secara detail tentang penggunaan *use case* tetapi hanya memberikan gambara singkat antara *use case*, aktor, dan sistem. Pada *use case* diagram yang dapat dilihat pada gambar 3.7, aktornya adalah hanya user yang terdaftar/admin pada aplikasi, karena kebutuhannya belum membutuhkan akses *privilege user* lain.



Gambar 3.7: *Use case diagram* aplikasi

Untuk mengetahui use case dari aktor user/admin dengan sistem yang dibuat, maka dibuat skenario-skenario yang dijelaskan pada tabel 3.1 s/d 3.7, yaitu:

1. Skenario *Use Case* Mengunggah Citra Medis
   Nama Use Case : Mengunggah Citra Medis
   Aktor : User / Admin

Deskripsi : Pada proses ini user dapat menggungah citra medis pasien yang belum terdaftar dalam data base.
Precondition : Aktor dapat memilih gambar citra terlebih dulu sebelum diupload ke sistem.
Postcondition : Setelah memilih gambar citra, tampilan citra akan ditampilkan ke sistem.

Tabel 3.1: Skenario *use case* mengunggah citra medis

Aktor	**Sistem**
1. Aktor mengisi username dan password pada halaman login	
	2. Sistem memverifikasi data akun dan menampilkan dashboard
3. Aktor memilih gambar citra untuk diunggah	
	4. Sistem menampilkan gambar citra yang sudah dipilih.

2. Skenario *Use Case* Membuat Data Citra Pasien
Nama Use Case : Membuat Data Citra Pasien
Aktor : User / Admin
Deskripsi : Pada proses ini user dapat membuat data citra medis pasien.
Precondition : Aktor dapat mengecek dan memberikan tambahan informasi pada citra yang akan dibuat ke database.
Postcondition : Setelah menambahkan informasi citra, aktor dapat langsung *create* data citra pasien dan menyimpannya ke database.

Tabel 3.2: Skenario *use case* membuat data citra pasien

Aktor	**Sistem**
1. Aktor mengisi username dan password pada halaman login	
	2. Sistem memverifikasi data akun dan menampilkan dashboard
3. Aktor mengecek dan dapat menambahkan informasi pada gambar citra pasien yang akan disimpan ke database	
	4. Sistem menyimpan citra dan infomarsi gambar citra pasien ke dalam database.

3. Skenario *Use Case* Memperbarui Data Citra Pasien

   Nama Use Case : Memperbarui Data Citra Pasien
   Aktor : User / Admin
   Deskripsi : Pada proses ini user dapat mempebarui data citra medis pasien yang sudah terdaftar pada database.
   Precondition : Aktor dapat mengubah isi formulir atau informasi data citra pasien.
   Postcondition : Setelah aktor mengubah informasi data citra pasien, data yang terisi dapat tersimpan pada database.

Tabel 3.3: Skenario *use case* memperbarui data citra pasien

Aktor	**Sistem**
1. Aktor mengisi username dan password pada halaman login	

	2. Sistem memverifikasi data akun dan menampilkan dashboard
3. Aktor mengubah isi formulir atau informasi data citra pasien	
	4. Sistem menyimpan data citra pasien dan menampilkannya pada dashboard.

4. Skenario *Use Case* Menghapus Data Citra Pasien

Nama Use Case	:	Menghapus Data Citra Pasien
Aktor	:	User / Admin
Deskripsi	:	Pada proses ini user dapat menghapus data pasien yang telah tersimpan pada database.
Precondition	:	Aktor dapat menghapus data pasien yang ditampilkan pada dashboard.
Postcondition	:	Setelah actor menghapus data pasien, data pasien yang dihapus tidak akan ditampilkan lagi pada dashboard.

Tabel 3.4: Skenario *use case* menghapus data citra pasien

Aktor	**Sistem**
1. Aktor mengisi username dan password pada halaman login	
	2. Sistem memverifikasi data akun dan menampilkan dashboard
3. Aktor menghapus data pasien yang ditampilkan di dashboard	
	4. Sistem menghapus data pasien dan kembali pada dashboard.

5. Skenario *Use Case* Memperbarui Data Pribadi Admin

Nama Use Case	:	Memperbarui Data Pribadi Admin
Aktor	:	User / Admin
Deskripsi	:	Pada proses ini user memperbarui data pribadi yang telah tersimpan pada database.
Precondition	:	Aktor dapat memperbarui data admin atau user yang diigunakan untuk l ogin dan ditampilkan pada dashboard.
Postcondition	:	Setelah aktor memperbarui data admin atau user, data pasien yang dihapus tidak akan ditampilkan lagi pada dashboard.

Tabel 3.5: Skenario use case memperbaharui data pribadi admin

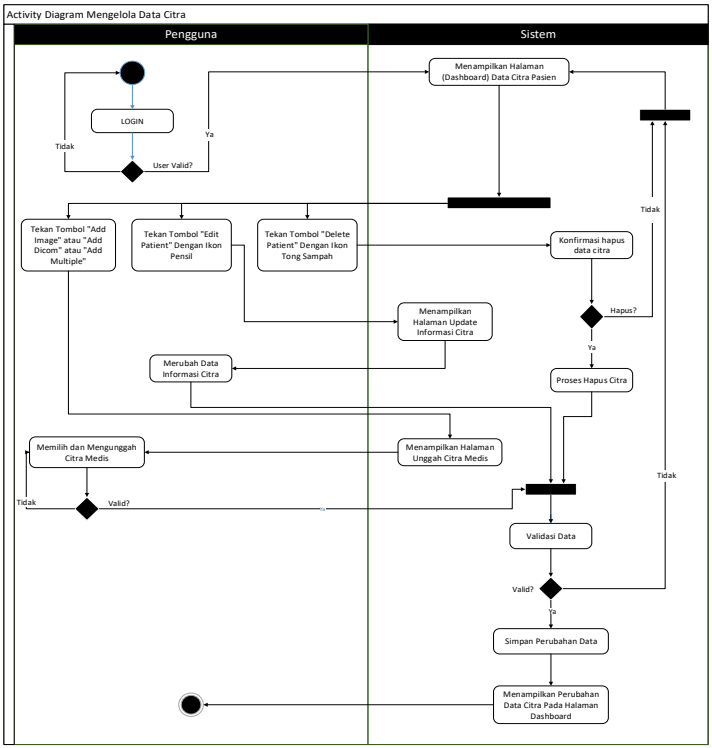
Aktor	**Sistem**
1. Aktor mengisi username dan password pada halaman login	
	2. Sistem memverifikasi data akun dan menampilkan dashboard
3. Aktor menghapus data pasien yang ditampilkan di dashboard	
	4. Sistem menghapus data pasien dan kembali pada dashboard.

3.1.5. *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 3.8 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 dashbord sesuai hak akses pengguna.

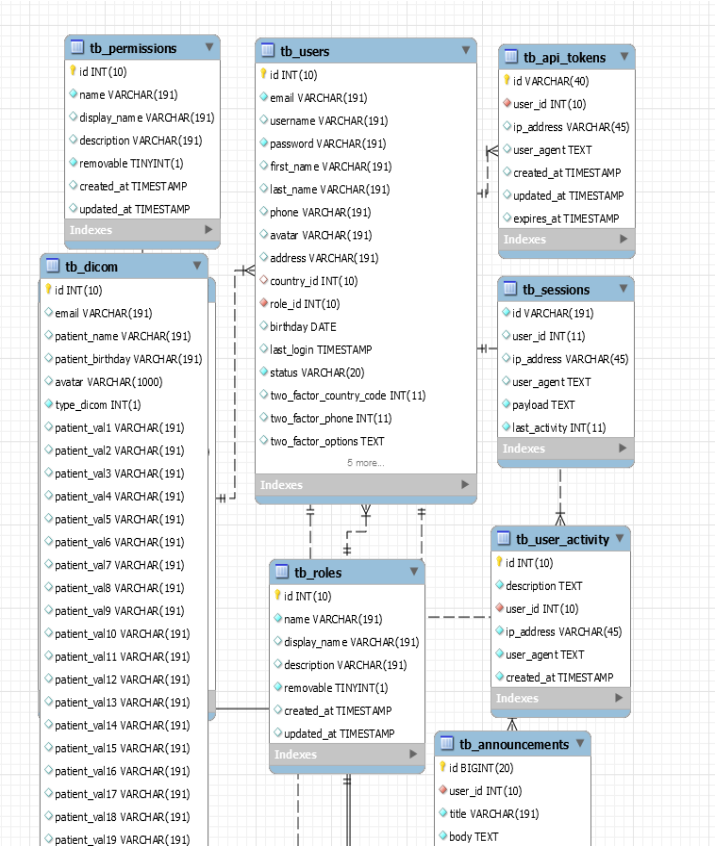
Pada dashboard admin, aktifitas yang dapat dilakukan adalah mengungangah citra medis, membuat atau mendaftarkan data citra pasien baru, melakukan penambahan informasi yang belum tersedia. Data citra pasien yang telah terdaftar sebagai juga dapat dilihat kembali dan dilakukan pembaruan informasi citra.



Gambar 3.8: *Activity* Diagram mengelola data citra pasien

3.2 Entity Relationship Diagram

ERD (Entity Relationship Diagram) adalah suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD dibuat untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol sesuai dengan gambar 3.9, agar sistem dapat berjalan dengan baik sesuai dengan workflow yang telah dirancang.



Gambar 3.9: *Entity relationship diagram* aplikasi

Tabel-tabel yang dibutuhkan pada perancangan sistem ini, yaitu:

1. Tabel pertama adalah *tb_users*, tabel ini adalah tabel yang digunakan untuk menghimpun seluruh data pengguna yang untuk mengakses fungsionalitas sistem aplikasi termasuk yang digunakan untuk data login ke sistem. Isi tabel ini meliputi nama pengguna, username, password, alamat, nomer telepon, tanggal lahir, gambar profil, role akses, tanggal lahir, status akun, hingga waktu terakhir login.
2. Tabel kedua adalah *tb_permissions*, tabel ini adalah tabel yang digunakan untuk menghimpun akses halaman pada aplikasi yang bisa diakses oleh pengguna berdasar role yang diberikan. Isi tabel ini meliputi nama halaman, nama halaman yang ditampilkan pada sistem, deskripsi, hingga waktu kapan data dibuat dan diupdate.
3. Tabel ketiga adalah *tb_roles*, tabel ini adalah tabel yang digunakan untuk menghimpun jenis hak akses yang diberikan pada pengguna aplikasi, apakah user biasa atau admin. Isi tabel ini meliputi jenis nama pengguna, nama yang pengguna yang ditampilkan sistem, hingga waktu kapan data dibuat dan diupdate.
4. Tabel keempat adalah *tb_api_tokens*, tabel ini adalah tabel yang digunakan untuk autentifikasi tambahan ketika pengguna mengakses aplikasi dengan memeriksa dan memverifikasi setiap token API yang digunakan untuk login pada aplikasi. Tabel tersebut termasuk komponen di dalam laravel yang dapat digunakan untuk pengembangan lebih lanjut. Isi tabel ini meliputi id pengguna, ip address, *user agent*, hingga waktu expired token.
5. Tabel kelima adalah *tb_sessions*, tabel ini adalah tabel yang digunakan untuk menghimpun semua *session* yang menyediakan cara untuk menyimpan semua informasi pengguna di semua permintaan halaman sistem, seperti menyimpan file, cookie, database, array, hingga data login. Isi tabel ini meliputi id pengguna, ip address, *user agent*, payload, hingga kapan aktivitas terakhir dilakukan pada sistem.
6. Tabel keenam adalah *tb_user_activity*, tabel ini adalah tabel yang digunakan untuk menghimpun semua data *log* pengguna kapan terakhir melakukan aktivitas login atau logout pada aplikasi, dan jenis *browser* yang digunakan. Isi tabel ini meliputi deskripsi login atau

logout, id pengguna login, ip address, *user agent*, dan waktu terakhir data dibuat.

7. Tabel ketujuh adalah *tb_announcements*, tabel ini adalah tabel yang digunakan untuk menghimpun data notifikasi yang dapat ditampilkan kepada pengguna aplikasi. Isi tabel ini meliputi, id pengguna, judul, isi deskripsi,, *user agent*, dan hingga waktu kapan data dibuat dan diupdate.
8. Tabel kedelapan adalah *tb_dicom*, tabel ini adalah tabel yang digunakan untuk menghimpun semua data citra dicom yang didaftarakan pada sistem. Isi tabel ini meliputi, nama pasien, email, tanggal lahir, gambar citra, tipe dicom, hingga data-data meta lainnya.

3.3 Implementasi Orthanc Sebagai PACS Server

Pada penelitian ini digunakan sistem operasi windows untuk melakukan implementasi Orthanc sebagai bagian dari PACS *server*. Implementasi dilakukan dengan cara melakukan instalasi orthanc pada sistem operasi dan pemasangan paket yang diperlukan oleh Orthanc agar dapat berjalan secara fungsional sebagai media penyimpanan citra melalui sistem aplikasi website yang dikerjakan.

Orthanc dapat diakses secara local melalui web browser dengan alamat *http://localhost:8042/*. Tampilan antarmuka Orthanc dapat dilihat pada gambar 3.10.

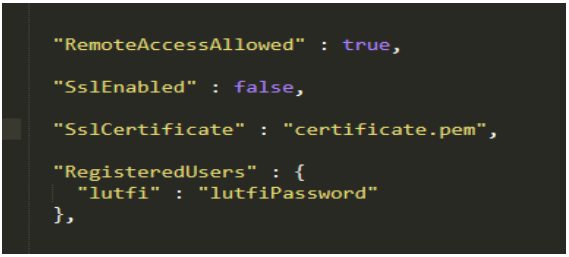
Gambar 3.10: Tampilan antarmuka Orthanc

3.3.1 Konfigurasi Jaringan

Pada Orthanc terdapat beberapa komponen Configuration File

yang berguna untuk mengatur akses dari setiap perangkat yang terhubung satu jaringan intranet dengan Orthanc untuk mendapatkan akses pada citra medis yang tersimpan.

Orthanc server dapat diakases melalui seluruh jaringan intranet yang terhubung dengan melakukan perubahan pada Configuration File. Yaitu dengan merubah salah satu variabel pada file orthanc.json yang awal bernilai false menjadi true dan dapat mendaftarkan user untuk mendapatkan akses.



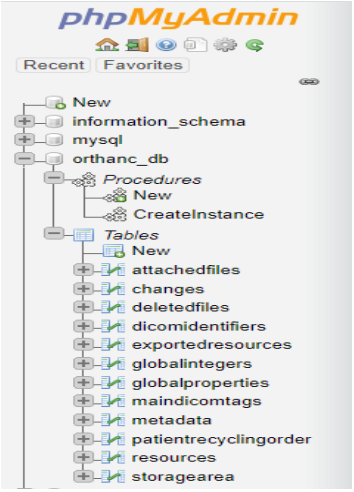
Gambar 3.11: Konfigurasi File pada Orthanc

Pengguna yang telah terdaftar pada *Configuration* File dapat melakukan akses pada Orthanc *server* dengan melakukan akses ip address dari Orhanc server melalui web browser, dan mengisikan id beserta password yang telah terdaftar dalam *Configuration* File. Dan untuk dapat melakukan pertukaran data citra medis dalam Orthanc *server* memerlukan penambahan *modality* dan ip address dari setiap perangkat yang terhubung ke orthanc server yang belum terdaftar pada *Configuration* file. Namun karena dalam pengerjaan penelitian ini Orthanc *server* tidak memerlukan penambahan perangkat tertentu untuk melakukan pertukaran data citra medis sehingga ip address tidak perlu ditambahkan, maka unggah citra cukup dilakukan melalui sistem aplikasi yang dibuat.

3.3.2 Implementasi MySQL Pada Orthanc

Pada penelitian ini implementasi MySQL pada Orthanc digunakan sebagai cara menggantikan SQLite standar Orthanc untuk menghubungkan database yang digunakan pada aplikasi dibuat dengan framework Laravel sehingga dapat berjalan dengan baik. Implementasi MySQL pada Orthanc dilakukan dengan cara melakukan pemasangan

paket MySQL pada sistem operasi Windows. Database serta tabel default milik Orthanc pada MySQL ditampilkan pada gambar 3.12



Gambar 3.12: *Database* Orthanc pada MySQL

Database Orthanc pada MySQL dapat berjalan pada sistem aplikasi yang dibuat apabila perubahan pada *Configuration File* telah dikonfigurasi pada *file* mysql.json Orthanc *server*. Konfigursi dapat dilakukan dengan mengubah beberapa variabel pada file tersebut yang awalnya false menjadi true, serta mengubah nama database, username, password sesuai dengan database yang ada pada Orthanc *server*.

```
"MySQL" : {
"EnableIndex" :true,
"EnableStorage" :true,
"Host" : "localhost",
"Port" : "3306",
"Database" : "orthanc_db",
"Username" : "root",
"Password" : "",
"Password" : false,
```

3.3.3 Implementasi Orthanc pada Laravel

Pada penilitian ini terdapat beberapa fungsi agar orthanc dapat digunakan sebagai media penyimpanan manajemen citra MRI DICOM melalui sistem aplikasi ketika pengguna melakukan unggah maupun saat menghapus citra medis melalui aplikasi. Fungsi yang digunakan adalah memberikan fungsi curl php dengan menyertakan alamat *instance* Orthanc *server* pada bagian *controller* laravel agar unggah maupun menghapus citra medis dapat berjalan dengan baik

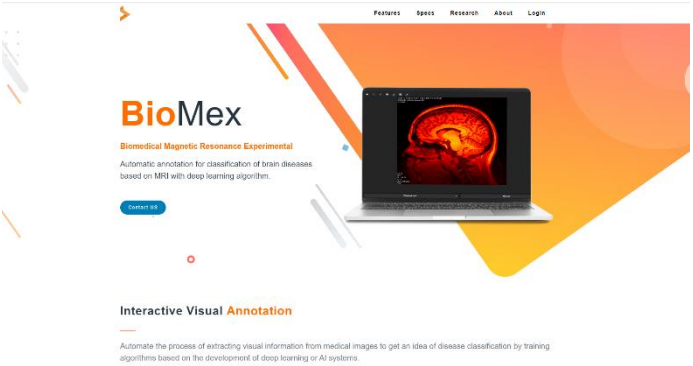
```
///fungsi unggah orthanc
$cmd='curl -X POST http://localhost:8042/instances --
data-binary @'.$file;
exec($cmd);
```

3.4 Rancangan *User Interface*

User interface merupakan bagian penting sebagai tampilan sebuah model yang menjebatani sistem dengan pengguna user. *User interface* diimplementasikan dalam sebuah platform aplikasi website sehingga dibutuhkan beberapa fungsi yang menyajikan data dengan mudah dimengerti oleh pengguna dalam mengakses layanan aplikasi. Layanan aplikasi ini terbagi menjadi beberapa halaman yaitu halaman beranda, halaman login, halaman daftar citra pasien, halaman penampil citra medis, dan halaman profil pengguna sesuai akses login ke aplikasi.

3.4.1 Desain Halaman Beranda Aplikasi

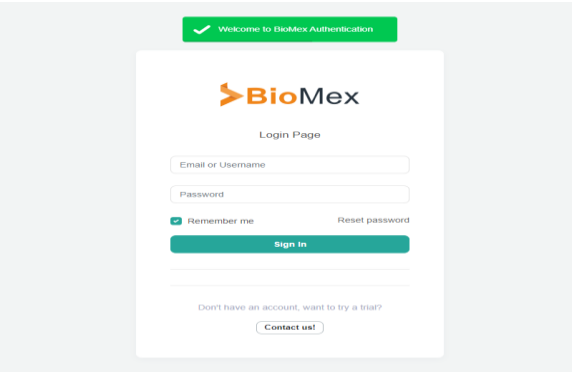
Halaman awal aplikasi yaitu halaman beranda atau portal aplikasi yang ditampilkan sebelum masuk ke halaman login. Halaman berisi informasi mengenai alur kerja secara umum aplikasi. Sehingga pengunjung umum yang mengakses aplikasi tahu mengenai aplikasi yang dirancang.



Gambar 3.13: Halaman beranda aplikasi

3.4.2 Desain Halaman Login

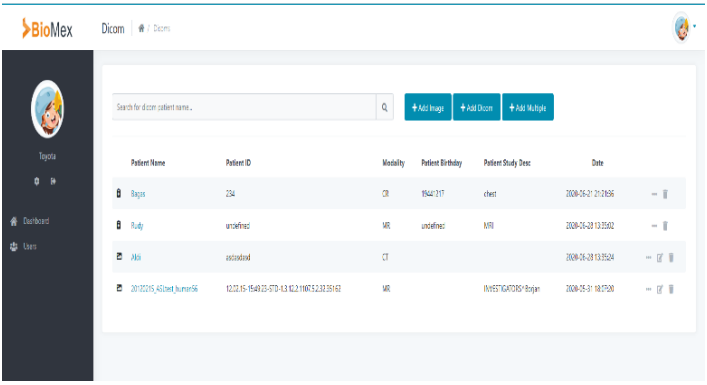
Halaman awal aplikasi juga termasuk halaman login yang dapat diakses dari menu halama portal aplikasi, yang didalamnya terdapat dua buah formulir yaitu username dan password. Dan sebuah tombol *Sign In* seperti yang ditampilkan pada gambar 3.14. Tombol tersebut berfungsi sebagai penghubung antara halaman login dan dashboard aplikasi. Sehingga Ketika tombol *Sign In* ditekan, pengguna akan dialihkan ke dashboard sesuai dengan hak akses pengguna.



Gambar 3.14: Halaman login aplikasi

3.4.3 Desain Halaman Daftar Citra Pasien

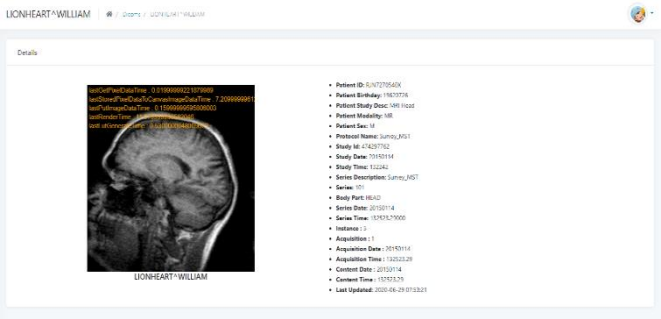
Halaman citra pasien akan ditampilkan ketika pengguna berhasil login. Halaman ini ditampilkan pada gambar 3.15. Pada halaman ini pengguna akan disajikan data-data pasien yang telah dibuat dan telah terdaftar pada database yang mencangkup id pasien, nama pasien, modality, tanggal lahir, dan tanggal data pasien dibuat. Pengguna juga dapat melakukan proses pembuatan data pasien atau menggunggah citra medis dengan klik dari salah satu tombol *Add Image*, *Add Dicom*, atau *Add Multiple*. Tombol Edit yang digambarkan dengan symbol pensil dapat digunakan untuk mengubah data pasien. Tombol menampilkan citra gambar digambarkan dengan simbol titik tiga *horizontal* dapat digunakan untuk melihat informasi dan citra gambar dari data pasien yang terdaftar. Dan tombol hapus digambarkan dengan simbol tong sampah dapat digunakan untuk menghapus data pasien.



Gambar 3.15: Halaman daftar citra pasien

3.4.4 Desain Halaman Penampil Citra Pasien

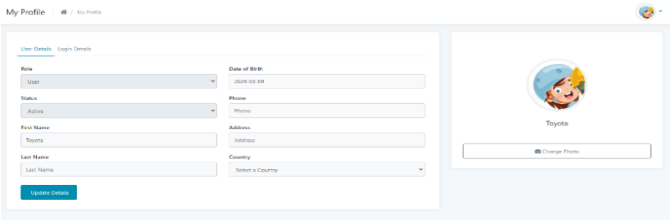
Halaman penampil citra pasien muncul ketika pengguna menekan tombol *view patient* setelah menekan simbol titik tiga *horizontal* atau dengan menekan nama pasien pada halaman daftar citra pasien. Halaman penampil citra pasien berfungsi menampilkan citra medis dan informasi-informasi yang dimiliki pasien, dan sebagai data pendukung. Tampilam halaman penampil citra medis ditunjukan pada gambar 3.12.



Gambar 3.16: Halaman penampil citra pasien

3.4.5 Desain Halaman Profil Admin

Halaman profil admin atau pengguna muncul ketika pengguna menekan tombol pengaturan yang terdapat di sebelah kiri dan bagian bawah gambar profil admin pada dashboard. Pada halaman ini pengguna dapat melakukan pembaruan data mengenai nama, tanggal lahir, alamat, negara, dan bahkan dapat mengubah detail email, username, dan password yang digunakan untuk login ke dashboard aplikasi. Tampilam halaman profil admin ditunjukan pada gambar 3.17.



Gambar 3.17: Halaman profil admin

BAB 4
PENGUJIAN DAN ANALISA

Pada bab ini dipaparkan hasil pengujian serta analisa dari desain sistem dan implementasi. Pengujian dilakukan guna mengetahui fungsionalitas dari sistem yang dibuat dan memastikan pengiriman data berjalan dengan baik sehingga dapat menarik kesimpulan dari sistem yang telah dibuat. Secara garis besar, pengujian yang dilakukan adalah sebagai berikut :

1. Pengujian Kesesuaian Sistem Aplikasi meliputi standar citra berdasatkan sistem yang telah dirancang.
2. Pengujian Performa Sistem meliputi waktu unggah citra me – dis dan waktu render pada aplikasi.
3. Pengujian Website atau Aplikasi meliputi *user experience* dan *user interface*.

Dalam pengujian ini secara keseluruhan sistem diuji dalam lingkungan 1 host (komputer) sebagai lokal server dengan spesifikasi perangkat lunak dan perangkat keras seperti pada tabel 4.1.

Tabel 4.1: Spesifikasi *Server*

Spesifikasi Server	
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

Pada bagian ini dilakukan pengujian penampil citra medis menggunakan library connerstone pada sistem menggunakan jaringan lokal atau intranet. Setelah dilakukan pengujian, hasil uji menunjukan hasil yang didapatkan dari pengujian standar citra DICOM yang dapat dibaca dan berjalan dengan baik oleh sistem sesuai skenario yang diharapkan.

4.1 Pengujian Kesesuaian Sistem Aplikasi

Pada pengujian kesesuaian sistem aplikasi, hal yang diperhatikan adalah apakah sistem aplikasi berjalan dengan desain sistem yang diinginkan. Pengujian dilakukan sebanyak lebih dari 100 kali uji scenario untuk mengecek seluruh proses dalam sistem aplikasi dan fitur yang ada apakah berjalan dengan baik atau tidak.

Tabel 4.2: 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

Dari pengujian tampilan citra medis dengan 3 jenis scenario uji sebanyak lebih 100 kali percobaan semua skenario dapat berjalan dengan baik dan tanpa ada kesalahan.

4.2 Pengujian Performansi Sistem

Pada bagian ini dilakukan pengujian untuk mengetahui performansi sistem aplikasi yang telah dirancang. Bagian yang diuji adalah pengujian waktu unggah citra medis dan pengujian waktu render citra medis. Pengujian ini dilakukan dalam lingkungan perangkat keras dengan spesisifikasi teknis yang ditampilkan pada tabel 4.3.

Tabel 4.3: Spesifikasi Komputer

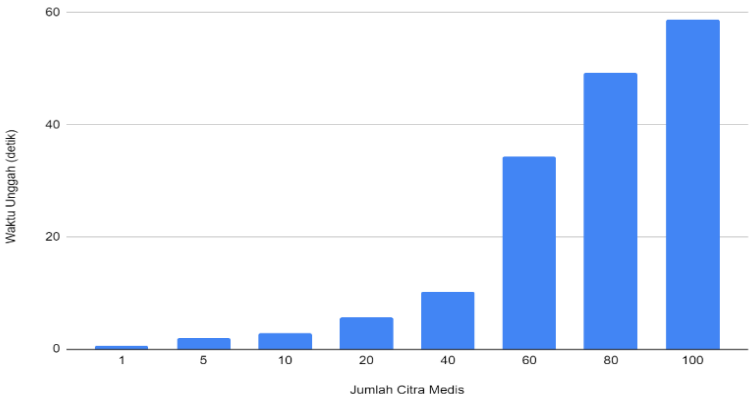
Spesifikasi Komputer	
Processor	Intel Core i7-9750H (2.6 GHz, 12M Cache)
RAM	8 GB DDR4
Storage	512GB INTELSSD PEKNW512G8
OS	Windows Pro 64 bit

4.2.1 Pengujian Waktu Unggah Citra Medis

Unggah atau sering dikatakan juga dengan upload, merupakan suatu cara untuk mengirimkan 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. Hasil pengujian waktu unggah citra medis dari sistem aplikasi yang telah dirancang adalah sebagai berikut.

Tabel 4.4: Pengujian waktu unggah citra medis

Uji	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



Gambar 4.1: Grafik pengujian waktu unggah citra medis

Dari tabel 4.4 dan grafik 4.1 menunjukan 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.

4.2.2 Pengujian Waktu Render Citra Medis

Render merupaka suatu proses membangun gambar dari sebuah model melalui program komputer. Waktu render gambar yang cepat akan memberikan pengalaman yang lebih baik bagi pengguna sistem. Pengujian waktu render citra medis dilakukan dengan menampilkan sebuah series 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 4.5.

Tabel 4.5: 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 time		0,414

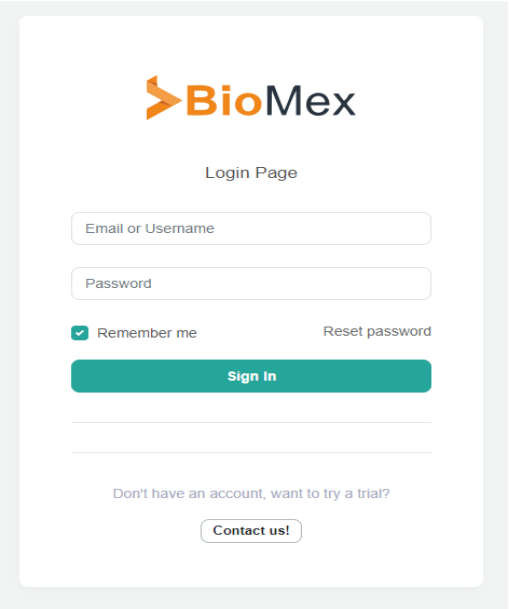
Dari tabel 4.5, ditunjukan bahwa waktu tercepat yang dapat diperolah ketika citra dapat divisulaisasikan oleh sistem aplikasi adalah 0,27 detik, dengan rata-rata interval 0,414 detik.

4.3. Pengujian Fungsionalitas Antarmuka Sistem

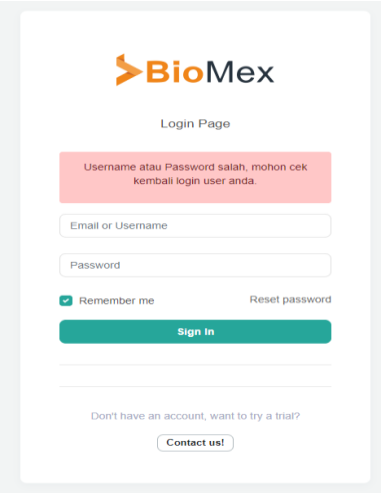
Pada bagian ini dilakukan pengujian untuk memastikan apakah sistem yang dirancang dapat berjalan dengan sesuai. Bagian yang diuji adalah pengujian fungsionslitas antarmuka sistem aplikasi.

4.3.1 Pengujian Halaman Login

Pada bagian ini, dilakukan pengujian pada apiklasi dengan uji coba pada halaman login untuk masuk ke halaman dashboard. Ketika pengguna menekan tombol *Sign In* dengan formulir yang kosong maka pengguna akan dikembalikan pada halaman login kembali. Ketika pengguna menekan tombol login dengan formulir yang telah terisi dengan hak akses yang telah terdafatar seperti yang ditampilkan pada gambar 4.1, maka pengguna akan dialihkan pada dashboard sesuai dengan hak aksesnya. Namun apabila pengguna salah memasukan informasi formulir login yang salah dan tidak terdaftar di database maka akan memunculkan notifikasi error, seperti pada gambar 4.2.



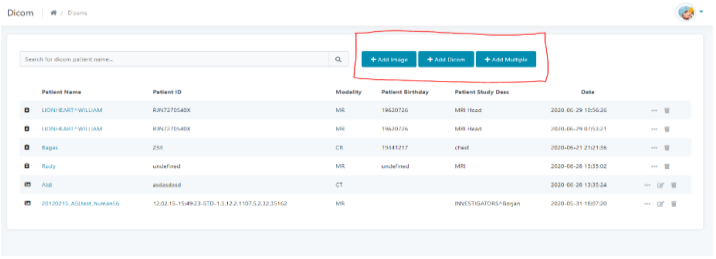
Gambar 4.2: Pengujian halaman login aplikasi



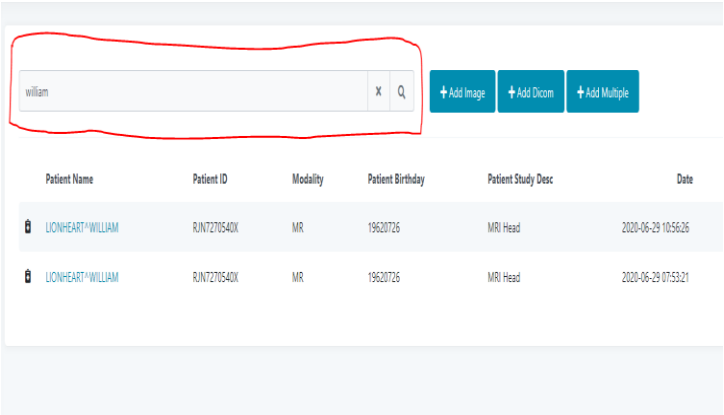
Gambar 4.3: Notifikasi informasi login yang tidak sesuai

4.3.2 Pengujian Halaman Citra Pasien

Pada bagian ini, ditunjukan beberapa citra pasien yang sebelumnya sudah terdaftar pada database. Pengujian pada aplikasi dengan uji coba pada halaman citra pasien. Pengguna dapat melakukan unggah citra atau membuat data pasien baru melalui tombol yang Add yang di sediakan pada tampilan sesuai pada gambar 4.3. Pengguna juga dapat melakukan pencarian data citra pasien yang telah terdaftar pada aplikasi melalui formulir pencarian di atas daftar citra pasien yang ditunjukkan pada gambar 4.4.



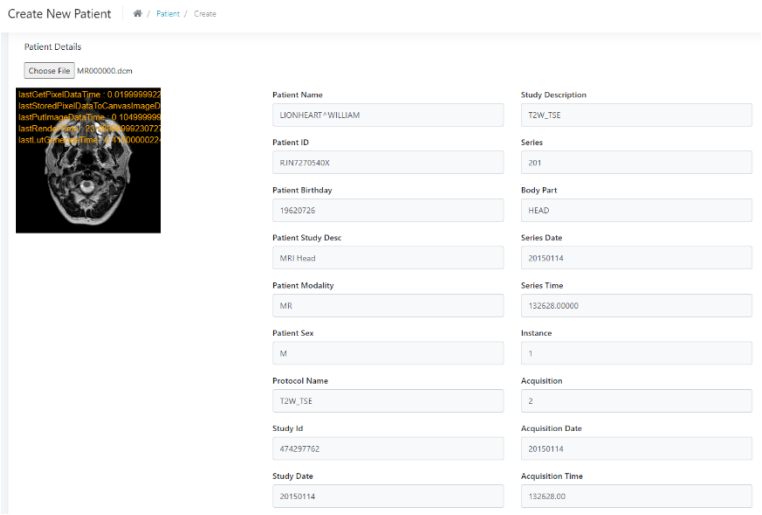
Gambar 4.4: Pengujian halaman citra pasien



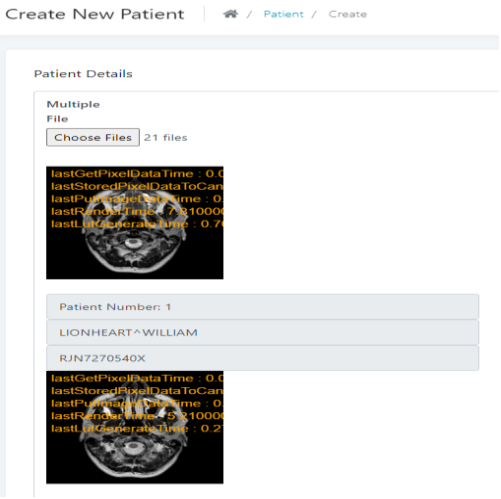
Gambar 4.5: Pengujian pencarian data citra pasien

4.3.3 Pengujian Halaman Unggah Citra Pasien

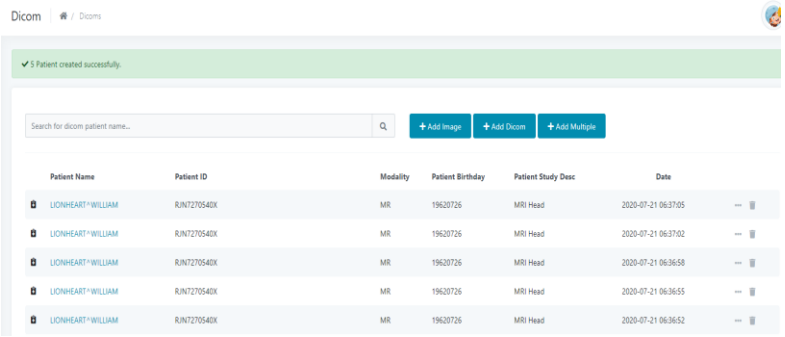
Pada bagian ini, ditunjukan pengujian aplikasi untuk membuat data citra pasien atau menggunggah data citra yang sebelumnya belum terdaftar pada database. Pengguna dapat melakukan unggah citra atau membuat, menambah data pasien baru yang belum ada melalui halaman unggah citra yang sesuai gambar 4.5. Namun bila gambar citra yang diupload memerlukan lebih dari satu gambar dapat menggunakan tombol menu *Add Multiple* yang ada pada halaman citra atau sesuai tampilan sebelumnya pada gambar 4.3. Pengujian *Add Multiple* ditampilkan pada gambar 4.6, dan penampilan citra medis yang sudah tersimpan di database ditunjukan sesuai pada gambar 4.7.



Gambar 4.6: Pengujian halaman unggah data citra *single* baru



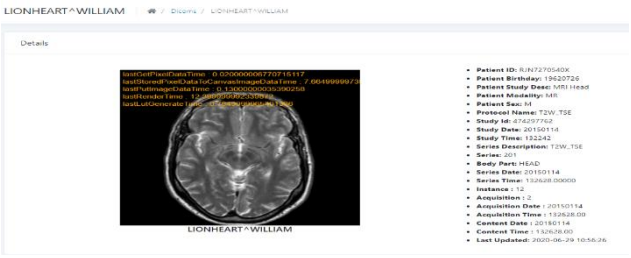
Gambar 4.7: Pengujian unggah dan membuat data citra *single* baru



Gambar 4.8: Hasil Pengujian unggah data citra baru

4.3.4 Pengujian Halaman Penampil Citra Pasien

Pada bagian ini, ditunjukan pengujian aplikasi untuk menampilkan data citra pasein yang sebelumnya sudah tersimpan atau terdaftar pada database. Pengguna dapat melihat data citra pasien dengan memilih data citra pada halaman citra pasien yang sebelumnya ditampilkan pada gambar 4.3, dengan mengklik nama pasien yang ada pada daftar citra. Pengujian unggah citra atau membuat, menambah data pasien baru yang belum ada melalui halaman unggah citra yang sesuai gambar 4.5. Namun bila gambar citra yang diupload memerlukan lebih dari satu gambar dapat menggunakan tombol menu *Add Multiple* yang ada pada halaman citra atau sesuai tampilan yang ditunjukan pada gambar 4.3. Pengujian *Add Multiple* ditampilkan pada gambar 4.6, dan hasil pengujian unggah citra ditunjukan sesuai pada gambar 4.7.



Gambar 4.9: Pengujian halaman penampil citra pasien

BAB 5
PENUTUP

5.1 Kesimpulan

Dalam penelitian ini telah diimplementasikan pembuatan aplikasi manajemen citra MRI berbasis framework laravel. Beberapa kesimpulan terkait penelitian yang dilakukan antara lain:

1. Pada tugas akhir ini telah berhasil dikembangkan aplikasi untuk manajemen citra MRI menggunakan framewok Laravel untuk informasi di file dicom.
2. Pengunggahan citra medis dapat dilakukan secara banyak *(multiple)* dalam sekali upload melalui sistem aplikasi.
3. Sistem belum bisa optimal untuk melakukan unggah citra bila jumlah citra lebih dari 40 dalam sekali upload karena membutuhkan waktu yang jauh relatif lebih lama bila dibandingkan unggah citra dengan jumlah 60 data, yang membutuhkan waktu sekitar 3,36 kali dibandingkan unggah 40 data citra dengan waktu unggah 10,2 detik.
4. Aplikasi manajemen citra sudah memenuhi kebutuhan standar fungsional dimana *user interface* dapat bekerja, unggah data citra sudah dapat tersimpan pada MySQL database termasuk pada Orthanc *server*, serta visualisasi data citra juga mampu ditampilkan dengan baik.

5.2 Saran

Demi pengembangan lebih lanjut mengenai tugas akhir ini, disarankan beberapa langkah lanjutan sebagai berikut:

1. Perlu peningkatan performa agar pengunggahan citra medis secara multiple/lebih banyak lagi, minimal 100 gambar citra sekali upload lebih cepat.
2. Perlu penambahan berbagai fitur yang memungkinkan penambahan anotasi untuk lebih memperjelas analisa citra medis saat citra medis ditampilkan.

3. Aplikasi memerlukan pengembangan lebih lanjut untuk dapat menampilkan atau menvisualisasikan data series, sehingga analisa citra dapat menjadi lebih jelas dan akurat.

DAFTAR PUSTAKA

[1] *Digital imaging and communications in medicine (DICOM)*. National Electrical Manufacturers Association, 1998. (Dikutip pada halaman 2, 7, 8).

[2] H. K. Huang and Faimbe, *PACS and imaging informatics.* John Wiley, 583AD. (Dikutip pada halaman 3).

[3] T. Rangga, *Implementasi PACS Orthanc Berbasis Postgresql dan Radiology Information System*, 2019. (Dikutip pada halaman 5).

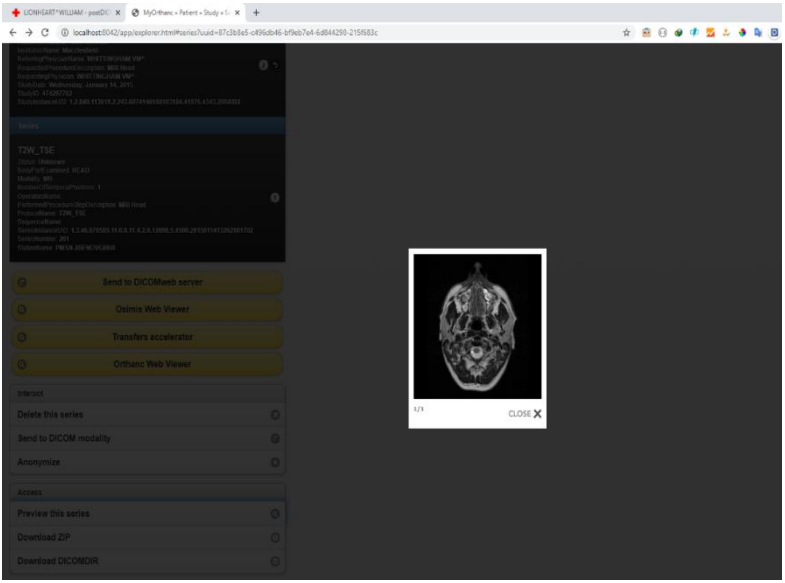
[4] N. Jacinto & G. Daniel, *Medical Imaging Multimodality Breast Cancer Diagnosis User Interface*, 2017. (Dikutip pada halaman 46, 49).

[5] Y. Neha, R. S. Dharmveer, D. K. Shri, *LARAVEL: A PHP Framework for E-Commerce Website*, 2019. (Dikutip pada halaman 20).

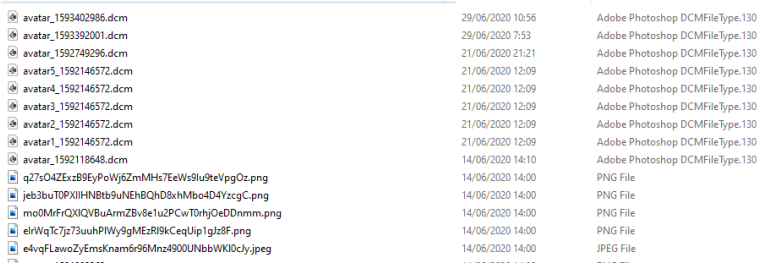
[6] Jodogne, Sébastien, et al., *Orthanc-A lightweight, restful DICOM server for healthcare and medical research*. IEEE 10th International Symposium on Biomedical Imaging. IEEE, 2013. (Dikutip pada halaman 19)

Halaman ini sengaja dikosongkan

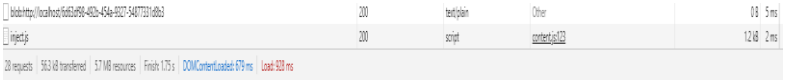
LAMPIRAN



Gambar 1: Citra medis yang diterima pada PACS *Server*



Gambar 2: File citra yang berhasil disimpan melalui aplikasi



Gambar 3: Waktu render citra medis

BIOGRAFI PENULIS

Lutfiadji Mazda Pradenta, lahir pada Tanggal 03 Maret 1995 di Klaten, Jawa Tengah. Penulis merupakan anak pertama dari dua bersaudara. Pada Tahun 2007 lulus dari SD Negeri Bayat. Tahun 2010 lulus dari SMP Negeri 1 Cawas Klaten, kemudian melanjutkan pendidikan ke SMA Neg Cawas Klaten hingga akhirnya lulus pada tahun 2013. Penulis kemudian melanjutkan Pendidikan S-1 ke Departemen Teknik Kom puter, FTE-ITS Surabaya. Saat di kuliah pe- nulis aktif di Laboratorium Telematika Komputer B201 sebagai asisten 2015/2017. Selama masa kuliah penulis juga aktif UKM KENDO ITS tahun 2013/2014. Penulis juga pernah memiliki pengalaman bergabung dengan komunitas start up yang bergerak di bidang konsultan IT 2016 sampai 2018. Bagi pembaca yang memiliki kritik, saran atau pertanyaan mengenai tugas akhir ini dapat menghubungi penulis melalui email: lutfidev29@gmail.com.

Halaman ini sengaja dikosongkan