

**UNIVERSITAS ESA UNGGUL**

**APLIKASI MANAJEMEN KONTEN UNTUK VIRTUAL TOUR BERBASIS WEB MENGGUNAKAN TEKNOLOGI MERN STACK**

**TUGAS AKHIR**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana**

**NAMA : YOGI PRASETYA**

**NIM : 20180801192**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS ESA UNGGUL**

**TAHUN 2021**

# HALAMAN PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini:

Nama : Yogi Prasetya

NIM : 20180801192

Jurusan : Teknik Informatika

Menyatakan bahwa Tugas Akhir ini merupakan murni hasil karya saya, Saya bersedia dikenakan sanksi pembatalan Tugas Akhir ini dan dibatalkan gelar keSarjanaan, apabila terbukti melakukan tindakan plagiarisme.

Demikian pernyataan ini saya buat dengan sebenarnya.

Yogi Prasetya

(20180801192)

# HALAMAN PENGESAHAN TUGAS AKHIR

Tugas Akhir ini diajukan oleh:

Nama : Yogi Prasetya

NIM : 20180801192

Program Studi : Teknik Informatika

Fakultas Ilmu Komputer - Universitas Esa Unggul

Judul Tugas Akhir : Aplikasi Manajemen Konten untuk Virtual Tour Berbasis Web Menggunakan Teknologi MERN Stack

**Telah berhasil dipertahankan dihadapan Tim Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika Fakultas Ilmu Komputer, Universitas Esa Unggul.**

**TIM PENGUJI**

|  |  |  |  |
| --- | --- | --- | --- |
| Pembimbing | : | Malabay, S.Kom, M.Kom | (  ) |
| Penguji I | : | Hani Dewi Ariessanti, S.Kom, M.Kom | (  ) |
| Penguji II | : | Diana Novita, ST, M.M | (  ) |
| Ditetapkan di | : | Jakarta |  |
| Ketua Program Studi | : | M. Bahrul Ulum, S.Kom, M.Kom | (  ) |
| Tanggal | : | 30 Agustus 2021 |  |

# HALAMAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

**UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademika Universitas Esa Unggul, saya yang bertanda tangan di bawah ini:

Nama : Yogi Prasetya

NIM : 20180801192

Program Studi : Teknik Informatika

Fakultas : Fakultas Ilmu Komputer

Jenis Karya Ilmiah : Tugas Akhir

Demi pengembangan ilmu pengetahuan, teknologi, dan seni, menyetujui untuk memberikan kepada Universitas Esa Unggul Hak Bebas Royalti Noneksklusif atas karya ilmiah saya yang berjudul:

**Aplikasi Manajemen Konten untuk Virtual Tour Berbasis Web Menggunakan Teknologi MERN Stack**

Beserta perangkat yang ada (apabila diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini, Universitas Esa Unggul berhak menyimpan, mengalihmediakan, mengelola dalam bentuk pangkalan data, merawat, dan mempublikasikan Tugas Akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Jakarta

Pada tanggal : 1 September 2021

Yang menyatakan

**(Yogi Prasetya)**

# KATA PENGANTAR

Puji syukur kehadirat Allah SWT Yang Maha Pengasih atas segala limpahan rahmat, taufik, serta hidayah-Nya sehingga penulis dapat menyelesaikan proposal tugas akhir yang berjudul “Aplikasi Manajemen Konten untuk *Virtual Tour* Berbasis Web Menggunakan Teknologi MERN Stack” ini sesuai dengan waktu yang telah ditentukan.

Tentunya banyak pihak telah memberikan bantuan baik moril maupun materil. Oleh karena itu penulis ingin menyampaikan ucapan terimakasih yang tiada hingganya kepada:

1. Bapak Habibullah Akbar, S.Si, M.Sc, Ph.D selaku Dekan Fakultas Ilmu Komputer.
2. Bapak Malabay, S.Kom, M.M selaku pembimbing materi.
3. Secara khusus penulis ingin mengucapkan kepada Ibunda, sehingga penulis dapat menjalani masa studi dengan baik.

Penulis menyadari bahwa proyek tugas akhir ini masih jauh dari kesempurnaan, maka saran dan kritik yang membangun dari semua pihak sangat diharapkan demi penyempurnaan selanjutnya.

Akhirnya hanya kepada Allah SWT kita kembalikan semua urusan dan semoga penulisan tugas akhir ini dapat bermanfaat bagi semua pihak, khususnya bagi penulis dan para pembaca pada umumnya.

Jakarta, September 2021

(Yogi Prasetya)

NIM: 20180801192

# ABSTRAK

|  |  |  |
| --- | --- | --- |
| Judul | : | Aplikasi Manajemen Konten untuk Virtual Tour Berbasis Web Menggunakan Teknologi MERN Stack |
| Nama | : | Yogi Prasetya |
| Program Studi | : | Teknik Informatika |

Salah satu unsur dari “Industri 4.0” itu sendiri adalah diimplementasikannya teknologi virtual reality pada bidang-bidang industri. Teknologi virtual reality ini merupakan teknologi yang berfungsi untuk mensimulasikan suatu lingkungan layaknya lingkungan nyata dan sebenarnya oleh computer atau biasa disebut *computer simulated environment*. Smarteye merupakan perusahaan yang menawarkan produk salah satunya virtual tour. Produk virtual tour pada Smarteye menggunakan device agar pengguna dapat berinteraksi dengan suatu lingkungan tanpa harus berada langsung di lingkungan tersebut. Device yang mahal, aplikasi masih lokal dan setting konten yang sulit sehingga dilakukan aplikasi manajemen konten untuk virtual tour berbasis web menggunakan teknologi MERN Stack untuk mengatasi masalah produk virtual tour perusahaan Smarteye.

**Kata kunci**: *Sistem Informasi, Virtual Tour, Web*

# ABSTRACT

|  |  |  |
| --- | --- | --- |
| Title | : | Content Management Application for Web Based Virtual Tour Using MERN Stack Technology |
| Name | : | Yogi Prasetya |
| Study Program | : | Teknik Informatika |

*One of the elements of "Industry 4.0" itself is the implementation of virtual reality technology in industrial fields. This virtual reality technology is a technology that functions to simulate an environment like the actual environment by a computer (computer simulated environment). Smarteye is a company that offers products, one of which is a virtual tour. The virtual tour product on Smarteye uses a device so that users can interact with an environment without having to be directly in that environment. The device is expensive, the application is still local and the content settings are difficult, so the development of a content management application for web-based virtual tours using MERN Stack technology is to solve the problem of the Smarteye company's virtual tour products.*

***Keywords****: Information Systems, Virtual Tour, Web*

DAFTAR ISI

[HALAMAN PERNYATAAN KEASLIAN i](#_Toc81640179)

[HALAMAN PENGESAHAN TUGAS AKHIR ii](#_Toc81640180)

[HALAMAN PERSETUJUAN PUBLIKASI KARYA ILMIAH iii](#_Toc81640181)

[KATA PENGANTAR iv](#_Toc81640182)

[ABSTRAK v](#_Toc81640183)

[ABSTRACT vi](#_Toc81640184)

[DAFTAR GAMBAR x](#_Toc81640185)

[BAB 1 PENDAHULUAN 1](#_Toc81640186)

[1.1 Latar Belakang Masalah 1](#_Toc81640187)

[1.2 Identifikasi Masalah 2](#_Toc81640188)

[1.3 Tujuan Tugas Akhir 2](#_Toc81640189)

[1.4 Manfaat Tugas Akhir 3](#_Toc81640190)

[1.5 Lingkup Tugas Akhir 3](#_Toc81640191)

[1.6 Kerangka Berpikir 3](#_Toc81640192)

[1.7 Sistematika Penulisan Tugas Akhir 4](#_Toc81640193)

[BAB 2 LANDASAN TEORI 6](#_Toc81640194)

[2.1 Teori Perancangan 6](#_Toc81640195)

[2.1.1 Pengertian Perancangan Sistem 6](#_Toc81640196)

[2.2 Model *Waterfall* 7](#_Toc81640197)

[2.3 *Virtual Reality* 8](#_Toc81640198)

[2.3.1 Pengertian *Virtual Reality* 8](#_Toc81640199)

[2.3.2 Pengertian *Virtual Tour* 9](#_Toc81640200)

[2.4 *Framework* 9](#_Toc81640201)

[2.5 MERN Stack (MongoDB, Express, React, Node) 10](#_Toc81640202)

[2.6 MongoDB 11](#_Toc81640203)

[2.7 Express 11](#_Toc81640204)

[2.8 React 11](#_Toc81640205)

[2.9 NodeJS 12](#_Toc81640206)

[2.10 NPM (Node Package Manager) 13](#_Toc81640207)

[2.11 JavaScript 13](#_Toc81640208)

[2.12 JSON 13](#_Toc81640209)

[2.13 Web 14](#_Toc81640210)

[2.14 Multimedia 14](#_Toc81640211)

[1.15 Media 360 Derajat 14](#_Toc81640212)

[2.16 *Functional Programming* 15](#_Toc81640213)

[2.17 NoSQL 15](#_Toc81640214)

[2.18 Unified Modelling Language 16](#_Toc81640215)

[BAB 3 METODE 21](#_Toc81640216)

[3.1 Rencana Penelitian 21](#_Toc81640217)

[3.2 Objek Penelitian 21](#_Toc81640218)

[3.2.1 Visi dan Misi Smarteye 22](#_Toc81640219)

[3.3 Teknik Pengumpulan Data 22](#_Toc81640220)

[3.4. Metode Pengembangan Sistem 23](#_Toc81640221)

[3.5 Tahapan dan Diagram Alir Penelitian 24](#_Toc81640222)

[BAB 4 HASIL DAN PEMBAHASAN 26](#_Toc81640223)

[**4.1.** **Perancangan Sistem** 26](#_Toc81640224)

[4.1.1. Gambaran Umum Aplikasi 26](#_Toc81640225)

[4.2.2. Perancangan Prosedur 26](#_Toc81640226)

[4.2 Spesifikasi Minimum Foto 27](#_Toc81640227)

[4.3 Pemodelan Diagram 27](#_Toc81640228)

[4.2.1 Activity Diagram Pendaftaran Akun 27](#_Toc81640229)

[4.2.2 Activity Diagram Login 28](#_Toc81640230)

[4.2.3 Activity Diagram Buat *Project* Baru 29](#_Toc81640231)

[4.2.4 Activity Diagram Tambah Room 30](#_Toc81640232)

[4.2.5 Activity Diagram Tambah Pintu Antar Ruang 31](#_Toc81640233)

[4.2.6 Activity Diagram *Tour View* 32](#_Toc81640234)

[4.3 Use Case Diagram 33](#_Toc81640235)

[4.4 Class Diagram 34](#_Toc81640236)

[4.5 Desain *Database* 34](#_Toc81640237)

[4.5 Hasil Implementasi Sistem 36](#_Toc81640238)

[BAB V KESIMPULAN 40](#_Toc81640239)

[5.1 Kesimpulan 40](#_Toc81640240)

[1. Implementasi 40](#_Toc81640241)

[2. Alternatif *VR Headset* 40](#_Toc81640242)

[5.2 Saran 41](#_Toc81640243)

[Daftar Pustaka 42](#_Toc81640244)

[Lampiran 1 Daftar Riwayat Hidup 45](#_Toc81640245)

[Lampiran 1 Keterangan Penelitian 46](#_Toc81640246)

# DAFTAR GAMBAR

[Gambar 1 Konsep MERN Stack 10](#_Toc71360077)

[Gambar 2 Waterfall Model 23](#_Toc71360078)

[Gambar 3 Activity diagram pendaftaran akun 28](#_Toc71360079)

[Gambar 4 Activity diagram login 29](#_Toc71360080)

[Gambar 5 Activity diagram buat project baru 30](#_Toc71360081)

[Gambar 6 Activity diagram tambah room 31](#_Toc71360082)

[Gambar 7 Activity diagram tambah pintu antar ruang 32](#_Toc71360083)

[Gambar 8 Activity diagram tour view 33](#_Toc71360084)

[Gambar 9 Use Case Diagram 33](#_Toc71360085)

[Gambar 10 Class diagram 34](#_Toc71360086)

[Gambar 11 Tampilan login 36](#_Toc71360087)

[Gambar 12 Halaman regristasi 37](#_Toc71360088)

[Gambar 13 Tampilan dashboard 37](#_Toc71360089)

[Gambar 14 Tampilan jelajah ruang 38](#_Toc71360090)

[Gambar 15 Tampilan room editor 38](#_Toc71360091)

[Gambar 16 Menu konfigurasi pintu 39](#_Toc71360092)

[Gambar 17 Tampilan View 39](#_Toc71360093)

# BAB 1 PENDAHULUAN

## 1.1 Latar Belakang Masalah

Sejak beberapa tahun terakhir, perusahaan industri telah melakukan sejumlah inisiatif untuk memigrasikan organisasinya ke era yang disebut "Revolusi Industri 4.0" yang mewujudkan gagasan bahwa teknologi digital secara fundamental mengubah aktivitas kerja pada berbagai industri dan kegiatan usaha.

Istilah “Revolusi Industri 4.0” sering digaungkan pemangku kebijakan, utamanya pada lima sektor prioritas Kementerian Perindustrian Republik Indonesia, seperti industri makanan dan minuman, industri otomotif, industri elektronik, industri kimia serta industri tekstil dan produk tekstil.

Salah satu unsur dari “Industri 4.0” itu sendiri adalah diimplementasikannya teknologi *virtual reality* pada bidang-bidang industri. Teknologi *virtual reality* ini merupakan teknologi yang berfungsi untuk mensimulasikan suatu lingkungan layaknya lingkungan yang sebenarnya oleh komputer (*computer simulated environment*). Dengan teknologi *virtual reality* ini pengguna dapat berinteraksi dengan suatu lingkungan tanpa harus berada langsung di lingkungan tersebut.

Adanya teknologi *virtual reality* ini memberikan dampak positif dalam dunia kerja seperti simulasi pelatihan, media pembelajaran, hingga digitalisasi media pemasaran.

Namun dalam penggunaannya, teknologi *virtual reality* sangat bergantung pada beberapa hal dan sulit terpisahkan. Beberapa hal di antaranya yaitu membutuhkan *device* khusus yang harganya tidak murah dan tidak praktis diaplikasikan pada industri tertentu, dari sisi teknis membutuhkan pihak ketiga yang ahli untuk mengimplementasikan, serta sulitnya untuk mengelola konten yang akan ditampilkan, terutama untuk kebutuhan *virtual tour* yang memerlukan perubahan konten yang cepat, mudah dan dinamis. Pada beberapa kebutuhan, memerlukan solusi “virtual reality” yang lebih mudah diakses dari berbagai perangkat di mana pun dan kapan pun, serta memberikan kemudahan dalam mengembangkan konten di dalam *virtual reality* tersebut.

Smarteye merupakan perusahaan yang menawarkan produk salah satunya virtual tour yang dalam penggunaannya membutuhkan *device*. Dengan produk *virtual tour* ini user dapat merasakan sensasi visual dengan *view* 360 derajat dan memiliki *experience virtual reality* seperti klik, audio, FX, pindah ruangan dan lain-lain. Produk *virtual tour* yang ditawarkan perusahaan Smarteye dalam penggunaannya masih memiliki kekurangan seperti membutuhkan perangkat khusus yang mahal, aplikasi masih lokal dan *setting* konten yang sulit, serta masih adanya batasan di ruang dan waktu. Batasan ruang dan waktu ini membuat user yang ingin menggunakan *virtual tour* ini harus datang ke lokasi langsung sehingga kurang digitalisasi. Sedangkan *virtual reality* sendiri merupakan sebuah teknologi yang sangat erat dengan *immersive teknologi*.

Berdasarkan berbagai masalah di atas, maka dilakukan penelitian tentang “**Aplikasi Manajemen Konten untuk Virtual Tour Berbasis Web Menggunakan Teknologi MERN Stack**”. Penulis berusaha mencoba memberikan solusi dengan teknologi web untuk implementasi *virtual tour* yang mudah diakses dari berbagai perangkat dan memungkinkan pengguna untuk mengelola kontennya dengan mudah dan dinamis.

## 1.2 **Identifikasi Masalah**

Berdasarkan latar belakang masalah di atas maka dibuatlah identifikasi masalah, yaitu:

1. Bagaimana mengembangkan teknologi *virtual tour* yang dapat diakses dari berbagai perangkat?
2. Apa solusi yang dapat memudahkan pengembangan konten dalam *virtual tour*?

## 1.3 Tujuan Tugas Akhir

Tujuan yang ingin dicapai dalam penulisan tugas akhir ini adalah sebagai berikut:

1. Menghasilkan prototype aplikasi manajemen konten untuk *virtual tour* berbasis web.
2. Memberikan gambaran dalam perancangan aplikasi manajemen konten untuk *virtual tour* berbasis web.
3. Memaparkan hasil rancangan.

## 1.4 Manfaat Tugas Akhir

Merujuk pada tujuan penelitian di atas, maka pengembangan aplikasi ini diharapkan dapat memberikan beberapa manfaat, yaitu:

1. Memberikan solusi atas terbatasnya teknologi virtual reality dalam hal ketergantungan pada perangkat khusus.
2. Memudahkan pengguna dalam mengembangkan konten yang ditampilkan pada virtual reality.

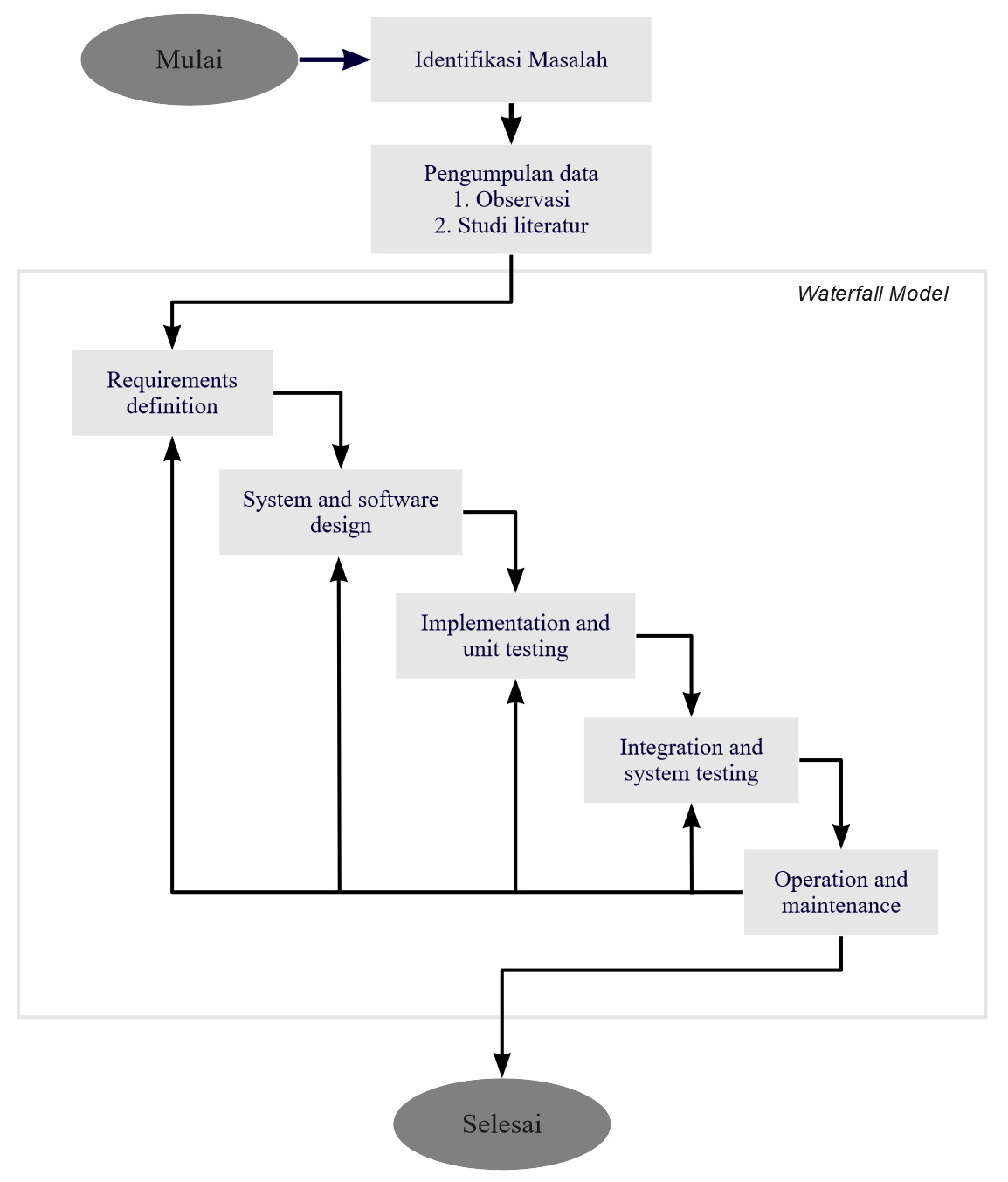
## 1.5 Lingkup Tugas Akhir

Berdasarkan identifikasi masalah yang ditemukan, batasan masalah pada penelitian ini adalah sebagai berikut :

1. Aplikasi dapat menghasilkan *virtual tour* dan dapat diakses pada berbagai perangkat dengan teknologi web.
2. Aplikasi mengelola konten dalam *virtual tour* secara cloud.
3. Perancangan aplikasi hanya mencakup pengaturan konten *virtual tour* menggunakan foto 360 hingga menampilkan hasil *virtual tour*.

## 1.6 Kerangka Berpikir

Dalam penyusunan skripsi ini penulis menceritakan alur penelitian, yang digambarkan dengan kerangka berpikir seperti pada gambar 1.1.



## 1.7 Sistematika Penulisan Tugas Akhir

Sistematika penulisan dapat dibagi menjadi lima bagian, yaitu awal, isi, dan akhir. Berikut adalah sistematika penulisannya :

BAB I PENDAHULUAN

Dalam bab ini membahas tentang Latar Belakang, Rumusan Masalah, Tujuan Penelitian, Batasan Masalah, Manfaat, Metodologi Penelitian serta Sistematika Penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini menjelaskan teori-teori yang digunakan sebagai dasar perancangan desain. Teori-teori tersebut diambil dari literatur yang sesuai dengan permasalahan.

BAB III METODE

Pada bab ini dijelaskan mengenai gambaran metodologi penelitian analisa dan pengembangan aplikasi yang akan dibuat.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini membuat implementasi meliputi implementasi sistem dan implementasi aplikasi, hasil pengujian aplikasi meliputi skenario pengujian, hasil pengujian dan pengujian fungsional.

BAB V KESIMPULAN DAN SARAN

Bab ini mengemukakan kesimpulan yang diambil berdasarkan sistem yang telah dibuat sesuai rancangan dan beberapa saran untuk kepentingan lebih lanjut agar menjadi lebih baik lagi.

# 

# BAB 2 LANDASAN TEORI

## 2.1 Teori Perancangan

Perancangan adalah sebuah proses untuk mendefinisikan sesuatu yang akan dikerjakan dengan menggunakan teknik yang bervariasi serta didalamnya melibatkan deskripsi mengenai arsitektur serta detail komponen dan juga keterbatasan yang akan dialami dalam proses pengerjaannya (Soetam Rizky, 2011).

Menurut Sandro Sembiring (2013:45), “Perancangan adalah analisis sistem, persiapan untuk merancang dan implementasi agar dapat menyelesaikan apa yang harus diselesaikan serta mengkonfigurasikan komponen-komponen perangkat lunak ke perangkat keras”.

Dalam jurnal, Dani Anggoro, dkk (2015:2) mengungkapkan bahwa “Perancangan adalah upaya untuk mengkontruksi sebuah sistem yang memberikan kepuasan akan spesifikasi kebutuhan fungsional, 7 memenuhi target, memenuhi kebutuhan dari segi performasi maupun penggunaan sumber daya, kepuasan balasan pada proses desain dari segi biaya, waktu dan perangkat”.

### 2.1.1 Pengertian Perancangan Sistem

Perancangan sistem adalah penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen terpisah ke dalam satu kesatuan yang utuh.

Menurut Satzinger, Jackson dan Burd (2012:5) perancangan sistem adalah sekumpulan aktivitas yang menggambarkan secara rinci bagaimana sistem akan berjalan. Hal itu bertujuan untuk menghasilkan produk perangkat lunak yang sesuai dengan kebutuhan user.

Adi Nurgroho dalam jurnal (Syahrial dan Sharipuddin, (2016:3), mengungkapkan bahwa “Perancangan sistem adalah hasil dari tahapan analisis yaitu model analisis yang menyediakan rincian pemahaman tentang spesifikasi kebutuhan pengguna”.

## 2.2 Model *Waterfall*

Menurut Ian Sommerville (2011, p30), metode waterfall memiliki tahapan utama dari *waterfall* model yang mencerminkan aktivitas pengembangan dasar. Terdapat 5 (lima) tahapan pada metode Waterfall, yaitu *requirement analysis and definition*, *system and software design*, *implementation and unit testing*, *integration and system testing*, dan *operation and maintenance*.

Adapun penjelasan dari tahapan-tahapan metode waterfall menurut Ian Sommerville tersebut sebagai berikut.

1). ***Requirement Analysis and Definition***

*Requirement Analysis and Definition* adalah tahapan penetapan fitur, kendala dan tujuan sistem melalui konsultasi dengan pengguna sistem. Semua hal tersebut akan ditetapkan secara rinci dan berfungsi sebagai spesifikasi sistem.

2). ***System and Software Design***

Pada Tahap *System and Software Design* ini akan dibentuk suatu arsitektur sistem berdasarkan persyaratan yang telah ditetapkan. Selain itu juga, dilakukan identifikasi dan penggambaran terhadap abstraksi dasar sistem perangkat lunak serta hubungan-hubungannya.

3). ***Implementation and Unit Testing***

Dalam tahapan *Implementation and Unit Testing* ini, hasil dari desain perangkat lunak akan direalisasikan sebagai satu set program atau unit program. Setiap unit akan diuji apakah sudah memenuhi spesifikasinya.

4). ***Integration and System Testing***

Dalam tahap *Integration and System Testing* ini, setiap unit program akan diintegrasikan satu sama lain dan diuji sebagai satu sistem yang utuh untuk memastikan sistem sudah memenuhi persyaratan yang ada. Setelah itu sistem akan dikirim ke pengguna sistem.

**5). *Operation and Maintenance***

Dalam tahap Operation and Maintenance ini, sistem diinstal dan mulai digunakan. Selain itu juga memperbaiki error yang tidak ditemukan pada tahap pembuatan. Dalam tahap ini juga dilakukan pengembangan sistem seperti penambahan fitur dan fungsi baru.

## 2.3 *Virtual Reality*

### 2.3.1 Pengertian *Virtual Reality*

1. (Hillis, 1999: xv)

Virtual Reality menyatukan dunia teknologi dan kemampuannya untuk mempresentasikan alam, dengan bidang yang luas dan tumpang tindih mengenai hubungan sosial dan makna.

2. (definisi-definisi Krieger, 1986; Benedikit, 1991; Biocca, 1992; Robinett, 1992; Pinsky, 1993)

Dalam sebuah jurnal yang disusun oleh Muhammad Jamil (2018), “Virtual reality (VR) merupakan simulasi gambar atau seluruh lingkungan yang dihasilkan komputer yang dapat dialami menggunakan peralatan elektronik khusus, yang memungkinkan penggunanya “hadir” di lingkungan alternatif seperti di dunia nyata terhadap objek dan informasi virtual tiga dimensi (3D) dengan data tambahan seperti grafik atau suara. Bentuknya berupa video 360 ° yang menangkap seluruh adegan di mana pengguna dapat melihat ke atas, bawah, dan sekitarnya dan memungkinkan pengguna berinteraksi dengan objek fisik dan virtual.”

Virtual reality didefinisikan secara luas sebagai suatu simulasi yang dihasilkan komputer atau presentasi dari lingkungan di mana pengguna mengalami rasa kehadiran fenomenologis atau keterlibatan dalam lingkungan.

### 2.3.2 Pengertian *Virtual Tour*

Dalam sebuah jurnal yang disusun oleh Osman, Wahab dan Ismail (2009, p173), *Virtual Tour* merupakan teknologi yang menempatkan user di dalam gambar dan memungkinkan *user* untuk meningkatkan kesadaran situasional serta meningkatkan daya lihat, tangkap dan menganalisa data *virtual* secara signifikan.

*Virtual tour* adalah sebuah simulasi dari suatu lingkungan nyata, biasanya terdiri dari kumpulan foto-foto panorama, kumpulan gambar yang terhubung oleh hyperlink, ataupun video, atau virtual model dari lokasi yang sebenarnya, serta dapat menggunakan unsur-unsur multimedia lainnya seperti efek suara, musik, narasi, dan tulisan (Valentina & Handjojo, 2013).

*Virtual Tour* merupakan sebuah simulasi dari sebuah lokasi yang terdiri dari rentetan. Rentetan gambar tersebut akan digabungkan (*stitch*) untuk menghasilkan foto panorama 360 derajat. *Virtual Tour* sendiri biasanya digunakan untuk memberi pengalaman “pernah berada” di suatu tempat hanya dengan melihat layar monitor. Penyajian virtual tour dapat dilakukan dengan cara memanfaatkan gambar ataupun *video*, selain itu dapat menggunakan model 3 dimensi. Untuk penyajian dengan menggunakan gambar, dapat digunakan foto panorama. Pemilihan jenis foto panorama juga mempengaruhi hasil *virtual tour* yang dihasilkan. Untuk panorama jenis *cylindrical*, bagian vertikalnya hanya dapat menangkap tidak lebih dari 180 derajat sedangkan jenis spherical, memungkinkan untuk melihat ke atas dan ke bawah (Highton, 2010).

## 2.4 *Framework*

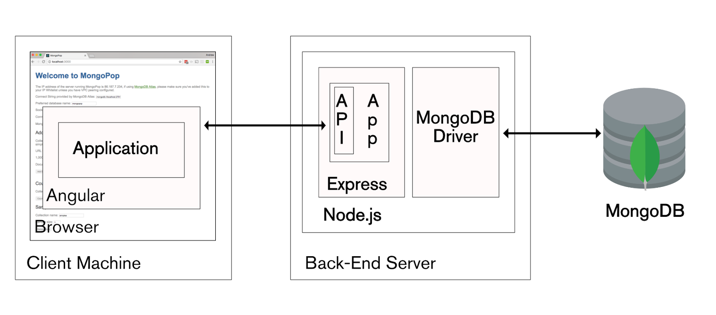
*Framework* adalah sekumpulan *script* atau fungsi dasar yang telah dibuat agar developer tidak perlu membuat suatu aplikasi dari nol sehingga developer dapat mempersingkat waktu dalam hal membuat suatu aplikasi (Beta Sidik, 2012).

Dengan kata lain, framework merupakan sekumpulan kode atau script dasar yang dibuat sedemikian rupa dengan memiliki tujuan agar developer dapat menggunakan kode atau *script* tersebut secara efisien dalam membuat atau mengembangkan suatu aplikasi.

## 2.5 MERN Stack (MongoDB, Express, React, Node)

Pembuatan website biasanya dibuat dengan beberapa teknologi menjadi satu. Kombinasi dari beberapa teknologi ini yang biasa disebut dengan “stack”. Dipopulerkan oleh LAMP stack yang merupakan akronim untuk Linux, Apache, MySQL, dan PHP yang semuanya merupakan komponen open-source.

Sebagai web yang pengembangan menjadi matang dan interaktivitas muncul kedepan yaitu Single Page Applications (SPA) menjadi lebih populer. SPA merupakan suatu paradigma aplikasi web yang menghindari *refresh* pada halaman web untuk menampilkan konten baru. Cara yang digunakan adalah melakukan panggilan ke server untuk mendapatkan beberapa data untuk memperbarui halaman web. Hasilnya terlihat cukup bagus jika dibandingkan dengan cara lama yang melakukan *refresh* halaman secara keseluruhan. Hal ini membawa peningkatan dalam kerangka front-end (Subramanian, 2019).



Gambar 1 Konsep MERN Stack

Sumber : (Andrew Morgan, 2017)

Pada Gambar 2.1 merupakan gambar konsep dari cara kerja MERN stack dimana React Js akan berkomunikasi dengan melakukan request terhadap Node Js yang menggunakan framework Express. Node Js akan menghubungkan back-end server dengan MongoDB yang digunakan sebagai database lalu memberikan respon dalam bentuk JSON terhadap front-end.

## 2.6 MongoDB

MongoDB adalah penyimpanan data yang powerful, flexible, dan scalable dalam basis data. MongoDB menggabungkan kemampuan untuk memperkecil skala dengan fitur seperti indeks sekunder, range queries, pengurutan, agregasi, dan indeks geospasial.

MongoDB merupakan document-oriented database, bukan merupakan database yang relasional. Alasan utama untuk bergerak jauh dari relasional adalah untuk membuat skala lebih mudah dan meningkatkan performa (Bradshaw et al., 2019).

## 2.7 Express

Express Js merupakan framework yang berada di atas fungsi server web Node.js untuk menyederhanakan API dan menambahkan fitur baru yang bermanfaat. Membuat lebih mudah untuk mengatur fungsionalitas aplikasi dengan middleware dan routing. Hal tersebut menambah manfaat utilitas untuk objek HTTP Node.js, memfasilitasi rendering tampilan HTML dinamis dan mendefinisikan standar ekstensibilitas yang mudah diimplementasikan (Hahn, 2016).

## 2.8 React

React, sering ditulis juga React.js atau ReactJS merupakan JavaScript *library* yang dikembangkan oleh Facebook untuk memfasilitasi pembuatan daripada komponen antarmuka yang interaktif, *stateful*, serta mudah untuk digunakan ulang. ReactJS sangatlah cocok digunakan untuk *rendering* antarmuka yang kompleks dengan performa tinggi (Kumar & Singh, 2016).

JavaScript *library* ini sendiri telah digunakan oleh Facebook untuk bagian *newsfeed* mereka. Selain itu, banyak situs-situs terkenal juga yang menggunakan ReactJS ini sebagai salah satu alat yang digunakan dalam mengembangkan situs tersebut, seperti Netflix, Paypal, Vevo, dan masih banyak lagi (M & Sonpatki, 2016). Hal ini membuktikan bahwa ReactJS merupakan salah satu *library* JavaScript yang sangat berkembang, banyak digunakan, serta sangat handal dalam melaksanakan tugasnya sebagai UI-*rendering* JavaScript *library*.

React merupakan bagian view dari konsep MVC (*model-view-controller*) (Kumar & Singh, 2016) yang berarti React hanya mengurusi bagian tampilan antarmuka dengan pengguna saja, tanpa mengurusi bagian cara mendapatkan data ataupun hubungan ke basis data.

Dalam pengembangannya, banyak sekali hal yang telah dapat dijalankan oleh React seperti menganimasikan suatu objek dengan efek transisi, menjalankan permainan di *web browser* yang sepenuhnya diprogram dengan menggunakan React, ataupun validasi form yang berjalan secara *real-time* sembari pengguna mengisikan data pada form tersebut. Pemakaian ReactJS dalam sebuah situs dapat dilihat dengan menggunakan alat tambahan pada Google Chrome yang bernama React Developer Tools yang dapat mendeteksi keberadaan atau penggunaan ReactJS dalam suatu *web* (M & Sonpatki, 2016).

## 2.9 NodeJS

Node.js® merupakan salah satu piranti pengembang yang bisa digunakan untuk membuat aplikasi berbasis Cloud. Node.js dikembangkan dari engine JavaScript yang dibuat oleh Google untuk browser Chrome / Chromium(V8) ditambah dengan lib UV serta beberapa pustaka internal lainnya. Dengan menggunakan Node.js, semua pengembangan akan dilakukan menggunakan JavaScript, baik pada sisi klien maupun server (Purnomosidi, 2013).

Node.js adalah platform perangkat lunak pada sisi peladen dan aplikasi jaringan. Ditulis dengan bahasa JavaScript dan dijalankan pada Windows, Mac OS X, dan Linux tanpa perubahan kode program. Node.js memiliki pustaka peladen HTTP sendiri sehingga memungkinkan untuk menjalankan peladen web tanpa menggunakan program peladen web seperti Apache atau Lighttpd.

## 2.10 NPM (Node Package Manager)

NPM adalah sebuah manajer paket untuk bahasa pemrograman JavaScript yang menjadi manajer baku untuk lingkungan *runtime* Node.js. NPM terdiri dari klien baris perintah yang disebut npm, dan sebuah pangkalan data daring yang berisi paket publik dan paket pribadi berbayar, yang disebut npm registry. Registry dapat diakses melalui klien, dan ketersediaan paket dapat dicari melalui situs web npm. Manajer paket dan registry dikelola oleh npm, Inc (Wikipedia, 2019).

## 2.11 JavaScript

JavaScript adalah bahasa *script* berdasarkan pada objek yang memperbolehkan pemakai untuk mengendalikan banyak aspek interaksi pemakai (sering disebut DOM) pada satu dokumen HTML. Objek tersebut dapat berupa suatu windows, frame, URL, dokumen, form, button, atau item yang lain. Terdapat dua piranti yang diperlukan dalam JavaScript yaitu browser dan text editor. Text editor adalah sebuah pengolah kata (word processor) yang menghasilkan file dalam format ASCII murni (Suryana & Koesheryatin, 2014).

JavaScript adalah bahasa scripting client side yang sangat populer. Hampir semua programmer web menggunakan JavaScript untuk memberi efek pemrograman di halaman. JavaScript tidak hanya berdiri sendiri, tapi JavaScript juga menjadi dasar yang bisa digunakan untuk teknologi lainnya, seperti Ajax, jQuery dan jQuery Mobile. JavaScript merupakan bahasa pemrograman paling populer di dunia. Ini karena JavaScript bisa dipakai di HTML, web, untuk server, PC, laptop, tablet, ponsel dan lainnya (Winarno dkk., 2014).

## 2.12 JSON

Javascript Object Notation(JSON) adalah sebuah general-purpose data encoding format yang populer. Penerapan JSON telah banyak digunakan pada database dan web service. Struktur dokumen JSON secara opsional dapat dibatasi berdasarkan skema yang terdiri atas dua hal yakni map (pemetaan struktur nilai berdasarkan klasifikasi jenisnya) dan list (pengelompokan nilai berdasarkan klasifikasi jenisnya) (Kleppmann & Beresford, 2017).

## 2.13 Web

World Wide Web (WWW) dapat juga disebut web adalah sistem informasi dan komunikasi *Hypertext* yang sangat populer digunakan pada jaringan komputer internet yang mampu menampilkan informasi secara grafis dan interaktif (Purbo, 2006).

Web merupakan sistem layanan informasi di internet yang berbasis grafis dan didasarkan pada konteks *hypertext*. Konsep *hypertext* ini sangat mirip dengan teks biasa kecuali satu aspek yang penting yaitu memungkinkan untuk konteks (link) di dalam dokumen itu sendiri atau koneksi ke dokumen lain.

## 2.14 Multimedia

Multimedia dapat memiliki arti sebagai penggunaan sejumlah media berbeda yang disatukan sebagai alat untuk menyampaikan pesan atau informasi dalam bentuk teks, grafis, audio, animasi maupun video (Rusman, dkk. 2011).

Menurut Munir, (2012:2) mendefinisikan bahwa “Multimedia merupakan penyatuan dua atau lebih media komunikasi seperti teks, grafik, animasi, audio, dan video dengan ciri-ciri interaktivitas komputer untuk menghasilkan satu presentasi yang menarik”.

1.15 Media 360 Derajat

Media 360° berupa 360° video dan gambar, merupakan cara yang bagus bagi pengembang untuk meningkatkan aplikasi tradisional dengan konten yang mendalam. Pengembang atau pengguna juga dapat membuat media 360° dalam format mono atau stereo. Gambar dan video umumnya perlu disimpan dalam format equirectangular-panoramic (equirect-pano), yang merupakan format umum yang didukung oleh banyak solusi pengambilan (Google, 2019).

## 2.16 *Functional Programming*

Pemrograman fungsional dimana program dibangun dengan menerapkan dan menyusun fungsi. Ini adalah paradigma pemrograman deklaratif di mana definisi fungsi adalah pohon ekspresi yang masing-masing mengembalikan nilai, bukan urutan pernyataan imperatif yang mengubah status program (Wikipedia, 2020).

Dalam pemrograman fungsional, fungsi diperlakukan sebagai *first-class citizen*, yang berarti bahwa mereka dapat terikat ke nama (termasuk pengenal lokal), diteruskan sebagai argumen, dan dikembalikan dari fungsi lain, seperti yang bisa dilakukan tipe data lainnya. Hal ini memungkinkan program untuk ditulis dalam gaya deklaratif dan dapat disusun, di mana fungsi-fungsi kecil digabungkan secara modular.

## 2.17 NoSQL

NoSQL memiliki pendekatan non-relasional. Menurut (Neal Leavitt, 2010), jenis non-relational database seperti hierarchical, graph, dan object oriented sudah ditemukan sejak akhir tahun 1960-an. Namun, nama NoSQL sendiri baru dicetuskan pada awal abad ke-21, dipicu oleh kebutuhan perusahaan-perusahaan Web 2.0, yaitu perusahaan dengan website yang menekankan keterlibatan antar user seperti website sosial media. Perusahaan Web 2.0 ini memiliki kebutuhan khusus akan infrastruktur dan data yang terus berkembang, seperti Amazon dan Google, untuk mendukung aplikasi big data dan real-time web. Terminologi NoSQL terkadang disebut juga ”Not Only SQL” yang menekankan bahwa NoSQL juga mendukung bahasa yang bisa melakukan query layaknya SQL. Teknologi-teknologi yang dimiliki NoSQL adalah sebagai berikut:

* Document-based stores. Basis data ini menyimpan dan mengelola data sebagai sekumpulan dokumen (collection), daripada sebagai tabel terstruktur dengan ukuran field yang seragam untuk setiap record.
* Key-value stores. Setiap item disimpan sebagai kunci (key) bersamaan dengan nilainya. Sistem ini dapat menangani data terstruktur maupun tidak terstruktur.
* Graph stores, digunakan untuk menyimpan informasi seputar jaringan data seperti koneksi sosial.
* Wide-column stores. Basis data ini teroptimisasi untuk query pada sejumlah dataset yang besar dan menyimpan kolom-kolom data bersamaan daripada sebagai baris.

## 2.18 Unified Modelling Language

*UML* merupakan salah satu bahasa rancangan sistem yang berorientasi objek. dipakai agar dapat menyederhanakan sebuah masalah yang lebih kompleks sehingga lebih mudah dimengerti. *Unified Modelling Language* (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumen-tasikan sistem piranti lunak (Malabay, 2018).

1. ***Usecase Diagram***

*Usecase Diagram* merupakan salah satu model untuk sistem yang akan dibuat. *Usecase Diagram* akan menjelaskan sebuah hubungan aktor dengan sistem yang akan dibangun. Sebuah *Usecase* akan menjelaskan sebuah hubungan antara aktor dengan sistemnya (Nugroho, 2010).

*Tabel 2. 1 Simbol Use Case Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Gambar | Nama | Keterangan |
| 1. |  | *Use Case* | *Use Case* merupakan deskripsi urutan aksi-aksi yang ditampilkan pada sistem yang menghasilkan suatu hasil yang urut atau terstruktur pada suatu *actor*. |
| 2. |  | *Actor* | Suatu *Actor* menggambarkan peran pengguna yang berinteraksi langsung dengan sistem, dan terhubung dengan *use case.* |
| 3. |  | *Association* | *Association* digunakan untuk menghubungkan di antara *Actor* dengan *Use Case*. garis yang |
| 4. |  | *Dependency* | *Dependency* digambarkan dengan sebuah garis panah putus-putus. |
| 5. |  | *Generalization* | *Generalization* merupakan objek data yang berada diatas objek induk |
| 6. |  | *Include* | *Include* menggambarkan suatu hubungan pada *use case* |
| 7. |  | *Extend* | *Extend* memperluas target *use case* |
| 8. |  | *System* | *System* merupakan paket yang menampilkan isi sebuah sistem secara terbatas |
| 9. |  | *Coloration* | *Coloration* biasa digunakan pada interaksi |

1. ***Activity Diagram***

*Activity diagram* yaitu menggambarkan alur kerja dari sebuah sistem. *Activity diagram* akan menggambar suatu aksi sistem. *Activity diagram* mendukung perilaku parallel (Nugroho, 2010). Berikut ini adalah simbol yang ada pada *activity diagram*:

*Tabel 2. 2 Simbol Activity Diagram*

| **Simbol** | **Nama** | **Deskripsi** |
| --- | --- | --- |
|  | Status awal | Merupakan status awal yang dimiliki diagram aktivitas. |
|  | *Activity* | Proses *activity* yang dilakukan sistem. |
|  | Penggabungan | Menggabungkan beberapa program dalam satu aktivitas. |
|  | Percabangan | Dimana jika ada pilihan aktivitas lebih dari satu dalam program disebut Asosiasi percabangan. |
|  | Status Akhir | Status akhir yang dilakukan system dimana sebuah diagram aktivitas memiliki sebuah status akhir. |
|  | *Swimlane* | Pemisah organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |

1. ***Sequence Diagram***

*Sequence Diagram* yaitu gambaran sebuah objek pada *usecase* dengan menjelaskan waktu dari objek dan pesan yang diterima antar objek (Nugroho, 2010). Berikut adalah simbol yang terdapat pada *Sequence Diagram*:

*Tabel 2. 3 Simbol Sequence Diagram*

| **Simbol** | **Deskripsi** |
| --- | --- |
| Aktor/Actor | orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi |
| Garis hidup / *lifeline* | Menyatakan kehidupan suatu objek |
| Objek | Menyatakan objek yang berinteraksi pesan |
| Waktu aktif | Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya |
| Pesan tipe *create* | Menyatakan suatu objek membuat objek lain, arah panah mengarah pada objek yang dibuat |
| Pesan tipe *call* | Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri |
| Pesan tipe *send* | Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lain, |

1. ***Class Diagram***

*Class Diagram* yaitugambaran dari struktur sistem yang akan dibuat untuk membangun sebuah sistem. *Class diagram* ini memiliki atribut, metode atau operasi (Nugroho, 2010). Berikut simbol-simbol yang digunakan *class diagram* :

*Tabel 2. 4 Simbol Class Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Gambar | Nama | Keterangan |
| 1. |  | *Class* | *Class* diagram merupakan blok-blok yang digunakan untuk membangun pada pemrograman berorientasi objek |
| 2. |  | *Association* | *Association* merupakan sebuah hubungan relasi umum yang menghubungkan 2 class yang berbeda |
| 3. |  | *Composition* | *Composition* digunakan jika ada sebuah class yang tidak bisa berdiri sendiri dan harus terhubung dengan class lain |
| 4. |  | Dependency | Seringkali sebuah class menggunakan class lain. |
| 5. |  | *Aggregation* | *Aggregation* digunakan untuk mengindikasi keseluruhan bagian dari *relationship,* |

# BAB 3 METODE

## Rencana Penelitian

Penelitian ini dilakukan di perusahaan *startup* internal Telkom *Innovative* yang berfokus pada layanan solusi Virtual dan *Augmented Reality* untuk bisnis dan perusahaan.

Lokasi penelitian ini dilakukan di kantor Smarteye yang beralamat di PT Metra-Net Mulia Business Park, Building J, Jl. Letjen MT Haryono Kav. 58-60 Pancoran, Jakarta Selatan, 12780. Adapun waktu penelitian dilaksanakan pada Oktober 2020.

## 3.2 Objek Penelitian

Pada penelitian ini akan dilakukan analisis dan Aplikasi Manajemen Konten untuk Virtual Tour Berbasis Web Menggunakan Teknologi MERN Stack di perusahaan Smarteye. Sejak 2011, smarteye.id berdedikasi mengembangkan konten dan memanfaatkan teknologi VR & AR untuk bisnis berbagai industri.

Smarteye.id adalah *startup* internal Telkom *Innovative* yang berfokus pada layanan solusi Virtual dan *Augmented Reality* untuk bisnis dan perusahaan. Dibentuk pada 10 Februari 2017 dari program Digital Amoeba, *start-up* yang pertama kali difokuskan pada bisnis *real estate property* untuk mempresentasikan dan menampilkan produk (rumah) mereka secara efektif dan efisien. Smarteye terus berkembang ke beberapa sektor termasuk transportasi, pariwisata dan hiburan.

Dengan menggunakan solusi Realitas Virtual, bisnis dan perusahaan dapat memperoleh peluang maksimum dari wawasan klien dengan menampilkan tampilan produk yang paling realistis dan imersif kepada mereka, tanpa melewati batas anggaran.

### 3.2.1 Visi dan Misi Smarteye

Visi:

Menjadi AR/VR spesialis yang memiliki kemampuan sebagai berikut:

1. A Translator: We translating what client needs
2. Many Services: We have many options for services
3. Customizable: Client can customize based on their needs

Misi:

1. Value Player: Offers lowest prices or tremendous value for similar product and services
2. Platform Services: Operates a shared resource or infrastructure others can utilize
3. Customizer: Uses insight and market intelligence to offer tailored products or services

## 3.3 Teknik Pengumpulan Data

Teknik pengumpulan data yang digunakan dalam penelitian ini adalah:

1. Observasi

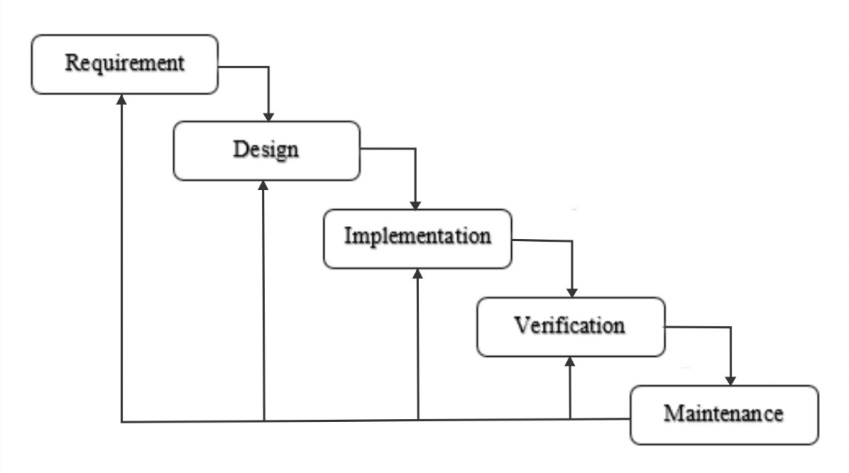
Observasi adalah pengamatan dan pencatatan secara sistematik terhadap unsur-unsur yang tampak dalam suatu gejala atau gejala-gejala pada objek penelitian (H. Hadari Nawawi & H. M. Martini Hadari, 1992). Dalam penelitian ini yang digunakan adalah observasi partisipan yang berarti peneliti ikut serta dalam kegiatan. Dalam teknik ini peneliti mencoba menggunakan produk Virtual Tour di kantor Smarteye.

1. Studi Pustaka

Studi pustaka adalah teknik pengumpulan data dengan mengadakan studi penelaah terhadap buku-buku, literatur-literatur, catatan-catatan, dan laporan-laporan yang ada hubungannya dengan masalah yang dipecahkan (Moh. Nazir, 2013). Dalam penelitian ini menggunakan metode pengumpulan data melalui buku, website dan literatur lainnya yang dibutuhkan untuk penelitian Aplikasi Manajemen Konten untuk Virtual Tour Berbasis Web Menggunakan Teknologi MERN Stack.

## 3.4. Metode Pengembangan Sistem

Pengembangan sistem pada penelitian ini menggunakan metode pengembangan Waterfall seperti terlihat pada gambar 3.1. Waterfall model merupakan salah satu model proses perangkat lunak yang mengambil kegiatan proses dasar seperti spesifikasi, pengembangan, validasi, evolusi, dan merepresentasikannya sebagai fase-fase proses yang berbeda seperti analisis dan definisi persyaratan, perancangan perangkat lunak, implementasi, pengujian unit, integrasi sistem, pengujian sistem, operasi, dan pemeliharaan. Adapun skema & penjelasanya sebagai berikut:



Gambar 2 Waterfall Model

1. Requirement

Mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan perangkat lunak yang akan dibangun.

1. Desain

Pada tahap ini dilakukan pembuatan use case diagram, class diagram, activity diagram, sequence diagram, desain perancangan data, desain perancangan antarmuka.

1. Implementasi

Pada tahap ini desain rancangan sistem yang akan dibuat dan diimplementasikan ke dalam bentuk kode program Javascript.

1. Verification

Setelah proses pengkodean selesai maka akan dilakukan proses verifikasi atau pengujian terhadap program yang dihasilkan untuk mengetahui apakah program yang sudah berjalan dengan benar dan sesuai dengan perancangan yang dilakukan. Teknik pengujian yang dilakukan pada penelitian ini menggunakan pengujian black box.

1. Pemeliharaan

Tahap akhir sesudah perangkat lunak dibangun dengan melakukan pemeliharaan, seperti penyesuaian atau melakukan perubahan sesuai dengan kondisi yang akan datang yang mungkin perlu dilakukan.

## 3.5 Tahapan dan Diagram Alir Penelitian

Dalam penelitian ini terdapat beberapa tahapan yang perlu dilakukan agar dapat berjalan dengan baik, adapun tahapan penelitian ini adalah sebagai berikut:

1. Survei Literatur

Melakukan pengumpulan informasi, tutorial, dan referensi yang relevan berkaitan dengan topik penelitian.

1. Identifikasi Masalah

Melakukan identifikasi masalah untuk menentukan solusi apa yang diperlukan, berkaitan dengan topik penelitian.

1. Perancangan Perangkat Lunak

Melakukan perancangan perangkat lunak yaitu melakukan persiapan software yang akan digunakan, membuat desain dan database.

1. Implementasi

Dari hasil perancangan perangkat lunak maka konversi ke dalam bahasa pemrograman Javascript agar terbentuk sebuah aplikasi sesuai perancangan.

1. Pengujian Perangkat Lunak

Pada tahap ini dilakukan pengujian perangkat lunak secara menyeluruh untuk memastikan fitur-fitur telah berjalan dengan baik sesuai yang diharapkan serta memastikan tidak ada bug ataupun logic error pada.

1. Kesimpulan

Kesimpulan ini adalah tahap akhir sistem yang diharapkan mampu menjawab rumusan masalah yang ada.

# BAB 4 HASIL DAN PEMBAHASAN

* 1. **Perancangan Sistem**

Perancangan sistem dibuat sebagai tahapan untuk mempersiapkan desain agar sesuai dan dapat memenuhi kebutuhan dari pengguna, Proses-proses yang diinginkan oleh pengguna dan nantinya akan dimiliki oleh *system* akan digambarkan secara jelas*.* Menggunakan pendekatan *functional programming*, maka dari itu alat bantu yang digunakan untuk menggambarkan seluruh proses adalah *Unified Model Language.*

### 4.1.1. Gambaran Umum Aplikasi

Aplikasi yang akan dibangun memiliki dua tipe pengguna dimana terdapat *content owner*, dan *viewer*.

Content owner dapat membuat dan mengembangkan konten *virtul tour* dan dapat membagikan tautan atau menyisipkan pada website dan aplikasi mobile.

### 4.2.2. Perancangan Prosedur

Perancangan prosedur yang dikembangkan melakukan beberapa perubahan pada proses manual yang dianggap rumit karena membutuhkan kemampuan pemrograman, prosedur yang diusulkan adalah sebagai berikut:

1. *Content owner* dapat membuat *virtual tour* secara mandiri menggunakan aplikasi berbasis web. Membuat ruang virtual dan pintu yang menghubungkan antara ruang.
2. Setelah konten selesai dibuat oleh *content owner*, virtual tour dapat dibagikan dalam bentuk URL, atau disematkan dalam aplikasi lain, baik web maupun mobile.
3. *Viewer* bisa mengakses konten virtual tour melalui URL yang dibagikan oleh *content owner.* Selain itu, *viewer* dapat menyematkan pada aplikasinya sendiri.

## 4.2 Spesifikasi Minimum Foto

Seperti aplikasi *virtual tour* pada umumnya, *virtual tour* berbasis web ini pun menjadikan foto 360 sebagai bagian utama dari keseluruhan fitur yang disediakan. Untuk mencapai ekspektasi pengguna serta memaksimalkan tujuan aplikasi, maka diperlukan foto dengan kualitas yang baik, namun tetap ringan secara ukuran.

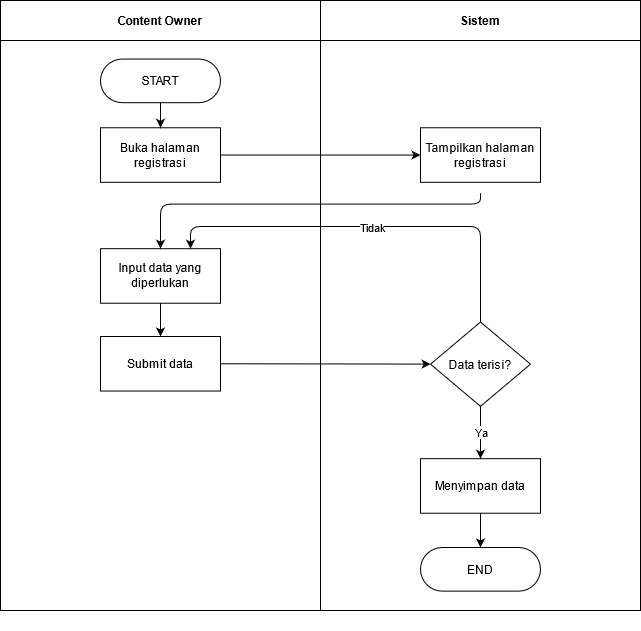
Foto yang dibutuhkan minimal memiliki dimensi 3840 x 1920 pixel, dengan resolusi yang direkomendasikan 72 dpi, untuk yang membutuhkan resolusi diatas 72 dpi, maka diperlukan server dengan spesifikasi yang lebih bagus untuk dapat melayani akses dengan lebih baik.

## 4.3 Pemodelan Diagram

Berdasarkan apa yang dihasilkan dalam pembahasan hasil penelitian, maka alur kerja (workflow) sebuah proses bisnis dan urutan proses aktivitas dalam sistem dapat dimodelkan secara visual.

### 4.2.1 Activity Diagram Pendaftaran Akun

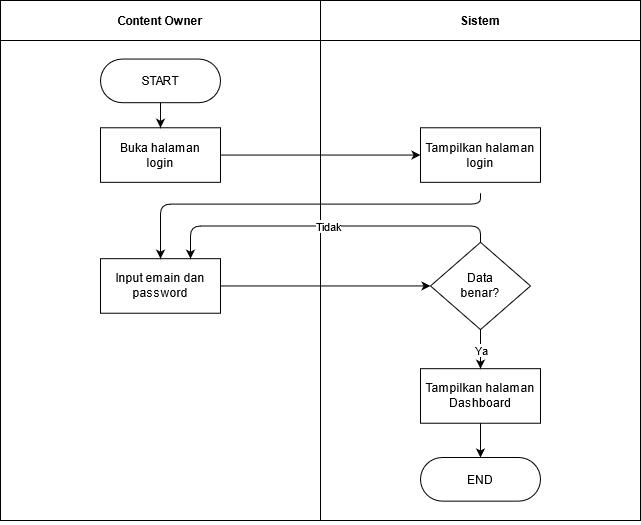
*Activity diagram* pendaftaran akun menjelaskan aktivitas yang ada di dalam *use case* pembuatan akun baru. Aktor yang berperan dalam aktivitas ini adalah *Content owner*. Adapun diagramnya adalah sebagai berikut:



Gambar 3 Activity diagram pendaftaran akun

### 4.2.2 Activity Diagram Login

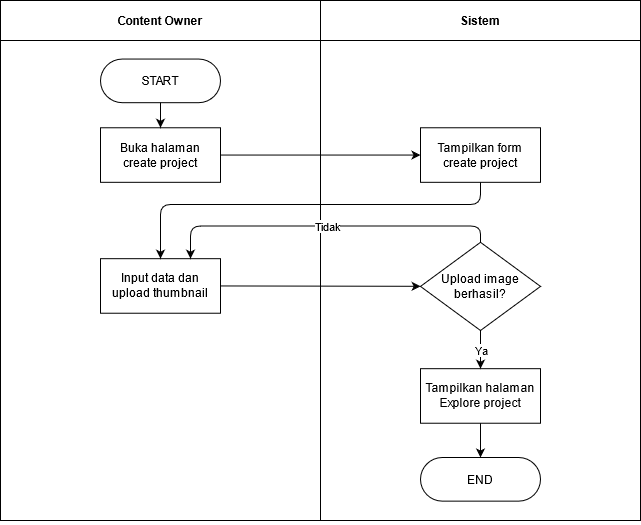
*Activity diagram* ini menjelaskan proses login. Aktor yang berperan dalam aktivitas ini adalah *Content owner*. Adapun diagramnya adalah sebagai berikut:



Gambar 4 Activity diagram login

### 4.2.3 Activity Diagram Buat *Project* Baru

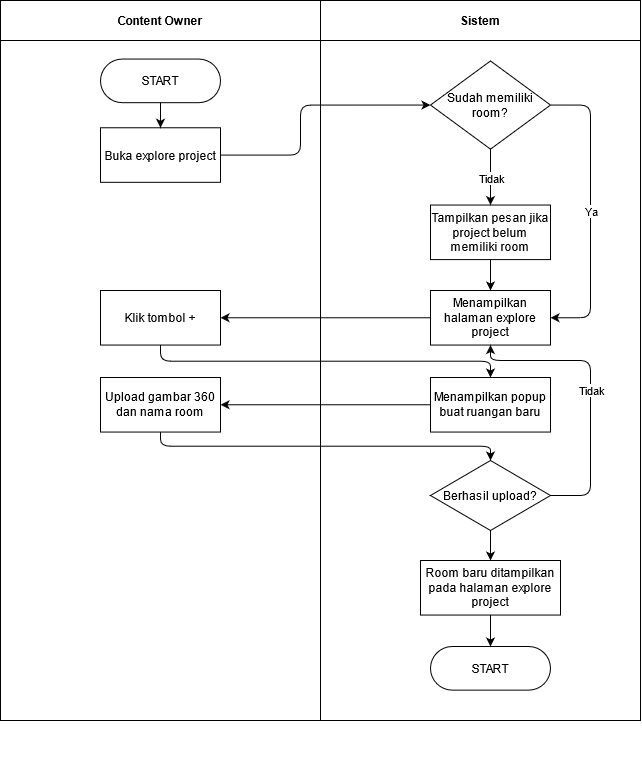
*Activity diagram* buat project baru adalah aktivitas pertama dalam *use case* perancangan konten *virtual*. Aktor yang berperan dalam aktivitas ini adalah *Content owner*. Adapun diagramnya adalah sebagai berikut:



Gambar 5 Activity diagram buat project baru

### 4.2.4 Activity Diagram Tambah Room

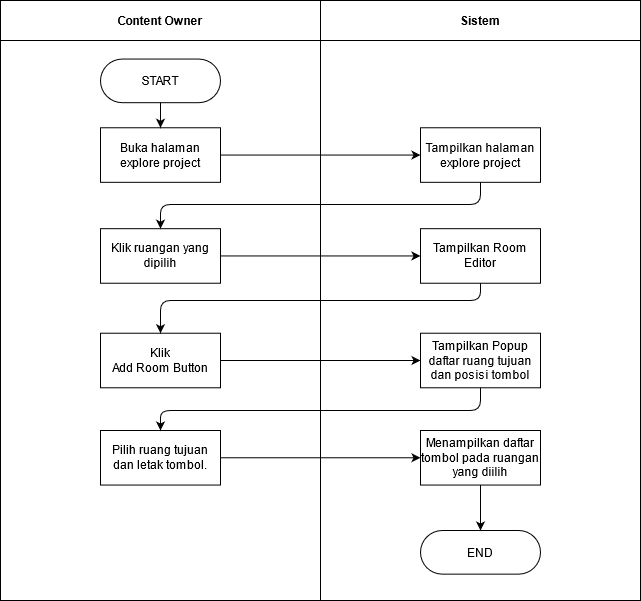
*Activity diagram* buat project baru adalah aktifitas utama dari serangkaian *use case* dalam sistem. Aktor yang berperan dalam aktivitas ini adalah *Content owner*. Adapun diagramnya adalah sebagai berikut:



Gambar 6 Activity diagram tambah room

### 4.2.5 Activity Diagram Tambah Pintu Antar Ruang

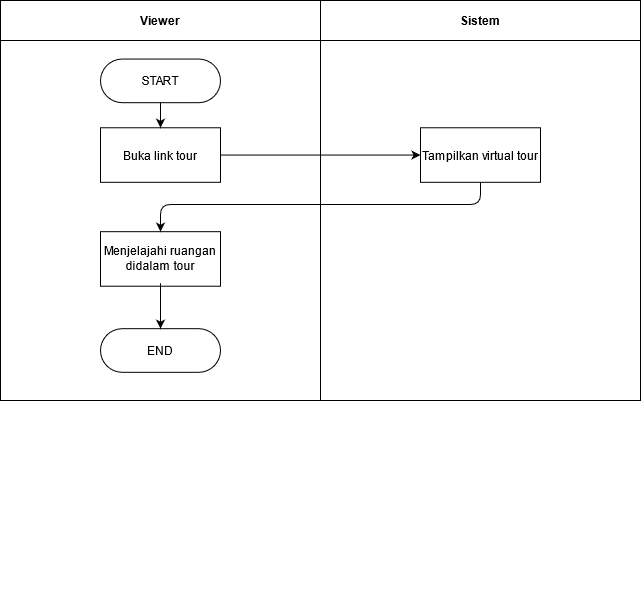
Pada *Activity diagram* ini menjelaskan aktivitas yang ada di dalam *use case* ketika *content owner* menambahkan tombol penghubung antar ruangan.



Gambar 7 Activity diagram tambah pintu antar ruang

### 4.2.6 Activity Diagram *Tour View*

*Activity diagram* tour view menjelaskan aktivitas yang ada di dalam *use case* ketika *audience* (*end-user*) menjelajahi *virtual tour* yang sudah dikembangkan *content owner*. Aktor yang berperan dalam aktivitas ini adalah *Viewer*. Adapun diagramnya adalah sebagai berikut:

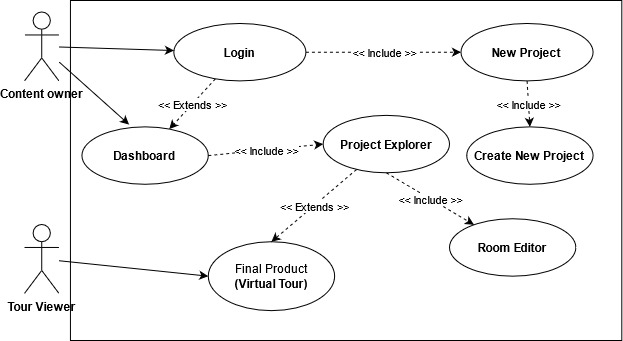


Gambar 8 Activity diagram tour view

## 4.3 Use Case Diagram

Pada sub-bab ini, penulis menjelaskan kegiatan yang dilakukan aktor dan sistem untuk mencapai suatu tujuan tertentu menggunakan *use case diagram*.

Use Case Diagram merupakan rangkaian tindakan yang dilakukan oleh sistem, aktor mewakili user atau sistem lain yang berinteraksi dengan sistem yang dimodelkan (John W. Satzinger dkk., 2016).



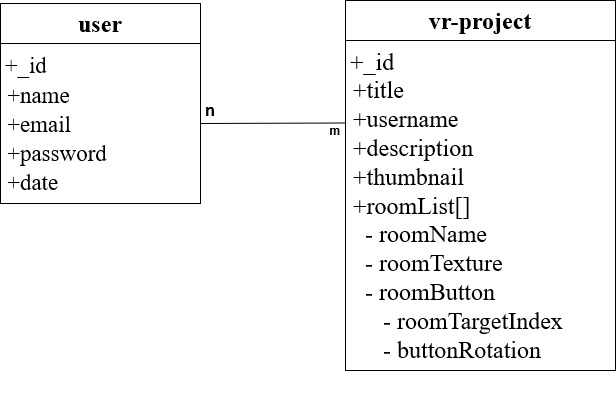
Gambar 9 Use Case Diagram

Peran dan aktivitas dari masing-masing aktor dalam sistem dapat dilihat pada gambar berikut:

## 4.4 Class Diagram

Untuk visualisasi struktur kelas dalam database dari sistem, penulis mencoba menyajikannya dalam bentuk Class Diagram. Diagram ini menggambarkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain.

*Class diagram* menggunakan model ERD (Entity Relationship Diagram) yang menggambarkan hubungan antar class dalam database.



Gambar 10 Class diagram

## 4.5 Desain *Database*

Kegiatan desain database ini memiliki tujuan untuk menghasilkan rancangan collection (istilah tabel dalam nosql) dengan relasi-relasi yang diimplementasikan pada *Database Management System* bernama *MongoDB*.

Desain database yang dibuat mengacu pada class, attribute, dan relasi pada class diagram yang ditunjukkan pada tabel-tabel dibawah ini.

|  |  |  |
| --- | --- | --- |
| **No** | **Nama *Collection*** | **Keterangan** |
| 1 | *user* | *Collection* untuk menyimpan data *content owner* |
| 2 | *vr-project* | *Collection* untuk menyimpan content *virtual tour* |

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama *Attribute*** | **Tipe Data** | ***Properties*** |
| 1 | \_id | *ObjectId* | *Primary Key* |
| 2 | name | *String* |  |
| 3 | email | *String* | *Unique* |
| 4 | password | *String* |  |
| 5 | date | *Date* |  |

Keterangan:

1. ***\_id*** : akan dihasilkan dari DBMS *MongoDb* sebagai primary key sistem sukses menyimpan data pengguna baru
2. ***Date:*** Memuat *timestamp* yang dibuat oleh sistem
3. ***Name:*** Inputan dari pengguna saat mendaftar
4. ***Email:*** email *unique* yang sudah divalidasi oleh sistem
5. ***Password:*** menyimpan data *hash* kata sandi, di encode sistem saat di buat, dan di *decode* kembali saat digunakan

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama *Attribute*** | **Tipe Data** | ***Properties*** |
| 1 | \_id | *ObjectId* | *Primary Key* |
| 2 | title | *String* |  |
| 3 | username | *String* |  |
| 4 | description | *String* |  |
| 5 | thumbnail | *String* |  |
| 6 | roomList | *Array* |  |
| - [].roomName | *String* |  |
| - [].roomTexture | *String* |  |
| - [].roomButton | *Array* |  |
| - []. -[].roomTargetIndex | *Int32* |  |
| - []. -[].buttonRotation | *Object* |  |

Keterangan:

1. ***\_id*** : akan di hasilkan dari DBMS *MongoDb* sebagai primary key sistem sukses membuat project pengguna baru
2. ***title*** :nama project
3. ***username*** : identitas pemilik konten
4. ***description:*** keterangan singkat
5. ***thumbnail*** : URL gambar pratinjau
6. ***roomList*** : daftar room yang berada di dalam project

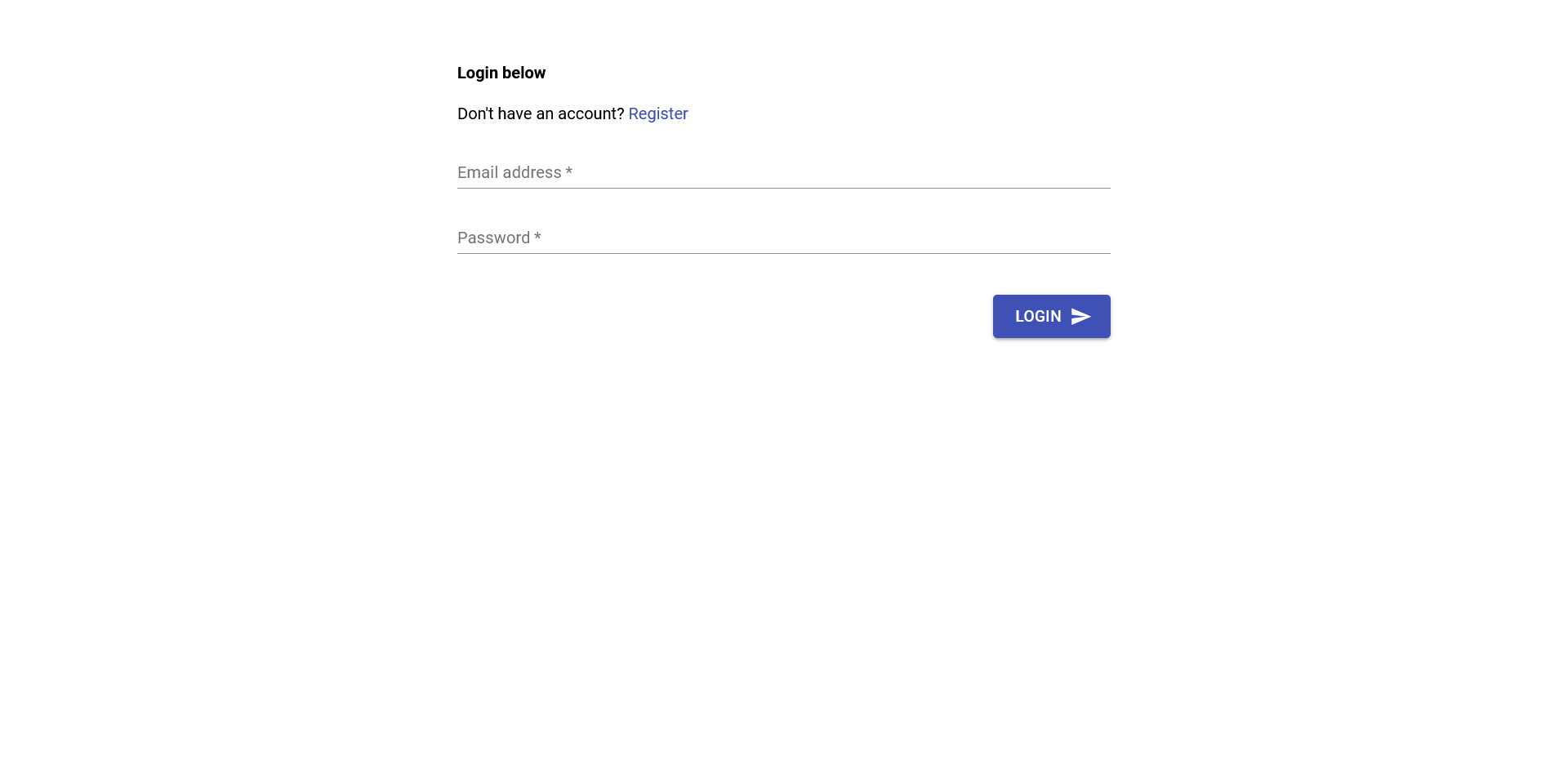
* ***roomName*** : Nama ruangan
* ***roomTexture*** : URL gambar 360
* ***roomButton*** : daftar tombol pintu untuk pindah antar ruang
* ***roomTargetIndex*** : nomor ruangan yang dituju
* ***buttonRotation*** : koordinat ruangan (x, y, z)

## 4.5 Hasil Implementasi Sistem

Untuk implementasi rancangan *interface* pada sistem dapat dilihat pada uraian dibawah ini:

1. Halaman Login

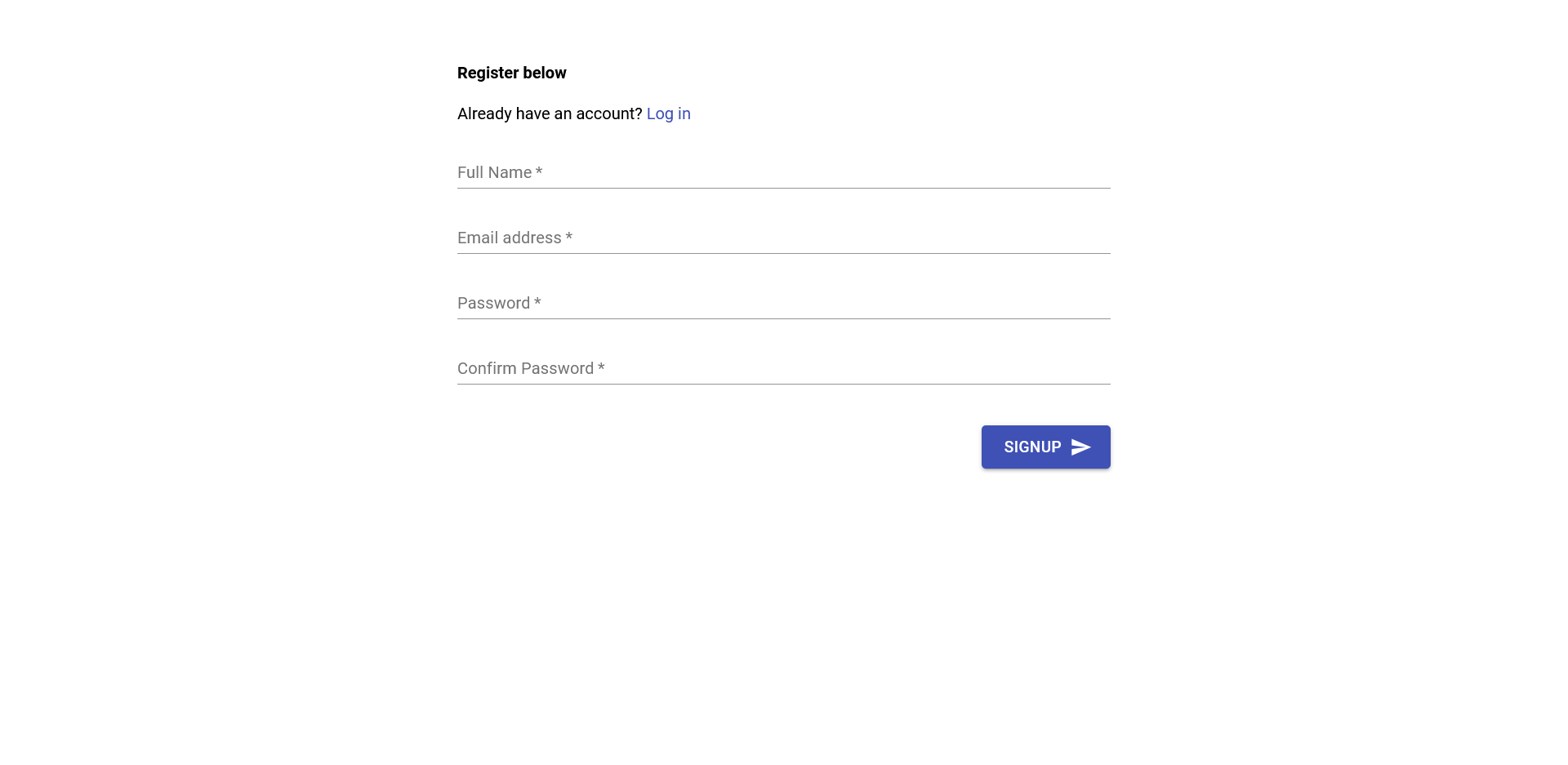
Tampilan Form login merupakan tampilan awal saat program dijalankan untuk menjamin bahwa user yang masuk kedalam sistem adalah user yang memiliki akun. Terdapat tombol untuk melakukan pendaftaran jika user belum memiliki akun.



Gambar 11 Tampilan login

1. Halaman Pembuatan Akun Baru

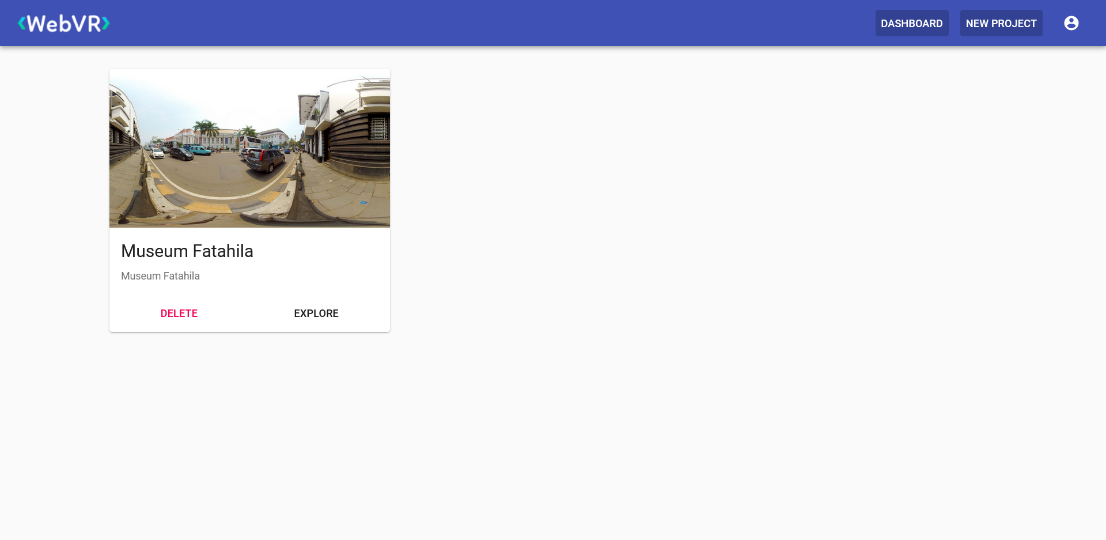
Jika *user* belum memiliki akun untuk login, *user* dapaat melakukan pendaftaran pada halaman ini dengan mengisi data diri dan password.



Gambar 12 Halaman regristasi

1. Halaman Dashboard

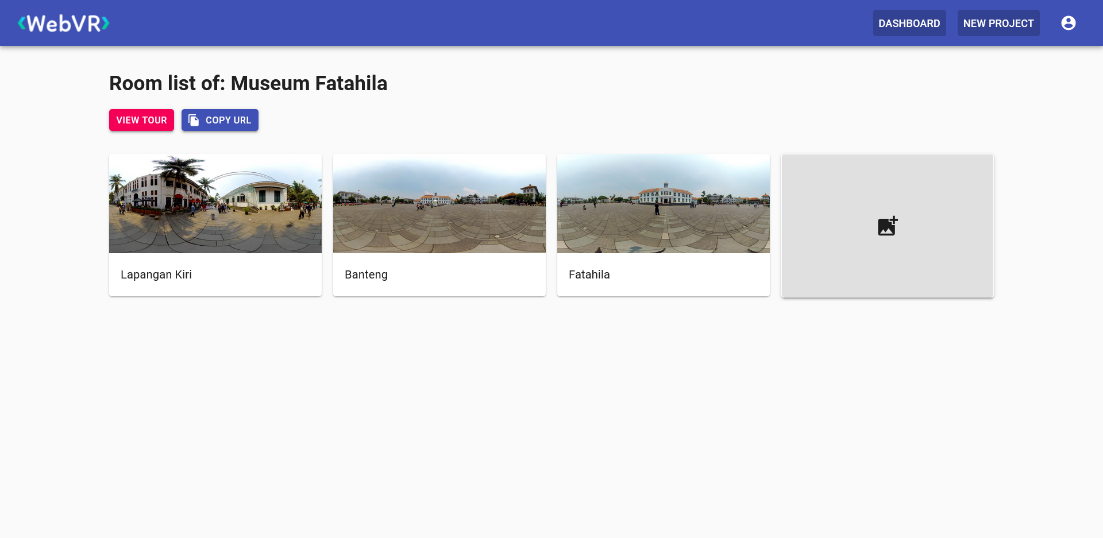
Daftar proyek yang sudah dibuat, jika belum ada maka tampilan kosong. User dapat memilih proyek mana yang akan dieksplorasi dengan klik proyek yang dimaksud.



Gambar 13 Tampilan dashboard

1. Halaman *Room Explore*

Menampilkan daftar ruangan yang sudah dibuat, jika belum ada maka sistem akan memunculkan pemberitahuan, user dapat membuat ruangan melalui tombol +.



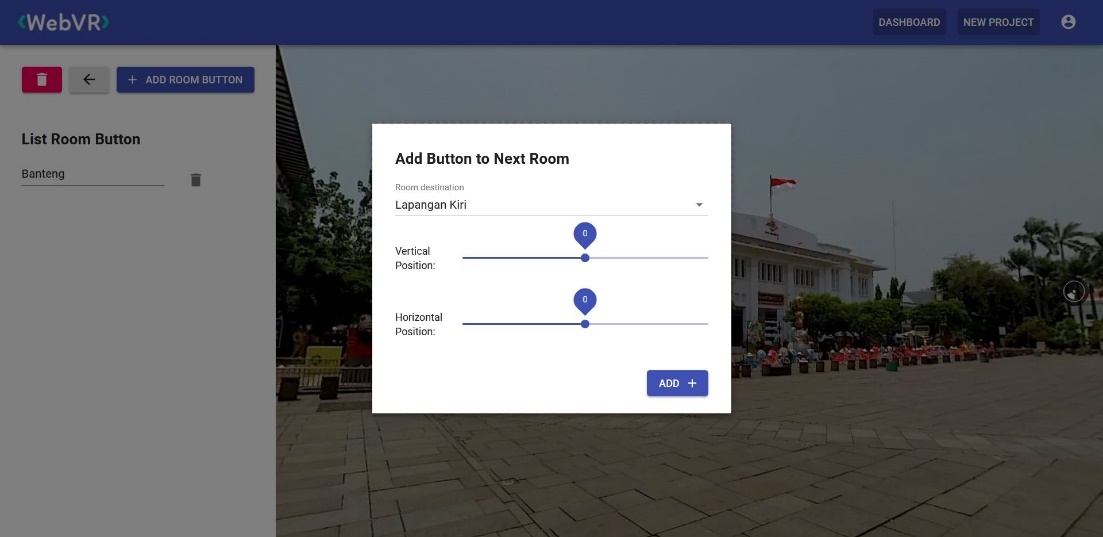
Gambar 14 Tampilan jelajah ruang

1. Halaman *Room Editor*

Merupakan halaman yang berfungsi untuk menambahkan tombol untuk berpindah antar ruangan. Di sisi kiri menampilkan daftar tombol yang sudah ada.



Gambar 15 Tampilan room editor



Gambar 16 Menu konfigurasi pintu

1. Contoh Hasil *Virtual Tour*

Demo tour yang sudah dikembangkan dan dapat dikunjungi oleh *viewer* melalui URL yang di-*share* *content owner*.



Gambar 17 Tampilan View

# BAB V KESIMPULAN

## 5.1 Kesimpulan

Dari hasil analisis masalah, implementasi *code* dan pengujian yang telah dilakukan peneliti, maka peneliti dapat mengambil beberapa kesimpulan, adapun kesimpulan-kesimpulan tersebut adalah sebagai berikut:

### Implementasi

Menggunakan serangkaian teknologi yang di analisis penulis melalui proses studi pustaka, bisa diaplikasikan untuk mengemembangkan sebuah sistem yang dapat menampilkan “dunia” virtual dalam bentuk *tour* berbasis web. Selain virtual tour itu sendiri, konten-konten yang ada di dalam tour tersebut dapat dibuat secara dinamis, dapat dilakukan mandiri oleh *content owner*.

### 2. Alternatif *VR Headset*

Solusi alternatif dari virtual tour dengan menggunakan VR headset yang memiliki batasan-batasan tertentu, diantaranya adalah:

1. Harga yang relatif mahal
2. Konten tidak terdistribusi secara cloud
3. Memiliki keterbatasan ruang dan waktu, dimana viewer harus menggunakannya di tempat.
4. Membutuhkan tenaga khusus untuk mengembangkan konten

Selain poin-poin di atas, dengan diimplementasikan berbasis web, virtual tour dapat disisipkan di aplikasi lain, seperti aplikasi *mobile* dengan menggunakan *web view*, ataupun website *menggunakan embedded iframe*.

## 5.2 Saran

Tentunya penelitian ini dibuat dan diselesaikan dengan banyak sekali kekurangan. Berdasarkan kesimpulan dan hasil-hasil terkait penelitian, maka peneliti akan memberikan beberapa saran yaitu sebagai berikut:

1. Virtual tour merupakan sistem yang memfasilitasi tour atau jelajah ruang yang sebelumnya dilakukan langsung pada tempat dan waktu tertentu, menjadi cara digital dan imersif. Sistem ini sangat bergantung kepada kualitas foto 360 sebagai objek utama pada konten.

Maka dari itu dibutuhkan kreatifitas saat mengunduh di internet ataupun memotret sendiri agar mendapatkan kualitas gambar yang maksimal.

2. Perlu ditambahkan beberapa fitur penting seperti penambahan spot informasi, kompresi gambar yang diunggah dll.

3. Perlunya dilakukan pemeliharaan terhadap teknologi-teknologi yang digunakan oleh sistem ini secara rutin agar tetap mendapatkan versi terbaik dan kompatibel dengan kemajuan browser atau aplikasi mobile.

# Daftar Pustaka

Andrew Morgan. (2017). *The Modern Application Stack – Part 1: Introducing The MEAN Stack*. Mongodb Blog. https://www.mongodb.com/blog/post/the-modern-application-stack-part-1-introducing-the-mean-stack

Beta Sidik. (2012). *Framework Codeigniter 2*. Informatika.

Bradshaw, S., Brazil, E., & Chodorow, K. (2019). *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage*. O’Reilly Media, Inc.

Dani Anggoro, dkk. (2015). Jurnal Rancangan Sistem Informasi Koperasi Simpan Pinjam Guru dan Pegawai Pada Koperasi SMK Manggala Tangerang. *Seminar Nasional Teknologi Informasi Dan Komunikasi*.

Google. (2019). *360° Media*. Google Dev Discover. https://developers.google.com/vr/discover/360-degree-media

H. Hadari Nawawi, & H. M. Martini Hadari. (1992). *Instrumen Penelitian Bidang Sosial*. Gadjah Mada University Press.

Hahn, E. (2016). *Express in Action : Writing, buildings, and testing Node.js application*. Manning Publications.

Highton, S. (2010). *Virtual Reality Photography Creating Panoramic and Object Images*. Virtual Reality Photography.

Ian Sommerville. (2011). *Software Engineering (Rekayasa Perangkat Lunak)*. Erlangga.

John W. Satzinger, Robert B. Jackson, & Stephen D. Burd. (2016). *Systems Analysis And Design In A Changing World* (7/E). Cengage Learning.

Kleppmann, M., & Beresford, A. R. (2017). A Conflict-Free Replicated JSON Datatype. *IEEE Transactions on Parallel and Distributed Systems*, *28*(10), 2733–2746. https://doi.org/10.1109/TPDS.2017.2697382

Kumar, A., & Singh, R. K. (2016). Comparative analysis of angularjs and reactjs. *International Journal of Latest Trends in Engineering and Technology*, *7*(4), 225–227. https://doi.org/10.21172/1.74.030

M, V. A., & Sonpatki, P. (2016). *ReactJS by Example - Building Modern Web Applications with React*. Packt Publishing.

Malabay. (2018). Model Rancangan Pembelajaran Aktif, Kreatif Dan Inovatif Dengan Pendekatan Unified Modeling Language. *Jurnal Ilmu Komputer*, *15 No 1*, 81–82.

Moh. Nazir. (2013). *Metode Penelitian*. Ghalia Indonesia.

Muhammad Jamil. (2018). *Pemanfaatan Teknologi Virtual Reality (VR) di Perpustakaan*.

Munir. (2012). *Multimedia Konsep & Aplikasi dalam Pendidikan*. Alfabeta.

Neal Leavitt. (2010). Will NoSQL Databases Live Up to Their Promise? *Computer*, *43*.

Nugroho, A. (2010). *Rekayasa Perangkat Lunak Menggunakan UML dan Java* (1st ed.). Andi Offset.

Purbo, O. W. (2006). *Buku Sakti Menjadi Programmer Sejati PHP dan MySQL*. Gava Media.

Purnomosidi, B. (2013). *Buku Cloud Node.js*.

Rusman, Kurniawan, D., & Riyana, C. (2011). *Pembelajaran Berbasis Teknologi Informasi dan Komunikasi : Mengembangkan Profesionalisme Guru*. Rajawali Pers. PT. Raja Grafindo Persada.

Satzinger JW, RB, J., & SD, B. (2012). *System Analysis and Design in a Changing World Seventh Edition*. Cengage Learning.

Sembiring, S. (2013). Perancangan Aplikasi Seganografi Untuk Menyisipkan Pesan Teks Pada Gambar Dengan Metode End of File. *Pelita Informatika Budi Darma*, *IV*, *No.2*.

Soetam Rizky. (2011). *Konsep Dasar Rekayasa Perangkat Lunak*. PT. Prestasi Pustakaraya.

Subramanian, V. (2019). Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node. In *Pro MERN Stack*. Apress.

Suryana, T., & Koesheryatin. (2014). *Aplikasi Internet Menggunakan HTML, CSS, & JavaScript*. PT ElexMedia Komputindo.

Syahrial, S. (2016). Analisis Dan Perancangan Sistem Informasi Koperasi Berbasis Web Pada Koperasi Unit Desa Pandan Jaya Geragai. *Manajemen Sistem Informasi*, *1 No. 1*.

Valentina, F., & Handjojo. (2013). *Perancangan Dan Implementasi Aplikasi Content Management System Dengan Format Virtual Online Tour*. *1*(2), 1–6.

Wikipedia. (2019). *npm (perangkat lunak)*.

Wikipedia. (2020). *Pemrograman fungsional*. https://id.wikipedia.org/wiki/Pemrograman\_fungsional

Winarno, E., Zaki, A., & SmithDev. (2014). *Pemrograman web berbasis Html5, PHP, dan Javascript*. Elex Media Komputindo.

# Lampiran 1 Daftar Riwayat Hidup

**DAFTAR RIWAYAT PENYUSUN**

**Data Pribadi**

|  |  |  |
| --- | --- | --- |
| Nama | : | Yogi Prasetya |
| No. KTP | : | 3212232504940003 |
| TTL | : | Indramayu, 25 April 1994 |
| Agama | : | Islam |
| Jenis Kelamin | : | Laki - Laki |
| Alamat | : | Dusun KB Randu 2. RT05/RW01.  Ds. Anjatan Baru, kec. Anjatan.  Kab. Indramayu - 45256 |
| No. Telp | : | 083873087154 |
| E-Mail | : | yogiprsetya@gmail.com |

****

**Riwayat Pendidikan**

1. SDN Anjatan Baru 3, Indramayu, Jawa Barat, 2001 - 2006
2. SMPN 1 Anjatan, Indramayu, Jawa Barat, 2007 - 2009
3. SMKN Losarang, Indramayu, Jawa Barat, 2010 - 2013
4. Universitas Esa Unggul Jakarta Barat, 2018 – Sekarang

**Riwayat Pekerjaan**

1. Quality Control, PT. Plymilindo Perdana, 2013 - 2014
2. IT Support, anekapetindo.com, 2015 – 2018
3. Software Engineer, smarteye.id, 2018 – sekarang

# Lampiran 1 Keterangan Penelitian

**SURAT KETERANGAN PENELITIAN**

**No: SE/001/XII/2020**

Yang bertanda tangan di bawah ini:

Nama : Fahmi Rahmadani, ST

Jabatan : Chief Executive Officer

Dengan ini menerangkan bahwa:

Nama : Yogi Prasetya

NIM : 20180801192

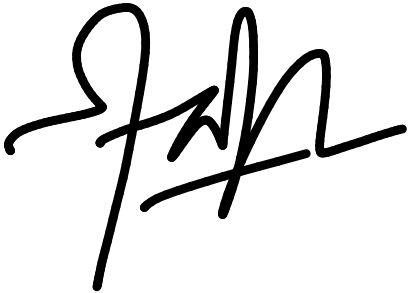
Program Studi : Teknik Informatika

Asal : Universitas Esa Unggul

Adalah benar telah melaksanakan penelitian di **smarteye.id** dalam rangka penulisan skripsinya yang berjudul “APLIKASI MANAJEMEN KONTEN UNTUK VIRTUAL TOUR BERBASIS WEB MENGGUNAKAN TEKNOLOGI MERN STACK”. Terhitung dari 13 Oktober 2020 s/d 3 Desember 2020, dan telah pula membahas materi hasil penelitiannya dengan kami.

Demikian surat keterangan penelitian ini kami buat untuk dapat dipergunakan sebagaimana mestinya.

Jakarta, 05 Desember 2020



**Fahmi Rahmadani, ST**

Chief Executive Officer