# BAB II TINJAUAN PUSTAKA

## Penelitian Terdahulu

Berikut adalah beberapa penelitian yang sudah dilakukan sebelumnya dan menjadi sebuah kajian penelitian :

Penelitian yang pertama yaitu “Analisa dan Perancangan Sistem Informasi *Quality Assurance* Berbasis Web pada Pt.Codigo Cyberlin Metadata Menggunakan *Unified Modeling Language*” penelitian ini bertujuan untuk mempermudah dalam membuat laporan *Quality Assurance* serta Membantu *Project Management Officer* terkait penjaminan mutu project (Riadi & Sartana, 2018). Kesimpulan pada penelitian tersebut adalah kesulitan dalam mengawasi data issue dan data fitur proyek dapat terorganisir dengan baik serta laporan dan informasi terkait issue di dalam proyek dapat disajikan dengan terorganisir.

Penelitian kedua berjudul “Rancang Bangun Sistem Informasi *Quality Assurance* (QA) pada Jurusan Teknik Elektro Politeknik Negeri Pontianak” dimana penelitian ini bertujuan untuk membantu pejabat QA untuk melakukan pengolahan data kinerja dosen dan teknisi pada Jurusan Teknik Elektro Politeknik Negeri Pontianak (Latifah et al., 2022). Kesimpulan dari penelitian tersebut adalah Aplikasi yang sudah dirancang peneliti diharapkan dapat membantu setiap penjamin mutu program studi dalam melakukan pengolahan data penilaian kepuasan pengguna layanan program studi terhadap dosen dan teknisi yang dilakukan oleh mahasiswa sebagai responden.

Penelitian ketiga yaitu “Rancang Bangun Sistem Informasi Manajemen Mutu Perusahaan (Studi Kasus: Bagian Pengendalian Dokumen Pt. Pura Barutama Divisi Boxindo, Kudus)” penelitian ini bertujuan untuk membantu dalam melakukan monitoring capaian mutu dan temuan audit, pendistribusian jadwal audit, serta membantu dalam pengelolaan dokumen ISO 9001:2008 (Fernando et al., 2019). Kesimpulan dari dari penelitian tersebut adalah aplikasi ini diharapkan dapat menjawab permasalahan pada pura Boxindo yaitu membantu dalam pendistribusian dokumen baik sasaran mutu, capaian mutu, dan program audit, membantu membuat analisa pencapaian sasaran mutu tiap departemen, serta membantu dalam memonitor temuan audit mana yang sudah maupun belum diselesaikan.

Penelitian keempat yaitu “Perancangan Sistem Informasi *Quality Control* Pada Pt. Sanly Industries Berbasis Web” penelitian memiliki tujuan untuk memudahkan dalam melaksanakan pengelolaan maupun pencarian data produk pada perusahaan Pt. Sanly Industries (Rizki & Malfiany, 2021). Kesimpulan dari penelitian tersebut adalah aplikasi ini diharapkan dapat menjadi jawaban atas masalah dan menjadi manfaat antara lain membantu operator *Quality Control* dalam membuat laporan harian dan pengecekan berkas serta penyajian informasi terkait produk plastic dari perusahaan.

Secara garis besar berdasarkan dari penelitian sejenis yang telah dijabarkan sebelumnya, perbedaan pada sistem yang akan peneliti buat adalah penggunaan *Framework* serta berfokus pada kebutuhan fitur *Test Management System* serta laporan yang dibutuhkan oleh *Quality Assurance Engineer*  sebagai wadah untuk mengumpulkan informasi terkait hasil *Test Case, Test Run*, hingga *Test Scenario* sehingga kegiatan penjaminan mutu produk berjalan lancar tanpa hambatan. Serta, tempat Studi kasus peneliti serta penggunaan *Framework* agar nantinya dimudahkan dalam pengembangan dan penambahan fitur di masa depan.

## Teori Terkait

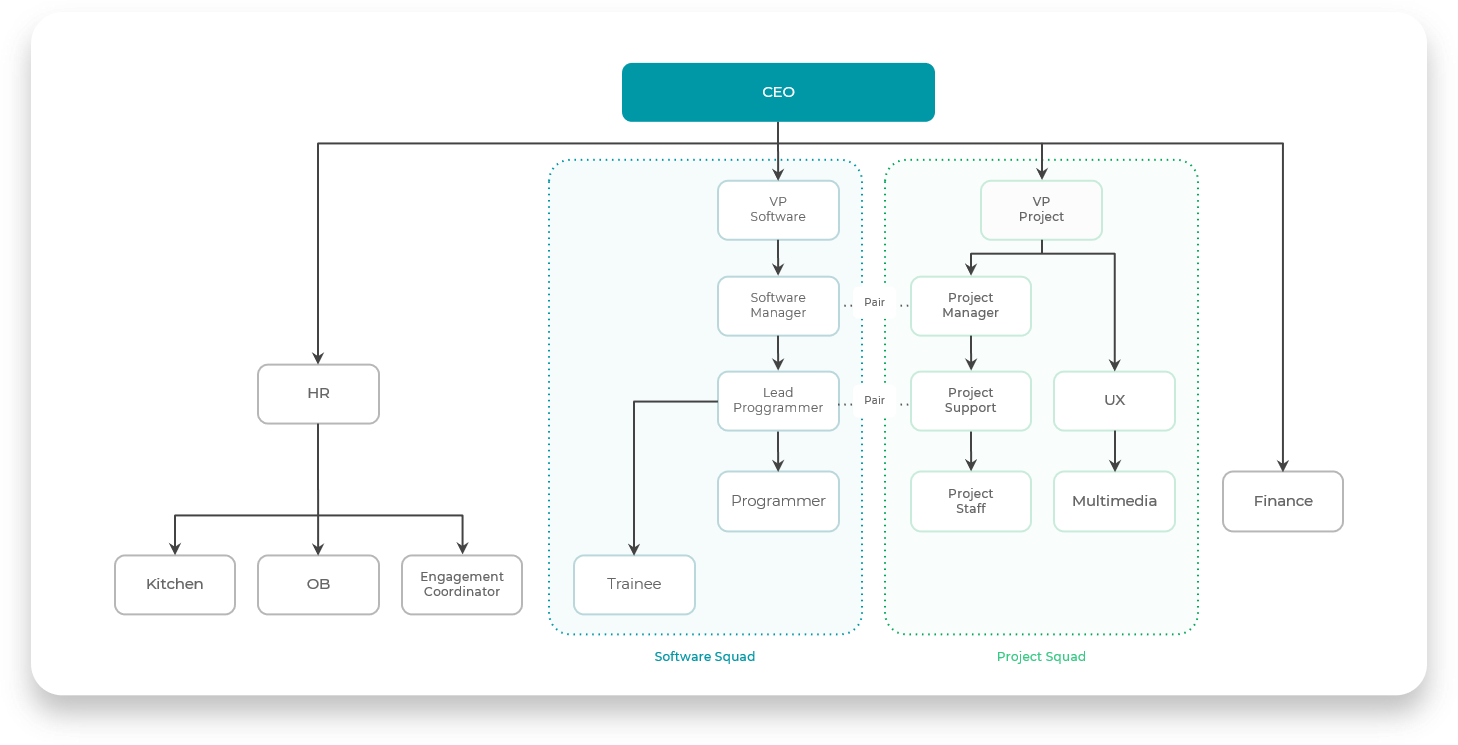
### Profil Perusahaan

Pt. Venturo Pro Indonesia merupakan merupakan perusahaan yang bergerak pada bidang IT Consultant yang memiliki banyak pengalaman dalam membuat dan mengembangkan aplikasi software untuk perusahaan maupun pemerintahan kota dan daerah di Indonesia sejak tahun 2013. Perusahaan ini fokus pada layanan Remote Working Software Talent. Perusahaan ini didirikan pada tahun 2013 oleh Yulianto Frandi di Malang. Kini Venturo berlokasi di Perumahan Griya Shanta Permata, N-524, Mojolangu, Kec. Lowokwaru, Kota Malang, Jawa Timur. Venturo membawa beberapa misi yaitu:

1. Selalu berinovasi dan beradaptasi sesuai dengan tuntutan customer dan perkembangan teknologi
2. Memberikan layanan / produk terbaik kepada pelanggan melalui sistem bisnis yang efektif dan efisien
3. Mengutamakan pengembangan karyawan sebagai aset utama perusahaan



Gambar 2. 1 Logo Perusahaan PT. Venturo Pro Indonesia



Gambar 2. 2 Struktur Organisasi PT. Venturo Pro Indonesia

### Quality Assurance

Secara sederhana, istilah *Quality Assurance* mengacu pada berbagai strategi dan teknik yang dapat diadopsi untuk menghasilkan kualitas terbaik pada system akhir yang sedang dikembangkan seorang pengembang (Walkinshaw, 2017). *Quality Assurance* memainkan peran penting dalam proses pengembangan. *Quality Assurance* memastikan bahwa produk yang dikembangkan berkualitas tinggi dan tidak mengandung kesalahan, cacat, maupun bug. *Quality Assurance* akan menguji, mengidentifikasi, dan menawarkan solusi untuk kesalahan yang teridentifikasi sebelum merilis produk. Hal ini akan membantu developer menghindari citra buruk karena aplikasi yang mereka buat tidak berfungsi seperti yang diharapkan.

### Software Testtng

*Software Testing* atau Pengujian perangkat lunak merupakan kegiatan penting dalam *software engineering*. Pengujian mengeksekusi perangkat lunak untuk melihat apakah itu sesuai dengan persyaratan yang ditentukan. Secara sederhana, kegiatan pengujian memproses dan menjalankan program untuk menemukan *bug*. Pengujian perangkat lunak akan melakukan serangkaian prosedur yang dirancang untuk memeriksa apa yang harus dilakukan dan (sebaliknya) apa yang tidak boleh terjadi pada perangkat lunak (Min et al., 2020). Ada berbagai teknik pengujian yang dapat digunakan untuk menguji perangkat lunak. Salah satunya adalah *end-to-end testing* yang biasa digunakan oleh perusahaan yang melakukan pengembangan aplikasi web. Pengujian *end-to-end* adalah metode yang digunakan untuk menguji apakah alur aplikasi berfungsi seperti yang dirancang dari awal hingga akhir (Min et al., 2020).

Implementasi pengujian *end-to-end* dapat menggunakan salah satu metode untuk pengujian skenario yang kompleks. Dalam praktiknya, pengujian *end-to-end* adalah teknik pengujian yang dapat dilakukan melalui *manual software testing* atau dengan *automated software testing*. Dalam pengembangan *Test Management System* ini, akan diimplementasikan metode *manual software testing*. *Manual testing* adalah sebuah metode pengujian dimana penguji akan menyiapkan beberapa *test scenario* atau *test cases* lalu mengeksekusi *test scenario* atau *test cases* tersebut untuk mengidentifikasi masalah dan *defect* di perangkat lunak yang akan diuji secara manual (Min et al., 2020). Secara umum, penggunaan *manual testing* saat mengimplementasikan pengujian *end-to-end* terdiri dari menjalankan pengujian dengan menjalankan perangkat lunak sesuai dengan skenario yang tertulis dalam kasus pengujian. Kemudian bandingkan hasil *ouput* dari aplikasi dengan *output* yang diharapkan untuk setiap kasus uji.

### UML

Bahasa pemodelan perangkat lunak UML (Unified Modelling Language) sejak pertama kali diperkenalkan pada tahun 1997, telah berkembang menjadi bahasa permodelan yang baku dalam pengembangan perangkat lunak (Kurniawan, 2018). UML digunakan dalam mengembangkan sistem perangkat lunak dengan pendekatan berorientasi objek. Intensitas penggunaan UML yang tinggi didukung oleh konsep pemodelan yang lebih matang yang diartikulasikan dalam setiap spesifikasi UML yang dikembangkan oleh Object Management Group (Kurniawan, 2018). Salah satu diagram UML yang penting untuk mengilustrasikan kebutuhan sistem adalah diagram *Use Case*, yang dengan jelas menjelaskan konteks interaksi antara aktor dan sistem. Setiap *Use Case* mendefinisikan perilaku/fungsionalitas dari sistem yang menjelaskan kebutuhan aktor untuk mencapai tujuannya.

Diagram *Use Case* dapat menjadi gambaran yang baik untuk menjelaskan konteks sistem sehingga batas-batas sistem terlihat dengan jelas. Ada dua elemen penting yang perlu dideskripsikan, yaitu aktor dan *Use Case*. Aktor adalah segala sesuatu yang berinteraksi langsung dengan sistem, bisa berupa orang (dari sudut pandang peran mereka, bukan nama/staf mereka) atau sistem komputer lain. Aktor diidentifikasi dengan figur orang – orangan atau *stickman*, dengan nama kata benda di bawah yang menunjukkan peran/sistem. Aktor bisa bersifat *primary*, yaitu mereka yang memulai *Use Case*, atau yang *secondary*, yaitu mereka yang membantu menjalankan *Use Case*. *Use Case* ditandai dengan simbol elips, yang berisi nama kata kerja aktif, yang menyatakan tindakan dari sudut pandang aktor. Setiap aktor dapat berinteraksi dengan sistem dalam banyak *Use Case*. Sebaliknya, setiap *Use Case* dapat diimplementasikan oleh lebih dari satu aktor (Kurniawan, 2018).

### Website

Website adalah sebuah platform yang berisi file gambar, teks, audio, video, dan animasi yang dapat diakses melalui internet. Website dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau bergerak, animasi, dan suara, baik yang bersifat statis maupun dinamis yang membentuk satu rangkain bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman (Tavakoli & Wijesinghe, 2019).

Sistem informasi adalah sekumpulan elemen yang bekerja sama untuk mengumpulkan, memproses, menyimpan, dan menyediakan informasi untuk memenuhi kebutuhan bisnis atau organisasi. Terdiri dari hardware, software, data, prosedur dan orang-orang yang bekerja sama untuk memproses informasi. Tujuannya adalah untuk meningkatkan efisiensi dan efektivitas dalam pengambilan keputusan dan operasi bisnis (Purnama, 2021).

### MVC

Arsitektur MVC (*Model, View* dan *Controller*) berasal dari tahun 1970-an dan dikembangkan oleh Trygve Reenskaug saat bekerja pada sistem Smalltalk. Tujuannya adalah memisahkan logika aplikasi dari tampilan dan mempermudah pemeliharaan dan pengembangan aplikasi. Hal ini memungkinkan para *Developer* untuk memfokuskan pada bagian spesifik dari aplikasi tanpa mempengaruhi bagian lain. Saat ini, MVC sering kali digunakan ketika pengembangan aplikasi web dan mobile (Pitt, 2021). MVC disusun sebagai solusi desain arsitektur untuk menyelesaikan masalah tertentu. Secara garis besar, MVC akan memisahkan beberapa bagian dalam aplikasi menjadi *Model, View* dan *Controller* (Sunardi & Suharjito, 2019).

1. *Model*

*Model* adalah bagian yang berisi perintah untuk mengolah berbagai data yang terdapat pada database. Contohnya seperti mengambil data, memasukkan data, mengubah data, dan menghapus data. Selain itu, Model juga berisikan *business logic* didalam aplikasi.

1. *View*

*View* adalah bagian yang berisikan dan menangani tampilan *user interface* dimana *view* merupakan bagian tersendiri dan berbeda dari *model* dan *controller.* Hal ini akan memudahkan web designer untuk membuat tampilan dari sebuah website sesuai dengan keinginan.

1. *Controller*

*Controller* adalah bagian yang dari aplikasi yang mengontrol aliran data antara *Model* (data) dan *View* (presentasi). *Controller* bertanggung jawab untuk mengkoordinasikan aksi dan interaksi antara *Model* dan *View*.

### Framework

*Framework* adalah kerangka kerja ang dapat digunakan sebagai dasar untuk membangun aplikasi atau sistem. Kerangka kerja ini dibuat untuk membantu pengembang web untuk membangun dan mengelola kode lebih cepat dan lebih jelas. Beberapa fungsi utama framework antara lain :

1. Memberikan kerangka kerja yang terstruktur untuk membangun aplikasi, sehingga membuat proses pengembangan lebih efisien dan teratur.
2. Menyediakan fungsi umum yang sering digunakan seperti autentikasi, validasi, routing, dan lainnya sehingga mempermudah pengembangan aplikasi.
3. Mempermudah dan mempercepat proses pengembangan aplikasi dengan menyediakan komponen dan fitur yang sudah teruji dan dapat digunakan kembali.
4. Meningkatkan konsistensi dan kualitas kode dengan memberikan standar yang harus diterapkan selama proses pengembangan.
5. Menyediakan solusi umum untuk masalah yang sering dijumpai selama pengembangan aplikasi, sehingga mempermudah proses pemecahan masalah.

Pengembangan aplikasi ini akan menggunakan 2 macam metode yaitu metode *back-end* dan *front-end* dalam pengembangannya. Pendekatan ini menyederhanakan dan mempercepat pengembangan aplikasi dengan memisahkan logika bisnis *back-end* dari perspektif pengguna *front-end.*

Dalam penelitian ini, bagian *back-end* akan dikembangkan dengan menggunakan *framework* Laravel. Sedangkan *front-end* akan menggunakan *frameworks* Angular

#### **Laravel**

Laravel adalah salah satu *framework* PHP terbaik yang dikembangkan oleh Taylor Otwell dan proyek Laravel dimulai pada April 2011. Awalnya *project* ini dibuat karena Otwell sendiri tidak menemukan *framework* yang up to date dengan versi PHP*.* Memperluas kerangka kerja yang ada juga bukan ide yang baik karena sumber daya yang terbatas. Karena beberapa keterbatasan tersebut, Otwell membuat *framework-*nya sendiri, Laravel (Yudhanto & Prasetyo, 2018). Laravel menggunakan pendekatan MVC.

*PHP* (*Hypertext Processor*) adalah bahasa pemrograman berbasis kode (script) yang digunakan untuk mengolah data dan mengirimkannya kembali ke web browser sebagai kode *HTML.* Dibandingkan dengan *HTML*, *PHP* bekerja dengan sisi server, dan kompilasi kode program dilakukan di sisi server. Script akan berjalan di komputer server, kemudian server akan mengirimkan hasilnya ke browser komputer *client*. Bahasa pemrograman sisi server termasuk *Perl, PHP, ASP, JSP, Coldfusion* dan lain-lain (Supono & Putratama, 2018).

#### **Angular**

Angular adalah sebuah *framework* untuk membuat aplikasi web yang menggunakan bahasa pemrograman *JavaScript*. Angular ini dibuat untuk membantu *developer* untuk membuat aplikasi dinamis dan interaktif dengan cara membuat halaman web yang reaktif terhadap aksi pengguna. Angular juga menyediakan alat dan sumber daya untuk membuat aplikasi skala besar dengan mudah. Seperti *framework javascript* pada lainnya, Angular menggunkan pendekatan MVC*.* Akan tetapi dalam implementasinya, menggunakan konsep yang berbeda dengan konsep asli MVC (Jubilee Enterprise, 2017).

TypeScript adalah bahasa pemrograman yang dikembangkan oleh Microsoft yang merupakan super set dari JavaScript. Ini memiliki tipe data statis dan fitur pemrograman modern yang memungkinkan pengembang untuk menulis kode dengan lebih efisien dan mudah dalam pemeliharaan. TypeScript dikompilasi ke JavaScript dan dapat berjalan di semua browser dan lingkungan JavaScript yang ada sehingga aplikasi yang dibuat menggunakan TypeScript dapat diterima oleh banyak platform.

Bahasa pemrograman yang ada didalam Angular antara lain adalah *Typescript* dan *CSS*. *TypeScript* adalah bahasa pemrograman yang dikembangkan oleh Microsoft yang merupakan super set dari *JavaScript*. *TypeScript* memiliki tipe data statis dan fitur pemrograman modern yang memungkinkan pengembang untuk menulis kode dengan lebih efisien dan mudah dalam pemeliharaan. TypeScript dikompilasi ke *JavaScript* dan dapat berjalan di semua browser dan lingkungan *JavaScript* yang ada sehingga aplikasi yang dibuat menggunakan *TypeScript* dapat diterima oleh banyak platform (Cherny, 2019). TypeScript menyediakan berbagai macam fasilitas diantaranya ada kelas, modul, dan interface yang membuat para *developer* untuk membangun dan merancang aplikasi yang kompleks dan rumit menjadi lebih mudah. Inilah yang membuat *Typescript* berbeda dengan *Javascript* (Cherny, 2019).

Sedangkan CSS adalah singkatan dari *Cascading Style Sheets*, yang merupakan bahasa *styling* untuk mengatur tampilan halaman web. Ini memungkinkan pengembang untuk menentukan bagaimana halaman web harus terlihat, seperti warna, tata letak, font, dan banyak lagi, sehingga halaman web dapat terlihat menarik dan menarik bagi pengguna. CSS diterapkan pada HTML untuk membuat halaman web yang responsif dan interaktif (Ndia et al., 2019). Dengan menggunakan *CSS,* para *developer* akan lebih mudah untuk menentukan keseluruhan tampilan web hanya dengan mengganti atribut atau perintah gaya CSS dengan atribut yang diinginkan.

### Database

*Database* adalah sistem penyimpanan informasi yang terkelola dengan baik dan saling terhubung satu sama lain. Dengan pengelolaan tersebut pengguna dapat mencari informasi atau suatu data, menyimpan data dan menghapus data dengan mudah. Database dapat menangani jumlah besar data dan memungkinkan pencarian cepat dan efisien melalui query. Database juga memiliki kemampuan untuk memastikan integritas data dengan memastikan bahwa data yang dimasukkan adalah benar dan memastikan bahwa data tidak hilang atau rusak. Database digunakan secara luas dalam berbagai bidang, seperti bisnis, teknologi informasi, dan manajemen sumber daya manusia (Andaru, 2018).

#### **MySql**

MySQL adalah salah satu jenis *Relational Database Management Systems* (RDBMS) yang populer. MySQL menyediakan solusi untuk penyimpanan, pengelolaan, dan akses terhadap data melalui konsep tabel yang terdiri dari baris dan kolom walaupun tidak menuntut sejumah *resource* yang besar (Christudas, 2019). Adapun kelebihan MySQL antara lain:

1. Gratis dan Open Source

MySQL merupakan perangkat lunak sumber terbuka sehingga mudah diakses dan dimodifikasi oleh siapa saja.

1. Skalabilitas

MySQL mampu menanganai jumlah besar data dan memastikan performa yang baik saat data bertambah besar.

1. Integrasi mudah dengan aplikasi

MySQL mudah digunakan dan diterapkan dalam berbagai aplikasi web seperti PHP dan Ruby on Rails.

1. Dukungan komunitas

Ada banyak sumber daya dan komunitas online yang membantu dalam mengatasi masalah dan menyediakan solusi untuk pengguna MySQL.

1. Kemampuan cross-platform

MySQL dapat berjalan pada berbagai sistem operasi seperti Windows, macOS, dan Linux.