

Bab II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Dalam pembuatan tugas akhir ini pengumpulan data didapatkan dengan cara studi literatur yaitu mencari referensi jurnal yang terkait dengan analisis komparasi web development stack. Berikut merupakan referensi jurnal dari penelitian sebelumnya.

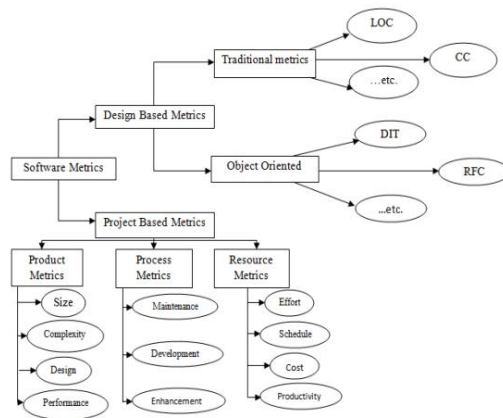
Thesis dari (Wohlgethan, 2018) yang berjudul “Supporting Web Development Decisions by Comparing Three Major JavaScript Frameworks: Angular, React and Vue.js” membahas gambaran tentang tiga teknologi yang dimiliki traksi tertinggi di pasar: Angular, React, dan Vue.js yang pada saat itu tidak cukup untuk studi menyeluruh dari seluruh ekosistem JavaScript, penelitian tersebut memberikan indikasi untuk pengambilan keputusan karena ketiga kerangka kerja ini semuanya mengikuti inti yang berbeda prinsip (Wohlgethan, 2018). Tujuan pada penelitian tersebut yaitu menguji salah satu framework yang paling menonjol sehingga dapat direkomendasikan untuk mayoritas kasus penggunaan dalam hal pengembangan web. Kriteria yang dianalisis pada penelitian tersebut diantaranya: (1) tingkat Stability setiap framework, ini mencakup versioning, release policy, maintainability, licensing; (2) Learning Curve meliputi Dokumentasi yang tersedia, persyaratan pengetahuan (HTML, CSS, Javascript); (3) Javascript Integration ini mencakup Best Practices and Stacks (routing, state management, compilers, bundling tools), Bahasa Pengembangan (Vanilla JavaScript (ES5/6/7), Typescript, JSX), dan fitur syntax. Analisis yang diambil menghasilkan beberapa perbandingan diantaranya: Pertama berupa fitur dan

Commented [JP8]: Gunakan titik koma

teknikal aspek, Kedua Support dan Accessibility, Ketiga Statistik Komunitas. Dari segi fitur dan teknikal aspek secara umum tidak jauh berbeda semuanya menggunakan konsep component based. Namun terdapat juga perbedaan yang bersifat subjektif contohnya seperti VueJs menawarkan developer lebih dari satu cara untuk menulis kode misalnya, ES5/6 atau TypeScript versi 2.5+, sementara pada Angular lebih membatasi (restrictive) penulisan kodenya yaitu wajib menggunakan Typescript. Dan untuk React sendiri sangat berbeda dari keduanya, react menggunakan JSX yang dimana ini adalah pendekatan yang paling beragam karena JSX menyiratkan HTML yang bercampur dengan Javascript. Adapun konsep state yang dapat menjadi penghalang untuk adaptasi. Vue dan Angular memiliki struktur komponen yang relative sama, keduanya memisahkan template(HTML), style(CSS), dan logic(JS). VueJs memisahkan template, style, dan logic dalam satu file yang berekstensi “.vue“, sedangkan Angular memisahkan template, style, dan logic ke tiga file terpisah. Data handling dari ketiga stack juga berbeda, React menggunakan pendekatan one-way binding, sementara dua stack yang lainnya menyarankan untuk menggunakan two-way binding. Two-way binding terlihat lebih nyaman ketika pertama kali dipakai, akan tetapi ketika aplikasi semakin kompleks maka penggunaan two-way binding akan sulit untuk melacak (tracking) data yang telah diubah. Satu hal yang penting dari klasifikasi umum yang disajikan pada angular adalah satu-satunya teknologi berfitur lengkap dalam perbandingan ini yang memang sesuai dengan yang disebut framework, sedangkan React dan Vue hanyalah library yang sering dinamai dalam konteks yang sama dengan framework. Kesimpulan dari penelitian ini menyatakan bahwa tidak

ada jawaban yang pasti dari framework yang paling menonjol dan dapat direkomendasikan untuk mayoritas developer dalam hal pengembangan web.

Adapun penelitian dari (Sharma, 2022) yang berjudul “A Survey on Software Design Based and Project Based Metrics” membahas tentang survei metrik perangkat lunak yang sudah ada sehingga metrik yang sesuai dapat dipilih tergantung pada kebutuhan proyek. Metrik adalah cara terbaik untuk mengukur sifat atau atribut internal dan eksternal perangkat lunak. Atribut internal spesifik untuk proses, sumber daya, atau produk itu sendiri terlepas dari lingkungan operasi. Pada bab pertama memberikan gambaran umum tentang produk atribut yang diukur dengan metrik. Metrik perangkat lunak yang sudah ada telah diklasifikasikan berdasarkan desain perangkat lunak (*software design*) dan penggunaan proyek (*project usage*). Pada penelitian tersebut telah menggunakan lima puluh makalah penelitian sebagai referensi untuk mencakup semua metrik yang ada dalam kategori yang telah disebutkan. Dengan menggunakan penelitian survei tersebut diharapkan dapat memperkuat penggunaan dan keterbatasan dari beberapa metrik, sehingga dapat menemukan suatu metrik yang cocok untuk tujuan aplikasi yang akan dikembangkan. Pada penelitian tersebut pengukuran bisa dilakukan secara langsung atau tidak langsung. Pengukuran langsung produk perangkat lunak meliputi: baris kode sumber, jumlah cacat yang terdeteksi, eksekusi kecepatan perangkat lunak, throughput per jam, waktu respons dan waktu penyelesaian. Pengukuran tidak langsung dari perangkat lunak merupakan efisiensi, portabilitas, kelengkapan, kesederhanaan, pemeliharaan, mudah dipahami. Berikut ini merupakan gambar dari klasifikasi metrik pada penelitian tersebut.



Gambar 2. 1 Klasifikasi software metrics (Sharma, 2022)

Klasifikasi terbagi menjadi dua yaitu Design Based Metrics dan Project Based Metrics. Design Based Metrics berbasis pada perangkat lunak fitur desain produk. Dua jenis utama dari metrik berbasis desain bersifat tradisional dan berorientasi objek. Tradisional Metriks adalah metrik yang diterapkan pada pengembangan fungsional konvensional, contohnya seperti LOC(*Line of Code*) dan CC(*Cyclomatic Complexity*). Object Oriented Metriks biasanya berhubungan dengan pengukuran dari sifat OOP, misalnya *Depth of Inheritance Tree* (DIT), *Number of Children*(NOC), *Coupling between Objects Classes*(CBO), *Response for a Class*(RFC). Project Based Metrics adalah jenis metrik yang berfokus pada kualitas produk yang dapat dikirim, juga kualitas proses dan sumber daya yang terlibat dalam membangun suatu produk. Biasanya berkaitan dengan kualitas dasar produk seperti ukuran produk(*size*), fitur desain(*design feature*), kompleksitas, dan kinerjanya (*performance*).

Terdapat juga penelitian yang berjudul “Analisis Perbandingan Bahasa Pemrograman PHP Laravel dengan PHP Native pada Pengembangan Website”

bertujuan untuk menguji dan mengkaji pengembangan website menggunakan framework Laravel dan menggunakan PHP Native untuk mendapatkan kelebihan dan kekurangan (Endra et al., 2021). Pada penelitian tersebut terdapat beberapa parameter metircs yang akan diadopsi pada penelitian ini diantaranya: jumlah baris kode, performa menggunakan chrome sebagai alat uji, Struktur Folder dan Arsitektur MVC dan OOP didalamnya.

Penelitian yang berjudul “Komparasi Penggunaan Framework CodeIgniter Vs PHP Native pada Sistem Informasi Manajemen Surat Sekertariat DPRD Pematang” bertujuan untuk mengetahui perbandingan Framework CodeIgniter dan PHP native yang diterapkan pada sistem informasi manajemen surat sekertariat DPRD Pematang (Padmanaba et al., 2020). Untuk bahan yang digunakan sebagai penelitian tersebut berupa data surat keluar, surat masuk, dan surat disposisi. Dan terdapat beberapa parameter untuk menguji sistem, diantaranya berupa komparasi implementasi koneksi database, performa, implementasi CRUD, dan requests per second.

Penelitian lain dari (Samra, 2015) yang berjudul “Comparing Performance of Plain PHP and Four of Its Popular Frameworks” bertujuan untuk mengevaluasi kinerja dari empat PHP popular framework Laravel, Symfony, CodeIgniter dan Phalcon bersama-sama dengan PHP biasa Eksperimen dilakukan dalam dua sesi. Sesi pertama membahas waktu eksekusi dan jejak tumpukan pengukuran sedangkan yang kedua meliputi pengukuran konsumsi penggunaan memori. Hasil dari percobaan yang telah dianalisis dan ditafsirkan untuk menjadi dasar keputusan dan kesimpulan penelitian ini untuk mengekspos kinerja kerangka kerja yang ditargetkan. Hasil eksperimen membuktikan bahwa target kerangka kerja berkinerja

berbeda dibandingkan satu sama lain dan PHP tumpukan. PHP dan Phalcon biasa berkinerja baik sementara tiga lainnya kerangka kerja memiliki kinerja yang biasa-biasa saja dan rendah.

Beberapa penelitian yang telah dilakukan sebelumnya akan diadopsi beberapa pengujian yang serupa namun dalam lingkup yang berbeda, dalam kasus ini yaitu pada penelitian (Wohlgethan, 2018) yang membahas pengujian salah satu framework paling menonjol (Angular, React, Vue) akan dikembangkan menjadi perbandingan *stack* MEAN, MERN, dan MEVN yang nantinya akan dibuat sebuah sistem informasi akademik sebagai media uji untuk dilakukan perbandingan. Dengan melakukan perbandingan maka dibutuhkan cara untuk mengukur sifat atau *property* internal dan eksternal pada suatu perangkat lunak (*framework*), dengan menggunakan penelitian dari (Sharma, 2022) metrik merupakan cara terbaik untuk mengukur sifat atau atribut pada suatu perangkat lunak. Beberapa metrik yang telah disebutkan pada penelitian (Sharma, 2022) akan dipilih *software metric* yang sesuai dengan penelitian yang akan dibuat. Penelitian dari (Endra et al., 2021) merupakan contoh umum *software metric* yang bisa digunakan untuk penelitian ini, diantaranya: jumlah baris kode (LOC), performa menggunakan chrome sebagai alat uji, Struktur Folder dan Arsitektur MVC dan OOP pada setiap *framework* yang diuji. Adapun *metric* yang lain seperti pada penelitian (Samra, 2015) perbandingan penggunaan memory, jumlah fungsi yang dipanggil setiap aksi CRUD, dan waktu eksekusi dari fungsionalitas CRUD. Penelitian (Padmanaba et al., 2020) akan dilakukan komparasi yang serupa diantaranya pengukuran performa dengan mengimplementasikan CRUD pada sistem informasi akademik.

Commented [KWP10R9]: OK

Commented [K9]: Di bagian akhir kajian literatur ini, jelaskan perbedaan/kebaruan antara penelitian yang Anda lakukan jika dibandingkan dengan penelitian-penelitian tersebut di atas.

2.2 Teori Terkait

2.2.1 Tech stack

Dilansir oleh Heap (Topics: Heap, 2021), *tech stack* adalah kombinasi teknologi yang digunakan perusahaan untuk membangun dan menjalankan aplikasi atau proyek. Kadang-kadang disebut “tumpukan solusi,” tumpukan teknologi biasanya terdiri dari bahasa pemrograman, *framework*, database, alat front-end, alat back-end, dan aplikasi yang terhubung melalui API.

Adapun definisi dari tech stack yang dilansir oleh Stackshare (Stackshare, 2021) yaitu sebagai kumpulan teknologi yang digunakan organisasi untuk membangun web atau aplikasi seluler. Ini adalah kombinasi dari bahasa pemrograman, *framework*, perpustakaan, pola, server, solusi UI/UX, perangkat lunak, dan alat yang digunakan oleh pengembangnya.

2.2.1.1 JavaScript

JavaScript bisa dibilang milik dunia bahasa pemrograman yang paling banyak digunakan. Menurut survei Stack Overflow pada tahun 2018 dari 73.248 responden JavaScript digunakan oleh 71,5% pengembang profesional menjadikannya bahasa yang paling banyak digunakan di dunia pemrograman dilansir oleh (Wirfs-Brock & Eich, 2020). Pada tahun ini JavaScript menyelesaikan tahun kesembilan berturut-turut sebagai bahasa pemrograman yang paling umum digunakan. Bagi sebagian besar pengembang, pemrograman adalah pemrograman web. Python bertukar tempat dengan SQL untuk menjadi bahasa terpopuler ketiga dilansir oleh (Insight Stack Overflow survey 2021, 2021).

2.2.1.2 TypeScript

TypeScript adalah bahasa pemrograman berorientasi objek yang dibangun di atas JavaScript. Menurut Boris Cherny (Black, 2020) TypeScript adalah bahasa pemrograman yang dikompilasi ke JavaScript dibuat oleh Microsoft, dengan kontribusi dari Google. Dengan menggunakan TypeScript akan membuat kode lebih mudah dibaca dan di *debug* dibandingkan menggunakan JavaScript. TypeScript juga merupakan bahasa pemrograman yang *Strongly Type* seperti Java, C#, dan C/C++.

2.2.2 Framework

Framework adalah sekumpulan fungsi, class, dan aturan-aturan yang bersifat menyeluruh mengatur bagaimana *developer* membangun aplikasi. Dengan menggunakan Framework memungkinkan *developer* untuk membangun aplikasi dengan lebih cepat karena *developer* dapat lebih memfokuskan pada pokok permasalahan (Ruli Erinton, Ridha Muldina Negara, 2017).

2.2.2.1 Front-end

Menurut Ilham Yusron & Antony Wibowo, Front-End dari sebuah situs web adalah bagian yang langsung terlihat oleh pengguna (Yusron & Wibowo, 2020). Dilansir dari medium (Makers Institute, 2017), *User* dapat berinteraksi secara langsung pada bagian ini dan biasanya dibangun menggunakan HTML, CSS, dan JavaScript.

Maka dari itu dapat diartikan bahwa frontend framework merupakan sekumpulan fungsi, class, dan aturan-aturan yang bersifat menyeluruh yang digunakan untuk mengatur bagian yang langsung dilihat oleh *user (client-side)*.

Ada banyak *framework* dan pustaka Front-End berdasarkan JavaScript karena untuk inovasi mesin V8 (Yusron & Wibowo, 2020). Berikut ini merupakan beberapa *framework front-end* yang akan digunakan pada penelitian ini:

1. Angular 13

Menurut (Hutagikar & Hegde, 2020), *framework* ini pertama kali diterbitkan pada tahun 2009, dan telah berganti nama menjadi AngularJS. Meskipun demikian, saat ini istilah ini saja mencerminkan pembaruan sebelumnya ke versi 2.0, versi ini bernama Angular yang dibangun menggunakan TypeScript.

Dikutip dari dokumentasi Angular *Official*, Angular adalah platform pengembangan, dibangun di atas TypeScript. Sebagai platform, Angular mencakup:

- *Framework* berbasis komponen untuk membangun aplikasi web yang skalabel
- Kumpulan *library* yang terintegrasi dengan baik yang mencakup berbagai fitur, termasuk perutean, manajemen formulir, komunikasi klien-server, dan banyak lagi
- Serangkaian alat pengembang untuk membantu *developer* mengembangkan, membangun, menguji, dan memperbarui kode *developer*.

Menurut R.Soni (Soni, 2017), AngularJS adalah library yang ditulis dalam JavaScript untuk pengembangan aplikasi web, dikelola oleh Google, ini adalah *framework* JavaScript sumber terbuka dan membahas tantangan aplikasi satu halaman (SPA).

Pada Aplikasi web AngularJS mengikuti Pola desain MVC, menghasilkan pengembangan aplikasi web yang dapat diperluas, dapat dipelihara, diverifikasi,

dan distandarisasi. Namun pada Angular 2, pola desain diubah menjadi MVVM (Model-View-ViewModel) yang memungkinkan pengembang untuk beroperasi dengan kumpulan data yang sama secara terpisah pada aplikasi yang sama (Hutagikar & Hegde, 2020).

2. React.js

Dikutip dari dokumentasi React *Official*, React merupakan *library* JavaScript yang deklaratif, efisien, dan fleksibel untuk membangun antarmuka pengguna. React memungkinkan *developer* untuk membuat antarmuka kompleks dari kumpulan kode yang kecil dan terisolasi yang disebut **komponen** (Facebook Inc, 2020).

Commented [JP11]: Kutipan diawali nama penelitiannya

React.js digunakan sebagai pendukung dari web-framework (Wali & Ahmad, 2018). React.js memiliki beberapa keunggulan diantaranya memberikan kecepatan, *simplicity*, dan *scalability* (Panjaitan & Pakpahan, 2021). React.js memungkinkan pengembang dapat membangun sebuah komponen UI yang lebih interaktif, *stateful*, & *reusable* (Panjaitan & Pakpahan, 2021).

3. Next.js

Dilansir dari medium (Agae, 2021) Next.js adalah sebuah React framework yang dibuat untuk mengatasi masalah *client-side rendering* yang dimiliki React. Untuk membangun aplikasi web lengkap dengan React dari awal, ada banyak detail penting yang perlu *developer* pertimbangkan:

- Kode harus dibundel menggunakan bundler seperti webpack dan diubah menggunakan kompiler seperti Babel.
- *Developer* perlu melakukan optimasi produksi seperti pemecahan kode.

- *Developer* mungkin ingin melakukan pra-render beberapa halaman secara statis untuk kinerja dan SEO. *Developer* mungkin juga ingin menggunakan rendering sisi server atau rendering sisi klien.
- *Developer* mungkin harus menulis beberapa kode sisi server untuk menghubungkan aplikasi React *developer* ke penyimpanan data *developer*.

Sebuah *framework* dapat memecahkan masalah ini. Tetapi *framework* seperti itu harus memiliki tingkat abstraksi yang tepat, jika tidak maka tidak akan terlalu berguna. Itu juga perlu memiliki "Pengalaman Pengembang" yang hebat, memastikan *developer* dan tim memiliki pengalaman yang luar biasa saat menulis kode.

NextJs memberikan solusi untuk semua masalah tersebut saat membangun aplikasi React Next.js bertujuan untuk memiliki pengalaman pengembang terbaik di kelasnya dan banyak fitur bawaan, seperti:

- Sistem perutean (*routing system*) berbasis halaman yang intuitif (dengan dukungan untuk rute dinamis)
- Pra-rendering, baik generasi statis (SSG) dan rendering sisi server (SSR) didukung pada basis per halaman
- Pemisahan kode otomatis untuk memuat halaman lebih cepat
- Perutean sisi klien (*client-side routing*) dengan pengambilan awal yang dioptimalkan
- Dukungan CSS dan SaSS *support*, dan dukungan untuk pustaka CSS-in-JS apa pun
- Lingkungan pengembangan dengan dukungan Fast Refresh
- Rute API untuk membangun titik akhir API dengan Fungsi Tanpa Server

- Dapat diperpanjang sepenuhnya (*fully extendable*)

4. Vue.js

Vue.js adalah *framework* sistematis untuk pengembangan antarmuka pengguna yang tidak seperti arsitektur monolitik seperti Angular, Vue dirancang agar dapat diadopsi oleh pengguna secara bertahap (Hutagikar & Hegde, 2020). Vue adalah seperangkat *progressive JavaScript framework* untuk membangun antarmuka pengguna, dan penggerak datanya diwujudkan melalui mode MVVM (Li & Zhang, 2021). Pada penelitian (Pšenák & Tibenský, 2020) menilai bahwa Vue.js merupakan *framework* yang *user-friendly* dan mudah digunakan untuk belajar mengembangkan web sebagai pemula. Beberapa keuntungan menggunakan Vue.js, diantaranya:

- Vue.js cepat dan sangat ringan. Ini menawarkan manfaat yang mencakup efisiensi yang lebih besar untuk sumber daya terbatas komputer (Hutagikar & Hegde, 2020)
- Dokumentasi yang bagus dan mudah dimengerti (Pšenák & Tibenský, 2020).
- *Developer* dapat mengintegrasikan secara bertahap pada halaman web yang sudah ada dan tetap menjaga agar halaman web yang lama tetap berfungsi sehingga perusahaan yang sudah memiliki website sebelumnya dapat mempertimbangkan pembaruan untuk masa mendatang (Pšenák & Tibenský, 2020).

2.2.2.2 Back-end

Backend adalah ruang logis dengan fungsional dan operasi dari aplikasi perangkat lunak atau sistem informasi. Salah satu implementasinya adalah sistem *cashless*. Semakin berkembangnya *cashless* dan *electronic payment*, kemudian

masyarakat retail yang belum menerapkannya harus memiliki *electronic data capture* terutama untuk melengkapi teknologi *cashless*. Melalui penelitian yang berjudul “Backend Server System Design Based on REST API for Cashless Payment System on Retail Community” telah membangun sebuah sistem teknologi *cashless* sebagai backend server berbasis REST API (Adam et al., 2019). Maka dari itu dapat diartikan bahwa backend framework merupakan sekumpulan fungsi, class, dan aturan-aturan yang digunakan untuk mengatur bagian tidak terlihat oleh pengguna secara langsung (*server-side*).

1. Node.js

Node.js yang dikembangkan oleh Ryan Dahl pada awal 2009 sebagai platform berbasis JavaScript open-source untuk membangun aplikasi server yang akan mampu menangani sejumlah besar klien secara bersamaan koneksi. Node.js menyediakan lingkungan pemrograman JavaScript dengan *library* yang mengekspos model I/O asinkron yang meresap. Ini menggabungkan antarmuka POSIX umum dengan JavaScript callback dan versi sederhana dari *event loop* browser. Implementasinya terdiri dari *Google's V8 JavaScript engine* dibungkus untuk penggunaan mandiri, pemuat Modul CommonJS, dan aset modul yang diimplementasikan C yang menyediakan versi non-blocking dari POSIX API dan lainnya operasi file dan jaringan tingkat yang lebih tinggi (Wirfs-Brock & Eich, 2020). Node.js menyediakan sejumlah fitur penting untuk mengembangkan aplikasi web termasuk protokol jaringan seperti HTTP. Ini juga mendukung pemasangan modul pihak ketiga yang membuat penggunaan npm (Node Package Manager) (Aggarwal & Verma, 2018). Node.js bersifat *asynchronous, event-driven engine*.

2. Express.js

Express.js adalah framework web minimalis yang sangat cepat dan *unopinionated* untuk Node.js dan merupakan aplikasi web framework untuk Node.js. Ini menyediakan banyak fitur yang membuat pengembangan aplikasi web super cepat dan mudah yang sebaliknya membutuhkan lebih banyak waktu hanya dengan menggunakan Node.js (Expressjs Official, 2021). Express.js dirancang menggunakan Node.js *middleware module* disebut *connect* yang mengimplementasikan *http module*. Selanjutnya sebagian besar modul *middleware* mendukung penyatuan dengan ExpressJS (Aggarwal & Verma, 2018).

2.2.3 Database

Basis data adalah kumpulan catatan atau data terstruktur yang disimpan dan dikelola sedemikian rupa berdasarkan ketentuan tertentu sehingga pengguna dapat dengan mudah mencari informasi, menyimpan informasi dan membuang informasi. Database relasional telah menjadi pilihan database de-facto di sebagian besar aplikasi TI. Dalam dekade terakhir telah terjadi peningkatan permintaan untuk aplikasi yang harus berurusan dengan data yang besar dan tidak dinormalisasi. Untuk memenuhi permintaan, ada perubahan besar untuk menggunakan database yang lebih santai dibentuk database NoSQL. Seiring dengan pergeseran ini, ada kebutuhan untuk memiliki metodologi yang terstruktur untuk mengubah data yang ada dalam database relasional (RDB) ke database NoSQL. Transformasi dari RDB ke database NoSQL menjadi lebih menantang karena tidak ada standar saat ini pada database NoSQL. Aturan transformasi Skema RDB ke berbagai NoSQL skema database, yaitu database berbasis dokumen, berbasis kolom dan berbasis grafik. Aturan diterapkan berdasarkan jenis hubungan yang dapat muncul dalam data

dalam database. Data dalam NoSQL berbasis dokumen disimpan sebagai dokumen, yang masing-masing diidentifikasi oleh tertentu kunci. Dokumen dikelompokkan ke dalam koleksi yang disimpan secara berurutan, dengan dokumen baru dapat ditambahkan ke dalam koleksi setiap saat. Ada dua cara untuk memodelkan hubungan dalam NoSQL berbasis dokumen, yang merujuk dan menyematkan. Referensi mirip dengan RDB ketika ID dokumen pengguna menjadi *foreign key* didokumen lain. Untuk penyematan, dokumen terkandung dalam dokumen lain dan dapat diakses bersama-sama (Alotaibi & Pardede, 2019).

MongoDB adalah database dokumen *open source* yang menyediakan persistensi untuk data aplikasi dan dirancang dengan keduanya skalabilitas dan kelincahan pengembang (Aggarwal & Verma, 2018). MongoDB termasuk tipe database NoSQL, *cross platform, document-oriented*. MongoDB data adalah dokumen seperti JSON yang menyimpan catatan data sebagai dokumen yang dikumpulkan bersama dalam koleksi. Penyimpanan database dapat memiliki satu atau lebih koleksi dokumen. Keuntungan menggunakan MongoDB adalah datanya bidang dapat bervariasi dari dokumen ke dokumen dan struktur data dapat diubah dari waktu ke waktu. Ini mendukung dinamis kueri pada dokumen termasuk bahasa sumber berbasis dokumen yang hampir sekuat SQL.

2.2.4 Deployment

Menurut Iman Tumorang, “Deployment adalah kegiatan yang bertujuan untuk menyebarkan aplikasi yang telah dikerjakan oleh para pengembang. Penyebarannya dapat melalui beragam cara tergantung dari jenis aplikasinya (Tumorang, 2019).” Pada penelitian ini akan menggunakan dua deployment yaitu *frontend* dan *backend*.

2.2.4.1 Vercel

Vercel adalah platform untuk *developer* frontend, memberikan kecepatan dan keandalan yang dibutuhkan inovator untuk menciptakan pada saat inspirasi (Vercel Inc, 2022). Pada penelitian ini, Vercel akan dimanfaatkan sebagai layanan hosting gratis untuk website yang bersifat statis, dalam hal ini aplikasi frontend seperti Angular, React, Vue akan di-*deploy* menggunakan Vercel. Untuk menjadikan *web static* menjadi *web dynamic*, maka diperlukannya API (*Application Programming Interface*) untuk mengambil data dari database.

2.2.4.2 Heroku

Heroku adalah platform cloud yang memungkinkan perusahaan membangun, mengirimkan, memantau, dan menskalakan aplikasi (Salesforce, 2022). Heroku adalah sebuah cloud platform yang menjalankan bahasa pemrograman tertentu, Heroku mendukung bahasa pemrograman seperti Ruby, Node.js, Python, Java, PHP, dan lain-lain (Hariyadi, 2017). Beberapa manfaat dari Heroku, diantaranya:

- Developer ingin melakukan live test aplikasi tapi tidak menggunakan domain terlebih dulu karena untuk testing dan masih dalam tahap developing ada testing online sebelum menuju production (Hariyadi, 2017).
- Developer ingin demonstrasikan app dengan client nya sehingga harus *online*-kan, namun developer tidak punya vps untuk mencoba app (Hariyadi, 2017).
- Vps tidak mempunyai service atau library yang dibutuhkan oleh app (Hariyadi, 2017).

Pada penelitian ini, Heroku akan dimanfaatkan sebagai layanan deployment backend aplikasi, sehingga API pada backend dapat diakses secara public.