

BAB 2

LANDASAN TEORI

2.1 Rancang Bangun Aplikasi

Rancang bangun suatu aplikasi dimiliki bertujuan untuk merancang dan membangun sistem – sistem yang baru untuk menyelesaikan permasalahan yang dialami oleh para pelaku usaha sebagai pilihan jalan alternatif menjadi sistem yang lebih baik. Rancang bangun adalah proses pembangunan sistem untuk menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun hanya sebagian (Yuntari. 2017).

Penelitian ini merancang dan membangun untuk pelaku usaha dalam mempermudah manajemen arsip karyawan dan pengajuan dari karyawannya untuk membuat permohonan – permohonan ke HRD di perusahaan tersebut. Nama aplikasi ini bernama “BBase” dalam kepanjangan “Business Base” adalah aplikasi ini basis bagi operasional bisnis yang terkait dengan manajemen karyawan seperti arsip – arsip karyawan, menyimpan pengajuan – pengajuan karyawan, pesan khusus dari perusahaan dan pengumuman yang tersimpan dalam server perusahaan. Pemilihan perancangan dan membangun aplikasi ini lebih efisien dengan penggunaan lintas platform berbasis android dan ios yang lebih murah untuk membangun aplikasi dengan sumber 1 *file project* kode programnya.

2.2 Ionic

Ionic adalah seperangkat alat antarmuka pengguna sumber terbuka untuk membangun aplikasi seluler yang berkualitas tinggi dan kinerja tinggi menggunakan teknologi web HTML, CSS, dan JavaScript, dengan integrasi *frameworks* seperti angular, react dan vue. Ionic adalah Kerangka Pengembangan Aplikasi Mobile berbasis HTML5. Kerangka ini mengelola tampilan dan kesan antarmuka pengguna (UI) dari aplikasi lintas platform[8].

Ionic salah satu pilihan menarik untuk pengembangan aplikasi lintas platform yang lebih efisien karena ionic mendapatkan keuntungan dari kode yang dapat digunakan dari framework ionic. Pengembangan menggunakan ionic dapat menggunakan kode pemrograman satu sumber untuk menghasilkan aplikasi yang

dapat di jalankan berbagai platform seperti android dan ios, sehingga dapat mengurangi kerumitan dalam pengembangan dan mempersingkat waktu baik dari pembaruan, pengembangan dan memelihara yang secara lintas platform.

2.3 React TypeScript

React TypeScript adalah suatu bahasa yang mengkompilasi dari JavaScript, pengguna TypeScript sudah tersedia di react dan node js. TypeScript bagian dari superset dari JavaScript yang dimaksud dapat digunakan dalam semua fitur yang tersedia di JavaScript plus yang berisikan fitur tambahan dari TypeScript. Dalam pengembangan memakai React TypeScript akan memudahkan memulai proses dari memanfaatkan fitur seperti kelas, fungsi, dan string tanpa perlu mengompilasi. React adalah memungkinkan pengembang untuk membuat aplikasi web berbasis SPA yang besar, cepat, skalabel, dan sederhana, yang dapat mengubah data tanpa memuat ulang halaman. React jauh lebih canggih dan lebih cepat daripada situs web biasa yang ditulis dengan HTML biasa[9].

Dalam segi bahasa pemrograman, TypeScript lebih aman untuk mengetik dan memiliki pemeriksaan kesalahan waktu kompilasi yang lebih baik. TypeScript lebih ke pengembangan pemrograman yang menggunakan fitur pemrograman berorientasi objek modern seperti kelas dan modul. TypeScript juga memiliki pemeriksaan yang bertipe statis yang berarti kompiler akan bekerja mendeteksi kalau ada kesalahan dalam penulisan kode – kode yang dibuat. Pemrograman TypeScript memiliki persamaan dalam sintaks di JavaScript tetapi memiliki fitur – fitur tambahan seperti tipe antarmuka, generik, dan *type inference* (yang berarti tidak perlu untuk mendefinisikan tipe data pada suatu variabel karena akan otomatis menterjemakan tipe data yang dapat digunakan).

2.4 SQL

Kepanjangan dari sql adalah *structured query language*. Database SQL (Structured Query Language) adalah sistem manajemen basis data relasional yang menggunakan SQL sebagai bahasa untuk mengelola dan memanipulasi data. SQL memungkinkan pengguna untuk membuat, membaca, memperbarui, dan menghapus data dalam basis data dengan cara yang terstruktur dan efisien[10]. Sql berfungsi sebagai media penyimpanan suatu data, maka sql digemari oleh

pemrograman sebagai pengelola data. Sql juga memiliki respons yang cepat dalam memproses data, kode perintah sql terbilang sederhana, sql juga bersifat *portability* yang berarti mudah dipindahkan, dan bersifat *multiple data view*.

Jenis – jenis tipe data sql digunakan sebagai menentukan jenis data yang diberikan pada setiap kolom tabel data. Pada tiap – tiap kolom harus memberikan jenis datanya yang akan digunakan. Contoh tipe data sql yang sering digunakan adalah int untuk bilangan bulat, float untuk tipe bilangan desimal, varchar untuk menyimpan karakter tulisan sebanyak 255 karakter.

2.5 TailWind CSS

Tailwind CSS adalah kerangka kerja CSS berbasis utilitas yang dirancang untuk pengembangan UI yang cepat. Alih-alih menyediakan komponen yang sudah dirancang sebelumnya, Tailwind CSS menawarkan serangkaian kelas utilitas yang komprehensif yang dapat digabungkan untuk membangun desain apa pun langsung di HTML Anda. Pendekatan ini memungkinkan fleksibilitas dan kustomisasi yang lebih besar, karena pengembang dapat menata elemen mereka dengan kontrol detail menggunakan nama kelas yang sederhana dan deskriptif. Tailwind CSS mempromosikan metodologi yang berbeda dibandingkan dengan kerangka kerja CSS tradisional dengan mendorong sistem desain berbasis kelas atom yang dapat digunakan kembali, bukan *stylesheet* khusus atau pustaka komponen.

Salah satu keuntungan utama Tailwind CSS adalah kemampuannya untuk merampingkan proses pengembangan dengan secara signifikan mengurangi kebutuhan akan CSS khusus. Dengan menggunakan kelas utilitas, pengembang dapat mencapai tingkat konsistensi dan keterluasan yang tinggi dalam desain mereka. Pendekatan ini tidak hanya mempercepat alur kerja pengembangan tetapi juga membuatnya lebih mudah untuk memelihara dan memodifikasi gaya. Orang-orang memilih menggunakan Tailwind CSS dibandingkan teknologi serupa lainnya karena beberapa alasan yang meyakinkan. Pertama, pendekatan berbasis utilitas Tailwind menawarkan fleksibilitas dan kontrol yang tak tertandingi. Tidak seperti kerangka kerja CSS tradisional yang dilengkapi dengan komponen yang sudah ditentukan sebelumnya, Tailwind menyediakan kelas utilitas tingkat rendah yang dapat digabungkan untuk membuat desain apa pun langsung di HTML. Ini menghilangkan kebutuhan untuk mengesampingkan gaya atau menulis CSS khusus

untuk komponen tertentu, menghasilkan kode yang lebih bersih dan lebih mudah dikelola.

2.6 Laravel

Laravel adalah kerangka kerja PHP sumber terbuka yang banyak digunakan dan mengikuti pola arsitektur Model-View-Controller (MVC). Kerangka ini dibuat oleh Taylor Otwell pada tahun 2011 dengan tujuan menyediakan alternatif yang lebih canggih untuk kerangka kerja CodeIgniter, yang kekurangan fitur penting seperti autentikasi dan otorisasi pengguna bawaan. Laravel dirancang untuk mempermudah dan meningkatkan efisiensi tugas pengembangan web dengan menyediakan serangkaian alat dan fitur yang kaya, termasuk sintaks yang elegan, ORM (Object-Relational Mapping) yang kuat bernama Eloquent, dukungan bawaan untuk caching, routing, sesi, dan sistem pengemasan modular dengan manajer dependensi khusus.

Salah satu keunggulan utama Laravel adalah sintaksnya yang ekspresif dan elegan, yang mudah dipahami dan mendukung pengembangan cepat. Ini membuat Laravel sangat populer di kalangan pengembang yang menghargai kode yang bersih dan mudah dipelihara. Laravel juga memiliki ekosistem yang komprehensif dengan alat-alat seperti Laravel Forge, layanan penyebaran otomatis dan manajemen server, serta Laravel Vapor, platform penyebaran tanpa server. Selain itu, mesin templating Blade Laravel menyediakan cara yang sederhana namun kuat untuk membangun aplikasi web dinamis, memudahkan pengembang untuk mengintegrasikan logika backend dengan desain frontend secara mulus.

2.7 PHP

PHP, atau *Hypertext Preprocessor*, adalah bahasa pemrograman sisi server yang banyak digunakan yang dirancang terutama untuk pengembangan web. Ia tertanam dalam HTML, memungkinkan pengembang untuk membuat halaman web dinamis dan konten interaktif secara mulus. Salah satu keunggulan utama PHP adalah kemudahannya dalam penggunaan dan aksesibilitasnya. Sintaks sederhana dan mengingatkan pada bahasa gaya C, sehingga relatif mudah bagi pengembang untuk mempelajarinya dan menggunakannya. Selain itu, PHP bersifat *open – source* dan didukung oleh komunitas pengembang yang luas di seluruh dunia,

berkontribusi pada perbaikan terus-menerus dan penerimaan yang luas.

Banyak pengembang memilih PHP karena fleksibilitas dan cakupan aplikasinya yang luas. Ia dapat digunakan untuk berbagai tujuan, termasuk *scripting* sisi server, *scripting* baris perintah, dan bahkan aplikasi desktop. Perpustakaan standar PHP yang luas dan kompatibilitasnya dengan berbagai sistem operasi dan server web membuatnya menjadi pilihan yang fleksibel untuk membangun berbagai jenis proyek. Selain itu, integrasi PHP dengan *database* populer seperti *MySQL* dan *PostgreSQL* memudahkan interaksi yang mulus dengan *database*, menjadikannya cocok untuk membangun aplikasi web dinamis dan berbasis data.

PHP telah menjadi kunci dalam pengembangan banyak aplikasi web dan platform terkenal. Misalnya, *WordPress*, sistem manajemen konten (CMS) terpopuler di dunia, sebagian besar dibangun dengan menggunakan PHP. Ia menggerakkan jutaan situs web dan blog di seluruh dunia, memperlihatkan skalabilitas dan kekuatan PHP. Selain itu, platform media sosial populer seperti Facebook dan Twitter awalnya menggunakan PHP dalam tahap awal pengembangannya, menunjukkan kemampuannya untuk menangani situs web dengan skala besar dan lalu lintas tinggi. Prestasi-prestasi ini menegaskan pentingnya PHP dalam membentuk lanskap web modern dan relevansi yang berkelanjutan di ranah pengembangan web.

2.8 User Experience Questuinnaire

User Experience Questuinnaire (UEQ) adalah alat berdasarkan ISO 9241-11 untuk mengukur pengalaman pengguna akhir saat berinteraksi dengan suatu produk. Kuesioner ini terdiri dari 6 skala pengukuran dengan 26 item. Setiap item merupakan diferensial semantik, diwakili oleh 2 kata yang berlawanan, dan diukur dari -3 (paling negatif) hingga +3 (paling positif), dengan 0 berarti netral. Setelah pengguna selesai berinteraksi dengan produk, kuesioner diberikan untuk menghindari pernyataan abstrak atau mengharuskan pengguna mengingat detail. Nilai rata-rata setiap item dari setiap skala kemudian dibandingkan dengan benchmark UEQ untuk mengetahui kualitas desain. Suatu desain dikatakan baik pada setiap skala apabila nilai rata-ratanya diatas 0,8. Berikut merupakan tolak ukur UEQ:

1. **Attractiveness**: Kesan keseluruhan terhadap produk (menjengkelkan / menyenangkan, baik / buruk, dll.).
2. **Perspicuity**: Kemudahan penggunaan (cepat / lambat, efisien / tidak efisien, dll.).
3. **Efficiency**: Kemudahan penyelesaian tugas (tidak dapat dipahami / dimengerti, dll.).
4. **Dependability**: Kontrol pengguna atas interaksi (tidak dapat diprediksi/dapat diprediksi, dll.).
5. **Stimulation**: Kenikmatan pengguna saat berinteraksi (berharga / rendah, membosankan / menyenangkan, dll.).
6. **Novelty**: Daya tarik produk (kreatif / membosankan, inovatif / konvensional, dll.).

2.9 Usability Testing

Usability testing merupakan teknik yang digunakan untuk mengevaluasi kemampuan produk dalam memberikan kemudahan navigasi, penggunaan, dan pemahaman. Usability testing melibatkan pengguna dalam menyelesaikan tugas-tugas melalui skenario yang telah ditentukan, dan metode ini dapat diterapkan pada tahap awal perancangan untuk menguji ide-ide baru. Tujuan utama dari usability testing adalah untuk mengidentifikasi masalah, bukan untuk mendemonstrasikan fenomena tertentu, dan masalah-masalah ini dapat ditemukan meskipun dengan jumlah responden yang sedikit[11]. Selama proses pengujian, semua tindakan pengguna diamati dan direkam untuk mengumpulkan data dan mengukur kepuasan pengguna. Wawancara dan kuesioner juga dapat digunakan untuk mengukur kepuasan pengguna dan membantu menjelaskan tindakan yang dilakukan.

Saat melakukan usability testing, terdapat lima prinsip yang dapat digunakan sebagai panduan untuk mengukur pengalaman pengguna:

1. **Memorability**: Apakah pengguna dapat dengan mudah mengingat cara menyelesaikan suatu tugas dan menggunakan informasi tersebut untuk tugas lain?

2. **Efficiency:** Berapa banyak waktu yang diperlukan pengguna untuk menyelesaikan setiap tugas? Apa yang menghambat penyelesaian tugas?
3. **Errors:** Jika terjadi kesalahan, apakah pengguna dapat dengan mudah memulihkannya?
4. **Learnability:** Apakah pengguna dapat memahami sistem dengan mudah?
5. **Satisfaction:** Apakah pengguna menyukai sistem atau produk yang ditawarkan?

