

BAB III. LANDASAN TEORI

3.1. *Framework*

Framework merupakan kumpulan dari beberapa *script* atau fungsi dasar yang dibuat pada suatu bahasa pemrograman untuk menambah fitur yang dimiliki oleh bahasa pemrograman tersebut. Fungsi pada *framework* dibuat sehingga dengan memanfaatkan fungsi tambahan yang disediakan oleh *framework* tersebut pengembang sistem tidak perlu membuat seluruh bagian sistem dari awal yang membuat pengembang dapat mempersingkat atau mengurangi waktu yang dibutuhkan untuk membuat suatu sistem informasi [4].

Framework harus di pasang pada perangkat yang digunakan untuk membangun sistem agar dapat menjalankan fungsi yang disediakan oleh *framework* karena agar dapat dibaca oleh bahasa program yang digunakan. Sebagian besar bahasa pemrograman memiliki *framework* yang dapat dipasangkan dan dapat dijalankan di sistem operasi *Windows*, *Linux* dan *macOS*.

3.2. *JavaScript*

JavaScript Merupakan bahasa pemrograman yang bersifat *client-side* yang digunakan bersamaan dengan *HTML* dan *CSS* untuk membuat sebuah *website*. *Javascript* pada digunakan sehingga *website* menjadi lebih dinamis dan interaktif seperti membuat suatu fungsi yang dijalankan dan lain-lain. *JavaScript* ditujukan untuk memproses beberapa kegiatan yang terjadi pada sebuah *website* seperti klik *event* serta proses *input-output* sebelum dikirimkan dan diproses pada *server* dan sebaliknya namun dikarenakan *Javascript* yang bersifat *client-side* ini sangat bergantung pada *web browser* pengguna sehingga pengguna dapat mengubah proses yang terjadi pada *javascript*.

Perkembangan saat ini telah membuat *javascript* juga bisa dijalankan secara *server-side* dengan menggunakan *framework nodejs*. Perkembangan ini membuat *javascript* bisa memproses *input-output* pada *server* tanpa terganggu oleh pengguna. *Javascript* digunakan oleh hampir seluruh *website* yang ada dikarenakan ketergantungan dari halaman untuk melakukan proses *input-output* sebelum di proses pada *server-side* serta hampir seluruh *web browser* yang tersedia di berbagai perangkat mendukung *javascript*.

3.3. React Native

React Native merupakan *framework* berbasis *JavaScript* buatan *facebook* yang digunakan untuk membuat sistem informasi berbasis *mobile* untuk *platform android* dan *iOS*. *React Native* dibuat berdasarkan gabungan dari *library React* dan *library native* untuk pembuatan sistem informasi pada *platform mobile* [1][4][7]. Penggunaan *React Native* bertujuan untuk mempersingkat waktu pengembangan sebuah sistem informasi di kedua *platform* hanya dengan satu pengkodean. *React Native* juga dapat memproses *input-output* pada sistem informasi untuk di proses pada *server* dan sebaliknya.

Pembuatan sistem informasi dengan *react native* memerlukan *library* dari *android studio* untuk dapat dijalankan pada *platform android* dan *library* dari *X Code* untuk dapat dijalankan pada *platform ios*. Dengan perbedaan *library* dari kedua *platform* tersebut tentunya membuat beberapa perbedaan yang diperlukan pada fungsi tertentu yang tersedia pada *react native* sehingga dapat memberikan sedikit perbedaan jika diperlukan.

React native dapat dijalankan pada sistem operasi *windows*, *linux* dan *macos* jika ingin menjalankan aplikasi pada *platform android* sesuai dengan ketersediaan *android studio* di ketiga sistem operasi tersebut dan hanya dapat dijalankan pada sistem operasi *macos* dengan aplikasi *X Code* yang hanya tersedia pada sistem operasi *macos* jika ingin menjalankan aplikasi pada *platform ios*.

3.4. *ReactJS*

ReactJS merupakan *library front-end javascript* yang dapat digunakan untuk pembuatan tampilan pada satu halaman atau lebih pada suatu sistem informasi menjadi lebih dinamis dan interaktif. *ReactJS* membuat tampilan dapat menyesuaikan antara tampilan desktop maupun tampilan secara *mobile*.

ReactJS dapat meningkatkan performa, mengurangi kompleksitas, dan memiliki skala yang cukup baik jika sistem menjadi besar. *Reactjs* memiliki beberapa fitur seperti dapat menggunakan fitur pada bahasa javascript untuk pengembangan fitur baru pada sistem dan dapat memudahkan dalam proses debugging [7].

3.5. *Node.js*

Node.js merupakan *runtime environment javascript engine* yang digunakan pada bahasa pemrograman *JavaScript* untuk pembuatan aplikasi *server* agar dapat menjalankan fungsi-fungsi yang dijalankan pada bahasa pemrograman *javascript* untuk keperluan tertentu yang bersifat *server-side* pada suatu *web server*. *Nodejs* bertujuan untuk memproses *input-output* yang terjadi pada *server-side* sebelum diproses oleh aplikasi *client-side*.

Nodejs memiliki *HTTP* protocol sehingga aplikasi *server-side* dapat jalankan sepenuhnya menggunakan *nodejs* sendiri dan dapat menjalankan *server web* secara independent. *Nodejs* dapat dijalankan pada sistem operasi *windows*, *linux*, dan *macos*.

3.6. Application Program Interface

Application Program Interface atau disingkat *API* merupakan sebuah aplikasi yang digunakan untuk mengirim dan menerima perintah dari beberapa perangkat pengguna kesuatu *server*. *API* dapat dibuat dengan hampir seluruh bahasa pemrograman yang tersedia.

Dengan adanya *API* suatu sistem informasi dapat berinteraksi dengan sistem informasi lain untuk tujuan tertentu yang melibatkan kedua sistem aplikasi tersebut seperti transaksi antar dua bank dan lain-lain. Sebuah *API* dapat dijalankan pada sistem operasi *windows*, *linux* dan *macos* sesuai dengan bahasa pemrograman yang digunakan.

3.7. Express.js

Express.js merupakan *framework* yang dibuat dari *Node.js* yang dirancang lebih sederhana dan fleksibel untuk pembuatan *API* yang akan digunakan untuk menghubungkan antara aplikasi pada perangkat pengguna dan *server*. *ExpressJs* bertujuan untuk pembuatan *server* yang dapat dilakukan dengan pendekatan *web*, dengan pendekatan ini *expressjs* memberikan fitur-fitur tambahan pada *server* yang tidak disediakan oleh *nodejs*.

Expressjs digunakan untuk mempersingkat dalam pembuatan kode *server* dan meningkatkan kinerja *server* dengan mengubah strtur pengkodean yang pada *nodejs*.

3.8. PostgreSQL

PostgreSQL merupakan sebuah aplikasi *open source Database Management System (DBMS)* berbasis relasional, yaitu kumpulan data dengan hubungan yang telah ditentukan membentuk satu set tabel dengan kolom dan baris. Tabel menyimpan informasi yang akan di presentasikan pada database. setiap kolom adalah atribut data dan nilai aktual, sedangkan setiap baris adalah nilai dari satu objek. Setiap baris memiliki identitas unik yang disebut kunci utama (Primary Key) dan baris data bisa dihubungkan dengan baris data lain pada database dengan menggunakan kunci asing (Foreign Key).

Sistem *PostgreSQL* dapat dijalankan pada sistem operasi *windows, linux, macOS*. Selain itu *PostgreSQL* memiliki fitur seperti proses pemulihan yang lebih mudah yang dapat membantu dalam mengolah data yang disimpan pada *DBMS* serta biaya perawatan yang lebih murah dikarenakan aplikasi bersifat *open source*.