

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Pada Jurnal Michael Sugiarto yang berjudul Rancang Bangun Sistem Kunci Berbasis Android dan Web menggunakan sebuah *server virtual* dan untuk komunikasi *real-time* menggunakan *Node.js* untuk komunikasi antara arduino *ethernet sheild* dan aplikasi. Arduino *ethernet sheild* akan membuka kunci dengan menggunakan transistor yang berfungsi sebagai saklar dengan perintah dari aplikasi tersebut, tetapi aplikasi ini belum memiliki penyimpanan data, hanya dapat mengecek kondisi pintu saja.

Sedangkan pada jurnal Dedeng Hirawan dan Mochamad Fajar Wicaksono yang berjudul Implementasi Kunci Pintar Berbasis *Smartphone* Android menggunakan sebuah *server local* dari *Raspberry Pi* dan untuk komunikasi data dilakukan hanya satu arah dengan koneksi bluetooth. *Raspberry Pi* dan aplikasi harus terhubung untuk membuka pintu tersebut, *Raspberry Pi* akan membuka pintu dengan menggunakan *relay* sebagai saklar untuk membuka *doorlock* solenoid dan *user* dapat melihat data yang ada di ruangan tersebut.

Berdasarkan jurnal diatas terdapat beberapa persamaan dan perbedaan. Persamaannya yaitu menggunakan *node.js* untuk komunikasi ke *database*. Sedangkan untuk perbedaan hanya pada tampilan *website* saja saya menggunakan *react.js* dan *database* menggunakan *mongoDB*.

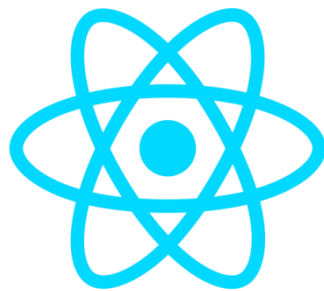
2.2 Pengertian Website

Website adalah sebuah halaman yang menyajikan informasi baik dalam bentuk tulisan, gambar, suara, atau video yang diletakkan di dalam sebuah *server/hosting* di mana untuk mengaksesnya diperlukan jaringan internet (Richardo, 2015).

2.3 Pengertian *React Javascript*

React Js adalah sebuah *library JavaScript* yang di buat oleh *facebook*. *React* bukanlah sebuah *framework MVC*. *React* adalah *library* yang bersifat *composable user interface*, yang artinya kita dapat membuat berbagai UI yang bisa kita bagi menjadi beberapa komponen (Musa : 2018). Dapat dilihat pada gambar 2.1.

Gambar 2.1 Logo *React Javascript*



2.4 Pengertian *Node.JS*

Node.js adalah perangkat lunak yang didesain untuk mengembangkan aplikasi berbasis *web* dan ditulis dalam sintaks bahasa pemrograman *JavaScript*. Bila selama ini kita mengenal *JavaScript* sebagai bahasa pemrograman yang berjalan di sisi *client/browser* saja, maka *Node.js* ada untuk melengkapi peran *JavaScript* sehingga bisa juga berlaku sebagai bahasa pemrograman yang berjalan di sisi *server* (Lutfi : 2017). Dapat dilihat pada gambar 2.2.



Gambar 2.2 Logo *NodeJS*

2.5 Pengertian *MongoDB*

MongoDB adalah salah satu produk *database noSQL Open Source* yang menggunakan struktur data *JSON* untuk menyimpan datanya. *MongoDB* adalah merupakan *database noSQL* yang paling populer di internet. *MongoDB* sering dipakai untuk aplikasi berbasis *Cloud, Grid Computing*, atau *Big Data* (Saputra : 2016). Dapat dilihat pada gambar 2.3.



Gambar 2.3 Logo MongoDB

2.6 Pengertian Material UI

Material-UI adalah proyek sumber terbuka yang menampilkan komponen Bereaksi yang menerapkan Desain Bahan Google. Ini dimulai pada tahun 2014, tidak lama setelah React keluar ke publik, dan telah semakin populer sejak itu. Dengan lebih dari 35.000 bintang di GitHub, Material-UI adalah salah satu pustaka antarmuka pengguna teratas untuk react di luar sana. (Alex : 2018). Dapat dilihat pada gambar 2.4.

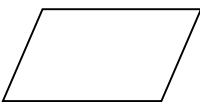
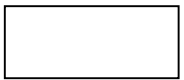
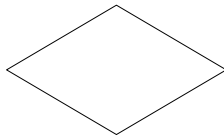
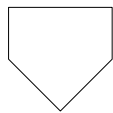
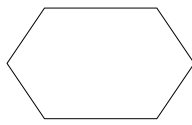


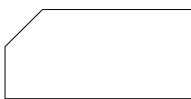
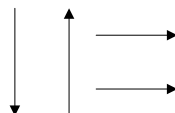
Gambar 2.4 Logo Material UI

2.7 Pengertian Flowchart

Flowchart adalah suatu bagan yang menggambarkan arus logika dari data yang akan di proses dalam suatu program dari awal sampai akhir. Bagan alir program yang berguna bagi *programmer* untuk mempersiapkan program yang lebih kompleks. Bagan alir terdiri dari simbol yang mewakili fungsi-fungsi langkah program dan garis alir yang menunjukkan urutan dari simbol yang akan dikerjakan (Jogiyanto : 2004). Dapat dilihat pada tabel 2.1.

Tabel 2.1 Simbol-simbol *Flowchart*

No	Simbol	Nama	Fungsi
1		<i>Terminal</i>	Menyatakan permulaan atau akhir suatu program.
2		<i>Input/output</i>	Menyatakan proses input atay output tanpa tergantung jenis peralatannya.
3		<i>Proccess</i>	Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.
4		<i>Decision</i>	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban: ya atau tidak.
5		<i>Connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama.
6		<i>Offline Connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda.
7		<i>Predefined process</i>	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.

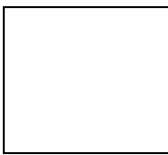
8		<i>Fatched Card</i>	Menyatakan input berasal dari kartu atau output data ke kartu.
9		<i>Document</i>	Mencetak keluaran dalm bentuk dokumen (melalui Printer)
10		<i>Flow</i>	Menyatakan jalannya arus suatu proses

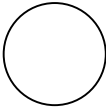
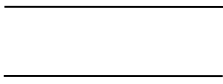
(Sumber: Jogiyanto : 2004).

2.8 Pengertian DFD (*Data Flow Diagram*)

Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. DFD ini sering disebut juga dengan nama Bubble chart, Bubble diagram, model proses, diagram alur kerja, atau model fungsi. (Selvina: 2011). Dapat dilihat pada tabel 2.2.

Tabel 2.2 Simbol-simbol *Data Flow Diagram*

No	Simbol	Nama	Fungsi
1		Terminator	Kesatuan diluar sistem (<i>external entity</i>) yang memberikan input ke sistem atau menerima output dari sistem berupa orang, organisasi atau sistem lain.
2		Aliran Data	Aliran data merupakan perpindahan data dari satu titik ke titik lain.

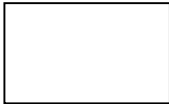
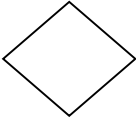
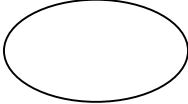
3		Proses	Proses biasanya selalu menunjukkan suatu perubahan data dan terjadi proses transformasi data.
4		<i>Data Store</i>	Penyimpanan data (Data Store) diberi nama dengan kata benda, sesuai dengan data yang disimpan di dalamnya.

(Sumber: Selvina : 2011).

2.9 Pengertian ERD (*Entity Relationship Diagram*)

Entity Relationship Diagram (ERD) merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol. (Selvina: 2011). Dapat dilihat pada tabel 2.3.

Tabel 2.3 Simbol-simbol *Entity Relationship Diagram*

No	Simbol	Nama	Fungsi
1		Entitas	Simbol yang berfungsi untuk memberikan identitas pada entitas yang memiliki label dan nama.
2		Relasi	Simbol yang berfungsi untuk mengetahui jenis hubungan yang ada pada 2 file.
3		Atribut	Karakteristik dari entitas atau relasi yang menyediakan penjelasan detail tentang entitas atau relasi tersebut.
4		Alur	Alur memiliki fungsi untuk menghubungkan atribut dengan entitas dan entitas dengan relasi.

(Sumber: Selvina : 2011).

2.10 *Black Box Testing*

Black Box Testing atau yang sering dikenal dengan sebutan pengujian fungsional merupakan metode pengujian Perangkat Lunak yang digunakan untuk menguji perangkat lunak tanpa mengetahui struktur internal kode atau Program. Dalam pengujian ini, tester menyadari apa yang harus dilakukan oleh program tetapi tidak memiliki pengetahuan tentang bagaimana melakukannya (Kurniawati : 2018).

2.11 Metode Pengembangan Sistem

a. Metode *System Development Life Cycle (SLDC)*

Metode ini adalah metode pengembangan sistem informasi yang pertama kali digunakan makanya disebut dengan metode tradisional. Metode ini *prototype* Adalah tahap-tahapan pekerjaan yang dilakukan oleh analis sistem dan *programmer* dalam membangun sistem informasi.

b. Metode *Waterfall*

Merupakan metode yang sering digunakan oleh penganalisa sistem pada umumnya. Inti dari metode *waterfall* adalah pengerjaan dari suatu sistem dilakukan secara berurutan atau secara linear. Jadi jika langkah ke-1 belum dikerjakan, maka langkah 2 tidak dapat dikerjakan. Jika langkah ke-2 belum dikerjakan maka langkah ke-3 juga tidak dapat dikerjakan, begitu seterusnya. Secara otomatis langkah ke-3 akan bisa dilakukan jika langkah ke-1 dan ke-2 sudah dilakukan.

c. Model *Prototyping*

Prototyping adalah proses *iterative* dalam pengembangan sistem dimana *requirement* diubah ke dalam sistem yang bekerja (*working system*) yang secara terus menerus diperbaiki melalui kerjasama antara *user* dan analis. *Prototype* juga

bisa dibangun melalui beberapa *tool* pengembangan untuk menyederhanakan proses.

d. *Model RAD (Rapid Application Development)*

RAD adalah penggabungan beberapa metode atau teknik terstruktur. *RAD* menggunakan metode *prototyping* dan teknik terstruktur lainnya untuk menentukan kebutuhan *user* dan perancangan sistem informasi selain itu *RAD* menekankan siklus perkembangan dalam waktu yang singkat (60 sampai 90 hari) dengan pendekatan konstruksi berbasis komponen.

e. *Model Spiral*

Model *spiral* pada awalnya diusulkan oleh Boehm, adalah model proses perangkat lunak *evolusioner* yang merangkai sifat *iteratif* dari *prototype* dengan cara kontrol dan aspek sistematis model *sequensial linier*. Model *iteratif* ditandai dengan tingkah laku yang memungkinkan pengembang mengembangkan versi perangkat lunak yang lebih lengkap secara bertahap.

f. *Object Oriented Technology*

Object Oriented Technology merupakan cara pengembangan perangkat lunak berdasarkan abstraksi objek-objek yang ada di dunia nyata. Dasar pembuatan adalah objek, yang merupakan kombinasi antara struktur data dan perilaku dalam satu entitas.

g. *Model Functional Decomposition*

Metodologi ini menekankan pada pemecahan dari sistem ke dalam subsistem-subsistem yang lebih kecil, sehingga akan lebih mudah untuk dipahami, dirancang dan ditetapkan.

Yang termasuk dalam kelompok metodologi ini adalah :

- *HIPO (Hierarchy plus Input Process Output)*
- *Stepwise Refinement (SR)* atau *Iterative Stepwise Refinement (ISR)*
- *Information Hiding*

i. *Prescriptive Methodologies*

Yang termasuk dalam metodologi ini adalah :

ISDOS (Information System Design dan Optimization System), merupakan perangkat lunak yang dikembangkan di University of Michigan. Kegunaan dari *ISDOS* adalah mengotomatisasi proses pengembangan system informasi. *ISDOS* mempunyai dua komponen, yaitu :

1. *PSL (Program Statement Language)*, merupakan komponen utama dari *ISDOS*, yaitu suatu bahasa untuk mencatat kebutuhan pemakai dalam bentuk machine readable form. *PSL* dirancang sehingga output yang dihasilkannya dapat dianalisis oleh *PSA*. *PSL* merupakan bahasa untuk menggambarkan sistemnya dan bukan merupakan bahasa pemrograman prosedural.
2. *PSA (Program Statement Analyzer)* merupakan paket perangkat lunak yang mirip dengan kamus data (*data dictionary*) dan digunakan untuk mengecek data yang dimasukkan, disimpan, dianalisis dan yang dihasilkan sebagai *output* laporan.

j. *Model V*

Model ini merupakan perluasan dari model *waterfall*. Disebut sebagai perluasan karena tahap-tahapnya mirip dengan yang terdapat dalam model *waterfall*. Jika dalam model *waterfall* proses dijalankan secara linear, maka dalam model V proses dilakukan bercabang.

k. *Metode End-user Development*

Disini pengembangan dilakukan langsung oleh *end-user*. Keterlibatan langsung *end-user* sangat menguntungkan, karena memahami benar bagaimana sistem bekerja. Artinya tahap analisis sistem dapat dilakukan lebih cepat. Kelemahan adalah pada pengendalian mutu dan kecenderungan tumbuhnya “*private*” sistem informasi dan integrasi dengan sistem yang lain menjadi sulit (Saputra, 2013).