

ABSTRAK

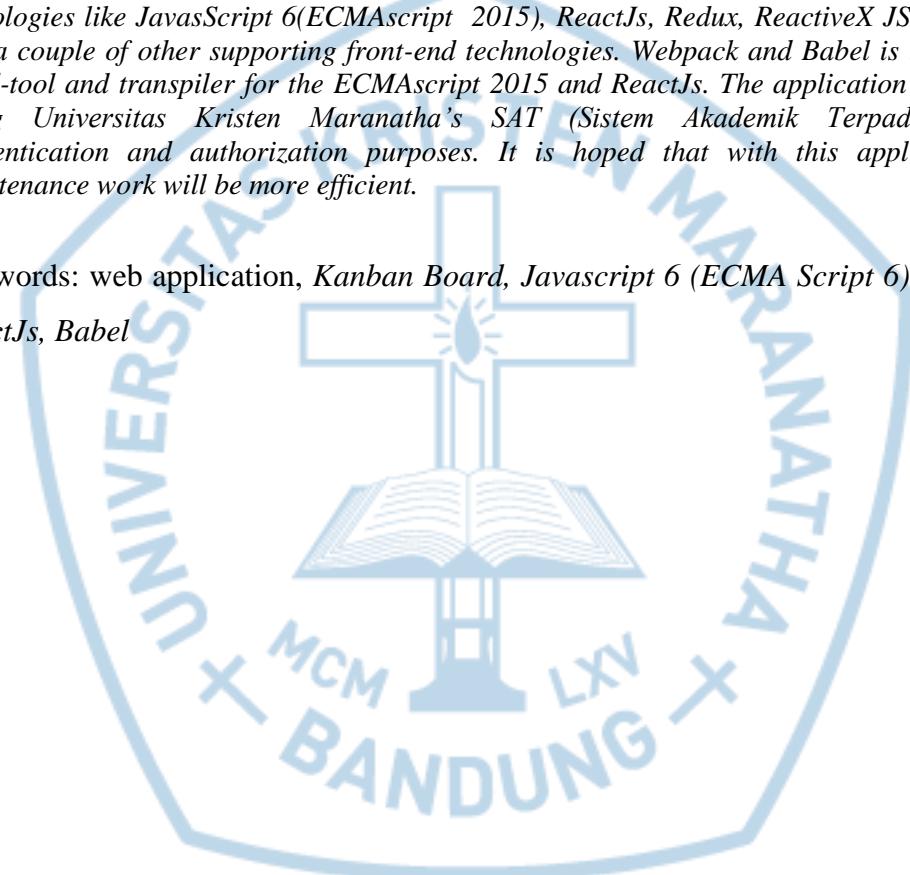
Seiring dengan berkembangnya Fakultas Teknologi Informasi Universitas Kristen Maranatha, maka semakin banyak punya peralatan serta fasilitas yang dimiliki dan digunakan. Oleh karena itu dibutuhkan pula kegiatan pemeliharaan (*maintenance*) yang dilakukan secara berkala untuk peralatan dan juga fasilitas tersebut. Pada dasarnya seluruh staff Fakultas Teknologi Informasi Universitas Kristen Maranatha dapat melakukan kegiatan *maintenance* tersebut, akan tetapi biasanya *maintenance* dilakukan oleh Staff Laboratorium Fakultas Teknologi Informasi. Kegiatan *maintenance* yang dilakukan oleh Staff Laboratorium biasanya dikerjakan dengan menggunakan metodologi *SCRUM* dan dibantu dengan penggunaan *Kanban Board*. Penelitian ini dilakukan untuk membuat digitalisasi dari *Kanban Board* fisik dan sistem penggerjaan *maintenance* yang telah dilakukan oleh Staff Laboratorium tersebut. Dalam pengembangannya, digunakan juga metode pemrograman reaktif (*reactive programming*) untuk membantu proses pengiriman dan pengambilan data antara *client* dan *server*. Hasil akhir dari penelitian berupa aplikasi berbasis web menggunakan bahasa pemrograman *PHP* dengan bantuan teknologi-teknologi front end seperti *Javascript 6 (ECMA Script 6)*, *ReactJs*, *Redux*, *ReactiveX JS (RxJS)* dan beberapa teknologi lainnya, serta menggunakan *Webpack* dan juga *Babel* sebagai *build-tool* dan *transpiler* untuk modul-modul *Javascript 6*. Otorisasi dan otentifikasi aplikasi ini akan dilakukan melalui SAT (Sistem Akademik Terpadu) milik Universitas Kristen Maranatha. Penulis berharap dengan adanya aplikasi ini, efisiensi dari kegiatan *maintenance* akan mengalami peningkatan.

Kata kunci : aplikasi *web*, *Kanban Board*, *Javascript 6 (ECMA Script 6)*, *PHP*, *ReactJs*, *Babel*

ABSTRACT

With the growth of Fakultas Teknologi Informasi Universitas Kristen Maranatha, rises the need for more equipments and better facilities. That in turn causes the need for a routine maintenance work. Generally every staff of Fakultas Teknologi Informasi Universitas Kristen Maranatha is able to perform maintenance work, but usually they are done by the laboratory staff of Fakultas Teknologi Informasi. The SCRUM methodology is applied to the maintenance process and a Kanban Board is used as an assisting tool. This research is done to provide a digitized form of the currently used physical Kanban Board and the current maintenance work process, in the form of a web-based application. The application will be using reactive programming to transfer data between clients and the server. The web-based application uses PHP and several front-end technologies like JavaScript 6(ECMAscript 2015), ReactJs, Redux, ReactiveX JS (RxJS) and a couple of other supporting front-end technologies. Webpack and Babel is used as build-tool and transpiler for the ECMAscript 2015 and ReactJs. The application will be using Universitas Kristen Maranatha's SAT (Sistem Akademik Terpadu) for authentication and authorization purposes. It is hoped that with this application, maintenance work will be more efficient.

Keywords: web application, *Kanban Board*, *Javascript 6 (ECMA Script 6)*, *PHP*, *ReactJs*, *Babel*



DAFTAR ISI

LEMBAR PENGESAHAN	i
PERNYATAAN ORISINALITAS LAPORAN PENELITIAN.....	ii
PERNYATAAN PUBLIKASI LAPORAN PENELITIAN	iii
PRAKATA.....	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Pembahasan	3
1.4 Ruang Lingkup dan Batasan-batasan	3
1.4.1 Ruang Lingkup Aplikasi	3
1.4.2 Batasan Perangkat Keras dan Perangkat Lunak.....	4
1.5 Sumber Data.....	4
1.6 Sistematika Penyajian	5
BAB 2 KAJIAN TEORI	6
2.1 SCRUM.....	6
2.1.1 Transparansi	6
2.1.2 Inspeksi	7
2.1.3 Adaptasi	7
2.2 Kanban Board.....	7

2.3 Sistem Informasi	8
2.3.1 Sistem.....	8
2.3.2 Informasi	9
2.4 UML (Unified Modeling Language).....	10
2.4.1 <i>Behavioral Diagram</i>	10
2.4.1.1 <i>Use Case Diagram</i>	11
2.4.1.1.1 <i>Actor</i>	11
2.4.1.1.2 <i>Use Case</i>	11
2.4.1.1.3 <i>Extend</i>	11
2.4.1.1.4 <i>Include</i>	12
2.4.1.2 <i>Activity Diagram</i>	12
2.4.1.2.1 <i>Activity Node</i>	12
2.4.1.2.2 <i>Activity Edge</i>	12
2.4.1.2.3 <i>Object Flow</i>	13
2.5 Rekayasa Perangkat Lunak	13
2.5.1 Flowchart	13
2.5.2 ERD.....	14
2.5.2.1 Entitas.....	14
2.5.2.2 Atribut	14
2.5.2.3 Relasi.....	14
2.5.2.3.1 Kardinalitas	14
2.5.2.3.2 Relasi Satu ke Satu.....	15
2.5.2.3.3 Relasi Satu ke Banyak.....	15
2.5.2.3.4 Relasi Banyak ke Satu.....	15
2.5.2.3.5 Relasi Banyak ke Banyak.....	15
2.6 Basis Data	15

2.6.1 RDBMS.....	15
2.7 Website.....	16
2.7.1 PHP	16
2.7.1.1 Slim	16
2.7.1.2 Medoo	17
2.7.2 HTML	17
2.7.3 CSS.....	17
2.7.4 JavaScript	18
2.7.4.1 ECMAScript 2015.....	18
2.7.4.2 Redux	19
2.7.4.2.1 Action.....	19
2.7.4.2.2 Reducer	20
2.7.4.2.3 Store	20
2.7.4.3 React JS.....	21
2.8 Pemrograman Fungsional (<i>Functional Programming</i>).....	21
2.8.1 First Class Functions	22
2.8.2 Fungsi murni (<i>Pure Functions</i>).....	22
2.8.3 Imutabilitas (<i>Immutability</i>)	22
2.9 <i>Reactive Programming</i>	23
2.9.1 ReactiveX Javascript (RxJS).....	24
2.9.1.1 Map	24
2.9.1.2 Flat Map	25
2.9.1.3 Switch Map	26
2.9.1.4 Delay	27
BAB 3 ANALISIS DAN RANCANGAN SISTEM.....	28
3.1 Profil Perusahaan	28

3.2 Proses Bisnis	28
3.2.1 Pembuatan <i>Task</i> Baru.....	28
3.2.2 Penggantian status sebuah <i>Task</i>	30
3.2.3 Aksi <i>Subscribe</i> pada sebuah <i>Task</i>	31
3.2.4 Aksi kolaborasi pada sebuah <i>Task</i>	32
3.3 <i>Entity Relationship Diagram</i> (ERD)	32
3.4 UML (<i>Unified Modeling Language</i>).....	33
3.4.1 Use Case.....	33
3.4.1.1 Use Case Sistem Kanban Board Digital.....	34
3.4.1.2 Use Case Manajemen Task	35
3.4.1.3 Use Case Manajemen Board	36
3.4.2 Activity Diagram.....	36
3.4.2.1 Login	37
3.4.2.2 Pembuatan <i>Task</i> baru	38
3.4.2.3 Perubahan <i>status Task</i>	39
3.4.2.4 <i>Subscribe</i> pada sebuah <i>Task</i>	39
3.4.2.5 Komentar pada sebuah <i>Task</i>	40
3.5 Desain Antarmuka.....	42
3.5.1 Form <i>Login</i>	42
3.5.2 Tampilan Board.....	42
3.5.3 Tampilan Detail Maintenance	43
BAB 4 IMPLEMENTASI.....	44
4.1 Implementasi ERD	44
4.1.1 Transformasi ERD menjadi <i>Table</i>	44
4.1.1.1 Tabel <i>Users</i>	45
4.1.1.2 Tabel <i>Maintenances</i>	45

4.1.1.3 Tabel <i>Subscriptions</i> (<i>maintenance_users</i>)	46
4.1.1.4 Tabel <i>Collaborations</i>	46
4.1.1.5 Tabel <i>Boards</i>	46
4.1.1.6 Tabel <i>Color Code</i>	47
4.1.1.7 Tabel <i>Logs</i>	47
4.2 Pengembangan aplikasi	47
4.2.1 Pengembangan <i>back-end</i> (sisi <i>server</i>).....	47
4.2.1.1 Implementasi <i>Slim</i>	48
4.2.1.2 Implementasi <i>Medoo</i>	49
4.2.1.3 Pengembalian Data dalam Bentuk <i>JSON</i>	50
4.2.2 Pengembangan <i>front-end</i> (sisi <i>client</i>)	51
4.2.2.1 Impementasi <i>React</i>	51
4.2.2.2 Implementasi <i>Redux</i>	56
4.2.2.3 Implementasi <i>ReactiveX JS</i> (<i>RxJS</i>)	56
BAB 5 PENGUJIAN	58
5.1 Pengujian <i>Form Login</i>	58
5.2 Pengujian <i>Form Penambahan Kartu Maintenance</i>	58
5.3 Pengujian Sinkronisasi Penambahan Kartu <i>Maintenance</i>	58
BAB 6 SIMPULAN DAN SARAN	59
6.1 Simpulan	59
6.2 Saran.....	59
DAFTAR PUSTAKA	60

DAFTAR GAMBAR

Gambar 2.1 Kanban Board Fisik.....	8
Gambar 2.2 Contoh <i>redux action</i>	20
Gambar 2.3 Contoh <i>redux reducer</i>	20
Gambar 2.4 Contoh <i>store</i>	21
Gambar 2.5 Komponen paling sederhana <i>React</i>	21
Gambar 2.6 Perbandingan antara <i>Observable</i> dan <i>Iterable</i>	24
Gambar 2.7 Ilustrasi fungsi <i>map</i>	25
Gambar 2.8 Ilustrasi fungsi <i>FlatMap</i>	26
Gambar 2.9 Ilustrasi fungsi <i>SwitchMap</i>	26
Gambar 2.10 Ilustrasi fungsi <i>Delay</i>	27
Gambar 3.1 <i>Flowchart</i> pembuatan <i>task</i> baru	29
Gambar 3.2 <i>Flowchart</i> penggantian status sebuah <i>task</i>	30
Gambar 3.3 <i>Flowchart</i> aksi <i>subscribe</i> pada sebuah <i>task</i>	31
Gambar 3.4 Penambahan kolaborasi pada sebuah <i>Task</i>	32
Gambar 3.5 <i>Entity relationship diagram</i>	33
Gambar 3.6 <i>Use case</i> Sistem <i>Kanban Board Digital</i>	34
Gambar 3.7 <i>Use Case Manajemen Task</i>	35
Gambar 3.8 <i>Use Case Manajemen Board</i>	36
Gambar 3.9 <i>Activity diagram login</i>	37
Gambar 3.10 <i>Activity diagram</i> pembuatan <i>task</i> baru	38
Gambar 3.11 <i>Activity diagram</i> perubahan <i>status task</i>	39
Gambar 3.12 <i>Activity diagram</i> <i>subscribe</i> pada <i>task</i>	40
Gambar 3.13 <i>Activity diagram</i> pemberian komentar pada <i>task</i>	41
Gambar 3.14 Tampilan halaman <i>login</i>	42
Gambar 3.15 Tampilan <i>active board</i>	42
Gambar 3.16 Tampilan detil kartu <i>maintenance</i>	43
Gambar 4.1 Implementasi <i>ERD</i> pada <i>database MySQL</i>	44
Gambar 4.2 Instansiasi dan pengambilan referensi <i>Container</i>	48
Gambar 4.3 Struktur <i>file</i> aplikasi	48
Gambar 4.4 Kode untuk meng- <i>include</i> seluruh route yang tersedia	49

Gambar 4.5 Contoh sebuah <i>route Slim</i>	49
Gambar 4.6 Instansiasi Medoo dan <i>dependency injection</i> kedalam container.....	50
Gambar 4.7 Contoh <i>query</i> dengan menggunakan <i>medoo</i>	50
Gambar 4.8 <i>UI Form Login</i>	51
Gambar 4.9 <i>Layout</i>	52
Gambar 4.10 <i>Form Container</i>	54
Gambar 4.11 <i>Login Form</i>	55
Gambar 4.12 Penggunaan <i>mapStateToProps</i> dan <i>mapDispatchToProps</i>	56
Gambar 4.13 Contoh <i>epic</i>	57



DAFTAR TABEL

4.1 Tabel <i>Users</i>	45
4.2 Tabel <i>Maintenances</i>	45
4.3 Tabel <i>Subscriptions</i> (<i>maintenance_users</i>)	46
4.4 Tabel <i>Collaborations</i>	46
4.5 Tabel <i>Boards</i>	46
4.6 Tabel <i>Color Code</i>	47
4.7 Tabel <i>Logs</i>	47

