

## I. PENDAHULUAN

### 1.1 Latar Belakang

PT. Microdata Indonesia merupakan penyedia layanan dibidang jasa konsultasi teknologi informasi yang bertempat di Bandar Lampung dan berdiri sejak 2010. PT. Microdata Indonesia memiliki 20 orang tenaga ahli di bidang teknologi informasi. Berpusat pada peningkatan solusi teknologi informasi dan produk untuk segmen lembaga pemerintah, dan lain-lain. Hasil kerja yang memuaskan pun dengan cepat mendapatkan perhatian dari perusahaan – perusahaan besar serta lembaga pemerintahan. Lembaga pemerintahan yang tertarik untuk menjadi klien guna mengembangkan teknologi informasi pada daerahnya, salah satunya adalah Kabupaten Mesuji. Penulis berkesempatan menjadi salah satu *freelancer front end web developer* setelah praktik kerja lapangan di PT. Microdata Indonesia dan diikut sertakan dalam proyek *web* yang sedang dibangun yaitu *web* Kabupaten Mesuji. Penulis menjadikan proyek tersebut sebagai laporan untuk tugas akhir mahasiswa.

Kabupaten Mesuji wilayah yang tidak terpisahkan dari sejarah Provinsi Lampung dan Sumatera Selatan. Pada tahun 1865, Sirah Pulau Pandang Kayu Agung. Kabupaten Mesuji merupakan salah satu wilayah Otonom Baru (DOB) Provinsi Lampung yang merupakan pemekaran dari Kabupaten Tulang Bawang. Pada masa itu pemerintahan Tulang Bawang sangat besar, dan kabupaten Mesuji masih kecamatan Mesuji. Oleh karena itu Kabupaten Tulang Bawang yang terlalu jauh dari pusat pemerintahan di Menggala dipisahkan dari kabupaten Mesuji atas dasar percepatan dan pemerataan pembangunan, peningkatan kesejahteraan dan keterjangkauan pelayanan publik 218.400 Hektar. Menurut data BPS Kabupaten Mesuji bahwa jumlah penduduk di wilayah Kabupaten Mesuji pada tahun 2016 berjumlah 192.913 orang dalam tujuh kecamatan dengan jumlah desa 75 desa merupakan sumber daya manusia yang potensial di samping potensi sumber daya alam yang beragam, berlimpah dan menjanjikan, terutama di bidang perkebunan, pertambangan, pertanian dan agribisnis lainnya. Diresmikan oleh Menteri Dalam Negeri Mardiyanto pada 29

Oktober 2008. Undang-Undang Nomor 49 Tahun 2008 yang menetapkan Kabupaten Mesuji di Provinsi Lampung.

Kabupaten Mesuji kabupaten yang berjalan dalam pembangunan untuk kemajuan daerah dan masyarakatnya. Dan dalam pembangunan tersebut tidak lupa dalam pembangunan *web* daerah. Pembangunan *web* daerah berfungsi untuk melakukan promosi terhadap daerah maupun memberikan informasi kepada masyarakat. Menurut hasil survei Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) pada triwulan II-2020 menunjukkan jumlah pengguna Internet Indonesia mencapai 196,7 juta. Peningkatan jumlahnya adalah 23,5 juta atau sebanyak 8,9%. Provinsi Lampung ke-9 dengan total konsumen internet sebanyak 5,3 juta orang. *Web* ini dalam rancangannya akan berisikan berbagai dinas – dinas, sekretariat, dan badan lainnya di Kabupaten Mesuji yang terhubung kedalam *web* ini. Penulis berfokus kepada *front end* pada salah satu dinas yaitu Dinas Koperasi dan UKM Kabupaten Mesuji dan sekaligus digunakan sebagai bahan Tugas Akhir Mahasiswa.

*Web* dirancang didalamnya terdapat berbagai informasi terkait, berita, dan kebutuhan lainya untuk mempromosikan dan pusat sarana daerah Kabupaten Mesuji khususnya dalam Dinas Koperasi dan UKM Kabupaten Mesuji. Implementasi *web* ini menggunakan *library ReactJS*. *Web developer* khususnya *front end developer* saat ini banyak menggunakan *ReactJS*. *ReactJS* dapat didesain sederhana pada setiap *level* aplikasi. Popularitasnya dapat diukur oleh dengan aplikasi yang menggunakan *ReactJS* seperti *Facebook*, *WhatsApp*, dan ratusan penyedia jasa pengembangan aplikasi *web* menggunakan *ReactJS*. Dengan alasan tersebut PT. Microdata Indonesia memutuskan untuk menggunakan *ReactJS*. Untuk bagian *back end* sudah disediakan oleh pihak PT. Microdata Indonesia sehingga penulis berperan dalam membuat bagian *front end*. Untuk menghubungkan bagian *front end* dengan bagian *back end* supaya mendapatkan data – data yang diperlukan dalam *web* Dinas Koperasi dan UKM menggunakan API.

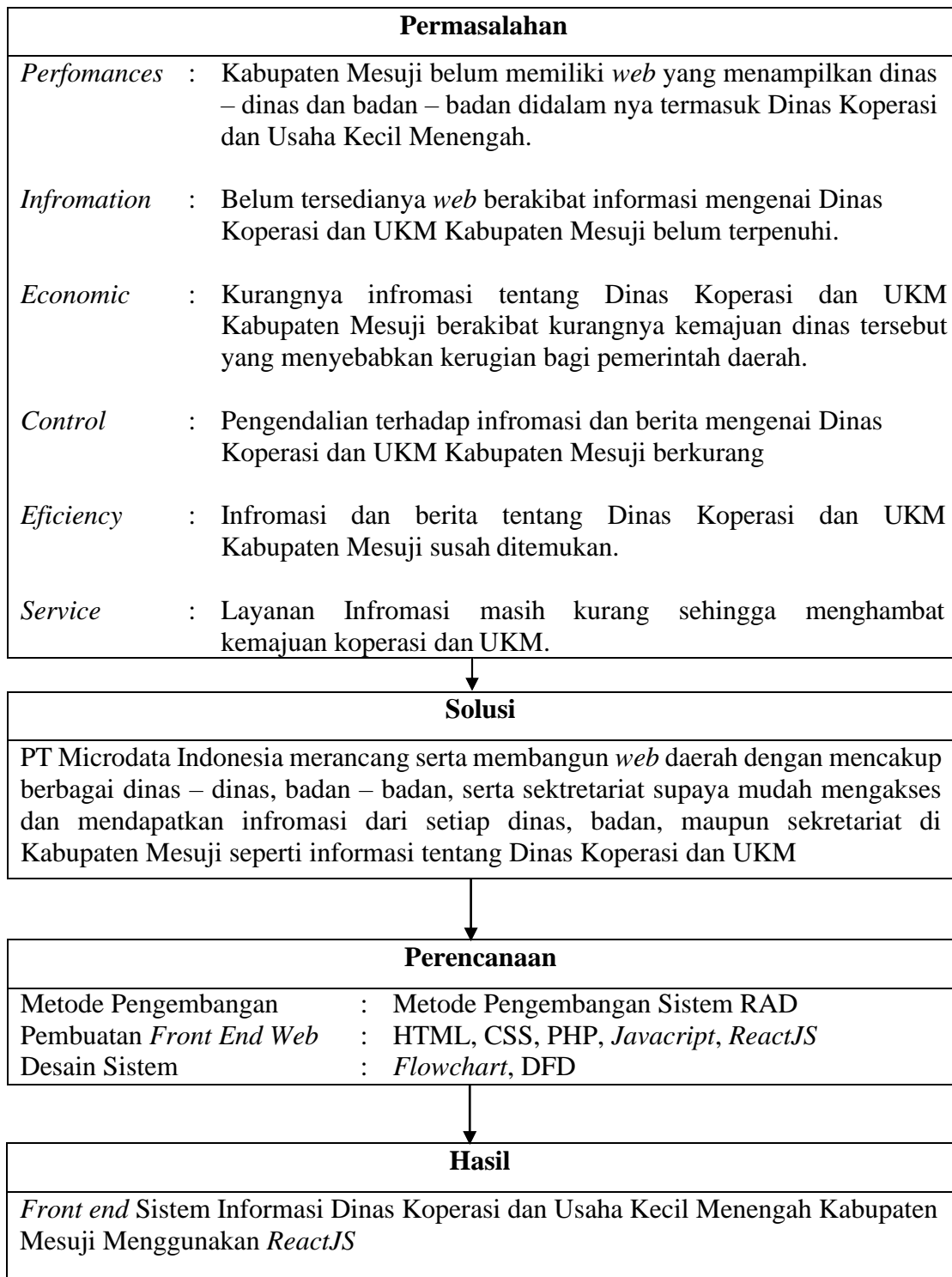
## **1.2 Tujuan**

Tujuan yang dicapai sebagai berikut.

1. Menyelesaikan *front end* sistem informasi Dinas Koperasi dan Usaha Kecil Menengah (UKM) Kabupaten Mesuji.
2. Mengimplementasikan *front end* sistem informasi Dinas Koperasi dan Usaha Kecil Menengah (UKM) Kabupaten Mesuji.

## **1.3 Kerangka Pemikiran**

Berdasarkan dari latar belakang, didapatkan suatu susunan kerangka pemikiran yang di sajikan gambar 1.



Gambar 1 . Kerangka Pemikiran

## 1.4 Kontribusi

Kontribusi dari pengerjaan Tugas Akhir Mahasiswa kepada :

1. PT. Microdata Indonesia

Proyek *web* Dinas Koperasi dan UKM Kabupaten Mesuji dapat tuntas.

2. Kabupaten Mesuji

Penyebaran informasi terkendali dan mudah didapatkan serta mudah mengakses *web* Dinas Koperasi dan UKM di Kabupaten Mesuji.

3. Dinas Koperasi dan UKM Kabupaten Mesuji

Memudahkan pemberian informasi terkait Dinas Koperasi dan UKM serta dapat diakses oleh masyarakat dengan cepat dan mudah.

4. Penulis

Diharapkan mampu meningkatkan pengetahuan serta keterampilan dalam memberikan serta menyajikan informasi mengenai *front end web* yang dirancang.

## II. TINJAUAN PUSTAKA

### 2.1 *Front end*

*Front end* adalah segala sesuatu yang menyatukan antara pengguna dengan sistem *back end*. *Front end* merupakan sebuah antarmuka pengguna dimana pengguna akan berhubungan dengan sistem. (Gloldbot, 2016).

### 2.2 *Web*

*Web* merupakan gabungan halaman dari multi halaman terdiri informasi berupa data digital baik berupa gambar, video, foto, teks, serta animasi lainnya yang tersedia melalui internet. (Hasugian, 2018)

### 2.3 *Javascript*

*Javascript* adalah kumpulan skrip yang dieksekusi dalam dokumen HTML. Bahasa pemrograman ini merupakan bahasa yang memberikan fungsionalitas tambahan pada HTML dengan memungkinkan perintah dieksekusi disisi *browser*.(Lavarino dan Yustanti, 2016:74)

### 2.4 *NodeJS*

*NodeJS* ialah *software* yang dikembangkan untuk pengembangan aplikasi *web* dalam penulisan sintaks *Javascript*. *Javascript* sebagai bahasa pemrograman yang berjalan pada *client-side* / *browser-side*. *NodeJs* ada untuk menyempurnakan tugas *Javascript* dan juga berfungsi sebagai bahasa pemograman *server-side*, seperti PHP, *Ruby*, dan lainnya. Tidak seperti bahasa pemrograman sisi *server* umum yang bersifat sinkronis dan *blocking*, *NodeJS* bersifat asinkronis atau *non-blocking* sebagaimana halnya *Javascript* berfungsi. (Firdaus, 2019)

### 2.5 *JSON (Java Script Object Notation)*

JSON adalah format ringkas untuk bertukar data komputer. Formatnya berbasis teks, dapat dibaca manusia, dan mewakili struktur data sederhana dan objek. Format JSON banyak digunakan untuk menstransmisikan data struktural melalui suatu koneksi jaringan dalam proses yang disebut serialisasi. (Syahputra, 2018)

## **2.6 ReactJS**

*ReactJS* merupakan *library Javascript* untuk mengembangkan komponen *user interface* (UI) yang dapat digunakan kembali. *ReactJS* pada dasarnya membolehkan pengembangan aplikasi *web* yang kompleks serta besar yang dapat mengganti data tanpa pembaruan halaman (Aggarwal, 2018).

### **2.6.1 Virtual DOM**

*Virtual DOM* (*Virtual document object model*) dari *ReactJS*. *Virtual DOM* mirip dengan *document object model* (DOM) yang dihasilkan oleh *browser*, dengan perbedaan menghematnya memori. *Virtual DOM* bekerja cukup sederhana. Setiap kali permintaan perubahan konten halaman dibuat, perubahan adalah direfleksi ke memori yang berada di *Virtual DOM* terlebih dahulu setelah itu algoritma *diff()* membandingkan *Virtual DOM* dan DOM di *browser* (Aggarwal, 2018).

### **2.6.2 JSX**

JSX sangat mirip dengan XML. Tidak diwajibkan menggunakan JSX saat mengembangkan aplikasi berbasis *React* tetapi sangat populer diantara para pengembang karena *Short hand* yang membuat pengembang mudah, setiap kali menulis *mark – up* untuk komponen dan untuk mengikat peristiwa yang sesuai. Sehingga JSX mudah digunakan dan tidak rumit (Aggarwal, 2018).

## **2.7 Rapid Application Development (RAD)**

RAD merupakan pengembangan suatu sistem informasi dalam waktu yang tidak lama. Dalam metode *Rapid Application Development* (RAD) terdapat fase perencanaan kebutuhan dimana pengguna terlibat dalam perancangan sistem dan mendirikan sistem (aktivitas ini dilaksanakan secara terus menerus hingga mencapai kesepakatan bersama) dan terakhir fase implementasi. (Supriyatna, 2018)

### **2.7.1 Requirements Planning**

Tahap *Requirements Planning* adalah tahap permulaan pada pembuatan sistem, aktivitas yang dilakukan pada tahapan ini adalah mengidentifikasi serta mengobservasi, kemudian mengumpulkan data.

### **2.7.2 User Design**

Didasari dari solusi yang didapatkan dalam *Requirements Planning*, setelah itu dimulai dengan perancangan yang meliputi perancangan sistem, DFD, *flowchart* dan perancangan tampilan

### **2.7.3 Contruction**

*Contruction* merupakan tahapan pengekseskusion dari perancangan yang telah dibentuk pada tahap *User Design* kedalam bentuk *script* program dan merupakan lanjutan dari tahapan sebelumnya.

### **2.7.4 Cutover**

Pada tahapan *cutover* sistem atau komponen dari sistem diuji. Apabila beberapa aspek dan nonkinetis yang dibutuhkan telah tepenuhi dan disetujui maka akan di implementasikan.

## **2.8 API (Application Programming Interface)**

API sama dengan *software interface* yang terdiri dari serangkaian perintah yang disimpan dalam serangkaian perintah yang disimpan dalam bentuk *library* dan menjelaskan bagaimana supaya suatu *software* dapat hubungan dengan *software* lain. (Firdaus, 2019).

## **2.9 REST (REpresentational State Trasnfer)**




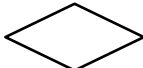




REST (*REpresentational State Trasnfer*) merupakan arsitektur metode komunikasi yang sering diterapkan dalam pengembangan layanan berbasis *web*. REST melakukan interaksi antara *client* dan *server* yang difasilitasi oleh sejumlah tipe operasional (verba) dan *Universal Resource Identifiers* (URIs) yang unik bagi tiap-tiap sumber daya. Masing – masing verba *GET*, *POST*, *PUT*, dan *DELETE* memiliki makna operasional khusus untuk menghindari ambiguitas. Arsitektur REST yang *decoupled* (terpisah) serta beban komunikasi yang ringan antara produsen dan konsumen menjadikanya populer didunia *cloud-based API*, *web* yang menggunakan arsitektur semacam itu dinamakan RESTful APIs (*Application Programming Interfaces*) atau REST APIs.(Firdaus,2019)



### 2.10 Flowchart

Menurut Wibawanto (2017) *Flowchart* merupakan diagram untuk menggambarkan alur proses secara terperinci dan berhubungan dari proses ke proses lainnya, Langkah-langkah yang dibuat dengan simbol-simbol tertentu dari suatu prosedur pemecah masalah. Simbol-simbol *flowchart* disajikan pada Tabel 1.

Tabel 1. Simbol-simbol *Flowchart*

Simbol (1)	Nama (2)	Keterangan (3)
	Terminal	Mendeklarasikan awal dan akhir suatu program
	<i>Input / Output</i>	Mendeklarasikan <i>input</i> dan <i>output</i>
	Proses	Mendeklarasikan proses suatu sistem .
	<i>Decision</i>	Mendeklarasikan suatu kondisi yang menghasilkan dua kemungkinan
	<i>Document</i>	Menyatakan keluaran dalam bentuk dokumen <i>hardcopy</i> (melalui <i>printer</i> )
	<i>Connector</i>	Menyatakan koneksi yang menghubungkan pada kepada proses lainnya di sama halaman
	<i>Offline Connector</i>	Menyatakan koneksi yang menghubungkan pada kepada proses lainnya di berbeda halaman
	<i>Flow</i>	Untuk menggambarkan aliran arus suatu proses.


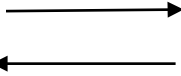
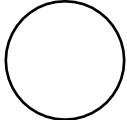

sumber: (Santoso dan Nurmalina, 2017).

### 2.13 DFD

*Data flow diagram* (DFD) adalah gambaran dari sistem operasional yang kompleks dan mudah dimengerti dengan memperlihatkan komponen secara lengkap.

DFD dapat menganalisis dan meningkatkan kinerja sistem (Maniah & Hamidin, 2017). Simbol DFD disajikan pada Tabel 2.

Tabel 2. Simbol-simbol DFD





Simbol	Nama	Keterangan
	Eksternal <i>Entity</i>	Seperti orang, organisasi, atau unit bersangkutan berinteraksi pada sistem, namun tidak dalam sistem
	Alur Data	Alur Data adalah aliran data pada sistem yang menunjukkan arah khusus dari sumber ke tujuan.
	<i>Process</i>	<i>Process</i> dilakukan oleh orang atau unit menggunakan transformasi data atau mentransformasikan data
	<i>Data store</i>	Data Store digunakan untuk penyimpanan data yang yang terkomputerisasi, seperti basis data maupun file.

Sumber : Sukamto dan Shalahuddin, 2018




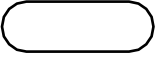
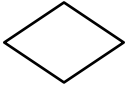

#### 2.14 Mapping Chart

*Mapping Chart* merupakan sebuah aliran data dokumen dalam suatu program atau prosedur yang menunjukkan aliran dokumen dalam bentuk informasi (Rinadiharsa & Yamasari, 2016). Simbol-simbol pada *mapping chart* disajikan pada tabel 3.

Tabel 3. Simbol *Mapping Chart*

Simbol (1)	Nama (2)	Keterangan (3)
	Dokumen	Dokumen <i>input</i> dan <i>output</i>
	Kegiatan Manual	Untuk menggambarkan kegiatan yang dilakukan secara manual
	Proses	Untuk menggambarkan proses yang dilakukan
	<i>Keyboard</i>	Masukan pada <i>keyboard</i> khusus yang terkomputerisasi

Tabel 3. Lanjutan

(1)	(2)	(3)
	<i>Database</i>	Untuk menggambarkan penyimpanan data yang terkomputerisasi
	Garis Alir	Menunjukkan arah dari setiap proses
	<i>display</i>	Untuk menyatakan peralatan <i>output</i> seperti <i>monitor</i>
	Terminasi	Untuk menandakan awal dan akhir suatu aliran
	<i>Decision</i>	Untuk menentukan keputusan berupa ya atau tidak
	Penghubung	Penghubung di halaman yang sama maupun halaman berbeda

Sumber : Kristanto, 2018

### 2.15 Sistem Informasi

Sistem informasi yaitu kombinasi yang terorganisir dari orang-orang, perangkat keras, perangkat lunak, jaringan komunikasi, dan sumber daya data yang mengumpulkan, mengubah, dan mendistribusikan informasi dalam suatu organisasi. (Firman, 2016)

### 2.16 Kamus data

Kamus data (*data dictionary*) digunakan untuk mengklasifikasikan aliran data yang dijelaskan oleh DFD. Kamus data merupakan daftar kelompok komponen data yang mengalir melalui sistem perangkat lunak guna masukan dan keluaran dapat dipahami secara umum. (Sukanto & Shalahuddin, 2018)

### ***2.17 Performance Testing***

*Performance Testing* merupakan penentuan kinerja aplikasi *web*, membangun model uji beban merupakan prasyarat penting untuk mendapatkan kemampuan beban aplikasi *web*. (Qian J, 2018)