

I. PENDAHULUAN

1.1 Latar Belakang

Teknologi Informasi dan Komunikasi (TIK) merupakan salah satu teknologi yang berkembang sangat cepat dan menyeluruh bagi kehidupan masyarakat. Perkembangan teknologi internet dalam kehidupan sehari-hari memiliki dampak yang positif pada semua aspek kehidupan seperti peluang dibidang pemasaran, pendidikan, pemerintahan dan membuka peluang dibidang bisnis. Lebih dari separuh warga Indonesia saat ini telah terkoneksi dengan internet. Data ini diambil dari hasil survei Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) yang dilaksanakan pada tanggal 2 sampai 25 Juni 2020. Survei tersebut menunjukkan bahwa menurut Badan Pusat Statistik (BPS) dari data pada triwulan II tahun 2020 jumlah pengguna internet di Indonesia naik menjadi 73,7% dari populasi atau sama dengan 196,7 juta pengguna. Dibandingkan dengan jumlah pengguna internet di Indonesia pada tahun 2018 yang hanya 64,8%, jumlah pengguna meningkat 8,9 % atau sama dengan 25,5 juta. Peningkatan jumlah pengguna ini disebabkan oleh infrastruktur internet yang cepat dan merata, semakin banyaknya pembelajaran *online* dan bekerja dari rumah (*work from home*) akibat adanya pandemi *covid-19* (APJII, 2020).

PT Lintas Data Multimedia adalah perusahaan yang bergerak di bagian jasa internet. PT Lintas Data Multimedia menyediakan layanan teknologi berbasis internet untuk perusahaan yang bermitra serta menyediakan internet untuk kebutuhan setiap masyarakat Indonesia. Perusahaan yang berlokasi di jalan Gunung Galunggung Raya No 13 Perumnas Way Halim ini bekerja sama dengan beberapa perusahaan seperti Asosiasi Penyelenggara Jasa Internet Indonesia (APJII), PT Media Akses Global Indo, *Indonesia Network Information Center* (IDNIC), *Interlink*, Moratelindo, Indosat, Pgncom, Telkom Indonesia, Telematika Media Solusi (TMS) dan *Greenet*.

PT Lintas Data Multimedia juga memiliki klien yang bermitra mengembangkan proyek instalasi jaringan seperti dinas pemerintahan. Proyek tersebut memiliki beberapa bagian tenaga kerja seperti supervisor dan teknisi. Supervisor memiliki tanggung jawab mengelola dan mengawasi alur kerja (*work flow*) tim dan setiap individu yang ada di dalam tim agar semua pekerjaan dapat dijalankan secara efektif. Teknisi bertugas melakukan instalasi jaringan, memantau dan memelihara sistem jaringan tersebut. Ketika akan melakukan pemasangan jaringan, supervisor memberikan jadwal dan *form* pemasangan jaringan di pagi hari kepada teknisi, kemudian disiapkan kebutuhan barang sesuai *form* pemasangan setiap dinas yang ada di provinsi Lampung. Kegiatan ini menyita waktu teknisi dalam pemasangan jaringan, karena menunggu pembagian jadwal dari supervisor. Teknisi kemudian memberikan bukti pemasangan berupa foto yang dikirimkan melalui *whatsapp* setelah pemasangan selesai. Data foto tersebut belum tersimpan dalam *database*, sehingga rentan kehilangan data yang dapat mempersulit supervisor dalam pelaporan pemasangan jaringan. Proses pengecekan pemasangan masih harus didatangi langsung ke lokasi oleh supervisor, hal ini membutuhkan biaya dan memerlukan waktu yang lama. *Form* pemasangan juga masih menggunakan kertas, yang dapat mengakibatkan terjadinya kehilangan, penggandaan data dan kerusakan berkas tersebut.

Solusi yang diperlukan berdasarkan permasalahan diatas adalah membuat aplikasi monitoring pemasangan jaringan menggunakan *microservice* arsitektur yang terdiri dari *frontend*, *backend*, dan *database*. Pada penulisan laporan tugas akhir ini penulis hanya memfokuskan pada desain *frontend*. Sistem ini akan memudahkan dalam pengolahan data oleh supervisor dan teknisi melalui desain *lay out* yang mudah dipahami oleh pengguna, serta akan diimplementasikan layanan *microservice* arsitektur untuk mengembangkan aplikasi. Arsitektur *microservices* adalah struktur baru yang fungsinya untuk menyelesaikan masalah yang terjadi pada sistem yang sedang berjalan. Arsitektur *microservice* membantu pengembangan perangkat lunak yang lebih cepat dan mudah yang berinteraksi secara terpisah satu sama lain melalui jaringan (Suryotrisongko, 2017).

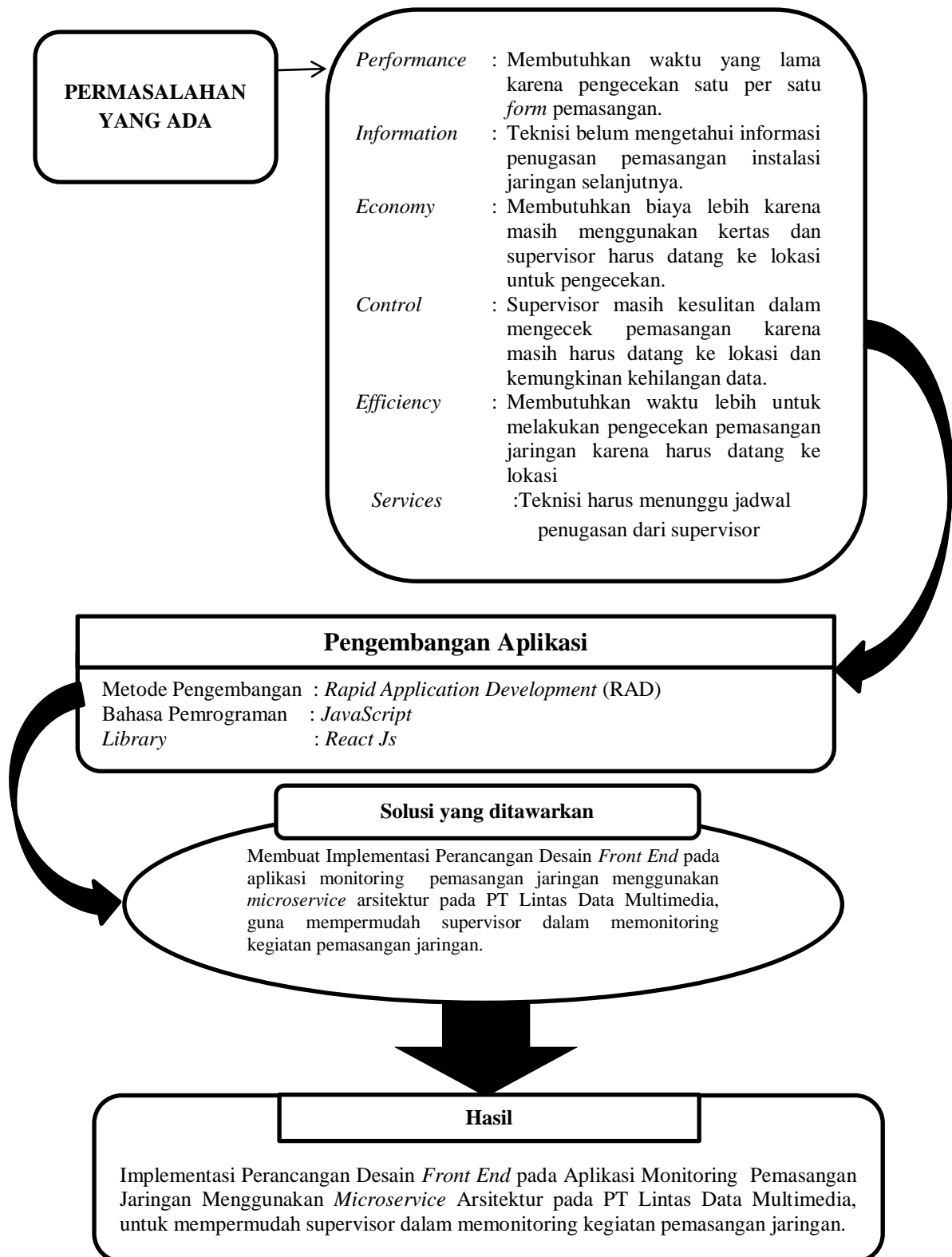
1.2 Tujuan

Tujuan yang telah dicapai dalam penulisan tugas akhir ini adalah menghasilkan aplikasi monitoring pemasangan jaringan untuk membantu PT Lintas Data Multimedia dalam melakukan monitoring kegiatan pemasangan jaringan.

1.3 Kerangka Pemikiran

Proses pengumpulan data foto pada pemasangan jaringan PT Lintas Data Multimedia masih dikirimkan melalui *whatsapp* dan belum tersimpan dalam *database* yang beresiko kehilangan data, sehingga dapat mempersulit supervisor untuk pelaporan pemasangan jaringan. Proses pengecekan pemasangan jaringan masih harus didatangi langsung ke lokasi oleh supervisor, mengakibatkan tidak efisiennya biaya dan waktu.

Permasalahan di atas membutuhkan sebuah solusi yang dapat digunakan untuk membantu supervisor dalam melakukan pelaporan kegiatan tersebut, yaitu dengan membuat “Implementasi Perancangan Desain *Frontend* pada Aplikasi Monitoring Pemasangan Jaringan Menggunakan *Microservice* Arsitektur pada PT Lintas Data Multimedia”. Metode yang digunakan adalah *Rapid Application Development* (RAD). Tahapan kerangka pemikiran dalam pembuatan aplikasi monitoring pemasangan jaringan ini disajikan pada Gambar 1.



Gambar 1. Kerangka Pemikiran

1.4 Kontribusi

Aplikasi ini memiliki kontribusi kepada beberapa pihak yang diuraikan sebagai berikut :

1. Mempermudah supervisor memonitoring kegiatan pemasangan jaringan.
2. Memudahkan teknisi dalam mendapatkan jadwal serta memberikan bukti pemasangan jaringan.

II. TINJAUAN PUSTAKA

2.1 Aplikasi Website dan Frontend

Aplikasi merupakan program yang disediakan untuk melakukan suatu kegiatan fungsi untuk pengguna jasa aplikasi dan pemakaian aplikasi yang lainnya untuk digunakan oleh pelaku sasaran yang menjadi tujuan (Fansuri, 2015). *Website* menurut Hastanti (2015) adalah sekumpulan halaman yang berisi informasi yang disimpan dalam internet, dapat diakses dan dilihat pada perangkat yang dapat mengakses internet. Kata *web* adalah penyederhanaan dari *World Wide Web* (www) yang merupakan sistem jaringan dengan berbasis *client-server* dengan HTTP (*Hypertext Transfer Potokol*) dan TCP (*Transmission Control Protokol / Intenet Protocol*) sebagai medianya. Fungsi *website* secara umum yaitu (1) alat komunikasi; (2) media informasi; (3) media *intertainment*; (4) sarana transaksi. Alasan mengapa banyak perusahaan atau individu menggunakan *website* antara lain (Soejono dkk., 2018) :

1. *Website* adalah sarana komunikasi pengenalan sebuah produk kepada masyarakat lebih luas, seperti visi misi perusahaan, dan jenis produk yang ditawarkan.
2. Internet merupakan media informasi tanpa batas, teknologi semakin maju segala sesuatu dilakukan memakai situs *website*.
3. *Website* didesain secara menarik, sehingga dapat meningkatkan citra bisnis perusahaan dan memudahkan pekerjaan.

Frontend menurut Aziz dkk., (2020) merupakan suatu bagian dari sistem yang menyuguhkan tampilan kepada pengguna. *Frontend* bertanggung jawab pada tampilan antar muka pengguna dan bertugas meningkatkan bagian-bagian visual dari suatu sistem. *Frontend* adalah *developer* yang bekerja dibagian desain *layout* dalam *web* dan aplikasi. *Developer* harus membuat *user intetrface* agar desain menarik perhatian dan tidak membosankan pada sisi pengguna. *Frontend* dibangun dengan menggunakan banyak komponen seperti HTML, CSS, dan *JavaScript*.

2.1.1 Monitoring

Monitoring adalah kegiatan memantau secara bersama pada setiap kondisi dan kegiatan yang bertujuan untuk data masukan yang didapat dari informasi pengamatan dapat menjadi dasar untuk mengajukan keputusan serta tindakan seterusnya. Keputusan tersebut dapat diambil jika dalam monitoring pelaksanaan kegiatan menunjukkan adanya hal yang tidak sesuai dengan yang direncanakan sebelumnya. Monitoring dapat dilakukan dengan cara rapat bersama, laporan harian, dan survei lapangan, kemudian fokus pelaksanaan perencanaan kegiatan dan pelaporan akhir (GRHASIA, 2014).

Monitoring digunakan untuk mengamati sebuah kemajuan dalam kegiatan dan mengidentifikasi suatu masalah dan langkah pemecahan masalah tersebut, maka tujuan dari monitoring adalah :

1. Mendapatkan informasi dari data perkembangan dari kegiatan secara terus menerus dalam mencapai tujuan kinerja serta permasalahan yang sedang dijalani.
2. Mengidentifikasi masalah dalam kegiatan agar dapat ditangani segera mungkin.
3. Menjadi penyempurna dalam suatu perencanaan kegiatan selanjutnya lewat hasil pengamatan.

2.1.2 Jaringan

Jaringan adalah suatu komponen komputer yang saling terhubung menjadi satu kesatuan yang dapat mengirimkan data dan informasi melalui kabel-kabel atau tanpa kabel (*wireless*) yang memungkinkan pengguna dapat saling terhubung bertukar dokumen dan data. Jaringan komputer digunakan untuk memperlancar dan menyebarkan informasi, jaringan sangat dibutuhkan pada berbagai instansi seperti sekolah, kampus, perusahaan dan konsumen umum lainnya (Ardianto & Akbar, 2017).

2.2.1 Microservices

Berdasarkan Rezaldy & Asror (2018), *microservices* arsitektur adalah sistem yang mengembangkan perangkat lunak yang dibuat dari susunan beberapa

service kecil, yang berkomunikasi dengan mekanisme sederhana melalui HTTP sebagai media komunikasi API. *Microservice* arsitektur membuat pengembangan aplikasi lebih mudah, lebih cepat dan dapat diperbarui secara terpisah. Kelebihan *microservice* menurut Mufrizal & Indarti (2019) yaitu (a) pembuatan kode aplikasi yang lebih sedikit dan mandiri; (b) perangkat lunak yang mudah pemeliharannya; (c) mudah diperluas; (d) memudahkan pengembang menggunakan setiap bahasa pemrograman dan *framework*.

Menurut Rezaldy & Asror (2018), *microservice* adalah arsitektur perangkat lunak yang digunakan untuk membangun suatu aplikasi yang dibuat dari susunan komponen kecil dari *service*, yang berjalan dengan metodenya sendiri, dan saling berhubungan dengan metode yang sederhana, biasanya menggunakan HTTP sebagai media komunikasi API. Sedangkan menurut Handayani (2020) menerangkan bahwa *microservices* adalah sebuah arsitektur yang pengembangan aplikasi berlangsung dalam bentuk layanan *web* yang lebih kecil, yang berkomunikasi satu dengan yang lain. Arsitektur *microservice* adalah arsitektur yang menguraikan aplikasi berdasarkan fungsi tertentu menjadi beberapa layanan, dan setiap layanan dirancang untuk bekerja secara independen, dan dapat menggunakan teknologi yang berbeda sesuai kebutuhan. Manfaat menggunakan *microservices* adalah :

1. *Technology Heterogenity*

Melalui sistem yang terdiri dari beberapa layanan, pengembang dapat menentukan penggunaan teknologi dari setiap layanan. Hal ini memungkinkan pengembang untuk memilih teknologi yang tepat untuk setiap layanan, dari pada menggunakan teknologi berperforma rendah yang berfungsi untuk semua orang. *Microservice* juga memungkinkan pengembang untuk mengadopsi teknologi baru dengan cepat dengan resiko yang minimal.

2. *Resiliensi*

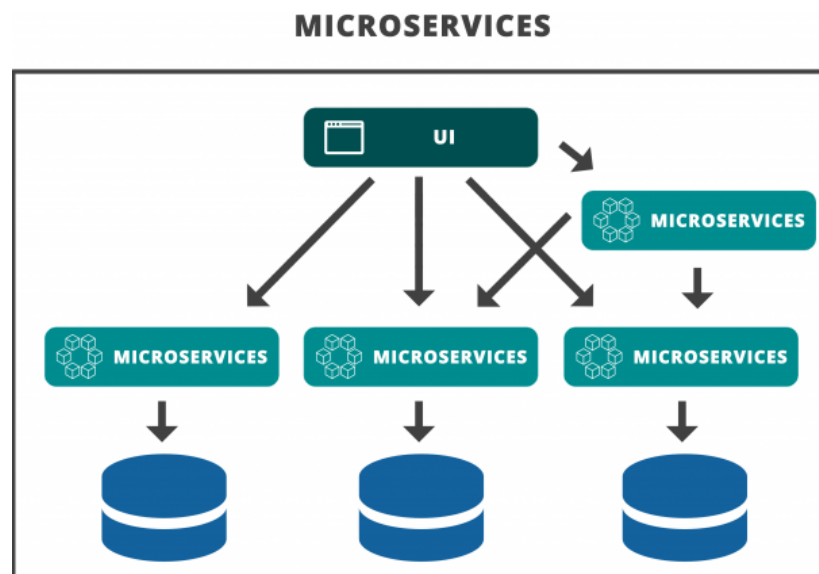
Penggunaan arsitektur *microservice* menjadikan aplikasi lebih tangguh, mencegah terjadinya kesalahan. Kerusakan dapat mudah terdeteksi dan hanya dialokasikan ke layanan tertentu. Dibandingkan dengan aplikasi monolitik biasanya kesalahan yang terjadi mengarah ke seluruh aplikasi untuk berhenti bekerja.

3. *Scalling*

Ketika beban *server* melebihi kapasitas, kinerja *server* membutuhkan proses *scalling* agar kinerja *server* tetap terjaga. Dalam aplikasi monolitik *scalling* hanya men *deploy* beberapa *service* saja. Hal ini menjadikan proses *scalling* lebih efisien karena bisa berjalan pada perangkat yang lebih kecil. Dengan karakteristik ini, kelebihanannya adalah :

1. Lebih sedikit kode dan bersifat mandiri jadi dapat dilakukan pengujian aplikasi dengan sendiri.
2. Mudah dalam perawatan perangkat lunak.
3. Dapat melakukan proses pengiriman *software* yang bersifat independen.
4. Mudah skalabilitas.
5. Pengembang dapat membangun aplikasi dengan berbagai bahasa pemrograman dan *framework*.

Microservice arsitektur disajikan dalam gambar 2



Gambar 2. *Microservice* Arsitektur
Sumber : (Ghozali, 2021)

3 teknik pengujian dalam *microservice* arsitektur menurut Suryotrisongko (2017), yaitu :

1. *Unit testing*

Pada *unit testing* setiap *service* wajib memiliki komponen *unit test* yang berkaitan. *Unit test* menguji bagian kecil dan mengisolasi dari fungsi.

2. *Componen test*

Terdapat pada tingkat pengujian *intefrface* eksternal dari *service* individual atau pada pengujian subsistem dari kelompok *service*. Pengujian ini digunakan untuk mengekspos *metric* dan *logging* melalui API pada saat pengujian.

3. *End-to-end testing*

Pengujian ini memastikan seluruh sistem dapat berjalan. Pengujian ini menguji kinerja dari seluruh fungsional bisnis yang dirancang.

2.2.2 API

Menurut Yanti & Rihyanti (2021), *Application Programming Intefrface* (API) adalah integrasi dari dua komponen sistem aplikasi yang terdiri dari elemen *function*, *protocols* dan lainnya. *Web API* mendukung fungsi CRUD (*Create, Read, Update, Delete*) yang berjalan melalui *HTTP protocol* dengan *method* GET, POST, PUT, dan DELETE. Kelebihan dari API adalah :

1. Memiliki *response Accept Header* dan *HTTP status code*
2. *Response* dengan format JSON,XML
3. Mempunyai metode MVC seperti *routing*, *controllers*, *action results*, filter model, *IOC container*, dll. *Web API* berjalan untuk semua ragam *server* seperti *Apache* dan yang lainnya. *Web API* mendukung segala bahasa yang dipakai dalam pemrograman.

2.2.3 REST

Representational State Transfer (REST) adalah *web service* yang memakai konsep perpindahan antar *state*, *server* akan mengirimkan *state* halaman *web* saat ini ke *browser*. Cara kerja REST yaitu dengan bernavigasi dari *link-link* HTTP untuk melakukan aktifitas tertentu, seperti terjadi perpindahan *state* satu sama lain. *Rest* dipakai oleh *browser* bisa dianggap sebagai bahasa internet dan API untuk menunjukkan layanan *web* yang akan dipakai pada *smartphone* berbasis *android*. Perintah HTTP yang biasa digunakan yaitu fungsi GET, POST, PUT, dan DELETE. Balasan yang dikirimkan dalam bentuk XML sederhana tanpa protokol pemaketan data, maka dari itu informasi yang didapat memudahkan pengguna untuk membaca dan menguraikan data. Kelebihan dari REST adalah

lebih mudah dimengerti, lebih sederhana dan simpel untuk dibangun dibandingkan SOAP (Fauziyah dkk., 2014). Ada empat prinsip dalam REST, yaitu :

1. Identifikasi *resources* melalui *Uniform Resource Identifier* (URI), REST *Web service* mencari gabungan berbagai sumber daya yang ditemukan interaksi antar *client*.
2. Memanipulasi *resource* melalui representasi.
3. Pesan *Self descriptive*, dapat mengakses berbagai format sebab sumberdaya informasi tidak terikat (HTML, XML, PDF, JPEG dan lainnya).
4. Memakai *hypermedia* sebagai *engine* dari *application state*.

2.3 RAD

Pengertian dari metode *Rapid Application Development* (RAD) yaitu suatu cara yang digunakan dalam pembangunan perangkat lunak tambahan dengan jangka waktu pengembangannya relatif cepat dan tidak mengurangi kualitas sistem tersebut. Langkah-langkah pada RAD terdapat tiga bagian, meliputi ketentuan apa saja untuk analisis kebutuhan, desain rancangan aplikasi, serta implementasi dan penyelesaian proyek untuk tahap yang terakhir (Aini dkk., 2019).

2.4 Perancangan dan Desain

Perancangan adalah langkah mendeskripsikan, merencanakan, menggambarkan dan mensketsa atau menyusun beberapa komponen independen menjadi satu menjadi fungsi yang utuh. Perancangan sistem dapat digambarkan dalam bentuk diagram alir sistem (*system flowchart*), yaitu suatu alat grafis yang dapat menampilkan proses sistem secara runtut.










Sedangkan desain adalah proses mendeskripsikan, menggambarkan, merencanakan dan menyusun beberapa komponen independen menjadi satu kesatuan fungsional yang lengkap. Desain atau rancangan memiliki dua tujuan, yaitu untuk memberikan gambaran yang jelas bagi programmer komputer dan pakar lainnya dan memenuhi kebutuhan pengguna sistem. Desain juga melibatkan

eksperimen atau beberapa percobaan seperti membuat sketsa, membuat *prototype* dan mencoba beberapa konsep dan ide (Fariyanto & Ulum, t.t.).

2.4.1 Flowchart

Pembuatan rancangan *flowchart* merupakan metode penggambaran algoritma sistem dalam bentuk simbol-simbol yang saling terhubung. Tujuan dari pembuatan *flowchart* yaitu menerangkan proses hubungan dan dokumentasi yang terjadi pada aplikasi (Putra dkk., 2019).

Tabel 1. Simbol *flowchart* program

Simbol	Nama	Fungsi
	<i>Input/Output</i>	Menyatakan <i>input</i> atau <i>output</i> disesuaikan jenis peralatannya
	<i>Offline Connector</i>	Menyatakan lanjutan dari prosedur ke prosedur dari halaman yang berbeda
	<i>Decision</i>	Menyatakan kondisi yang menentukan 2 kemungkinan (iya atau tidak).
	<i>Connector</i>	Menyatakan lanjutan dari prosedur ke prosedur dalam halaman yang sama
	<i>Terminal</i>	Menyatakan awalan atau akhir program.
	<i>Flow</i>	Menyatakan jalannya arus.
	Dokumen	Mencetak dokumen sumber atau laporan
	<i>Punched Card</i>	Menyatakan jalinan dari prosedur ke prosedur dalam halaman yang sama
	<i>Process</i>	Menyatakan suatu tindakan/ proses dari komputer







Sumber: (Ipapedia, 2018)

2.4.2 Mapping chart

Menurut Putra dkk., (2019), *mapping chart* adalah gabungan dari ilustrasi peta konsep dan *flowchart* yang menunjukkan pergerakan aliran dokumen dari suatu tempat dengan tempat lain, digambarkan menggunakan simbol-simbol standar untuk tahap pemecahan masalah secara sederhana, jelas dan rapih. *Mapping chart* disusun menggunakan simbol- simbol,dan simbol ini digunakan menjadi alat untuk mengilustrasikan proses dalam program. Petunjuk yang harus diperhatikan dalam membuat *flow map* adalah :

1. *Mapping chart* diilustrasikan dari bagian atas ke bawah dari kiri ke kanan.
2. Alur kegiatan yang diilustrasikan harus mudah dipahami oleh pembacanya.
3. Aktivitas yang berjalan harus dimulai dan berakhir secara jelas.
4. Langkah dalam tahapan dari aktivitas harus berada pada urutan yang benar.
5. Lingkup dan urutan dari alur kegiatan yang diilustrasikan harus diteliti dengan hati-hati.
6. *Flow map* digambarkan dengan simbol-simbol yang standar yang mudah dipahami.

Tabel 2. Simbol *mapping chart*

Simbol	Keterangan
	Dokumen masukan dan keluaran untuk prosedur yang manual maupun komputer
	Kegiatan manual
	Kegiatan dalam proses dari operasi dikomputer
	<i>Input</i> dan <i>output</i> menggunakan file
	Arus aliran dari proses
	Tempat penyimpanan data secara arsip

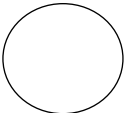

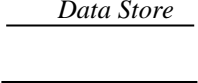

Sumber : (Dewi, 2019)

2.4.3 DFD (Data Flow Diagram)

Data Flow Diagram adalah diagram alir data yang dipakai untuk membangun model alur kerja sistem. *Data Flow Diagram* adalah aliran sistem

untuk menggambarkan proses kerja sistem sehingga membantu untuk secara logis, tersusun dengan jelas. *Data Flow Diagram* dapat menganalisis sistem, memperbaiki sistem dan meningkatkan kinerja sistem. DFD dibagi menjadi DFD level 0, DFD level 1 dan DFD level 2. DFD level 0 menjelaskan proses aliran data secara umum. Sedangkan DFD level 1 dan 2 menerangkan aliran data secara lengkap (Putra dkk., 2019)

Tabel 3. Simbol DFD

Simbol	Keterangan
<p>Proses</p> 	Menampilkan aktivitas yang dicoba oleh <i>user</i> atau komputer.
<p>External Entity</p> 	Menampilkan <i>entity</i> dalam lingkup luar sistem yang bisa dibentuk oleh <i>user</i> , organisasi, maupun sistem lain yang terletak pada luar area yang hendak memberikan <i>input</i> ataupun menerima <i>output</i> dari sistem.
<p>Data Store</p> 	Menampilkan simpanan dari informasi yang dapat berbentuk, file dalam <i>database</i> sistem komputer, arsip ataupun catatan manual.
<p>Data Flow</p> 	Menampilkan arus aliran informasi yang bisa berupa masukan dalam sistem ataupun hasil dari proses di sistem.

Sumber : (Dewi, 2019)

2.5 JavaScript

Javascript merupakan bahasa pemrograman yang tidak memerlukan *server* dalam menjalankannya. Semua kode hanya dituliskan dalam teks editor seperti *notepad*, *wordpad*, dan juga *visual studio code* yang hasilnya bisa langsung ditampilkan. *Javascript* juga dapat ditulis pada *file* terpisah dari *file* html yang dibuat, saat akan menjalankannya maka dengan kode tertentu *javascript* dipanggil melalui *file* html tersebut (Widodo, 2013).

Keunggulan dari *javascript* menurut (Romadhon, 2021) adalah :

1. Sangat ringan dan kecil, karena menggunakan memori yang sedikit.
2. Dapat dikolaborasikan dengan bahasa pemrograman lain.

3. Bersifat dinamis, yaitu dapat mengembangkan *web* secara fleksibel dan mudah dalam menambahkan fitur.
4. Dapat dijalankan di berbagai *platform* serta mudah dipelajari.
5. Mudah menemukan dan memperbaiki *error*.

2.6 *ReactJs*

ReactJs adalah struktur kerja *open source* dengan *library javascript* untuk membangun antarmuka pengguna. *React* biasanya digunakan dalam menangani pengembangan sistem aplikasi *single-page* dan juga *mobile*. Keuntungan menggunakan *Reactjs* adalah struktur kerja menyediakan kecepatan, kesederhanaan dan skalabilitas. *React* dikembangkan oleh *facebook* untuk membantu pengembang dalam membangun komponen UI yang lebih interaktif, *stateful* dan dapat digunakan kembali. Dalam pedoman MVC (*Model View Controller*), *react* hanya menyajikan pada bagian *view* yang paling baik dalam penyederhanaan (Nursaid dkk., 2020).

2.7 JSON

JavaScript Object Notation (JSON) adalah sebuah format yang digunakan untuk menyimpan serta melakukan penukaran informasi secara berurutan antara *server* dan *client*. File dalam *json* hanya memuat teks dan berekstensi *.json*. Terdapat dua komponen inti dari objek JSON yaitu *key* dan *value*. *Key* berbentuk *string* yang diapit oleh dua tanda kutip dalam penulisan. Sedangkan *value* dapat berbentuk *array*, *objek*, *string*, *Boolean*, angka atau *null* (Yanti & Rihyanti, 2021).

Menurut Fansuri (2014), struktur data JSON bersifat umum, karena menggunakan bahasa pemrograman yang biasa digunakan oleh *programmer*, yang mendukung struktur data dalam bentuk yang sama atau berbeda, hal ini mudah dipertukarkan dengan bahasa pemrograman dan struktur data ini. Berikut *parsing* data menggunakan JSON :

1. Aplikasi *backend* mengakses data dari *database*.

PHP akan mengakses data pada *database* dan mempresentasikannya dalam bentuk *array*, kemudian di *encode* kedalam format JSON menggunakan perintah *json_encode*, maka akan menghasilkan *array* dalam format *json*.

2. Aplikasi *frontend* mengakses aplikasi *backend*.

Frontend melakukan akses URL dari file PHP yang ada pada *backend* dan menangkap data yang sudah diubah ke dalam bentuk JSON kemudian disimpan pada *database frontend* bersifat sementara sebelum ditampilkan ke pengguna.

2.8 *System Usability Scale (SUS)*

System Usability Scale (SUS) menurut Brook (2013) digunakan untuk mengukur bagaimana persepsi kegunaan sistem komputer atau mengukur tingkat kepuasan seseorang setelah melakukan pengujian dengan mengerjakan tugas dan melihat hasil dari responden. Manfaat dari *System Usability Scale (SUS)* adalah :

1. Dapat membandingkan sistem lama dengan sistem baru yang akan diterapkan.
2. Menentukan kelayakan sistem.
3. Dapat mengidentifikasi masalah agar mudah dalam melakukan perbaikan.
4. Menentukan standar efektifitas dan efesiensi dari sistem.

Tujuan dari *System Usability Scale (SUS)* adalah :

1. Untuk melihat ukuran dari pemahaman subjektif dari seseorang dari manfaat sistem.
2. Untuk melakukan evaluasi secara cepat.
3. Efektifitas, efisiensi dan kepuasan pelanggan.

Beberapa kriteria item kuesioner dalam membangun *System Usability Scale (SUS)* adalah :

1. Harus ada timbal balik dari semua item yang dipilih.
2. Jumlah item harus dibatasi karena keterbatasan waktu untuk mengerjakan kuesioner.
3. Responden harus memikirkan tanggapan yang tepat, tidak terburu-buru walaupun kami menginginkan responden dapat mengisi dengan cepat.

2.9 **Artikel Ilmiah Terkait**

Penelitian terkait adalah teori dari beberapa sumber penelitian yang dijadikan sebagai referensi penyusunan tugas akhir ini. Referensi yang diambil

dari jurnal tersebut dapat berupa kesamaan studi kasus atau penggunaan metode pengembangan sistem. Dalam penataan pustaka ini, penulis menjadikan beberapa tulisan sebagai bahan literatur dalam pembuatan “Implementasi Perancangan Desain *Frontend* pada Aplikasi Monitoring Pemasangan Jaringan Menggunakan *Microservice* Arsitektur pada PT Lintas Data Multimedia”, diantaranya jurnal dari:

1. Saputra, dkk., (2019) yang berjudul “Aplikasi Monitoring Berdasarkan Laporan Kegiatan Organisasi Perangkat Desa Kabupaten Pringsewu Studi Kasus pada Badan Perencanaan dan Pembangunan Daerah (Bappeda)”. Aplikasi ini dibuat karena pihak Organisasi Perangkat Desa hanya mengirimkan laporan kegiatan di akhir tahun saja, sehingga mempersulit Bappeda dalam melakukan monitoring kegiatan. Hasil dari penelitian ini adalah dibuatnya aplikasi monitoring dan pelaporan untuk membantu Bappeda dalam memonitoring kegiatan OPD dan membantu OPD dalam mengirimkan laporan kegiatan kepada Bappeda. Metode yang digunakan dalam pengembangan aplikasi tersebut adalah *Rapid Application Development* (RAD).
2. Mulyono, dkk., (2019) dengan judul “Desain dan Implementasi *Microservices* Studi Kasus pada Layanan *Taking Order* (Aplikasi *E-Commerce* PT XYZ)”. Alasan dibuatnya aplikasi ini karena pengembangan sistem masih menggunakan layanan arsitektur *monolithic*, yang akan memperlambat kecepatan pengembangan aplikasi dan mempersulit penambahan pengembangan baru. Hasil dari perancangan sistem ini menggunakan layanan *microservice* untuk menciptakan aplikasi *e-commerce* guna memecahkan masalah pengembangan aplikasi sebelumnya. Metode yang digunakan dalam pengembangan aplikasi tersebut adalah *Rapid Application Development* (RAD).
3. Parjiono, dkk., (2019) yang berjudul “Aplikasi *Taking Order* (*Front End*) Menggunakan *Microservices* pada PT XYZ”. *Monolithic* arsitektur masih digunakan dalam pengembangan sistem *taking order* pada PT XYZ, semua proses yang dilakukan didalam satu *server* maka ketika *database* sedang dilakukan perbaikan, semua akses dari induk *server* tidak dapat dijalankan. Solusi yang diterapkan adalah dengan menggunakan layanan *microservice*

arsitektur dengan tujuan memudahkan dalam kerja sama dengan pengembang lain dan dapat menjalankan *REST API* dari pengembang lain. Aplikasi *e-commerce (Front End)* menggunakan *microservices*, dengan melakukan pendekatan pemrograman berbasis teks dengan rancangan *Object Oriented Programming (OOP)*. Metode yang digunakan untuk mengembangkan aplikasi tersebut adalah *Design Science Research Methodology (DSRM)*.

4. Handayani, (2020) dalam jurnal yang berjudul “Pengorganisasian Kerja Sistem Parkir Menggunakan Arsitektur *Microservice*”. Alasan dibuatnya aplikasi ini karena kurang efisiennya menggunakan arsitektur monolitik karena jika terjadi gangguan pada salah satu modulnya, maka seluruh sistem harus berhenti bekerja untuk dilakukan perbaikan. Objek yang diambil dalam penelitian tersebut adalah mesin parkir otomatis yang kemudian diimplementasikan dengan arsitektur *microservice*.
5. Munthe dkk., (2019) dengan jurnal yang berjudul “Aplikasi Monitoring dan Pelaporan Jaringan *Fiber Optic* Berbasis *Web* pada PGN COM Lampung”. Aplikasi ini dibuat karena dalam pelaporan situasi dan kondisi *fiber optic* yang dilakukan setiap akhir bulan, teknisi patrol memberikan laporan dalam bentuk *hardcopy*. Solusi yang diterapkan adalah dengan membuat aplikasi monitoring dan memberikan laporan jaringan *fiber optic* berbasis *web*, yang digunakan untuk memudahkan teknisi patroli dalam memonitoring dan memberikan laporan jaringan *fiber optic*.

Tabel 4. Penelitian Terkait

Judul	Peneliti	Tahun	Hasil Penelitian
Aplikasi Monitoring Berdasarkan Laporan Kegiatan Organisasi Perangkat Desa (OPD) Kabupaten Pringsewu Studi Kasus pada Badan Perencanaan dan Pembangunan Daerah (Bappeda)	Saputra, dkk	2019	Aplikasi yang dapat menolong Organisasi Perangkat Daerah (OPD) dalam proses pelaporan dan membantu Bappeda dalam melaksanakan monitoring kegiatan Organisasi Perangkat Daerah (OPD).

Desain dan Implementasi <i>Microservices</i> Studi Kasus pada Layanan <i>Taking Order</i> (Aplikasi <i>E-Commerce</i> PT XYZ)	Mulyono dkk.	2019	Aplikasi <i>e-commerce</i> dengan layanan <i>microservice</i> arsitektur guna memecahkan masalah pengembangan aplikasi sebelumnya
Aplikasi <i>Taking Order (Front End)</i> Menggunakan <i>Microservices</i> pada PT XYZ	Parjiono, dkk	2019	Aplikasi <i>e-commerce (Front End)</i> menggunakan <i>microservices</i> , dengan pendekatan pemrograman berbasis teks dengan rancangan <i>Object Oriented Programming (OOP)</i>
Pengorganisasian Kerja Sistem Parkir Menggunakan Arsitektur <i>Microservice</i>	Handayani,	2020	Aplikasi mesin parkir otomatis yang diimplementasikan dengan arsitektur <i>microservice</i> .
Aplikasi Monitoring dan Pelaporan Jaringan <i>Fiber Optic</i> Berbasis <i>Web</i> pada PGN COM Lampung	Munthe dkk.,	2019	Aplikasi monitoring dan pelaporan jaringan <i>Fiber Optic</i> berbasis <i>Web</i>
